

МЕТОДИ И АЛГОРИТМИ ЗА ИНТЕЛИГЕНТНА ЛОКАЛИЗАЦИЯ И НАВИГАЦИЯ В ИНТЕРИОРНИ И ЕКСТЕРИОРНИ СРЕДИ



РОСЕН ИВАНОВ

УНИВЕРСИТЕТСКО ИЗДАТЕЛСТВО
ГАБРОВО
2026

Росен Стефанов Иванов

**МЕТОДИ И АЛГОРИТМИ ЗА ИНТЕЛИГЕНТНА
ЛОКАЛИЗАЦИЯ И НАВИГАЦИЯ В ИНТЕРИОРНИ
И ЕКСТЕРИОРНИ СРЕДИ**

Монография



УНИВЕРСИТЕТСКО ИЗДАТЕЛСТВО

“ВАСИЛ АПРИЛОВ”

ГАБРОВО, 2026

Всички права запазени. Никаква част от тази монография не може да бъде възпроизвеждана или разпространявана под каквато и да е форма без писменото съгласие на автора.

© доц. д-р инж. Росен Стефанов Иванов - автор, 2026

Рецензенти: проф. дн Евелина Пенчева, проф. дмн Стоян Капралов

© Университетско издателство “В. Априлов” – Габрово, 2026

ISBN: 978-954-683-737-0

Съдържание

УВОД.....	1
ГЛАВА 1: ТЕОРЕТИЧНИ ОСНОВИ И СЪВРЕМЕННО СЪСТОЯНИЕ НА ПРОБЛЕМА.....	9
1.1 Основни понятия и дефиниции.....	9
1.1.1 Навигационни системи и техните компоненти	9
1.1.2 Локализация и позициониране	10
1.1.3 Контекстно-осъзнати системи.....	11
1.1.4 Достъпност и универсален дизайн	11
1.2 Класификация на навигационните системи	12
1.2.1 Екстериорни навигационни системи	12
1.2.2 Интериорни позиционни системи	13
1.2.3 Хибридни навигационни системи	14
1.2.4 Специализирани навигационни асистиращи системи	16
1.2.5 Перспективи.....	16
1.3 Критичен анализ на съществуващите решения.....	17
1.3.1 Предимства и недостатъци на съвременните подходи	17
1.3.2 Технически ограничения	20
1.3.3 Потребителски изисквания и очаквания	21
1.3.4 Предиизвикателства.....	23
1.3.5 Перспективи за бъдещо развитие.....	23
ГЛАВА 2: АЛГОРИТМИ ЗА ОБРАБОТКА И ОПРОСТЯВАНЕ НА GPS ДАННИ.....	25
2.1 Проблеми при обработката на GPS данни в реално време	25
2.1.1 Шум в GPS сигналите	25
2.1.1.1 Атмосферни грешки.....	25
2.1.2 Специфики при различни режими на движение	27
2.1.3 Изисквания за ефективност при мобилни устройства.....	28
2.2 Matlab инструментариум за визуализация на GPS данни.....	30
2.2.1 Съществуващи разработки.....	31
2.2.2 Описание на предлаганият Matlab инструментариум	32
2.2.3 Заключение	38
2.3 Алгоритъм RSTS за опростяване на GPS траектории в реално време	39
2.3.1 Въведение	39
2.3.2 Архитектура на алгоритъма за опростяване на GPS траекторията	41

2.3.3 Експериментални резултати	53
2.3.4 Заключение	59
ГЛАВА 3: МОДЕЛИРАНЕ НА СГРАДИ С ЦЕЛ ИНТЕРИОРНА НАВИГАЦИЯ	61
3.1 Въведение	61
3.2 BIM-базирани навигационни системи	62
3.3 Предлаган подход	64
3.4 Описание на предлагания модел	66
3.4.1 Геометричен модел	71
3.4.2 Семантичен модел	72
3.5 Внедряване на BIM	74
3.5.1 Контекстен модел.....	75
3.5.2. Модел за достъп до данни	75
3.5.3. Модел за запитвания, базирани на местоположението	76
3.5.4 Алгоритъм за намиране на път	77
3.5.5. Евакуация	77
3.6 Експерименти	78
3.7 Заключение	83
ГЛАВА 4: АЛГОРИТЪМ ЗА НАМИРАНЕ НА К НАЙ-КРАТКИ ПЪТИЩА В МНОГОЕТАЖНИ СГРАДИ .	85
4.1 Въведение	85
4.2 Анализ на проблема	85
4.3 Предлаган подход за намиране на маршрут	91
4.3.1 Моделиране на свързаността.....	92
4.3.2 Изграждане на модел на свързаност	94
4.3.3 Изчисляване на възможните пътища	101
4.4 Експериментални резултати	104
4.4.1 Намиране на път в пълни ненасочени графи	105
4.4.2 Намиране на маршрут в многоетажни сгради	106
4.5 Заключение	112
ГЛАВА 5: АЛГОРИТЪМ ЗА ЗАОБИКАЛЯНЕ НА ПРЕПЯТСТВИЯ.....	115
5.1 Въведение.....	115
5.2 Описание на предлагания подход.....	116
5.2.1 Полигонно буфериране	116
5.2.2. Буфериране на полилинии	120

5.2.3 Алгоритъм за заобикаляне на препятствия	121
5.3 Експерименти	123
ГЛАВА 6: ИНТЕРИОРНИ НАВИГАЦИОННИ СИСТЕМИ, БАЗИРАНИ НА RFID И NFC ТЕХНОЛОГИИ	127
6.1 Технологии RFID и NFC	127
6.1.1 Принципи на работа и технически характеристики	127
6.1.2 Предимства при интериорна навигация	128
6.1.3 Ограничения и предизвикателства	128
6.2 Система RSNAVI за интериорна навигация на слепи хора	130
6.2.1 Въведение	130
6.2.2 Архитектура и компоненти на предлаганата система	131
6.2.3 Моделиране на сградите	134
6.2.4 Планиране на маршрути	136
6.2.5 Навигация	138
6.2.6 Заключение	140
ГЛАВА 7: ЛОКАЛИЗАЦИЯ С BLUETOOTH LOW ENERGY МАЯЦИ	141
7.1 Въведение	141
7.1.1 Технически характеристики на BLE маяците	141
7.1.2 Методи за измерване на разстояние чрез оценка на RSSI	142
7.1.3 Шум в радио каналите	143
7.2 Алгоритъм за автоматично разполагане на маяци в сгради	145
7.2.1 Анализ на проблема	145
7.2.2 Описание на предлагания алгоритъм	148
7.2.3 Експерименти	156
7.2.4 Заключение	165
7.3 Алгоритъм за избор на маяци с цел точна локализация	166
7.3.1 Анализ на проблема	166
7.3.2 Описание на предлагания алгоритъм	168
7.3.3 Експерименти	172
7.3.4 Заключение	177
7.4 Алгоритъм за микролокализация в големи обществени сгради	178
7.4.1 Анализ на проблема	178
7.4.2 Описание на предлагания алгоритъм	182

7.4.3 Експерименти.....	190
7.4.4 Заключение	203
ГЛАВА 8: СИСТЕМА ЗА ПРЕВЕНЦИЯ НА ДЕМЕНЦИЯ ЧРЕЗ ТЕХНОЛОГИИ ЗА НАВИГАЦИЯ	205
8.1 Въведение	205
8.2 Анализ на проблема	208
8.3 Концептуална рамка	211
8.4 Архитектура на системата	215
8.5 Програмно осигуряване за ESP32-S3-BOX3.....	216
8.5.1 Wi-Fi мрежова свързаност	216
8.5.2 BLE сканиране	217
8.5.3 MQTT клиент	219
8.5.4 Дисплей	219
8.5.5 Диагностика и мониторинг	221
8.6 Бизнес логика	221
8.6.1 Технологичен стек	222
8.6.2 Реализация на системата	222
8.7 Визуализация на данните в реално време	223
8.8 Запис на данните от баджовете в база данни	224
8.9 Визуализация и експортиране на данните от базата данни	226
8.9.1 Описание на програмното осигуряване	226
8.9.2 Предварителна обработка на данните и експортиране.....	228
8.10 Заключение	230
ГЛАВА 9: БЪДЕЩИ НАПРАВЛЕНИЯ И РАЗВИТИЕ	231
9.1 Тенденции в технологичното развитие.....	231
9.2 Подобрения на предложените методи и алгоритми	234
9.3 Научни и технологични предизвикателства	236
Литература	239
Публикации на автора	253
Оценка на качеството на публикации на автора	255
Приноси на автора	256

Списък на фигурите

Фиг. 2.1	Опитна постановка	32
Фиг. 2.2	Изглед на главния прозорец на	34
Фиг. 2.3	GPS данни и разположение на видимите сателити за избран момент от време (сателитите, използвани за определяне на позицията с загълнени с червено).....	34
Фиг. 2.4	Получаване на размита оценка за нивото на шума в GPS данните	36
Фиг. 2.5	Получаване на start/stop точките на маршрут.....	37
Фиг. 2.6	Диаграма на процеса на алгоритъма за опростяване на траекторията	42
Фиг. 2.7	Грешки в GPS позиционирането в региони „движение-спиране-движение“: отклонение от 6,6 м за регион 1 и отклонение от 23,1 м за регион 2	44
Фиг. 2.8	Изглед към видимите сателити: Връзка между HDOP, средна стойност на SNR и средна височина при различни позиции на 3 видими сателита	47
Фиг. 2.9	Функции за принадлежност.....	48
Фиг. 2.10	Матрица на размити правила.....	49
Фиг. 2.11	Параметри на алгоритъма за опростяване на траекторията.....	51
Фиг. 2.12	Визуално сравнение на следите: а) Траектория 1 (RSTS вляво, DP вдясно), б) Траектория 2 (RSTS вляво, DP вдясно), в) Траектория 3 (RSTS вляво, DP вдясно)	55
Фиг. 2.13	Сравнителен анализ на нивото на батерията	59
Фиг. 3.1	Структура на предложения модел на данни (съставни части и връзки между тях)	65
Фиг. 3.2	Опростена UML клас диаграма на предложената BIM.....	67
Фиг. 3.3	Геометрична част от предлагания BIM.....	71
Фиг. 3.4	Геометрична част от предлагания BIM.....	74
Фиг. 3.5	Модел за достъп до данните.....	76
Фиг. 3.6	XML скрипт, описващ топологията на сграда.....	78
Фиг. 3.7	Тестване на услугата в симулирана 2D среда	80
Фиг. 4.1	Модел на свързаност: (а) нива на абстракция; (б) нива на йерархия	93
Фиг. 4.2	Намерете възможни пътища, обединяващи подграфи за различни нива на йерархията: „Dep-1“, „Dep-2“ и „Вертикална свързаност“ (виж Фиг. 4.1б).....	96
Фиг. 4.3	Инициализиране или разширяване на графика: (а) алгоритъм <i>addNode</i> ; (б) алгоритъм <i>addLink</i>	97

Фиг. 4.4 Премахване на възел или връзка от модела на свързаност: (а) алгоритъм <i>removeNode</i> ; (б) алгоритъм <i>removeLink</i>	98
Фиг. 4.5 Алгоритъм <i>insertTargetVirtualNode</i>	100
Фиг. 4.6 Алгоритъм <i>calculatePaths</i>	102
Фиг. 4.7 Максимален размер на използваната динамична памет (а) и относително време, прекарано в GC; (б) за пълен граф с $n=11$ и K от 100 до 5000	106
Фиг. 4.8 Многоетажни тестови среди: а) сграда „Ректорат“; б) сграда „Библиотека“	107
Фиг. 4.9 Сравнение на времето за намиране на маршрути при сценарий за намиране на път в рамките на един етаж.....	109
Фиг. 4.10 Сравнение на време за намиране на път между етажите: а) тестов сценарий 1; б) тестов сценарий 2.....	110
Фиг. 4.11 Сравнение на изчислителното време за сценарий 1 за намиране на път между сгради: а) $K = 1$, $K = 3$ и $K = 16$; б) $K = 5400$	111
Фиг. 4.12 Сравнение на време за намиране на път между сгради при сценарий 2: а) намиране на пътища от възел „стая-3414“ до възел „на открито“; б) намиране на пътища от възел „на открито“ до възел „хранилище-4-f3“	112
Фиг. 5.1 Буфериране на полигон	117
Фиг. 5.2 Пример за външно буфериране на полигон: а) алгоритъм в действие; б) краен резултат	120
Фиг. 5.3 Пример за буфериране на полилинии: а) алгоритъм в действие; б) краен резултат	120
Фиг. 5.4 Пример за заобикаляне на препятствие	121
Фиг. 5.5 Буфериране на стая: а) потребителят е близо до обекти 2 и 3; б) потребителят е близо до обекти 1 и 9	123
Фиг. 5.6 Последователност на работа на алгоритъма за заобикаляне на препятствия.....	124
Фиг. 5.7 Търсене на оптимален микроскопичен път в сграда	125
Фиг. 6.1 Архитектура на RSNAVI.....	132
Фиг. 6.2 Моделиране на сградата и потребителите	135
Фиг. 6.3 Пример за планиране на маршрут.....	138
Фиг. 7.1 Алгоритъм <i>polygonPartitioning</i> – просто праволинейно разделяне на многоъгълник на неприпокриващи се правоъгълници.....	150
Фиг. 7.2 Пример за разделяне на полигони (коридор).....	151
Фиг. 7.3 Алгоритъм <i>abd</i> – автоматично разполагане на маяци	153

Фиг. 7.4 Пример за разполагане на маяци: триъгълна форма; максималното разстояние между маяците е 6 m; вдлъбнати ръбове (зелени кръгчета); и маяци (червени кръгчета)	154
Фиг. 7.5 Алгоритъм <i>obd</i> – оптимизация на разполагането на маяци.....	155
Фиг. 7.6 Разделяне на коридора на неприпокриващи се правоъгълници (червени пунктирани линии): а) разлагане по ос <i>x</i> (7 правоъгълника); б) разлагане по ос <i>y</i> (10 правоъгълника)	157
Фиг. 7.7 Автоматично разполагане на маяци: а) стая с правоъгълна форма, триъгълно разполагане; б) стая с правоъгълна форма, правоъгълно разполагане.....	158
Фиг. 7.8 Разположение на маяците в правоъгълна зона (20 м x 10 м) за триъгълна форма и различна стойност за максималното разстояние между маяците	159
Фиг. 7.9 Разполагане на маяци в коридор със сложна форма, триъгълно разположение	159
Фиг. 7.10 Тестов сценарий 3 – тестова среда (3D модел)	160
Фиг. 7.11 Разположение на маяците: а) преди оптимизация (31 маяка); б) след оптимизация (21 маяка); и в) експертно разположение (24 маяка)	161
Фиг. 7.12 Навигация по маршрут при експертно разполагане на маяците: възможен маршрут (синя линия) и приблизителен маршрут, изчислен въз основа на локализация чрез маяци (червена линия)	163
Фиг. 13 Сравнение на сценарии за разполагане на маяци с помощта на алгоритъм за локализация NLLS (а) и Trilateration (б): диаграма „box-и-whisper“ (средна стойност – зелена линия; горен и долен мустак – черна кутия; горна съседна стойност – червени лин	165
Фиг. 7.14 Диаграма на предложения алгоритъм	168
Фиг. 7.15 Размити функции на принадлежност	169
Фиг. 7.16 Матрица на размити правила.....	171
Фиг. 7.17 Сценарии за тестване на разполагането на маяци ($D_{max}=5m$ за левите изображения, $D_{max}=7m$ за десните изображения): а) разполагане във формата на триъгълник; б) разполагане във формата на правоъгълник	173
Фиг. 7.18 Точност на локализация за правоъгълна форма на разположение (графика „box-и-whisper“): медианна стойност – зелена линия; горни (75%) и долни (25%) мустаци – сив правоъгълник; горна съседна стойност – червена; долна съседна стойност – синя; изключения – червени кръгчета	174
Фиг. 7.19 Точност на локализация за триъгълна форма на разположение (диаграма „box-и-whisper“): а) $D_{max}=5m$ (14 маяка); б) $D_{max}=7m$ (11 маяка)	174
Фиг. 7.20 Тестова среда: правоъгълна стая с размери 20 м x 10 м; триъгълна форма на разположение на маяците; $D_{max}=7$ м; 4 препятствия	175

Фиг. 7.21 NLOS сценарий: $n = 4, k = 3$; най-близък възел - $B2$; избрани маяци $\{B2, B1, B6, B5\}$, реална позиция на целевия възел – син кръг; приблизително разстояние до най-близкия маяк – червен пунктиран кръг; приблизителна позиция на целевия възел след корекция на RSSI – цианов кръг	176
Фиг. 7.22 Точност на локализация в сценарий NLOS: ляво – точност на локализация без използване на предложените алгоритми; дясно – точност на локализация с използване на предложените алгоритми	177
Фиг. 7.23 Размити функции на принадлежност	184
Фиг. 7.24 Матрица с размити правила	184
Фиг. 7.25 Локализация с помощта на алгоритъм ALG-1: (а) Как работи алгоритъмът: валидни пресечни точки (сини точки); невалидни пресечни точки (червени точки); приблизително местоположение на целевия възел pt (червен ромб); (б) Алгоритъм в действие: $\Delta RSSI = \pm 6 dBm$; целева точка (син кръг); кръг с $r = 1m$ (тъмножълт)	186
Фиг. 7.26 Локализация с помощта на алгоритъм ALG-2: (а) Как работи алгоритъмът: целева точка (синя окръжност); точки на пресичане (червени окръжности); (б) Алгоритми ALG 2A и ALG 2B в действие: $\Delta RSSI = \pm 3dBm$; кръг с $r = 1m$ (тъмножълт); приблизителна позиция на целевия възел, използвайки алгоритъм ALG-2A (зелен триъгълник нагоре); и приблизителна позиция на целевия възел, използвайки алгоритъм ALG-2B (зелен триъгълник надолу)	188
Фиг. 7.27 Модел на загуба на сигнала в лекционна зала, $\gamma = 2,533$ (95% доверителни граници)	191
Фиг. 7.28 Java приложение за предварителна обработка и извличане на данни от маяци	192
Фиг. 7.29 Предварителна обработка на данните от маяк – необработени данни (зелено), филтрирани данни (MA+AKF): средни стойности (зелени кръгчета), максимална стойности (червено) и минимални стойности (синьо): а) 1 m разстояние между референтния и целевия възел; б) 8 m разстояние между възлите	193
Фиг. 7.30 Предварителна обработка на данните от маяка в динамична среда и движещ се целеви възел: а) използва се неадаптивен KF; б) използва се предложеният AKF	195
Фиг. 7.31 Сравнение на точността на локализация на алгоритмите: а) Разстоянието между маяците е 3 m; б) Разстоянието между маяците е 6 m; в) Разстоянието между маяците е 9 m	197
Фиг. 7.32 Сравнение на различни типове разположение на маяци – диаграма „box-and-whisper“ ($n = 50, \Delta RSSI = \pm 5dBm$): а) Правоъгълна форма, $d = 5m$; б) Триъгълна форма, $d = 5m$; в) Правоъгълна форма, $d = 10m$; г) Триъгълна форма, $d = 10m$	199

Фиг. 7.33 Тестова среда – лекционна зала: маяци (червени правоъгълници); тестови точки (сиви кръгчета); мебели (жълти и зелени правоъгълници)	200
Фиг. 7.34 Точност на локализация за всички анализирани алгоритми за локализация във всички тестови точки	201
Фиг. 7.35 Предполагаеми маршрути спрямо идеалния маршрут (тъмно сиво).....	202
Фиг. 7.36 Сравнение между различни алгоритми за локализация според точността на локализация	203
Фиг. 8.1 Статистика на анализираните документи по: а) година на публикуване и б) държава в която работят авторите	208
Фиг. 8.2 Структура на концептуалната рамка.....	212
Фиг. 8.3 Архитектура на системата.....	215
Фиг. 8.4 Примерен телеметричен пакет.....	218
Фиг. 8.5 Примерен локационен пакет	218
Фиг. 8.6 Свързване с Wi-Fi AP на EDGE GATEWAY	220
Фиг. 8.7 Въвеждане на име и парола за достъп до домашната Wi-Fi мрежа: а) Помощна информация на дисплея на шлюза; б) Форма за задаване на името и паролата на домашната Wi-Fi мрежа през браузър	220
Фиг. 8.8 Информация, визуализирана на дисплея при нормална работа на шлюза	221
Фиг. 8.9 Данни, получени от ESP32-S3-BOX3: а) телеметрични данни; б) данни за локацията на баджа	223
Фиг. 8.10 Визуализиране на данните от баджовете в реално време.....	224
Фиг. 8.11 Съдържание на MongoDB колекция за конкретен бадж	226
Фиг. 8.12 Филтриране на данните от базата данни	227
Фиг. 8.13 Визуализиране на данните: а) данни от акселерометъра; б) температура; в) състояние на бутоните	228

Списък на таблиците

Табл. 2.1 Случайни грешки в GPS данните.....	46
Табл. 2.2 Описание на GPS траекториите.....	53
Табл. 2.3 Прагови стойности на параметрите	54
Табл. 2.4 Резултати след филтриране на GPS данни въз основа на знания.....	56
Табл. 2.5 Значение на параметъра <i>OUT</i>	56
Табл. 2.6 Значение на етапа „Адаптивно филтриране на Калман“	57
Табл. 2.7 Намаляване на флаш паметта.....	58
Табл. 3.1 Семантичен модел (опростен).....	73
Табл. 4.1 Сравнение на най-използваните алгоритми SSP и KSP.....	91
Табл. 4.2 Средно време за изпълнение на търсени на маршрути при пълни графики за предложените алгоритми и алгоритъма на Йен.	105
Табл. 4.3 Брой възли и връзки за различни нива на йерархията.....	108
Табл. 5.1 Избор на оптимален възможен път.....	125
Табл. 7.1 Обобщение на резултатите – минимални, средни и максимални стойности на грешката (отклонение от референтните точки) и стандартно отклонение	164
Табл. 7.2 Обобщение на резултатите, получени в динамична среда с използване на статичен целеви възел.....	194

Използвани съкращения

Съкращение	Значение на английски	Превод на български
AI	Artificial Intelligence	Изкуствен интелект
AKF	Adaptive Kalman Filter	Адаптивен филтър на Калман
AR	Augmented Reality	Разширена реалност
ASR	Automatic Speech Recognition	Автоматично разпознаване на речта
BFS	Breadth-First Search	Търсене в широчина
BIM	Building Information Modeling	Информационно моделиране на сгради
BLE	Bluetooth Low Energy	Bluetooth с ниска енергия
CAD	Computer-Aided Design	Компютърно подпомогнато проектиране
CSV	Comma-Separated Values	Стойности разделени със запетая
DGPS	Differential GPS	Диференциален GPS
DOP	Dilution of Precision	Разреждане на точността
EKF	Extended Kalman Filter	Разширен филтър на Калман
GAE	Google App Engine	Платформа за приложения на Google
GAP	Generic Access Profile	Общ профил за достъп
GC	Garbage Collector	Java технология за почистване на паметта
GIS	Geographic Information System	Географска информационна система
GNSS	Global Navigation Satellite System	Глобална навигационна сателитна система
GPS	Global Positioning System	Глобална система за позициониране
GPX	GPS Exchange Format	GPS формат за обмен
HDOP	Horizontal Dilution of Precision	Хоризонтално разреждане на точността
IFC	Industry Foundation Classes	Индустриални основни класове
IIR	Infinite Impulse Response	Безкрайна импулсна характеристика
IMU	Inertial Measurement Unit	Инерционна измерителна единица

Съкращение	Значение на английски	Превод на български
INS	Inertial Navigation System	Инерционна навигационна система
IoT	Internet of Things	Интернет на нещата
IPS	Indoor Positioning System	Система за позициониране в затворени помещения
IR	Infrared	Инфрачервен (канал, датчик)
KF	Kalman Filter	Филтър на Калман
LBS	Location-Based Services	Услуги, базирани на местоположение
LBSN	Location-Based Social Network	Социална мрежа базирана на местоположение
LOS	Line of Sight	Пряка видимост
MA	Moving Average	Подвижна средна
MEMS	Micro-Electro-Mechanical Systems	Микро-електро-механични системи
MIMO	Multiple Input Multiple Output	Множество входи множество изходи
MMSE	Minimum Mean Square Error	Минимална средно квадратична грешка
MQTT	Message Queuing Telemetry Transport	Транспортен протокол за пренос на телеметрични данни с опашки за съобщения
MR	Mixed Reality	Смесена реалност
NFC	Near Field Communication	Комуникация в близко поле
NLLS	Non-Linear Least Squares	Нелинейни най-малки квадрати
NLOS	Non-Line of Sight	Без пряка видимост
NMEA	National Marine Electronics Association	Национална асоциация на морската електроника
PSRAM	Pseudo Static Random Access Memory	Псевдо статична памет с произволен достъп
RAM	Random Access Memory	Памет с произволен достъп
RF	Radio Frequency	Радиочестотен
RFID	Radio Frequency Identification	Радиочестотна идентификация
RSS	Received Signal Strength	Сила на получения сигнал

Съкращение	Значение на английски	Превод на български
RSSI	Received Signal Strength Indicator	Индикатор за силата на получения сигнал
RTK	Real-Time Kinematic	Кинематика в реално време
SBAS	Satellite-Based Augmentation System	Сателитна система за подобрене
SLAM	Simultaneous Localization and Mapping	Едновременно локализиране и картиране
SOAP	Simple Object Access Protocol	Прост протокол за достъп до обекти
TDOA	Time Difference of Arrival	Времева разлика на пристигане
TLV	Type-Length-Value	Тип-дължина-стойност
TTS	Text-to-Speech	Текст към реч
UCD	User-Centered Design	Дизайн центриран на потребителя
UKF	Unscented Kalman Filter	Филтър на Калман без допълнителни предположения за линейност
URI	Uniform Resource Identifier	Единен идентификатор на ресурс
UUID	Universally Unique Identifier	Универсален уникален идентификатор
UWB	Ultra-Wideband	Ултра широколентов
VIO	Visual-Inertial Odometry	Визуално-инерционна одометрия
VR	Virtual Reality	Виртуална реалност

УВОД

Актуалност на проблема

Навигацията и локализацията стават все по-значими с разрастването на градовете и преобразуването им до умни градове. Необходимо е използване на съвременните технологии с цел създаване на надеждни и точни методи за навигация и локализация. Системите за навигация на закрито и открито са неделима част от концепцията за умни домове и градове и са свързани с качеството на живот на хората.

Пазарните проучвания установяват ръст в областта на интериорното позициониране и навигация, което обосновава необходимостта от изследвания в тази сфера. Глобалният пазар за интериорно позициониране и навигация нараства от \$13.74 милиарда през 2024 г. на \$16.90 милиарда през 2025 г. и ще продължи да се увеличава с годишен темп 22.52%, като достигне \$46.50 милиарда до 2030 г. ([Research и Markets, 2025](#)). Съществуват и други данни - пазарът на системите за интериорно позициониране се оценява на \$18.91 милиарда през 2024 г. и ще достигне \$160.35 милиарда до края на 2030 г. при годишен темп на растеж 42.8% ([Virtue Market Research, 2024](#)). Северна Америка е водещ регион в сегмента на интериорното позициониране и навигационни системи през 2025 г., което показва високо технологичното развитие в региона ([Straits Research, 2025](#)).

Този растеж се дължи на внедряването на Internet of Things (IoT) технологии в градовете и използването на изкуствен интелект при градското планиране. Технологиите за интериорно позициониране са компонент на приложенията за изкуствен интелект (AI) и IoT ([Zhang и колектив, 2024](#)). През последните години има нужда от интериорни услуги свързани с локализация, което обуславя развитието на технологични решения в областта. Умните градове внедряват технологии и данни с цел подобряване на жизнените условия на гражданите и работата на градските услуги, а навигационните системи имат специфични функции в тази област ([Zhou и колектив, 2024](#)).

Global Positioning System (GPS) не функционират в интериорни среди заради сателитните сигнали, които не достигат вътрешността на сградите ([Zafari и](#)

колектив, 2019). При липса на сателитни сигнали в интериорните пространства изследванията търсят технологии за интериорно позициониране. Този въпрос засяга градски структури - болници, летища, търговски центрове, образователни институции, музеи и обекти, където се изисква безопасна локализация.

Indoor Positioning Systems (IPS) се прилагат за решаване на тези ограничения. Съществуват три типа информация в основата на системите за интериорно позициониране: 1) Радиочестотни маяци с фиксирани позиции, които заместват GPS в интериорни среди; 2) Fingerprinting техники, които се базират на характеристиките на радиочестотната среда (всяко място има специфична радиочестотна карта - комбинация от Wi-Fi мрежи, Bluetooth устройства и безжични сигнали) и 3) Инерционни системи, които използват сензори като акселерометър, компас и жирокоп (Li и колектив, 2024). Изследванията въвеждат технологични рамки, базирани на модели с шест слоя за проучване на системите за интериорно позициониране, включващи слой устройство, комуникационен слой, мрежов слой, слой данни, слой методи и слой приложения (Tong и колектив, 2023).

IPS използват разнообразни технологии включително Wi-Fi, Bluetooth Low Energy (BLE), Radio Frequency Identification (RFID), Near Field Communication (NFC), компютърно зрение, и хибридни подходи, всяка с предимства и ограничения по отношение на точност, разходи и сложност на внедряване (Brena и колектив, 2017). Развитието в тази област се характеризира с преход към адаптивни системи, които интегрират сензорни източници и използват алгоритми за машинно обучение за подобряване точността и надеждността на локализацията.

Значение придобиват навигационните системи, които предоставят информация за местоположението и интегрират семантична информация за околната среда, потребителските предпочитания и условията. Изследователските тенденции в умните градове и интернет на нещата показват нарастване, като анализът на 14,309 статии от базата данни Scopus за периода 2010-2024 г. показва научното внимание към областта (Zhou и колектив, 2024).

В контекста на пандемията COVID-19 и социалните промени, важноста на навигационните системи се увеличи, тъй като те имат роля в проследяване на заразени хора, управление на потоците от хора и поддържане на социална дистанция в обществени пространства. Това създава изисквания към навигационните системи

по отношение на мониторинг в реално време, предсказващ анализ и управление на трафика, които надхвърлят функциите за локализация и навигация.

Специфични предизвикателства при навигацията на хора със зрителни увреждания

Навигацията е проблем за хората със зрителни увреждания, които не могат да се движат лесно в пространството. Незрящите хора имат трудности при придвижване, тъй като те нямат защита от опасни обекти. Световната здравна организация посочва, че над 285 милиона души по света живеят със зрителни увреждания, като 39 милиона са слепи ([WHO, 2019](#)). Тази популация се нуждае от мобилност и изисква технологични решения за решаване на проблема.

Средствата като бели бастуни и кучета водачи работят в някои ситуации, но имат ограничения и не дават информация за препятствията ([Kuriakose и колектив, 2017](#)). Белият бастун засича препятствия в радиус един метър пред потребителя, което не стига за навигация в среди с препятствия. Кучетата водачи изискват дълго обучение, разходи и не са достъпни за всички зрително увредени хора.

Проучвания сред експерти в областта на слепотата показват, че след външната навигация втората функция на електронните устройства за придвижване е интериорната навигация. За зрително увредени хора е трудно да се движат и да използват бял бастун за намиране на обекти като хора, животни, пътеки и тротоари. Това изисква разработване на системи, които дават информация за средата чрез слуха и допира.

Изискванията към навигационните системи за зрително увредени хора включват няколко важни характеристики ([Fallah и колектив, 2013](#)). Точността на локализацията трябва да е под 1 метър за да дава системата информация за близката околна среда. Реакцията в реално време е нужна за избягване на препятствия и предупреждение за опасности. Аудио интерфейси с пространствен звук дават възприемане на информация чрез способностите на слуха за локализация на звукове в пространството. Навигационните системи трябва да работят надеждно независимо от времето, светлината или звуците в пространството. Енергийната ефективност на навигационните системи е нужна за използване на мобилни устройства, а достъпността и разходите са важни за внедряването им.

Навигационни системи за хора с деменция

Паралелно с развитието на системи за зрително увредени хора, прогрес се наблюдава в областта на навигационните системи за хора с деменция. Деменцията засяга навигационните способности чрез нарушаване на пространствената памет, ориентацията и способността за планиране на маршрути, което поставя хората в риск от загубване и дезориентация.

Навигационните системи за хора с деменция имат следните цели: 1) проследяване на местоположението в реално време за предотвратяване излизането от безопасни зони; 2) анализ на модела на движение и походката за диагностика на когнитивни промени, мониторинг на дневните активности за оценка на функционалната независимост, както и навигационна помощ при дезориентация. Тези системи интегрират функции за уведомяване на лицата, които се грижат за болните при излизане от защитените зони (геозащити), GPS проследяване за външни пространства, Wi-Fi и Bluetooth технологии за интериорна локализация, както и сензори за анализ на биометрични показатели като скорост на движение, дължина на стъпките и стабилност на походката, които служат като индикатори за прогресия на заболяването.

[Pillette и колеги \(2022\)](#) правят преглед на литературата за устройства, които помагат на хора с деменция при интериорна, екстериорна и виртуална навигация. Анализът на 23 статии показва три типа важна информация: типовете навигационни съвети които устройствата дават; технологията на която се базират устройствата и методологията за оценка на устройствата и навигационните резултати.

Развитие на технологиите за позициониране и навигация

През последните години има развитие в областта на безжичните технологии и системи, което създава възможности за развитие на нови навигационни решения. BLE маяците са технология, която използва BLE пакети за излъчване на данни за идентификация и местоположение. Приложенията за смартфони засичат сигналите от маяците и доставят съобщения и информация на потребителя, създавайки възможности за взаимодействия според обстановката ([Crowdconnected, 2024](#)).

Концепцията за системи отчитащи контекста (Context-Aware Systems) е въведена от [Schilit и Theimer \(1994\)](#). Тя еволюира от засичане на местоположение към системи, които интегрират източници на контекстна информация включително време,

потребителски предпочитания, социални взаимодействия и характеристики на околната среда (Dey, 2001). Тази еволюция показва че навигацията изисква не само информация за местоположението, но и семантична информация за контекста на движението.

Системите базирани на местоположение (LBS) надграждат тези основи, предоставяйки услуги според локацията на потребителя и неговите предпочитания. Освен информацията за местоположението на потребителите, BLE маяците могат да предават контекстна информация чрез измерване на близостта до обект или област, което позволява създаването на приложения на базата на геолокация с персонализация.

Информационното моделиране на сгради - Building Information Modeling (BIM) - осигурява възможност за развитието на навигационните системи, както и за създаване на семантични модели на сгради, които съдържат геометрична информация, топологични връзки и функционални характеристики на пространствата (Eastman и колектив, 2018). Моделите се прилагат за планиране на маршрути, осигуряване на информация и създаване на навигационни решения, които засягат характеристиките на пространството.

Изкуственият интелект и машинното обучение засягат възможностите на навигационните системи и осигуряват адаптивно поведение, персонализация и предсказващи функционалности (Chen и колектив, 2020). Невронните мрежи и алгоритмите за дълбоко обучение се прилагат за подобряване точността на локализацията чрез извличане на модели от радиочестотните данни, разпознаване на модели в поведението на потребителите и генериране на персонализирано съдържание, което засяга потребностите и предпочитанията.

Цели и задачи на монографията

Целта на монографията е да представи научен подход за проектиране, разработка и внедряване на системи за навигация и локализация в интериорни и екстериорни среди и създаването на технологични решения.

Изследването си поставя задачи, които засягат характера на проблематиката. *Първата задача* е теоретичното обосноваване принципите и методите за създаване на навигационни системи за технологични платформи и приложни области, включвайки анализ на подходите и идентифициране на ограниченията. *Втората*

задача включва разработката на алгоритми за обработка GPS данни в реално време, оптимизация на интериорната локализация с BLE маяци, и интегриране на BIM за навигационни цели. *Третата задача* предвижда експериментална валидация на подходите чрез тестове в симулирани и реални условия, с количествена оценка на производителността и потребителското изживяване, използвайки показатели и методологии за оценка. *Четвъртата задача* се фокусира върху системната интеграция на технологиите в решения, които поддържат преход между интериорна и екстериорна навигация. *Петата задача* включва практическото приложение на системите в реални условия, включително обществени сгради и среди за хора със зрителни увреждания, с цел демонстриране практическа стойност.

За реализацията на задачите се използват множество методи и алгоритми:

- Анализът и усъвършенстването на алгоритми за опростяване GPS траектории цели намаляване изчислителната сложност и подобряване ефективността при мобилни устройства с ограничени ресурси.
- Разработката на нови методи за адаптивна селекция и оптимално разполагане на BLE маяци има за цел максимизиране точността при минимални разходи за нова инфраструктура.
- Създаването специализирани BIM, адаптирани за навигационни приложения със семантична информация, дава възможност за планиране маршрути и предоставяне на контекстна информация.
- Проектирането на йерархични алгоритми за планиране на маршрути в многоетажни сгради със сложна хоризонтална и вертикална свързаност адресира предизвикателствата при навигация в големи обществени сгради.
- Внедряването на контекстно-осъзнати системи за персонализирано съдържание, използвайки геофенсинг и изкуствен интелект, създава възможности за адаптивно потребителско изживяване.
- Експерименталната оценка на производителността, точността и потребителското приемане на системите предоставя емпирични доказателства за ефективността им.

Структура и организация на монографията

Информацията в монографията е структурирана в девет глави. Те са представени в логична последователност - от теоретичните основи през практическите

приложения до бъдещите направления в областта на навигацията и локализацията. Целта е да се предостави цялостен поглед върху системите за навигация и локализация.

Глава 1. Теоретични основи и съвременно състояние на проблема

Глава 1 въвежда читателя в теорията на навигацията и позиционирането – запознаване с основните понятия в областта на навигационните системи. В главата се анализират екстериорни и интериорни навигационни системи и специализирани приложения за навигация на хора с увреждания. Глава 1 включва критичен анализ на съществуващите технологии, техните предимства, недостатъци и предизвикателства.

Глава 2. Алгоритми за обработка и опростяване на GPS данни

В Глава 2 се анализират и разглеждат методи за обработка на GPS сигнали в реално време. Описани се техники за филтриране на шума в GPS данните, както и алгоритъм за опростяване на маршрути, наречен RSTS. Той е предназначен за работа в динамични среди и платформи с ограничени изчислителни ресурси.

Глава 3. Моделиране на сгради с цел интериорна навигация

Глава 3 разглежда използването на BIM технологии в системите за интериорна навигация. Целта е създаване на семантични модели на сгради. В главата се описват методи за интеграция на геометрични данни към BIM. Разгледани са и топологични данни за навигация в интериорни пространства.

Глава 4. Алгоритъм за намиране на K най-кратки пътища в многоетажни сгради

В Глава 4 се предлагат алгоритми за оптимизиране на маршрути. Алгоритмите са предназначени за сгради с множество етажи. Те отчитат както хоризонталната свързаност, така и вертикалната свързаност между интериорните пространства.

Глава 5. Алгоритъм за заобикаляне на препятствия

Глава 5 предлага техники за локална навигация. Описан е метод за полигонно буфериране. Разглеждат се алгоритми за заобикаляне на препятствия в реално време.

Глава 6. Интериорни навигационни системи, базирани на RFID и NFC технологии

В Глава 6 се анализира приложението на RFID при навигационните системи. Разглежда се и използването на NFC пасивни етикети за интериорна локализация. Акцентът е върху системата RSNavi. Тя е предназначена за навигация на хора със зрителни увреждания.

Глава 7. Локализация с Bluetooth Low Energy маяци

Глава 7 обхваща методи за локализация с BLE маяци. Предлагат се алгоритми за разполагане на маяците в интериорна среда. Прилагат се техники за селекция на маяци. Описано е внедряването на система за микролокализация в обществени сгради.

Глава 8. Система за превенция на деменция чрез технологии за навигация

Глава 8 съдържа описанието на система за предсказване на деменция в домашни условия. Системата функционира в реално време чрез анализ на биомаркери, получени от умни баджове.

Глава 9. Бъдещи направления и развитие

Глава 9 обхваща тенденциите в технологичното развитие на системите за навигация и позициониране. Установяват се възможности за подобряване на описаните в монографията методи и алгоритми. Описани са и предизвикателствата в областта на навигацията.

Всяка глава съдържа теоретично въведение. Следва описание на предложените методи и алгоритми. Прилагат се математически формулировки и алгоритмични детайли. Описани са получените резултати в контекста на изследванията.

Монографията е предназначена за изследователи, както и за специалисти в областта на компютърните науки. Може да се използва и от практики, които внедряват навигационни системи. Тя е структурирана така, че да бъде разбираема за читатели с различни нива на познания - от начинаещи до експерти, като същевременно предоставя достатъчно технически детайли за специалистите в областта.

ГЛАВА 1: ТЕОРЕТИЧНИ ОСНОВИ И СЪВРЕМЕННО СЪСТОЯНИЕ НА ПРОБЛЕМА

1.1 Основни понятия и дефиниции

1.1.1 Навигационни системи и техните компоненти

Навигационните системи се определят като технически решения, които включват хардуер и софтуер за установяване на местоположение, ориентация и движение на обекти в пространство. Развитието на тези системи се дължи на прилагането на сензори, методи за обработка на данни и елементи на изкуствен интелект ([Wagner и колектив, 2017](#)).

Технологичният прогрес може да се раздели на три поколения. *Първото поколение* се основава на радионавигация и ранни версии на GPS, при които точността е ограничена и обновяването на данните е бавно. *Второто поколение* включва използване на инерциални модули (IMU), ултразвукови и лазерни устройства за измерване на разстояния, което позволява локализация и при липса на сателитен сигнал ([Elsanhoury и колектив, 2021](#)). *Третото поколение* се свързва с прилагане на методи за машинно обучение, комбиниране на данни от различни сензори и адаптивни алгоритми, които се използват и в среди без налична предварителна информация ([Liu и колектив, 2024](#)).

Архитектурната на съвременните навигационни системи най-често включва пет основни функционални модула:

- **Сензорен модул.** Включва GNSS приемници, инерциални сензори, лидари, камери и BLE маяци. Те предоставят измервания с различна точност и честота на обновяване ([Irwansyah и колектив, 2024](#)).
- **Алгоритмичен модул.** Прилагат се филтри на Калман, particle filters, SLAM алгоритми и невронни мрежи за обработка на данни ([Wei и колектив, 2024](#)).
- **Модул за планиране на траектории.** Използват се графови методи и евристики за изчисляване на маршрути при отчитане на условията на средата.
- **Интерфейсен модул.** Осигурява взаимодействие с клиентите чрез визуални, аудио и тактилни канали.
- **Комуникационен модул.** Поддържа връзки с облачни услуги и протоколи за координация с други системи.

Развитието на навигационните технологии се насочва към комбиниране на класически позиционни системи и автономни платформи. В този случай навигацията има функция на елемент от общото поведение на системата (Ghasemieh и Kashef, 2024). Поради това се прилагат решения, които обработват данни за средата и се настройват според изискванията на задачата.

1.1.2 Локализация и позициониране

Локализацията и позиционирането се разглеждат като процеси в автономните системи. Чрез тях се установява местоположението на обекта, както и ориентацията и траекторията му. Точността на тези процеси се обуславя от използваните технологии, приложените алгоритми и условията, в които функционира системата (Nawaz и колектив, 2021).

Сателитната навигация (GPS, GLONASS, Galileo и BeiDou) се прилага за външна локализация и осигурява покритие с определена точност при наличие на подходящ сигнал. В градска среда, горски масиви и вътрешни пространства се отчита ограничение, тъй като сигналят може да бъде блокиран или отразен (Zou и колектив, 2016). Инерциалните навигационни системи се използват като автономно решение чрез използване на данни от акселерометър и жирокоп. При тях грешката се натрупва и се увеличава във времето, когато липсва външна корекция.

Радиочестотните технологии за локализация, като Wi-Fi, BLE и Ultra-Wideband (UWB), се прилагат във вътрешни пространства при точност при установяване на позицията под няколко метър до няколко сантиметра (Ye и колектив, 2023). При тях се използват методи базирани на Time Difference of Arrival (TDOA), Received Signal Strength Indicator (RSSI) и fingerprinting, чрез които местоположението се установява чрез данни от радиосигналите. Визуалните и лидарните технологии се комбинират със SLAM алгоритми. При този подход едновременно се изгражда карта на средата и се отчита позицията на обекта. Той се използва в среди, които имат структурирани елементи (Kumar и колектив, 2025).

Хибридните методи се разглеждат като подход за локализация чрез комбиниране на GNSS, инерциални и визуални данни с прилагане на Bayesian филтри и графова оптимизация. При тези системи се използват особеностите на различни сензорни източници, което позволява компенсиране на техните ограничения и установяване на определена точност (Liu и колектив, 2024). Развитието на 5G технологиите и предвижданото внедряване на 6G създават възможности за локализация чрез милиметрови вълни и антенни решетки. Това е начин за подобряване на качеството на навигацията в сгради (Elsanhoury и колектив, 2021).

1.1.3 Контекстно-осъзнати системи

Контекстно-осъзнатите навигационни системи се разглеждат като развитие спрямо класическите позиционни решения. При тях се използват допълнителни данни за средата, потребителя и задачата, чрез които се модифицира функционирането на системата ([Chen и колектив, 2022](#)). Тези системи имат функция за приспособяване на навигационните методи според условията и изискванията.

Контекстната информация се получава от различни източници, сред които карти в реално време, данни за трафик, детекция на препятствия чрез лидарни устройства и стерео визия, както и потребителски профили с ограничения и предпочитания за маршрути ([Zuo и Shen, 2023](#)). Обработката на тази информация се осъществява чрез алгоритми, в които се прилагат онтологични модели за семантично описание на средата, методи на машинно обучение за предвиждане на движението на обекти, както и системи с множество агенти за координация на работи или превозни средства.

Семантичното картографиране се разглежда като компонент на контекстно-осъзнатите системи, тъй като позволява интерпретация на геометричните и функционалните характеристики на средата ([Xiang и колектив, 2023](#)). Това включва идентифициране на различни обекти, определяне на тяхната функция и прогнозиране на поведението им във времето. Машинното обучение, включително дълбоките методи, има функция в тези процеси, като дава възможност за адаптиране към нови интериорни условия.

Контекстно-осъзнатите системи се използват при автономните доставки, мобилните услуги и умните градове. В тези случаи навигацията трябва да реагира на промени в средата и да се приспособява към различни ситуации ([Ghasemieh и Kashef, 2024](#)).

1.1.4 Достъпност и универсален гизайн

Принципите на достъпността и универсалния дизайн са основни изисквания при разработването на навигационни системи. Те осигуряват използваемостта на системите, включително от лица с увреждания ([Aharchi и Kbir, 2022](#)). Универсалният дизайн предполага създаването на интерфейси и алгоритми, които се използват без адаптация или обучение.

Мултимодалната обратна връзка представлява елемент за осигуряване на достъпност. Тя интегрира аудио, визуални и тактилни канали за комуникация с потребителя ([Ye и колектив, 2023](#)). Навигационните указания за незрящи и хора с

двигателни проблеми изискват разработването на адаптивни алгоритми. Те модифицират съдържанието и формата на информацията според потребностите на клиентите. Автоматичното откриване на препятствия и опасности в реално време се реализира чрез интеграция на компютърно зрение. Прилагат се лидарни технологии и машинно обучение за идентификация на заплахи (Mai и колектив, 2024).

Енергийната консумация и размерите на устройствата се разглеждат като фактори при носими навигационни решения за лица с увреждания. При тях интеграцията с мобилни приложения и IoT среди има за задача създаването на решения, които могат да се настройват спрямо потребителските нужди (Nawaz и колектив, 2021). В областта на носимите технологии, като часовници, очила и тактилни устройства, се отчита развитие, което дава възможност за прилагане на навигационна поддръжка.

Разработването на достъпни системи се обуславя от мултидисциплинарен подход, при който се съчетават технически знания и разглеждане на потребителски изисквания. В този процес се включва сътрудничество с организации на хора с увреждания, експерти по ергономия и специалисти по взаимодействие човек–компютър с цел изработване на решения, които осигуряват надеждна навигация и мобилност.

1.2 Класификация на навигационните системи

Навигационните системи се разработват чрез множество критерии в това число отчитане на околната среда и наличните технологии. На практика се използват четири категории системи, като всяка категория засяга специфични предизвикателства и изисквания: 1) Екстериорни навигационни системи; 2) Интериорни навигационни системи; 3) Хибридни решения за навигация и 4) Системи за навигация за лица с увреждания.

1.2.1 Екстериорни навигационни системи

Екстериорните навигационни системи се основават на глобалните сателитни навигационни системи (GNSS). В това число са GPS, GLONASS, Galileo и BeiDou, които осигуряват глобално покритие за позициониране на открито с определени нива на точност и надеждност. Архитектурната реализация на системите интегрира високочестотни приемници за сателитни сигнали. На практика се прилагат алгоритми за корекция на грешки като Differential GPS (DGPS) и Real-Time Kinematic (RTK) позициониране. Използва се и комбиниране с инерциални

навигационни системи (INS) за компенсиране на временните прекъсвания в сигнала ([Elsanhoury и колектив, 2021](#)).

Развитието на GNSS технологиите показва потенциал за подобряване на устойчивостта в условия като високи сгради, планински терени и области с растителност. Мултифреквентните приемници (приемници които работят на повече от една честотна лента) позволяват преодоляване на атмосферни грешки и грешки в йоносферата, докато асистирани GNSS (A-GNSS) технологии използват информационни канали (базови станции на мобилни оператори) за ускоряване на процеса на получаване на сигнал и подобряване на точността в урбанизирани среди.

Ниската цена, лесният достъп и световното покритие правят GNSS популярна технология. Но тя има и недостатъци. В градска среда високите сгради блокират сателитните сигнали. Гъстата растителност и подземните тунели създават същия ефект - загуба на връзката със сателитите. Резултатът е неточно определяне на позицията или пълно липсваща информация. Опасност представлява и човешкият фактор. Злонамерени лица могат да заглушат честотите (jamming) или да излъчват фалшиви сигнали (spoofing). При важни приложения това създава риск за сигурността ([Ghasemieh и Kashef, 2024](#)). Инженерите търсят решения чрез компютърните технологии. Изкуственият интелект може да научи как грешките се появяват в различни условия. Програмно се анализират записани данни от миналото и текущата ситуация. Така системата разбира кога възникват проблеми и ги поправя самостоятелно.

1.2.2 Интериорни позиционни системи

Затворените сгради представляват предизвикателство за сателитната навигация. Физическите бариери - стени, тавани, метални конструкции - прекъсват комуникацията между приемниците и сателитните системи. Резултатът е невъзможност за определяне на точната позиция в интериорни среди. Научната общност отговаря на проблема с разработка на алтернативни методи за локализация. Безжичните локални мрежи се използват за пространствено позициониране чрез геометрични изчисления - силата на радиосигнала от множество източници позволява математическо определяне на координатите. Bluetooth устройствата с ниско енергопотребление (BLE) се инсталират като стационарни маяци с предварително известни местоположения. UWB комуникационни протоколи осигуряват измервания с милиметрова точност. Оптичните и акустичните сензори предлагат алтернативни принципи за детекция

на разстояния. Инженерният избор между технологиите зависи от изискванията към прецизността, площта на покритие и ресурсните ограничения ([Nawaz и колектив, 2021](#)).

Bluetooth технологията с ниско енергопотребление (BLE) достига прецизност при локализация в диапазона от няколко метра при функциониране в комплексни архитектурни структури. Протоколите iBeacon, Eddystone и AltBeacon са тествани успешно в търговски комплекси, медицински заведения и транспортни възли ([Zou и колектив, 2016](#)). UWB комуникационните системи превъзхождат другите технологии по точност и устойчивост към електромагнитни смущения. Тези характеристики ги утвърждават като оптимален избор за производствени обекти и системи за проследяване на материални активи с непрекъснат мониторинг. При Wi-Fi мрежите се прилага различен подход - създават се цифрови отпечатъци на радиосредата (fingerprinting). Процесът включва картографиране на сигналните характеристики и използване на алгоритми за самообучение. Събраните данни от множество точки се обработват чрез изкуствен интелект за генериране на радиочестотни карти, които осигуряват прецизна локализация ([Dinh-Van и колектив, 2017](#)).

Визуалните данни се съчетават с измерванията от сензори за движение в SLAM архитектурата за реализиране на самостоятелна навигация при роботизирани системи и AR приложения ([Wei и колектив, 2024](#)). Функционирането се основава на паралелно изпълнение - създаване на карта от околната среда и установяване на местоположението на апарата в тази среда. Оптичните устройства фиксират образи от пространството, а инерциалните елементи отчитат промените в скоростта и посоката. Съвместната обработка на информацията води до триизмерно моделиране на територията и определяне на позицията на оборудването. Практическата реализация среща затруднения поради изискванията за специфична техника, многостепенни процедури за инициализация и постоянно техническо обслужване. Тези условия затрудняват масовото приложение на методологията. Настоящите проекти са ориентирани към премахване на ръчното конфигуриране. Самообучаващите се програмни решения на принципа на машинното познание осигуряват автономно адаптиране към вариращите особености на работното обкръжение.

1.2.3 Хибридни навигационни системи

Хибридните системи за определяне на местоположението представляват еволюционно развитие в навигационните технологии. Основният принцип се

базира на обединяване на различни типове сензори за постигане на стабилно позициониране при променливи работни условия. Техническата реализация обединява множество източници на данни. Сателитните приемници осигуряват глобална референция за позицията. Инерциалните датчици регистрират ускоренията и ъгловите скорости. Оптичните устройства анализират визуалната обстановка. Лазерните сканиращи системи измерват разстоянията до обектите. Радиолокационните системи детектират препятствията и движенията. Интегрирането на тези технологии създава устойчивост срещу технически неизправности. При загуба на сателитния сигнал останалите сензори компенсират липсващата информация. Резултатът е непрекъсната работоспособност на навигационната система дори в неблагоприятни условия ([Wagner и колектив, 2017](#)).

Автомобилите без водач функционират благодарение на комбинирани сензорни мрежи за навигация в сложната градска среда. Съчетаването на GPS позициониране с инерциални датчици и оптични сензори осигурява надеждно проследяване на пътя при проблемни зони - между високи сгради, в подлезите или при задръствания, където сателитните данни стават недостоверни ([Liu и колектив, 2024](#)). Автономните летателни апарата като дронове изискват различен технологичен подход за придвижване в триизмерното пространство. Конструкцията им включва оптични системи за създаване на карти и самоллокализация, сензори за атмосферното налягане и сензори за посоката. Обединените данни от тези източници гарантират устойчиво определяне на позицията при различни метеорологични условия.

Обединяването на данните от множество сензори се реализира чрез статистически техники. За целта може да се използват Extended Kalman Filters (EKF), Unscented Kalman Filters (UKF), Particle Filters и методи за граф-базирана оптимизация. Съвременните разработки включват изкуствени невронни архитектури и алгоритми за машинно обучение. Тези технологии прогнозираат поведенческите модели на системата и коригират нелинейните отклонения в сензорните показания. То този начин се постига висока прецизност и стабилност при променливи и непрогнозируеми работни режими ([Irwansyah и колектив, 2024](#)). Комбинирането на глобално позициониране, UWB комуникации и визуално-инерциална одометрия позволява достигането на сантиметрова прецизност. Особено ефективни са тези решения при преминаване от външна към вътрешна навигация и обратно - критичен момент за поддържане на непрекъсната локализация.

1.2.4 Специализирани навигационни асистиращи системи

Разработването на навигационни системи за хора със зрителни увреждания се разглежда като отделно направление в приложните изследвания. При тях се комбинират позиционни технологии и решения за подпомагане на мобилността на тази потребителска група. Системите интегрират мултимодални интерфейси за обратна връзка, включващи гласови указания, пространствени аудио сигнали и тактилна стимулация чрез носими устройства, с цел предоставяне на навигационна поддръжка ([Ye и колектив, 2023](#)).

Технологичната реализация на помощните навигационни системи се основава на комбинация от ултразвукови сензори, стерео камери и лидарни устройства. Чрез тях се откриват и класифицират препятствия, които се срещат в средата. Компютърното зрение и методите на машинно обучение се използват за разпознаване на обекти, при което се отчита информация за тротоари, пешеходни пътеки, светофари и входи на сгради ([Aharchi и Kbir, 2022](#)). Системи с BLE маяци се прилагат за ориентиране в обществени сгради. При тях се предоставят данни за разположение на услуги, стълбища, асансьори и изходи.

Навигационните системи трябва да имат универсален дизайн. Това позволява използване от всички потребители. Подходът е по-икономичен в сравнение с отделно разработване на системи за хора с увреждания и за хора без увреждания. Така се осигурява социална интеграция и устойчивост ([Mai и колектив, 2024](#)). Предполага се и използване на персонализирани системи на основа изкуствен интелект. Те се настройват според потребностите и възможностите на отделните лица.

1.2.5 Перспективи

На съвременният етап е перспективно разработването на хибридни навигационни системи. В част от разработките се използват контекстно ориентирани подходи. Използването на данни от множество сензори заедно с алгоритми на изкуствен интелект също е перспективно. Подобни системи се характеризират с висока точността при локализация, надеждност и достъпност ([Chen и колектив, 2022](#)).

Интеграцията на системи от различни категории обуславя развитието на навигационните технологии. Предизвикателствата пред тези системи са преходите между външна и вътрешна среда, използване на технологични ресурси и управление на енергийни ресурси. Възможности за това се откриват в развитието на 5G и 6G мрежи, което позволява свързване на навигационните функции с

облачни изчисления и изчисления на ръба (edge computing). Това ще позволи внедряване на практика на сложни навигационни услуги.

1.3 Критичен анализ на съществуващите решения

1.3.1 Предимства и недостатъци на съвременните подходи

Затворените архитектурни обекти изискват специални методи за установяване на местоположението. Съществуващите технологии работят по различни принципи и показват нееднаква ефективност при конкретните условия на експлоатация. Корените на проблема са във физиката на електромагнитното разпространение. Сателитните радиосигнали се поглъщат от строителните конструкции - стоманобетонни стени, метални покриви и подови плочи. Следствието е невъзможност за GPS локализация в сградни помещения, мазета и други покрити пространства. Експлоатационните изисквания усложняват инженерните задачи. Работната среда се променя непрекъснато - температурни вариации, радиочестотни смущения, както и преместващи се обекти. Позиционните устройства трябва да запазват точността си при тези динамични обстоятелства. Техническото развитие се ориентира към автономни решения, които използват локални датчици и оборудване за осигуряване на стабилно определяне на координатите.

1.3.1.1 Системи на базата на машинно зрение

Визуалните алгоритми за определяне на координатите в сградни интериори представляват научен интерес поради възможностите за обработка на сложна образна информация. Принципът се основава на анализ на естествените характеристики на пространството без изискване за специализирана техническа инфраструктура. Фотосензорните устройства реализират прецизно разчитане на архитектурните детайли и интериорните компоненти. Резултатът е ефективност при решаване на многостепенни локализационни проблеми в разнотипни затворени обекти ([Chen и колектив, 2022](#); [Liu и колектив, 2024](#)). Изследването на [Khan и колектив \(2022\)](#) доказва значимостта на образните техники за навигация в жилищни и обществени сгради. Основните ползи се изразяват в ниските капиталови вложения, опростената експлоатация, безпроблемната интеграция със съществуващи конфигурации и осъществимостта на реалното внедряване.

Икономическите предимства на подобни системи се коренят в достъпността на масово произвежданите камери и отсъствието на нужда от скъпа инфраструктура. Архитектурните елементи на сградите служат като готови референтни точки,

докато системата генерира тримерни модели на работното пространство. Комбинирането на образна интерпретация с данните от вътрешните датчици показва обещаващи перспективи за локализационното усъвършенстване (Liu и колектив, 2024; Ghasemieh и Kashef, 2024). Технологичният механизъм функционира посредством едновременен анализ на фотографските записи и сигналите от сензорите за ускорение - визуално-инерциалната одометрия. Научното изследване на Łukasik и колектив (2024) отчита засиленото внимание към визуалните методи. Стратегията обединява оптичните данни с информацията от датчиците за движение, безжичните комуникационни модули и устройствата за лазерно сканиране на разстояния (лидари). Комбинацията от различните канали води до подобряване на точността при локализация.

Оптичните системи за навигация имат множество ограничения. Работата им се нарушава при слаба светлина и силни контрасти. Изчислителните изисквания са високи за работа в реално време. Това създава проблеми при използването в мобилни устройства (Naseer и Burgard, 2017). Движещите се обекти в пространството предизвикват грешки при определянето на позицията. Необходими са специални алгоритми за решаване на този проблем (Liu и колектив, 2025).

1.3.1.2 Безжични комуникационни технологии

Безжичните технологии се използват за локализация в сгради чрез анализ на радиосигналите. Wi-Fi, Bluetooth, UWB и другите RF технологии имат предимства основно свързани с изграждане на инфраструктурата и цената за внедряване.

Обзорът на Pascacio и колектив (2021) за колаборативни системи показва че най-използваните технологии за позициониране са Wi-Fi, Ultra-Wideband и Bluetooth. Wi-Fi системите използват съществуващата инфраструктура и дават точност от няколко метра. Прилагат се техники като радио картографиране и трилатерация (Dinh-Van и колектив, 2017; Nguyen и колектив, 2016). Ultra-Wideband технологията има висока точност при измерване на разстояния. Това я прави подходяща за приложения с изисквания за прецизност (Nawaz и колектив, 2021; Liu и колектив, 2024). Основните техники включват Received Signal Strength Indication (RSSI), fingerprinting и Time of Arrival/Flight (ToA/ToF).

Анализът на Wichmann (2022) за системи за позициониране в болници показва следното разпределение на технологиите: Bluetooth Low Energy (n=17), Wireless-Fidelity (n=10), хибридни системи (n=4), Radio-Frequency Identification (n=4), UWB

(n=1), инфрачервени (n=1) и ZigBee (n=1). Безжичните технологии преобладават при локализация в специализирани среди.

Безжичните технологии имат и недостатъци. Те са чувствителни към интерференция и многопътно разпространение на сигнала. Металните обекти, електронните устройства и движещите се хора създават сложна радио среда. Това влияе на точността на локализацията (Doer и Trommer, 2020). Системите за fingerprinting изискват предварително картографиране на радио средата. Това е трудоемък процес. Картографирането трябва да се повтаря при инфраструктурни промени.

1.3.1.3 Инерциални сензори и обединяване на данни от множество сензори

Инерциалните сензори са компонент в системите за навигация в сгради. Те се интегрират и в мобилни устройства. Акселерометрите, жирокопите и магнитометрите дават информация за движението и ориентацията. Те функционират независимо от външни сигнали (Kulkarni и Beulet, 2023).

Предимството на инерциалните сензори е автономността. Могат да работят в среди с ограничена инфраструктура. Осигуряват високочестотни измервания за проследяване на движението. Micro-Electro-Mechanical Systems (MEMS) сензорите достигат добра точност при ниска цена и енергопотребление (Hou и Bergmann, 2022). Основен проблем за инерциалните сензори е натрупващата се грешка. Тя се увеличава с времето поради дрифт и шум. Това ги прави неподходящи за продължителна навигация без външно калибриране (Lan и колектив, 2015).

Обединяването на данни от различни сензори се използва за намаляване на ограниченията на отделните технологии. Съчетаването на визуална, инерциална и безжична информация се прилага за изграждане на системи за локализация (Gao и колектив, 2023; Irwansyah и колектив, 2024). В практиката се използват разширени филтри на Калман, particle филтри и методи на машинно обучение за обработка на сензорните данни (Elsanhoury и колектив, 2021). Изследването на Liu и колектив (2025), насочено към интериорна локализация чрез смартфони, показва, че комбинирането на сензори в интериорни условия представлява изследователски интерес и се основава на допълването между различните сензори.

1.3.1.4 LiDAR и структурирано светлинно сканиране

Light Detection и Ranging (LiDAR) технологията се използва за изготвяне на 3D карти на средата и за локализация в интериорни пространства. Чрез директно измерване на разстояния с висока точност и честота LiDAR системите намират

приложение при работи и автономни превозни средства ([Hensel и колектив, 2022](#); [Wei и колектив, 2024](#)).

Предимствата на LiDAR включват работа при различни условия на осветеност, точност на измерванията и създаване на 3D модели на средата. Съвременните LiDAR сензори имат обхват от стотици метри и точност до милиметри, което позволява използването им в различни приложения ([Mai и колектив, 2024](#)). Ограниченията на LiDAR технологията са свързани с цената, енергопотреблението и чувствителността към прозрачни и отразяващи повърхности. Размерът и теглото на LiDAR сензорите също ограничават употребата им в компактни мобилни устройства.

1.3.2 Технически ограничения

1.3.2.1 Точност и надеждност в сложни среди

В интериорната навигация има проблеми с точността и надеждността при различни условия на средата. Компонентите на интериора имат сложна форма, различна осветеност, отразяващи повърхности и подвижни обекти. Това затруднява работата на системите.

Проблеми възникват при многопътното разпространение на радиосигналите, при препятствия за визуалните сензори, при магнитни смущения върху електронните компаси и при натрупване на грешки в инерциални системи ([Pahlavan и колектив, 2015](#)). Различните материали и архитектурни елементи водят до разнообразна среда, което изисква използване на адаптивни алгоритми за локализация.

Изследванията се насочват към разработване на алгоритми, които могат да работят в различни условия. Прилагат се техники за машинно обучение за адаптиране към особеностите на средите и методи за откриване и корекция на грешки ([Xiang и колектив, 2023](#)). [Sariman и колектив \(2024\)](#) предлагат решение за навигация в помещения, което използва налични данни и технологии. Те описват обучение на алгоритъм за намиране на най-кратък път чрез дълбоко обучение. Това се прилага с цел подобряване на работата на системите за интериорна навигация.

1.3.2.2 Скалируемост и адаптивност

Има проблем и със скалируемостта на интериорните навигационни системи. Централизираните подходи имат ограничения при много потребители. Същото важи и за големи интериорни площи.

Разпределените архитектури могат да се използват за решаване на тези проблеми ([Wagner и колектив, 2017](#)). Използват се peer-to-peer комуникация и изчисления на

ръба. Прилагат се и разпределени алгоритми. [Pascacio и колектив \(2021\)](#) отбелязват нарастващ интерес към колаборативното позициониране. Наблюдава се и употреба на разпределени системи и системи без инфраструктура.

Има нужда от алгоритми, които са адаптивни към различните сгради. Те трябва да работят при различни интериорни условия. Необходимо е автоматично калибриране на сензорите.

1.3.2.3 Енергийна ефективност и ресурсни ограничения

Енергийната ефективност има значение при мобилни и носими устройства, които работят основно с батерии. Алгоритмите за обработка на сензорни данни, включително визуални и LiDAR системи, изискват много изчисления. Това води до по-голям разход на енергия. За ограничаване на този проблем се използват по-прости алгоритми и специализиран хардуер. Прилагат се методи като адаптивно семплиране, хибридни архитектури с изчисления в облака, както и сензори с ниска консумация ([Nawaz и колектив, 2021](#)). Мобилните устройства имат ограничени ресурси. Това налага компромиси между функционалност и производителност. Търсят се решения, които позволяват поддържане на висока точност при по-ниска консумация и ограничена памет.

1.3.2.4 Взаимодействие между системите и стандартизация

На този етап няма унифицирани стандарти и протоколи за интериорна навигация. Това води до проблеми при съвместната работа на различните системи. Производителите на системи за интериорна навигация използват свои решения и специфични технологии. Това силно затруднява интеграцията на подобни системи.

Трябва да се използват стандартизирани интерфейси, комуникационни протоколи и формати за данните. Изисква се общ подход при създаване на навигационни услуги. Едно възможно решение е да се използват общи API, стандартизирани формати и протоколи за обмен на данни за локализация.

1.3.3 Потребителски изисквания и очаквания

1.3.3.1 Специализирани потребителски групи

Анализът на потребителските изисквания показва разнообразие в нуждите на различни групи. Това налага разработване на специализирани решения. Хората с нарушено зрение имат нужда от навигационни системи с аудио-тактилна обратна връзка, информация за препятствията и маршрути с по-нисък риск от сблъсъци ([Aharchi и Kbir, 2022](#); [Ye и колектив, 2023](#)).

Прегледът на [Fernihio и колеги \(2023\)](#) относно методи за планиране на маршрути при интериорна навигация за хора с нарушено зрение установява, че разликите със зрящите не са основен фактор при повечето случаи. Посочва се, че маршрути, избягващи трудни за преминаване зони, преминаващи близо до стени, съдържащи ориентери и с по-малко завои, са използван подход за съобразяване с потребностите на тази група.

[Kim \(2025\)](#) в своя преглед на електронни помощни средства за хора с нарушено зрение отбелязва, че липсва обединен подход за интегриране на различни функции в тези системи за интериорна навигация. Анализът на [Ху и колеги \(2023\)](#), посветен на носими електронни помощни средства за избягване на препятствия, показва, че повечето използват камери и ултразвук за възприемане на средата. Най-често прилаганата обратна връзка е акустична.

За хора с когнитивни нарушения навигационните системи трябва да предоставят опростени инструкции, визуални ориентери и помощ при вземане на решения. Нуждите включват различни форми на представяне, специализирано съдържание и точно време за предаване на информация.

Професионалните потребители като спасителни екипи, поддръжка на сгради и логистични работници имат изисквания за точност и надеждност. В тези случаи системите трябва да работят в по-тежки условия и да се интегрират с професионални инструменти.

1.3.3.2 Потребителски опит и приемане

Опитът на потребителите се отчита при внедряване на навигационни технологии. Разработването на безопасни решения има за цел създаване на използваеми системи. Потребителският интерфейс се изисква да бъде прост за приемане от различни потребители. За да се подобри използваемостта на навигационните системи може да се доставя съдържание, съобразено с всеки потребител. Доверието в услугите за локализация се дължи основно на тяхната надеждна и прозрачна работа. От потребителя се изисква разбиране на действието на системата и контрол върху нейното използване.

1.3.3.3 Интеграция с умни сгради и IoT екосистеми

Тенденциите в умните сгради отварят възможности за интеграция на навигационни системи с IoT инфраструктура. Използват се сензорни мрежи за мониторинг на средата, системи за осветление и автоматизирано управление на климата. Интеграцията на навигационни системи с модел на сградите (BIM) дава достъп до

архитектурна информация за навигация и локализация ([Xiaodong и колектив, 2018](#)). Данните се използват за изготвяне на карти, настройка на алгоритми за локализация и предоставяне на контекстна информация.

Свързването с други системи в сградата позволява услуги като автоматизиран достъп, управление на климата и насочване към свободни паркоместа или стаи. В прегледа на [Shaharuddin и колеги \(2023\)](#) за IoT сензори в умни сгради при пожарни сценарии са определени пет теми: засичане чрез машинно зрение, автоматизация, евакуация и интериорна навигация, ранно откриване и превенция на пожар, както и BIM-свързани въпроси.

1.3.4 Преглед на извикателствата

Тъй като липсват единни методи за оценка на интериорни навигационни системи това е спънка за развитието и имплементирането им. Различните методики за тестване и различните метрики правят трудно директното сравняване на отделните решения. В наличните научни изследвания се прилагат различни критерии за оценка. Често се измерват точност на локализация, време за придвижване и енергопотребление. В някои случаи се включват и мнения на потребители. Налага се въвеждане на подход, който обхваща технически параметри и практически фактори заедно с потребителския опит. За тази цел са нужни методологии с референтни среди, стандартизирани протоколи и единни метрики. [Khan и колеги \(2022\)](#) отбелязват, че големият брой различни процедури прави сравняването трудно и показва необходимост от уеднаквени методи.

1.3.5 Перспективи за бъдещо развитие

Анализът на тенденциите показва няколко основни направления за развитие на интелигентни системи за навигация и локализация. Комбинирането на данни от различни сензори се утвърждава като водещ подход. Очаква се бъдещи решения да използват по-широк набор от технологии. Развитието на изкуствен интелект и машинно обучение позволява адаптация към различни среди и нужди ([Kumar и колектив, 2025](#)). Персонализацията и адаптивността ще бъдат важни свойства. Алгоритмите ще се учат от поведението на потребителите и ще се настройват според техните предпочитания. Това включва модели на движение, алгоритми за маршрути и препоръки според контекста. Интеграцията с разширена и виртуална реалност ще допълни функционалността на системите. AR интерфейсите ще наслагват информация върху реалната среда. VR ще се използва за обучение и симулация. Изчисленията на ръба и разпределените архитектури ще осигурят по-

бърза обработка и по-малка зависимост от централна инфраструктура. Така се постига по-добра защита на данните и по-висока надеждност.

ГЛАВА 2: АЛГОРИТМИ ЗА ОБРАБОТКА И ОПРОСТЯВАНЕ НА GPS ДАННИ

2.1 Проблеми при обработката на GPS данни в реално време

Обработката на GPS данни в реално време е проблематично и се характеризира с множество проблеми. Тези проблеми възникват от естеството на самата GPS технология, особеностите на средата за разпространение на сигналите и ограниченията на мобилните устройства, използвани за навигация и локализация.

2.1.1 Шум в GPS сигналите

GPS системата, въпреки високата си точност при оптимални условия, е изложена на различни източници на грешки и шум, които значително влияят върху качеството на получаваните данни. Основните категории грешки могат да бъдат класифицирани в няколко групи според тяхното произход и характеристики.

2.1.1.1 Атмосферни грешки

Разпространението на GPS сигналите през атмосферата е един от най-значимите източници на неточности. Грешки възникват основно при преминаване на GPS сигналите през йоносферата и тропосферата.

Поради взаимодействието на радиовълните с електрически заредените частици в йоносферата на височина между 50 и 1000 километра над земната повърхност грешките могат да достигнат до 50 метра при GPS приемници, които използват един честотен канал, и показват силна зависимост от слънчевата активност и времето от денонощието ([Klobuchar, 1987](#)). Йоносферите грешки имат характерна дневна вариация, като максималните стойности се наблюдават около обяд, когато йонизацията достига най-високи нива.

Грешките в тропосферата се дължат на рефракцията на сигналите в долните слоеве на атмосферата и са пряко свързани с метеорологичните условия - температура, налягане и влажност. За разлика от йоносферите грешки, тропосферните не могат да бъдат компенсирани чрез използване на два честотни канала и изискват специализирани модели за корекция ([Hopfield, 1969](#)). Грешката при преминаване на сигналите през тропосферата варира между 2 и 25 метра в зависимост от позицията на спътниците.

2.1.1.2 Разпространение на сигналите по множество пътища

Тъй нареченият многопътен (multipath) ефект представлява един от най-сложните за моделиране източници на грешки в GPS измерванията. Той възниква когато GPS сигналът достига до приемника по повече от един път - директно от спътника и чрез отражения от различни обекти в околната среда като сгради, терен или водни повърхности. Това е особено проблематично в градска среда, където високите сгради създават „каньони“, водещи до значителни грешки в позиционирането.

2.1.1.3 Геометрични фактори

Геометричното разпределение на спътниците оказва съществено влияние върху точността на GPS измерванията. Този ефект се оценява количествено чрез параметъра Dilution of Precision (DOP), който се разделя на няколко компонента:

- GDOP (Geometric DOP) - общ геометричен фактор;
- PDOP (Position DOP) - отчита позиционната точност;
- HDOP (Horizontal DOP) - отчита хоризонталната точност;
- VDOP (Vertical DOP) - отчита вертикалната точност;

GDOP измерва колко добро е разположението на GPS спътниците. Когато спътниците са разпръснати равномерно във всички посоки, получаваме точни позиции. Когато са сгрупани в една област, точността се влошава. Например, ако всички спътници са само на изток заради сгради на запад, позицията ще бъде неточна, защото няма информация от различни посоки. Стойности на GDOP под 2 означават отлично разположение, 2-5 е добро, а над 10 - неприемливо за точна навигация.

2.1.1.4 Шум от приемника

Вътрешният шум на GPS приемника и нестабилността на часовниците въвеждат допълнителни неточности. Времеви грешки възникват от различията между високо прецизните атомни часовници на спътниците и часовниците с кварцово-стабилизирана честота в потребителските GPS приемници. Тези грешки се третираат като допълнителна неизвестна в навигационните уравнения, но тяхната временна нестабилност въвежда шум в измерванията (Enge, 1994).

Всяка LBS услуга изисква определена точност на GPS данните. Един възможен начин за подобрене на точността на GPS позицията на клиентите е да се използва диференциална GPS (DGPS) или Satellite Based Augmentation System (SBAS) - EGNOS за Европа, WAAS за САЩ и MSAS за Япония. В този случай се изисква GPS

приемникът да е DGPS/SBAS съвместим и съответната сателитна система да има покритие за позицията на клиента. Подобряване на точността на GPS позицията е възможна и чрез обработка на GPS данните, най-често със статистически алгоритми. На настоящият етап използваният с най-голям успех статистически метод за филтриране на GPS данни е цифровият филтър на Калман и неговите разновидности.

2.1.2 Специфики при различни режими на движение

Различните режими на движение въвеждат специфични предизвикателства при обработката на GPS данни, изискващи адаптивни алгоритми и стратегии за филтриране.

2.1.2.1 Статично позициониране

При статично позициониране (обектът не се придвижва за определен период от време) основното предизвикателство е да се осигури дългосрочна стабилност на резултатите и да се намалят систематичните грешки. Тъй като приемникът остава неподвижен, може да се използва усредняване на данните във времето, което подобрява точността – стандартното отклонение на позицията намалява приблизително с $\sqrt{n}$, където n е броят на направените измервания (Hofmann-Wellenhof и колектив., 2007). При дълги периоди на наблюдение е важно да се моделират и компенсират бавно променящи се грешки, като например многопътното разпространение на сигнала и атмосферни влияния. За постигане на точност под 1m алгоритмите за статично позициониране често използват филтри на Калман с ниска динамика.

2.1.2.2 Придвижване на пешеходци

Пешеходното движение се характеризира с относително ниски скорости (1-2 m/s), чести промени в посоката и периоди на застой. За пешеходно движение се използват модели с постоянна скорост или постоянно ускорение. Характерното за този режим е високата непредсказуемост на траекторията и необходимостта от адаптивно филтриране. Алгоритмите трябва да разграничават периодите на движение от статичните интервали за оптимизиране на параметрите на филтрирането (Brown и Hwang, 2012).

2.1.2.3 Движение с превозни средства

Движението с автомобили създава специфични ограничения и особености, които могат да се използват за повишаване на точността на позиционирането. Тъй като автомобилите се движат основно по пътища, това поставя геометрични ограничения върху възможните им траектории. Алгоритмите за позициониране

използват тази особеност, като коригират GPS координатите чрез съпоставяне с дигитални карти и пътната мрежа (map matching). При високи скорости се налага прилагането на по-сложни модели на движение, които отчитат инерцията и кинематичните ограничения на автомобила. Често се използват модели с постоянен радиус на завиване и модели за координирани завои, които по-точно описват динамиката на превозното средство.

2.1.2.4 Високоскоростно движение

При високи скорости на движение (над 100 км/ч) възникват специфични предизвикателства, свързани с доплеровия ефект и ограничената честота на обновяване на GPS данните. Поради относителното движение между спътниците и приемника се получава изместване на честотата на сигнала. При скорости около 300 км/ч това изместване може да достигне до ± 5 kHz, което налага използването на специализирани алгоритми за корекция и компенсация на честотните промени (Kaplan и Hegarty, 2017). Освен това стандартната честота на обновяване на GPS данните (обикновено 1 Hz) не е достатъчна за точно проследяване на обекти, движещи се с висока скорост. За да се подобри точността, се използват алгоритми за интерполация и екстраполация, които изчисляват междинните позиции между реалните GPS измервания.

2.1.3 Изисквания за ефективност при мобилни устройства

Мобилните устройства налагат строги ограничения върху алгоритмите за обработка на GPS данни, изискващи балансиране между точност и ефективност на изчисленията.

2.1.3.1 Енергийна ефективност

Енергийното потребление е ключов фактор при мобилните устройства, тъй като GPS приемниците са едни от най-енергоемките компоненти. Постоянната обработка на данни допълнително ускорява изтощаването на батерията. За оптимизация се използват няколко подхода:

- Адаптивно четене от приемника – честотата на GPS измерванията се регулира динамично според скоростта на движение и необходимата точност. Например, при неподвижно устройство честотата може да се намали до 0.1 Hz, докато при високи скорости може да се увеличи до 10 Hz за по-точно проследяване.
- Хибридни методи – GPS данните се комбинират с информация от вградени сензори (акселерометри, жирокопи и магнитометри), които консумират

значително по-малко енергия. Чрез интегриране на ускорението и ъгловата скорост тези сензори позволяват навигация (инерциална) при кратки прекъсвания на GPS сигнала, като така се намалява нуждата от постоянен GPS достъп.

- Интелигентно управление на захранването – модерните системи използват модели за временно изключване на GPS приемника в зони с лошо покритие и го активират отново на базата на прогнози за движението на устройството.

2.1.3.2 Изчислителни ограничения

Мобилните устройства разполагат с ограничени изчислителни ресурси, което налага оптимизация на алгоритмите за навигация и обработка на GPS данни в реално време. Основните подходи за постигане на ефективност включват:

- Паралелизация на изчисленията – съвременните мобилни процесори поддържат многоядрена архитектура и GPU ускорение, което позволява едновременно обработка на данни и значително ускорява алгоритмите за навигация.
- Адаптивна сложност на алгоритмите – за да се балансира между точност и производителност, алгоритмите могат динамично да намаляват своята изчислителна сложност при ниски нива на батерията или при високо системно натоварване, като същевременно поддържат приемлива точност.

2.1.4 Бъдещи направления

Съвременните решения за ефективна обработка на GPS данни в реално време се основават на интегриран подход, който комбинира различни техники и технологии за подобряване на точността, надеждността и ефективността:

- Комбиниране на данни от различни сензори – чрез използване на информация от GPS, инерциални навигационни системи (INS), и други сензори се постига по-висока точност и надеждност (Groves, 2013). Тази интеграция позволява взаимно компенсиране на слабостите на отделните системи и осигурява непрекъсната навигация, дори при временна загуба на GPS сигнал.
- Машинно обучение – алгоритмите за машинно обучение все по-често се използват за прогнозиране и коригиране на GPS грешки. Те са особено ефективни в среди с ограничена видимост на спътници, като градски

каньони, и могат да моделират сложни нелинейни зависимости в данните, адаптирайки се към динамични условия.

- Облачни изчисления – прехвърлянето на тежките изчислителни операции към облачни платформи позволява разтоварване на мобилните устройства, като същевременно се поддържа високо качество на обработката. Този подход е особено полезен за методи като Real-Time Kinematic (RTK) позициониране и обработка на диференциални корекции, изискващи висока точност и бърза реакция.

Предизвикателствата при обработката на GPS данни в реално време могат да бъдат ефективно решени само чрез комплексен подход, който съчетава теоретични принципи на сигналната обработка с практическите ограничения на мобилните устройства. Изследвания следва да се фокусират върху разработване на още по-ефективни и адаптивни алгоритми, способни да работят надеждно в разнообразни условия при минимални ресурсни изисквания.

2.2 Matlab инструментариум за визуализация на GPS данни

В последните няколко години рязко се увеличи броя и значението на мобилните услуги, които се базират на местоположението на клиентите, например: Location Based Services (LBS), Location Based Social Networking (LBSN) и Geographic Information Systems (GIS). Клиентите на подобни услуги трябва да разполагат с мобилни телефони с вграден GPS приемник, или външен GPS приемник с Bluetooth интерфейс. В зависимост от услугата програмното осигуряване от страна на клиента изпраща към сървъра през определен интервал от време необходимите GPS данни, например: географска ширина, дължина и височина; скорост и посока на придвижване; оценка за моментната точност на GPS приемника и др. Тази информация се извлича от данните, които GPS приемникът генерира във формат National Maritime Electronics Association (NMEA) 0183, най-често през 1 секунда.

При по-голяма част от LBS услугите се налага предварителна обработка на GPS данните от мобилните клиенти, преди те да се предадат към сървъра. Това се прави с цел подобряване на оценката за позицията на клиента и оптимизиране на трафика между клиентите и сървъра. Например, ако се налага on-line следене на клиентите на дадена услуга, количеството информация, която се предава към сървъра, е прекалено голяма и това води както до натоварването му, така и до генериране на голям трафик, който се заплаща. В този случай е подходящо да се използва алгоритъм, който намалява броя на точките с които се описва маршрута на клиента

- към сървъра се предават само координатите на точките от маршрута в които има реална, значима за услугата промяна в посоката на придвижване. Ако услугата предполага off-line обработка на маршрута, например GIS като Google Maps и Google Earth, големият брой данни води до много бавен отговор на сървъра или до блокиране на брауъра от страна на клиента.

Разработването на LBS услуга предполага обработка на GPS данните от страна на клиента, преди те да бъдат изпратени към сървъра за анализ. Това налага разработването на програмно осигуряване за математически анализ и визуализация на GPS данни с цел по-бързо разработване на необходимите алгоритми. Предлага се Matlab инструментариум чрез който в off-line режим могат да се обработват чрез статистически методи и анализират всички GPS данни, които GPS приемниците генерират. За целта е необходимо тези данни да се запишат в NMEA формат, който се явява входен за приложението. На базата на математическа обработка и визуална преценка на GPS данните, разработчиците на LBS приложения могат да намалят времето за разработване на необходимите алгоритми.

2.2.1 Съществуващи разработки

Съществуват множество приложения за анализ, визуализация и моделиране на данните от GPS сателити и приемници. Те могат да бъдат систематизирани в следните групи: комерсиални или безплатни; за on-line или off-line обработка на GPS данните; професионални или с масово приложение; платформено независими или зависими.

Програмното осигуряване, предназначено за научни цели, в повечето случаи е с много висока цена и е насочено предимно към анализ и моделиране на данните, които GPS сателитите генерират. Типични представители на този клас програмно осигуряване са: BERNESE (GPS/GLONASS Post Processing Package) на астрономическият институт в Берн, GAMIT на катедрата по атмосферни и планетарни науки към MIT и GIPSY/OASIS-II на Jet Propulsion Laboratory, NASA.

Повечето от програмните приложения за анализ на данни от GPS приемници са платформено зависими. Типичен пример за това е GPSy на Global Mapping System, която работи само под Macintosh.

Платформената независимост може да се постигне по няколко начина: обработката, анализът и визуализацията на GPS данните да се реализира на WEB сървър или да се използват платформено независими среди като Java и Matlab. По-голямата част от Matlab инструментариума за обработка на GPS данни е насочен

към обработка на информацията от GPS сателитите, например Constellation Toolbox на Constell Inc., GNSS Toolbox на Orion Dynamics & Control Inc., GPS receiver Toolbox на Data Fuzion Corp., GPS Software Toolbox на Lupash Consulting, GPS Toolbox на L3NAV Systems, GPS Matlab Toolbox на Draper Laboratory и др.

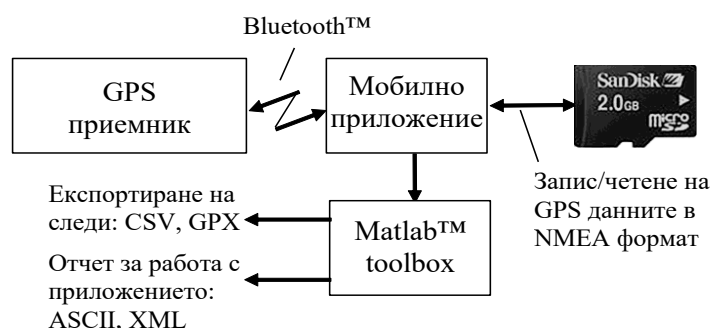
Основните недостатъци на малкото съществуващи платформено независими програмни приложения за анализ и визуализация на данни от GPS приемници са: прекалено висока цена на професионалните версии; невъзможност за промяна на изходния код; ограничени функционални възможности (само визуализация или само моделиране).

2.2.2 Описание на предлаганият Matlab инструментариум

Разработен е Matlab инструментариум, който дава възможност за: 1) декодиране на всички GPS данни от NMEA изразите; 2) визуализация на всички данни, генерирани от GPS приемника; 3) визуализация на позициите на видимите сателити в избран момент; визуализация на следа на маршрут без или с GPS карта; 4) определяне на режима на придвижване; 5) анализ и оценка на шума в GPS данните; 6) филтриране на позициите чрез адаптивен филтър на Калман; 7) редуциране на броя на точките, описващи следата на маршрута; въвеждане и запис на еталонен маршрут; 8) сравняване на маршрути; 9) експортиране на следи; 10) изчисляване на GPS разстояние, посока на придвижване и др.

2.2.2.1 Описание на основните характеристики на приложението

Входните данни с които предлагания Matlab инструментариум работи, трябва да бъдат в NMEA 0183 формат. Те се получават чрез специално разработено мобилно приложение, което позволява прехващане и експортиране на данните от GPS приемници в NMEA формат (виж Фиг. 2.1).



Фиг. 2.1 Опитна постановка

Освен експортиране на GPS данните в NMEA формат мобилното приложение дава възможност за включване и изключване на статичната навигация и разрешаване и забрана на използване на DGPS.

Всички експерименти са реализирани с GPS приемник Nokia LD-3W с вграден Bluetooth интерфейс. За да се ускори работата с приложението GPS данните в NMEA формат се конвертират до MAT файл.

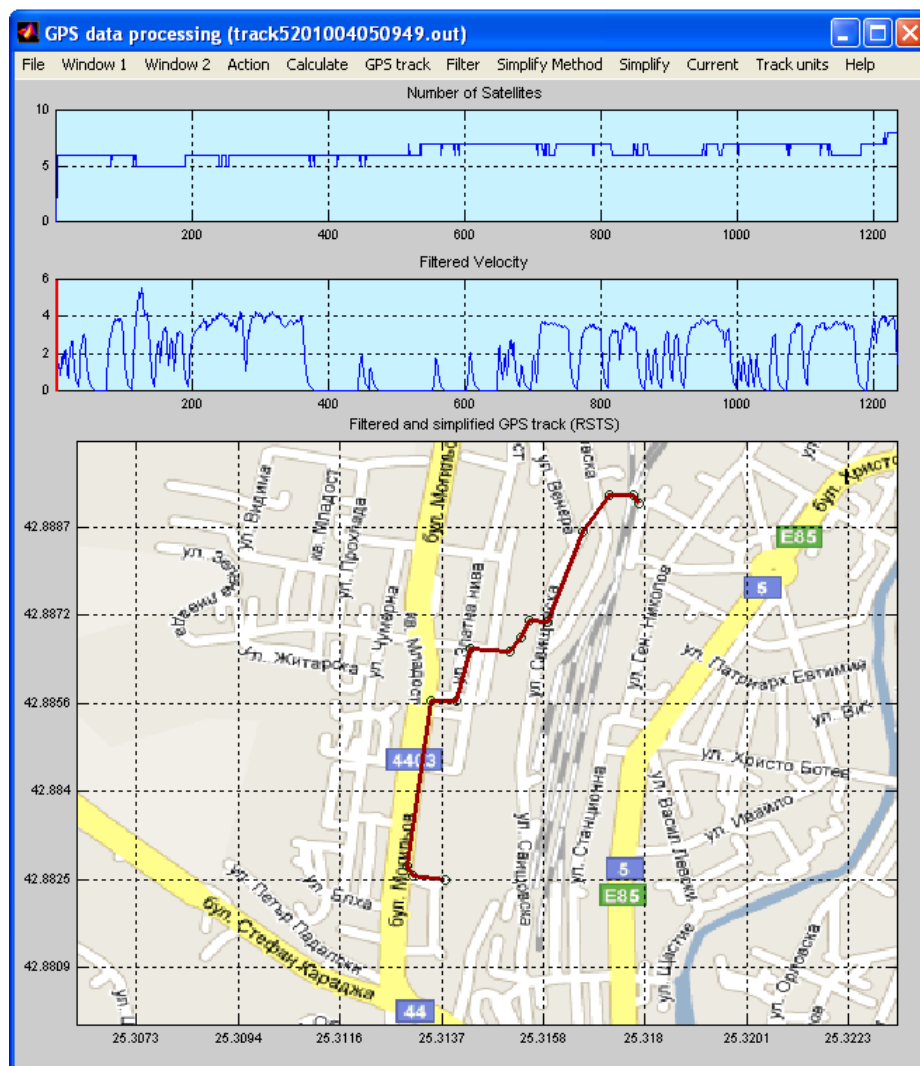
Приложението позволява експортиране на маршрути на еталонни или реални маршрути след редуцирани на броя на точките, с които се описват, в GPS Exchange (GPX) или Comma-Separated Value (CSV) формат. След завършване на работа с приложението се генерира отчет (ASCII текстов файл за потребителя и XML файл за машинна обработка) в които са записани всички резултати, получени по време на работа с приложението.

2.2.2.2 Визуализация

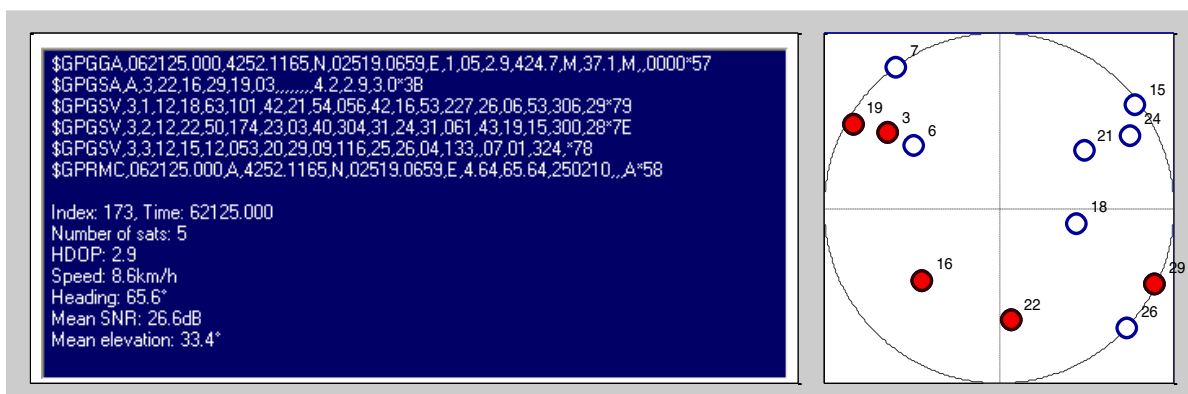
Приложението позволява визуализирането и произволно мащабиране на следните GPS данни: брой на видимите сателити; скорост и посока на придвижване; средна стойност на SNR, елевация и азимут за видимите сателити; оценка за хоризонталната и вертикална точност в зависимост от моментното пространствено разположение на видимите сателити и GPS приемника (HDOP и VDOP). Във всеки момент от време в прозорци 1 и 2 се визуализират два избрани от потребителя параметри. Всички визуализирани данни могат да бъдат филтрирани с рекурсивни и нерекурсивни цифрови филтри (виж Фиг. 2.2).

Приложението позволява визуализация на следата на изминатия от клиента маршрут (прозорец 3). Възможно е използването на GPS карти. Картите се свалят чрез Google Maps Static API. Маршрутите могат да се визуализират с точки, линии или комбинация от точки и линии.

За всеки един момент от време потребителят на приложението може да получи разположението на всички видими сателити и тези, чрез които се изчисляват GPS данните, както и данните, които GPS приемника генерира в NMEA формат и в декодиран вид (виж Фиг. 2.3).



Фиг. 2.2 Изглед на главния прозорец на приложението



Фиг. 2.3 GPS данни и разположение на видимите сателити за избран момент от време (сателитите, използвани за определяне на позицията с запълнени с червено)

2.2.2.3 Оценка за нивото на шума в GPS данните

Приложението дава възможност за количествена оценка за нивото на шума в GPS данните във всеки момент от време. Оценката се получава чрез размита логика. Използват се две лингвистични променливи:

- MSNR - средна стойност за SNR на видимите сателити;
- SEDOP - параметър, който обединява информация за броя на видимите сателити и разположението им в пространството (елевация и азимут).

Исходната лингвистична променлива *OUT* дава оценка за шума в GPS данните. За получаването ѝ се използват 12 размити правила. След деразмиване *OUT* приема стойности в интервала [0, 100]: 0-минимално ниво на шума, 100-максимално ниво на шума. На Фиг. 2.4 е показан част от изходния код на функция *gpsFuzzy* чрез която се получава оценка за шума (*res*) на базата на моментните стойности на SEDOP и MSNR.

2.2.2.4 Филтриране на следа на маршрут

Маршрута се описва с точки, които отговарят на позицията на клиента в даден момент от време. За да се намали броя на точките, с цел по-компактно описание на маршрута, приложението позволява да се използват два алгоритъма от клас *track simplification*: алгоритъм на Douglas-Peucker ([Hershberger и Snoeyink, 1992](#)), който е най-добрият за момента алгоритъм за off-line редуциране на броя на точките, описващи маршрут, и предлагания on-line алгоритъм RSTS който ще бъде описан в секция 2.3. Алгоритъм RSTS използва локални техники за редуциране на броя на точките и предварително филтриране на позициите на клиента чрез адаптивен филтър на Калман. Намаляването на броя на точките, които описват маршрута, при запазване на неговата форма, е над 90% при всички режими на придвижване. Размерът на изходният файл със следата на маршрута се намалява средно 30 пъти.

2.2.2.5 Получаване на режима на придвижване

За много от LBS и LBSN услугите е важен текущият режим на придвижване на клиента (пеша, с колело, с автобус или кола). Приложението класифицира придвижването на клиента като „пеша“ или „с превозно средство“. За целта се използва алгоритъм, който анализира моментната скорост на придвижване и нейните филтрирани стойности.

```

function res = gpsFuzzy(x1, x2)

% ----- Init variables
S=1; M=2; H=3; L=1; ML=2; MB=3; B=4; lCenter=0; mlCenter=30; mbCenter=60;
bCenter=90;

% ----- Fuzzificate SEDOP
y1=[]; y1(3)=0;

if (x1 < 9), y1(S) = 1.0;
elseif (x1 < 18), y1(M) = (x1-9)/9; y1(S) = 1-((x1-9)/9);
elseif (x1<27), y1(H) = (x1-18)/9; y1(M) = 1-((x1-18)/9);
else y1(H) = 1.0;
end

% ----- Fuzzificate MSNR
...
% ----- IF-THEN rules:
outL=[]; outML=[]; outMB=[]; outB=[]; lIndex=1; mlIndex=1; mbIndex=1; bIndex=1;
% ----- Rule 1
if (y1(S)~=0 & y2(L)~=0),
    outB(bIndex) = min(y1(S),y2(L));
    bIndex = bIndex + 1;
end
...% ----- Rule 12
if (y1(H)~=0 & y2(B)~=0),
    outL(lIndex) = min(y1(H),y2(B));
    lIndex = lIndex + 1;
end

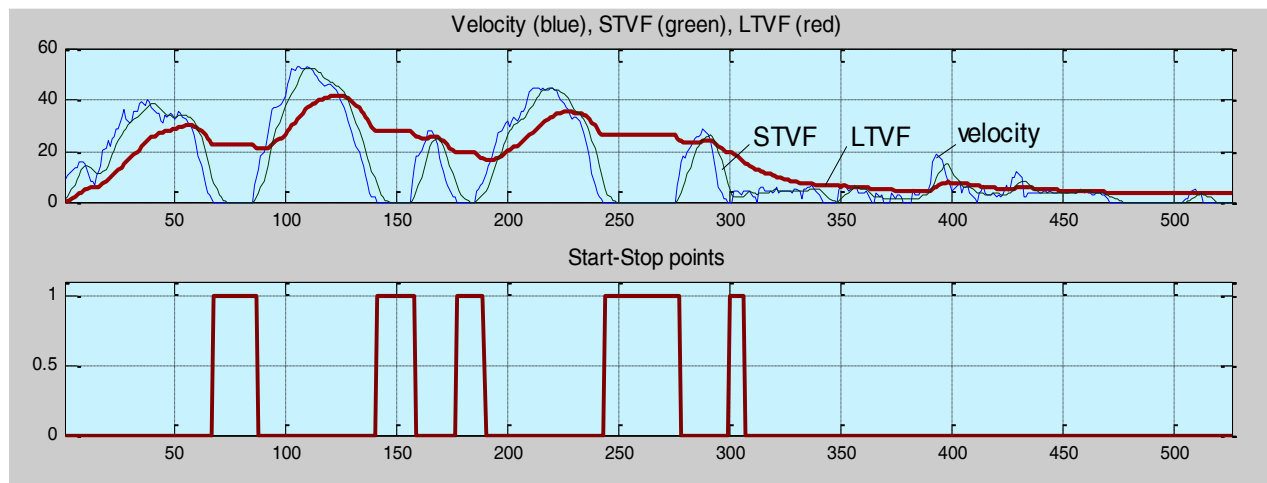
% ----- Defuzzification
lRSS=0; mlRSS=0; mbRSS=0; bRSS=0;
% ----- outL
for(i=1:length(outL)),
    lRSS = lRSS + outL(i)*outL(i);
end
lRSS = sqrt(lRSS);
% ----- outML
for(i=1:length(outML)),
    mlRSS = mlRSS + outML(i)*outML(i);
end
mlRSS = sqrt(mlRSS);
% ----- outMB
for(i=1:length(outMB)),
    mbRSS = mbRSS + outMB(i)*outMB(i);
end
mbRSS = sqrt(mbRSS);
% ----- outB
for(i=1:length(outB)),
    bRSS = bRSS + outB(i)*outB(i);
end
bRSS = sqrt(bRSS);
% ----- Get result (centroid)
res = (lCenter*lRSS+mlCenter*mlRSS+mbCenter*mbRSS+bCenter*bRSS) / ...
    (lRSS+mlRSS+mbRSS+bRSS);

```

Фиг. 2.4 Получаване на размита оценка за нивото на шума в GPS данните

2.2.2.6 Получаване на start/stop точките от маршрута

За някои LBS услуги е важно да се знаят точките от маршрута в които клиентът спира и отново тръгва (start/stop точки). Чрез тези точки, ако придвижването е пеша, може да се направи оценка какво представлява интерес за клиента в зависимост от това колко дълго се е задържал в дадена точка. При други услуги определянето на моментите в които клиентът тръгва или спира е важно, тъй като за тях грешката в GPS данните е максимална и се предполага, че те трябва да бъдат или пропуснати, или силно филтрирани. При конкретното приложение start/stop точките се откриват с вероятност не по-малка от 95% на базата на анализ моментната скорост, кратковременно филтрираната скорост - Short Term Velocity Filtering (STVF) и дълговременно филтрираната скорост - Long Term Velocity Filtering (LTVF). На Фиг. 2.5 е показан резултатът, който алгоритъмът генерира, при маршрут, съдържащ придвижване с автобус и пеша.



Фиг. 2.5 Получаване на start/stop точките на маршрут

2.2.2.7 Въвеждане на референтна следа на маршрут

Приложението позволява въвеждане на референтна (еталонна) следа на маршрут с цел количествена оценка на алгоритми за редуциране на броя на точките, описващи реален маршрут. Въвеждането на следата се реализира чрез мишката, като се описва чрез последователност от точки. За целта е необходимо да се разполага с GPS карта. След въвеждане на следата тя може да се експортира в GPX или CSV формат.

2.2.2.8 Сравняване на следи на маршрути

Приложението позволява получаване на количествена оценка при сравнение на два маршрута. За целта се получава RMS оценка за сумата на перпендикулярните

разстояния от всяка точка на референтния маршрут до следата на реалния маршрут.

2.2.2.9 Други Възможности

Предвидена е възможност за изчисляване на GPS разстояние между две точки, въведени чрез мишката, най-краткото разстояние от точка до следа на маршрут и посока на придвижване (при 3 въведени точки) в градуси спрямо посока север. Възможно е търсене на точка от маршрута на клиента, която отговаря на определени изисквания (време, стойност на даден GPS параметър) и др.

2.2.3 Заключение

Разработен е Matlab инструментариум за off-line обработка на данни от GPS приемници. Целта на този инструментариум е бързо разработване и експериментирание на алгоритми за LBS приложения на базата визуален и математически анализ на първични GPS данни.

Основните предимства на предложения инструментариум са следните:

1. Платформена независимост на кода на приложението. Независимост от типа на GPS приемника. Отворен код, който позволява развитие.
2. Визуален анализ на всички GPS данни. Търсене на GPS данни, които отговарят за конкретна точка от маршрута. Търсене на точки от маршрута, които отговарят на зададени стойности на GPS данните. Анализ на значението на моментното геометричното положение на видимите сателити (азимут и елевация) за качеството на GPS данните.
3. Математически анализ на GPS данните: филтриране на GPS данни; получаване на количествена оценка за моментното ниво на шума в GPS данните на базата на размита логика; филтриране на следата на маршрут чрез експертни правила и адаптивен филтър на Калман; вградени алгоритми за on-line и off-line намаляване на броя на точките чрез които се описва следата на маршрута; количествена оценка за качеството на алгоритми за редуциране на броя на точките на базата на RMS оценка за сумата на перпендикулярните разстояния от всяка точка на референтна следа до следата на реалния маршрут.
4. Автоматично генериране на отчет за всички получени резултати по време на използване на приложението.

2.3 Алгоритъм RSTS за опростяване на GPS траектории в реално време

2.3.1 Въведение

Повечето от съществуващите GPS навигационни системи за хора с увреждания на зрението и слепота използват цифрови търговски GPS карти от трети страни или GIS. Те включват Wayfinder Access (Wayfinder Systems AB), Trekker и BrailleNote GPS (Human Ware), MobileGeo (Code Factory) и Drishti (Университет на Флорида).

Такива системи са неподходящи за региони, в които няма GPS карти или картите не са достатъчно точни. Това е типично за повечето малки градове и региони извън градовете. Основните слабости на съществуващите GPS навигационни системи за слепи са следните: 1) Системите са с твърде висока цена и не съответстват на покупателната способност на хората със зрителни увреждания. Повечето навигационни системи изискват специализиран хардуер и/или мобилни телефони като смартфони, екранни четци като Talks и Mobile Speak, достъп до GIS и GPS карти, които трябва да се актуализират периодично. 2) Навигацията по маршрут, ако се поддържа, в повечето случаи не е адаптирана за използване от хора със зрителни увреждания.

Когато липсва GPS карта или тя не е достатъчно точна или подробна, единствената алтернатива е навигацията по маршрут. Повечето GPS навигатори и мобилни GPS навигационни системи позволяват проследяване на маршрути. Проблемът е, че много малко от тях позволяват навигация по маршрут. Причината е трудността при определянето на точките, в които има реална промяна на посоката (критични точки), главно поради грешки в GPS данните. Грешното идентифициране на критичните точки е фатално за навигацията на хората с увреждания на зрението.

Друг основен проблем при получаването на проследяване на маршрута е голямото количество памет, необходимо за записване на трасето, особено при ходене. Това налага ограничение на броя на запазените трасета и максималния брой точки, с които те се описват. Например, GPS навигаторите на Garmin за пешеходци, които не поддържат GPS карти, могат да запамятват 10 или 20 траектории, като всяка траектория може да съдържа до 10 000 точки. Решението на този проблем е да се използват алгоритми за опростяване на траекторията в реално време.

Тъй като всяка траектория се описва чрез поредица от точки, може да се приеме, че алгоритмите за опростяване на траекториите са част от алгоритмите за опростяване на линии. Те могат да бъдат класифицирани в пет категории: 1)

Алгоритми за независими точки; 2) Алгоритми за локална обработка; 3) Алгоритми за неограничена разширена локална обработка; 4) Алгоритми за ограничена разширена локална обработка и 5) Алгоритми за глобална обработка. Първите четири типа алгоритми принадлежат към алгоритмите за локално опростяване на линии, а последният – към алгоритмите за глобално опростяване на линии (Shi и Cheung, 2006).

Алгоритми за локално опростяване: При тези алгоритми се анализира връзката между всеки две или три последователни точки от маршрута. Тези видове алгоритми са много прости и бързи, но поради локалната обработка на данни е много трудно да се получат оптимални резултати. По-често използваните алгоритми за локално опростяване на траектории са: n -та точка – запазва се само една точка между n последователни точки; алгоритми, базирани на праг на разстояние – всички точки, за които разстоянието до предходната точка на траекторията е по-малко от предварително определения праг, се изтриват; алгоритми, базирани на промени в посоката – точката се запазва, ако промяната в посоката е по-голяма от предварително определения праг; и алгоритми, базирани на перпендикулярно разстояние – точката се запазва, ако разстоянието от точката до сегмента, образуван от предходната и следващата точка, е по-голямо от предварително определения праг. Към локалните алгоритми принадлежат и алгоритми, които анализират участъка от последователни точки, например: алгоритъм на Reumann-Witkam, алгоритъм на Opheim и алгоритъм на Lang. Целта е да се постигне по-добра производителност въз основа на анализ на по-голям брой точки от траекторията.

Глобални алгоритми за опростяване: В тази категория алгоритми решението коя точка да се изтрие и коя да се запази се основава на взаимоотношенията между всички точки, които принадлежат към траекторията. Глобалните алгоритми извличат критичните точки по-точно от локалните алгоритми, но изискват повече памет и са много по-бавни. Това ги прави неподходящи, когато са необходими алгоритми за опростяване на траекторията в реално време, които се изпълняват на платформи с ограничени ресурси. В момента, главно в картографията и GIS, се използват два глобални алгоритъма и техните модификации: алгоритъмът на Дъглас-Пюкер (DP) (Douglas и Peucker, 1973; Hershberger и Snoeyink, 1992) и алгоритъмът на Висвалингам-Уайат (Visvalingam и Whyatt, 1993). Експериментите показват (Cheung и Shi, 2006), че алгоритъмът DP дава най-точно опростяване на траекторията с минимални деформации на формата. Основен недостатък на алгоритъма DP е времето за изчисление. Средната сложност на алгоритъма е $O(n^2 * \dots)$

$\log m$), а в най-добрия случай е $O(n * \log n * \log m)$, където n е общият брой точки, а m е броят запазени точки.

Повечето мобилни приложения, които позволяват записване на траектории (WeTravel, gpsVP, Odgps, AFtrack, GPSTrack, GPSTrack, GPSTrack, Mobile Trail Explorer) и алгоритми за опростяване на траекториите в реално време (Barbeau и колектив, 2008; Chen, и колектив, 2009; Zhung и колектив, 2008), използват локални алгоритми, базирани главно на промяната на курса и разстоянието между съседните точки. Тези алгоритми могат да намерят практическо приложение само ако няма шум в GPS данните и следователно не могат да се използват за навигация на хора с увредено зрение.

Предлага се алгоритъм за опростяване на траекторията, който работи в реално време и чиято точност е сравнима с точността на DP. Алгоритъмът използва комбинация от няколко локални техники за опростяване: промени в посоката, перпендикулярно разстояние и праг на разстоянието. За да се намали чувствителността на алгоритъма към грешки в GPS данните, се предлага алгоритъм за адаптивно филтриране на GPS позициите. Той комбинира предварителна обработка на GPS данни, базирана на знания, оценяване на шума в GPS данните и филтриране чрез адаптивен филтър на Калман. Тъй като алгоритъмът е адаптивен към скоростта, той може да се използва не само за разработване на приложения за навигация за хора с увредено зрение, но и в GPS навигатори, приложения за услуги, базирани на местоположението (LBS), приложения за социални мрежи, базирани на местоположението (LBSN), и услуги, базирани на GIS.

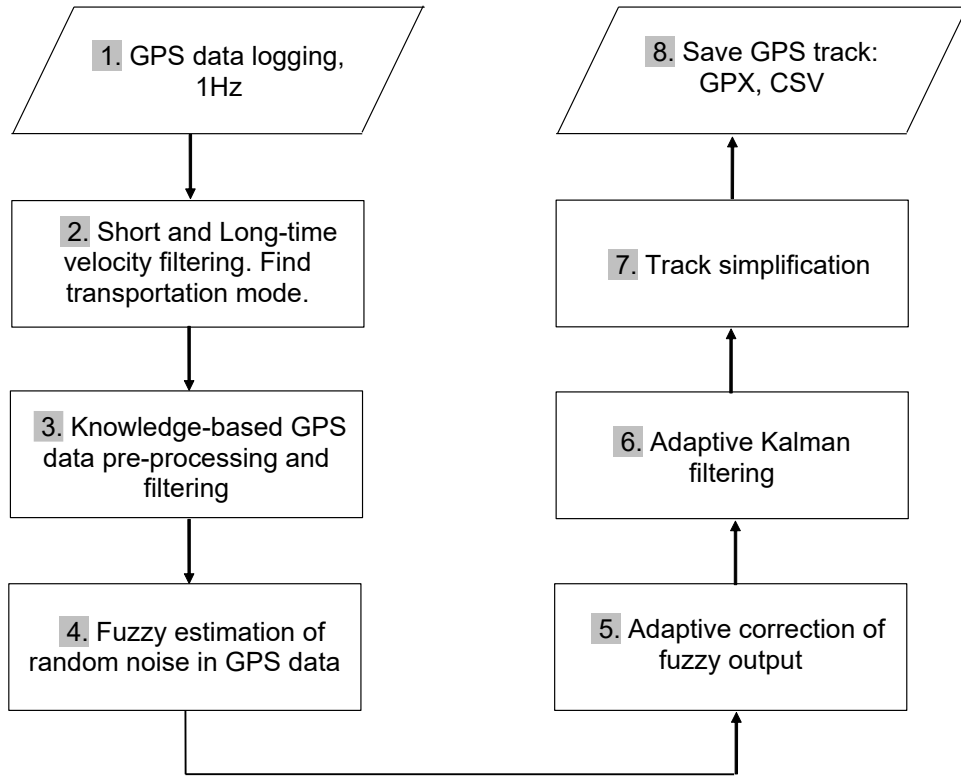
2.3.2 Архитектура на алгоритъма за опростяване на GPS траекторията

Фиг. 2.6 показва последователността на изпълнение на предложени алгоритъм под формата на диаграма на процеса.

По-долу са описани етапите на изпълнение на алгоритъма:

2.3.2.1. Записване на GPS данни

Етапът се изпълнява от NMEA 0183 парсер, който анализира данните от GPS приемника (NMEA команди GPGLA, GPGLA, GPGLV и GPRMC) и извлича необходимите за предложени алгоритъм GPS данни: брой видими сателити; позиция на сателитите - азимут и височина; SNR за всеки сателит; хоризонтално отклонение на точността (HDOP) – оценка на хоризонталната точност на GPS приемника; позиция на потребителя WGS 84 (дължина, ширина) и скорост.



Фиг. 2.6 Диаграма на процеса на алгоритъма за опростяване на траекторията

2.3.2.2. Филтриране на скоростта. Намиране на транспортния режим

Филтрирането на скоростта се използва за определяне на транспортния режим – „ходене“ или „не ходене“ и в други етапи на предложения алгоритъм. За определяне на транспортния режим се използва филтър от първи ред с обратна импулсна характеристика (IIR) – филтриране на скоростта за кратко време (STVF):

$$STVF_i = \alpha STVF_{i-1} + (1 - \alpha)v_i, \text{ ако } v_i > 0,$$

$$STVF_i = STVF_{i-1} \text{ ако } v_i = 0. \quad (2.1)$$

където α е параметър, който контролира степента на филтриране, а v_i е скоростта в дискретно време i . Приема се, че начинът на придвижване е „ходене“, ако $STVF < 7\text{km/h}$.

Дългосрочното филтриране на скоростта (LTVF) се изчислява като средната скорост за интервала от последните M дискретни (секунди):

$$LTVF_i = \frac{1}{M} \left(\sum_{j=i-M+1}^i v_j \right). \quad (2.2)$$

2.3.2.3. Предварителна обработка и филтриране на GPS данни на базата на знания

Задачата на този етап е да се филтрират грешките в GPS данните, които не могат да бъдат отстранени на етапа „Филтриране чрез адаптивен алгоритъм на Калман“. Това се осъществява на базата на знания и опит за суровите GPS данни под формата на предварително определени правила.

Експериментите показват, че най-големите грешки в GPS данните са възниквали при следните събития: преход от невалидни към валидни GPS данни, промени в броя на видимите сателити, преходи „движение-спиране-движение“, ходене с много ниска скорост (по-малко от 3 км/ч), което е типично за слепите хора, и когато е активна статичната GPS навигация.

Правило 1: Брой видими сателити и точност на GPS приемника.

За оценка на точността на GPS позициите се използва параметърът HDOP. Колкото по-малка е неговата стойност, толкова по-добра е точността. Тази стойност зависи от отношението между позициите на видимите сателити и GPS приемника. Параметърът HDOP не взема предвид качеството на сигнала на отделните сателити. На практика се оказва, че дори при високи стойности на HDOP, позицията може да бъде достатъчно точна, ако броят на видимите сателити (N) е достатъчно голям. Правило 1 определя кои GPS данни да се използват за по-нататъшна обработка:

$$N > 5 \text{ OR } (N > 3 \text{ AND HDOP} < 3.25)$$

Правило 2: Промени в валидността на GPS данните.

Анализът на суровите GPS данни показва, че грешката в позицията е най-голяма при прехода от невалидни (без GPS фиксиране) към валидни (3D GPS фиксиране) данни за $N > 3$. Отклонението от действителната позиция може да бъде над 50 м. След известно време тази грешка се минимизира. Правило 2 минимизира този тип грешка, както следва:

Пропуснете следващите 5 GPS позиции И филтрирайте максимално следващите 5 GPS позиции

Правило 3: Промени в броя на видимите спътници.

Промяната в броя на видимите спътници води до изчисляване на позицията въз основа на данни, включващи различни комбинации от спътници. Поради това точността се подобрява или влошава. В резултат на това отклонението от предишната позиция е в рамките на 10-30 м. Това може да доведе до въвеждането на излишни точки. Правило 3 трябва да премахне повечето от тези точки:

Пропуснете GPS позициите, ако $|N_i - N_{i-1}| > 2$ И филтрирай максимално следващите 3 GPS позиции

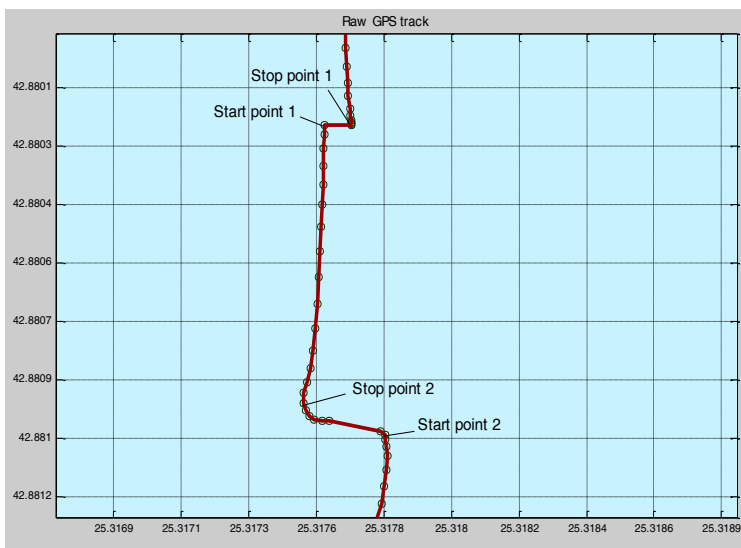
Правило 4: Статус на статичната навигация.

Повечето GPS приемници не са проектирани специално за пешеходци и статичната навигация е активирана по подразбиране. В този режим GPS данните се филтрират от фирмения софтуер на GPS приемника и когато скоростта е по-ниска от предварително зададена прагова стойност, GPS приемникът започва да генерира последното изчислено местоположение. Когато потребителят върви с ниска скорост и често спира, за някои участъци от пътя (10-40 м) информацията за позицията се пропуска. Предлаганият алгоритъм за филтриране на GPS позициите се основава на факта, че GPS приемникът генерира данни в един и същ интервал от време (една секунда). Това може да доведе до големи грешки в региони, съдържащи плавна крива. Правило 4 е отговорно за елиминирането на тези грешки:

Деактивирайте статичната навигация ИЛИ филтрирайте минимално следващите 3 GPS позиции

Правило 5: Преходни зони „движение-спиране-движение“

За тези зони грешките в GPS позицията са големи, независимо от вида транспорт. Най-голямата грешка е 10-40 м, ако видът транспорт е „не ходене“ и статичната навигация е активна. Фиг. 2.7 показва грешката в позицията в две зони от трасето, съдържащи преходи от типа „движение-спиране-движение“.



Фиг. 2.7 Грешки в GPS позиционирането в региони „движение-спиране-движение“: отклонение от 6,6 м за регион 1 и отклонение от 23,1 м за регион 2

Правило 5 има за цел да премахне GPS позициите в преходните региони:

Пропуснете всички точки между началната и крайната точка И филтрирайте максимално следващите 3 GPS позиции.

За тази цел се търсят началната и крайната точка на тези области.

Получаване на начална точка: За интервала на M дискретни точки се изчислява броят на точките (*counter*), за които е изпълнено следното условие:

$$LTVF_{i-j} < LTVF_{i-j-1}, j = 0 \div M - 1. \quad (2.3)$$

Ако $counter = M$, се приема, че точка i е начална точка на региона.

Получете точка на спиране: За да бъде взета под внимание точката на спиране, тя трябва да отговаря на следните условия:

$$STVF_i > Th_1 \text{ AND } (LTVF_i < Th_2/3 \text{ OR } v_i = 0), \quad (2.4)$$

където праговете са както следва:

$$Th_1 = 15 \text{ km/h}, \quad Th_2 = \min(Th_1, 0.7 * Th_3), \quad Th_3 = \frac{1}{M/2} \sum_{j=0}^{(M/2)-1} LTVF_{i-j}. \quad (2.5)$$

2.3.2.4. Оценяване на случайния шум в GPS данните

За правилното функциониране на етапа „Филтриране чрез адаптивен филтър на Калман“ е необходимо точно оценяване на грешките в измерванията на GPS данните. Колкото по-големи са тези грешки, толкова по-силно трябва да бъде филтрирането на данните и обратно – ако грешката е малка, филтрирането трябва да бъде минимално. Грешките в GPS данните са от два вида ([Ogle и колектив, 2002](#)): 1) Систематични грешки и 2) Случайни грешки.

Систематичните грешки се дължат главно на малкия брой видими сателити, високата стойност на Position Delution of Precision (PDOP) и разположението на антената на GPS приемника. Систематичните грешки се идентифицират и отстраняват лесно.

Случайните грешки са резултат от промяна в орбитите на спътниците, грешки в часовниците на спътниците, ефекти в тропосферата и йоносферата и многократно отражение на сигнала. Поради своята същност случайните грешки са трудни за идентифициране и филтриране. Табл. 2.1 показва средното отклонение в позицията за различни видове случайни грешки ([Parkinson и Spilker, 1996](#)).

Табл. 2.1 Случайни грешки в GPS данните

Тип на грешката	Средна грешка, m	Времева зависимост
Йоносферни ефекти	± 5.0	Повече от 1 час
Промяна в орбитата на спътника	± 2.5	-
Грешки в сателитните часовници	$\pm (1.5-2.0)$	-
Тропосферни ефекти	± 0.5	Повече от 1 час
Ефекти от многократно отражение	± 0.5	(0.5-10) минути

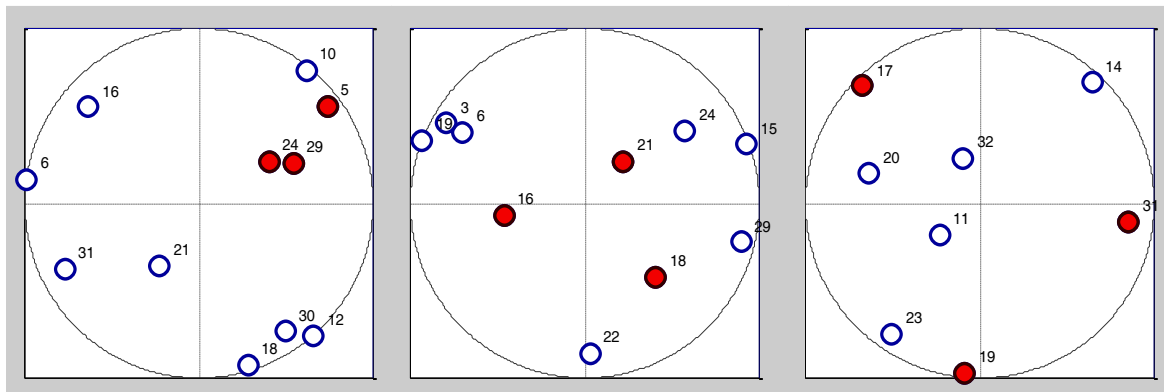
За да се сведе до минимум ефектът от случайните грешки, най-често се използват статистически техники за изглаждане ([Hastie и колектив, 2001](#)). Понастоящем най-добрата статистическа техника за свеждане до минимум на ефекта от случайните грешки в GPS данните е филтърът на Калман. Дискретният филтър на Калман приема като вход поток от шумови измервания на GPS приемника. Той използва тези измервания, за да оцени състоянието на приемника. Ако входната информация е поток от GPS позиции, филтърът на Калман може да предскаже точна информация за позицията. Основният проблем е точното оценяване на грешката в измерването, тъй като тя е случайна по своята същност.

Предлага се да се използва размита логика за оценяване на шума в GPS данните. В предложения алгоритъм се използват две входни размити променливи:

- Средна стойност на SNR (MSNR);
- Брой сателити, височина и азимут (NSEA).

Входната променлива MSNR се изчислява като средна стойност на SNR за всички видими сателити. Стойностите на MSNR се разделят на 4 сегмента: S (малък), MS (средно малък), MH (средно висок) и H (висок). Приема се, че колкото по-голяма е стойността на MSNR, толкова по-вероятно е шума в GPS данните да е нисък. Лингвистичната променлива NSEA отчита три параметъра: брой видими сателити, среден азимут и средна височина. По този начин броят на лингвистичните променливи на входа и времето за реализация на алгоритъма се намаляват. Точността на GPS приемниците е пропорционална на сателитната геометрия – относителните позиции на сателитите и GPS приемника. Оценката на тези позиции се дава от параметъра Delution of Precision (DOP). Стойността на DOP е

обратно пропорционална на обема на тетраедъра, който образуват спътниците и GPS приемникът. Колкото по-ниска е стойността на DOP, толкова по-точен е GPS приемникът. Грешката в GPS данните е по-малка, колкото по-голяма е площта на основата на тетраедъра и колкото по-ниска е височината (виж Фиг. 2.8).



HDOP=12.5, Mean elevation=46.3°,
Mean SNR=29.0dB

HDOP=2.8, Mean elevation=62.3°,
Mean SNR=26.0dB

HDOP=1.4, Mean elevation=20.7°,
Mean SNR=34.0dB

Фиг. 2.8 Изглед към видимите сателити: Връзка между HDOP, средна стойност на SNR и средна височина при различни позиции на 3 видими сателита

За да се получи по-голям обем на тетраедъра, е необходима ниска височина на спътника. Но грешките, дължащи се на процесите в тропосферата и йоносферата, са големи, ако височината е ниска. Например, ако височината на спътника е до 5°, грешката, която влияе на измерването, е около 3 до 10 пъти по-голяма от грешката, ако спътникът е в зенита.

Тъй като параметърът DOP не отчита влиянието на SNR и височината на спътниците, той няма да се използва директно. Стойността на NSEA се получава чрез даване на точки за всички параметри, които участват в изчислението му. Максималният брой точки за всеки параметър е 12. Стойностите на променливата NSEA са S (малка), M (средна) и H (висока). Колкото по-голяма е стойността на NSEA, толкова по-нисък е шума в GPS данните.

$$NSEA = N1 + N2 + N3, \quad (2.6)$$

където $Nj, j = 1 - 3$ е броят точки за параметър j .

Параметър 1: Брой видими сателити. Всеки сателит получава 1 точка; следователно $N1 = N$, където N е броят на видимите сателити.

Параметър 2: Положение на сателитите над хоризонта. Придаваме по-голямо значение на сателитите, които са на ниска височина:

$$N_2 = \frac{N_{\max}}{N} \sum_{j=1}^N \left(1 - \frac{\delta_j}{90} \right), \quad (2.7)$$

където δ_j е стойността на височината за спътника j , а $N_{\max}$ е максималният брой видими спътници. В дадена точка от Земята не могат да бъдат видими повече от 12 сателита, $N_{\max} = 12$.

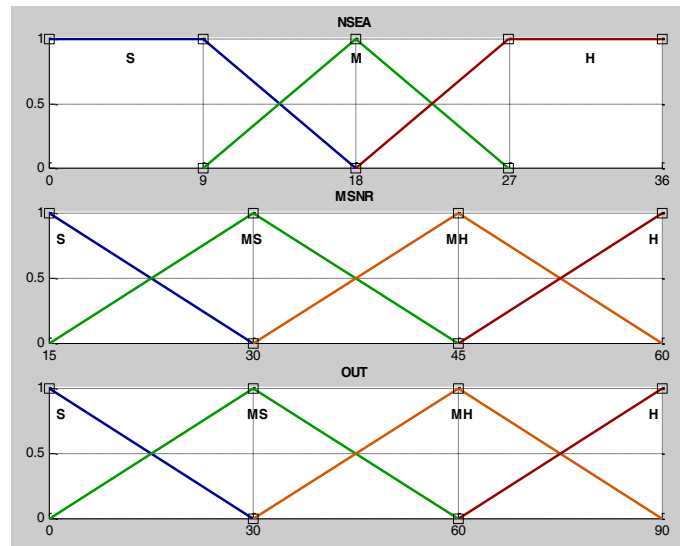
Параметър 3: Положение на спътниците според техния азимут. Предполага се, че в зависимост от стойността на азимута спътниците попадат в един от следните четири квадранта: $[0,90)$, $[90,180)$, $[180,270)$ и $[270,360)$. Придаваме по-голямо значение, ако спътниците са разположени равномерно в квадрантите:

$$N_3 = 3 \sum_{j=1}^4 \left[\min(n_j, N_m) / \max(n_j, N_m) \right], \quad (2.8)$$

където n_j е броят на сателитите в квадрант j , $N_m = N/4$.

Размитата изходна променлива *OUT* дава приблизителна оценка за грешката в GPS данните. Възможните стойности за тази променлива са S (малка), MS (средно малка), МН (средно висока) и Н (висока). Колкото по-голяма е стойността на *OUT*, толкова по-силно трябва да се филтрират GPS позициите, тъй като шума в тях е по-голям.

Фиг. 2.9 показва функциите на принадлежност на входните и изходните лингвистични променливи.



Фиг. 2.9 Функции за принадлежност

За да се получи лингвистичната променлива *OUT*, се използват 12 размити правила от следния тип:

Ако $NSEA = value$ И $MSNR = value$ то $OUT = value$.

Фиг. 2.10 показва използваната матрица с размити правила.

OUT		MSNR			
		S	MS	MH	H
NSEA	S	H	MH	MH	ML
	M	H	MH	ML	L
	H	MH	ML	L	L

Фиг. 2.10 Матрица на размити правила

За да се изведат логически продукти за всяко правило, се използва методът Root-Sum-Square (RSS). Деразмиването на данните в ясен резултат се осъществява чрез комбиниране на резултатите от процеса на извеждане и изчисляване на размития център на площта.

2.3.2.5. Корекция на нивото на грешката, получено чрез размита логика

Когато транспортният режим е „не пеша“, е необходимо коригиране на реакцията на неточната система, тъй като е възможно прекомерно изглаждане на кривите. Коригирането се активира, когато стойността на STVF е над 50 km/h:

$$OUT_i = OUT_i(1 - reduction), \quad (9a)$$

където

$$reduction = 0.9 \text{ if } STVF_i > 150 \text{ km/h}$$

$$reduction = [10 + 0.8(STVF_i - 50)]/100 \text{ if } STVF_i > 50 \text{ km/h} . \quad (2.9b)$$

2.3.2.6 Адаптивно филтриране по Калман

Филтърът на Калман е оптимален метод за оценка, който се използва широко в GPS навигационните системи (Mohamed и Schwarz, 1999; Kelly, 1994). При GPS навигацията нивото на шума при измерването зависи от околната среда и движението на потребителя. В този случай филтърът на Калман не е оптимален и може да доведе до ненадеждни резултати, например прекомерно изглаждане на кривите. Едно от решенията на проблема е адаптивното филтриране на Калман (AKF). Повечето съществуващи техники за онлайн идентифициране на матрицата на ковариацията на шума (Congwei, и колектив, 2003; Dah-Jing Jwo и Chung-Min

Huang, 2007) не могат да бъдат реализирани в реално време, когато приложението се стартира на мобилни устройства.

Предложеният АКФ алгоритъм се използва за филтриране само на позициите на потребителя. АКФ изглажда позициите чрез рекурсивно модифициране на стойностите на грешките в измерванията. Оценката на шума при измерването се получава в етап 4, „Неясно оценяване на случайния шум в GPS данните“. Алгоритъмът на филтъра на Калман може да бъде описан като двуетапен процес: етапът на прогнозиране (актуализиране на времето), който използва научената динамика на системата, за да предвиди следващата стъпка на системата, и етапът на корекция (актуализиране на измерването), който взема ново наблюдавано измерване и го използва, за да коригира динамиката и състоянието на системата.

Дискретните уравнения на филтъра на Калман, които описват стъпката на прогнозиране, са:

$$\begin{aligned}\hat{x}_k^- &= A\hat{x}_{k-1} + Bu_k \\ P_k^- &= P_{k-1}A^T + Q,\end{aligned}\tag{2.10}$$

където k е дискретно време, $\hat{x}_k^-$ е априорна оценка на състоянието, $\hat{x}_k$ е апостериорна оценка на състоянието, P_k^- е априорна оценка на ковариацията на грешката, P_k е апостериорна оценка на ковариацията на грешката, A е матрица на времевия преход, u_k е допълнителен известен входен параметър, а Q е ковариация на шума в процеса, $p(Q) \sim N(0, q)$.

Дискретните уравнения на филтъра на Калман, които описват стъпката на корекция, са:

$$\begin{aligned}K_k &= P_k^- H^T (HP_k^- H^T + R)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-) \\ P_k &= (I - K_k H)P_k^-\end{aligned}\tag{2.11}$$

където K_k е матрицата на Калман, H е матрицата на времевия преход за процеса на наблюдение, z_k са наблюдаваните данни (действителните измервания), а R е ковариацията на шума при измерването, $p(R) \sim N(0, r)$.

Тъй като се филтрира само позицията (дължина, ширина), u_k в (2.10) става нула (едномерен филтър на Калман). Данните от GPS се анализират за една секунда и следователно матриците на времевия преход A и H са 1 секунда. В този случай шума

на процеса е същият като шума на измерването. По този начин уравненията на филтъра на Калман се редуцират до следната форма:

$$\begin{aligned}\hat{x}_k^- &= \hat{x}_{k-1} \\ P_k^- &= P_{k-1} + q\end{aligned}\tag{2.12}$$

$$\begin{aligned}K_k &= P_k^- / (P_k^- + r) \\ \hat{x}_k &= \hat{x}_k^- + K_k (z_k - \hat{x}_k^-)\end{aligned}\tag{2.13}$$

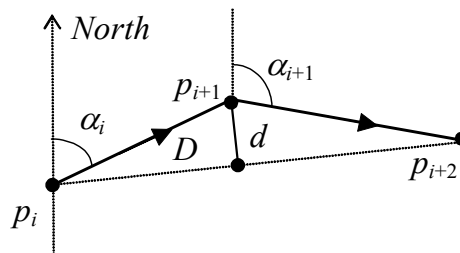
$$P_k = (1 - K_k) P_k^- .$$

При прилагането на алгоритъма се приема, че вариацията на шума в процеса q е постоянна, а вариацията на шума в измерването r е стойността на параметъра *OUT* от размитата система. Филтрираната позиция се получава въз основа на оценка на разликата между измерената позиция z_k и априорната оценка на състоянието (прогнозиран остатък).

2.3.2.7 Опростяване на траекторията

Благодарение на филтрирането на позицията с АКФ е възможно да се използва локален алгоритъм за опростяване на траекторията. Това осигурява реализация в реално време, като същевременно се запазва възможността за извличане на необходимите за навигационния процес критични точки на траекторията.

Алгоритъмът се основава на анализ на връзката между всеки 3 последователни точки: последната точка, принадлежаща към траекторията p_i , и следващите две точки p_{i+1} и p_{i+2} (виж Фиг. 2.11). Основната задача на алгоритъма е да реши дали точка p_{i+1} принадлежи към траекторията или не. Това се реализира въз основа на три параметъра: разстоянието D между последната точка на траекторията и точка p_{i+1} ; промяната в посоката α , $\alpha = |\alpha_i - \alpha_{i+1}|$, и разстоянието d от точка p_{i+1} до сегмента, който образуват точки p_i и p_{i+2} .



Фиг. 2.11 Параметри на алгоритъма за опростяване на траекторията

Изборът на следващата точка по траекторията се извършва въз основа на три правила:

Правило 1: Избор на кандидат-точка. Предполага се, че p_{i+1} е кандидат-точка, ако разстоянието от последната въведена точка p_i до точка p_{i+1} е по-голямо от D , където:

$$D = 15 * HDOP_{i+1} ,$$

$$D_{\max} = 20 \text{ ако придвижване е „пеша“} . \quad (2.14)$$

Стойността на D зависи от текущата точност на GPS приемниците. По този начин е възможно да се филтрират точки, получени при ниска точност на GPS.

Правило 2: Анализ на промяната в посоката на движенията. Анализиранията точка принадлежи към траектория, ако е изпълнено следното условие:

$$\alpha > \alpha_{Th1} \text{ AND } \alpha < \alpha_{Th2} . \quad (2.15)$$

Стойността на параметъра α_{Th1} определя степента на редуциране на точките. При малки стойности на този параметър траекторията се описва по-точно, но с по-голям брой точки. Параметърът α_{Th2} има за задача да филтрира точките, получени в резултат на грешки в GPS данните при малки стойности на параметъра D .

Правило 3: Анализ на важността на параметъра d . Правило 2 може да пропусне критични точки за региони, съдържащи гладки криви. Правило 3 има за цел да предотврати това. Ако разстоянието d е над предварително определена прагова стойност, се приема, че p_{i+1} принадлежи към траекторията:

$$d > COFF * HDOP_{i+1} , \quad (2.16)$$

където параметърът $COFF$ зависи от скоростта:

$$COFF = 0.5 \text{ if } STVF_{i+1} < 30 \text{ km/h} ,$$

$$COFF = 1.7 \text{ if } STVF_{i+1} > 100 \text{ km/h}$$

$$COFF = 0.7 + (STVF_{i+1} - 30)/70 \text{ if } STVF_{i+1} \geq 30 \text{ km/h} . \quad (2.17)$$

Накрая, точка p_{i+1} принадлежи към траекторията, ако:

$$Rule_1 = true \text{ and } (Rule_2 = true \text{ or } Rule_3 = true) . \quad (2.18)$$

2.3.3 Експериментални резултати

Разработеният алгоритъм се използва в мобилно приложението за навигация по маршрут за хора с увредено зрение и слепи. Това приложение не изисква GPS карти и достъп до уеб сървър и може да работи както при наличие, така и при отсъствие на DGPS (WAAS, EGNOS, MSAS).

2.3.3.1 Настройка

За да се анализират характеристиките на предложения алгоритъм за 6 месеца, са записани 200 траектории – по 50 записи за всеки от 4-те типа траектории. Параметрите на траекториите са описани в Табл. 2.2.

Табл. 2.2 Описание на GPS траекториите

Траек- тория	Тип придвижване	Оптимален брой точки	Средна дължина на траекторията, m	Среден HDOP интервал
1	пеша	13	1099	0.9 – 7.8
2	пеша	16	1267	0.9 – 5.6
3	автобус и пеша	9	1641	1.2 – 5.5
4	кола	112	13141	1.1 – 7.6

Оптималният брой точки, които описват маршрута, се получава ръчно с помощта на GPS картата. За запаметяване на траекториите се използва мобилно приложение, което записва GPS данни във формат NMEA 0183 на флаш диска на мобилния терминал с интервал от 1 секунда. GPS данните се анализират с помощта на специално разработен GPS инструментариум за Matlab (описан в Секция 2.2), който позволява: визуализация и филтриране на GPS данни, изглед към небето, получаване на оптималния брой точки, описване на траекторията, използване на GPS карти, програмно изпълнение на предложения алгоритъм за опростяване на траекторията, възможност за сравняване на резултатите, получени от алгоритмите RSTS и DP, GPS разстояние, GPS азимут и др.

Табл. 2.3 описва праговите стойности на параметрите, които се използват при прилагането на алгоритъма.

Табл. 2.3 Прагови стойности на параметрите

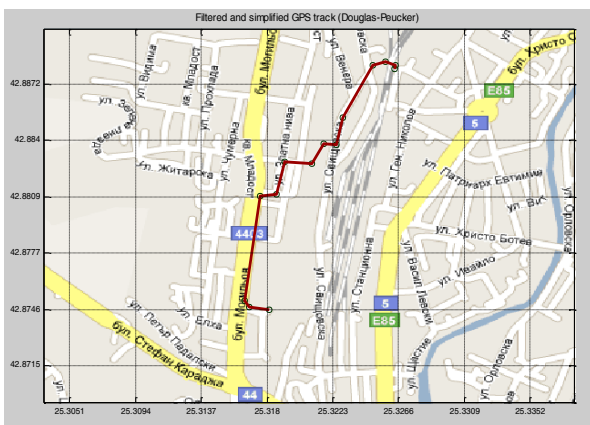
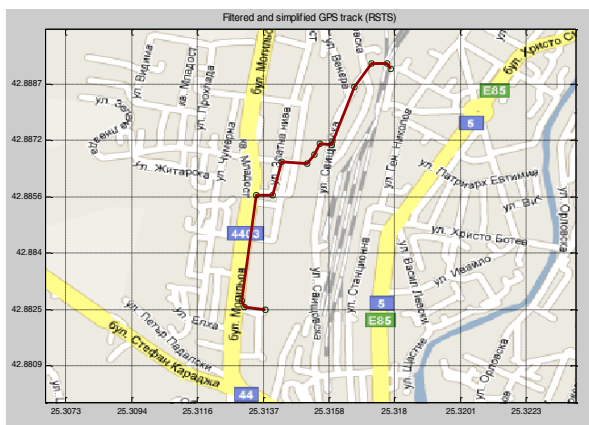
Етап на алгоритъма	Параметри
Краткосрочно и дългосрочно филтриране на скоростта. Откриване на транспортен режим.	$\alpha=0.95$
Предварителна обработка и филтриране на GPS данни на базата на знания	$M=8$
Опростяване на траекторията	$\alpha_{Th1}=25, \alpha_{Th2}=45$

2.3.3.2. Резултати

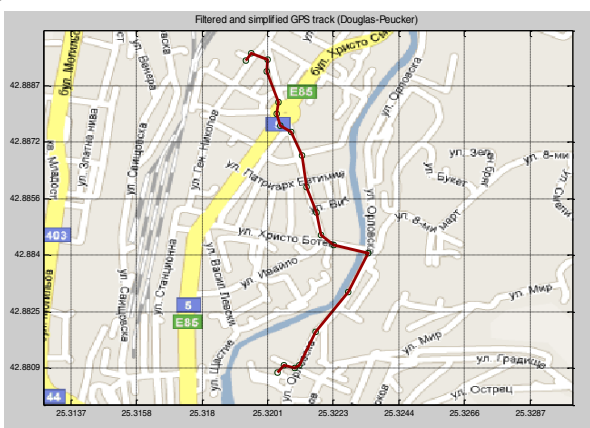
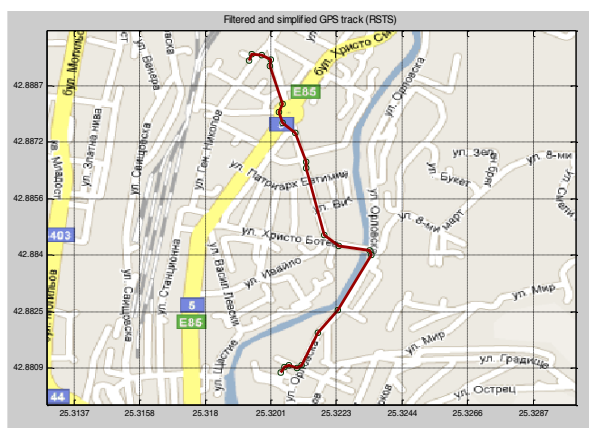
По-долу е описана важността на различните етапи на внедряване на предложения алгоритъм за неговите характеристики.

2.3.3.2.1 Тест на алгоритъма, базиран на визуално сравнение

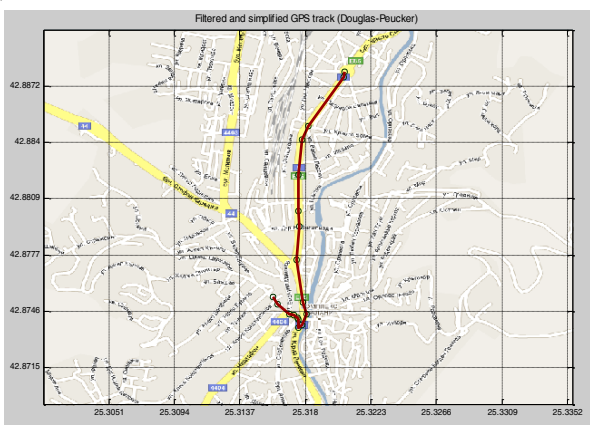
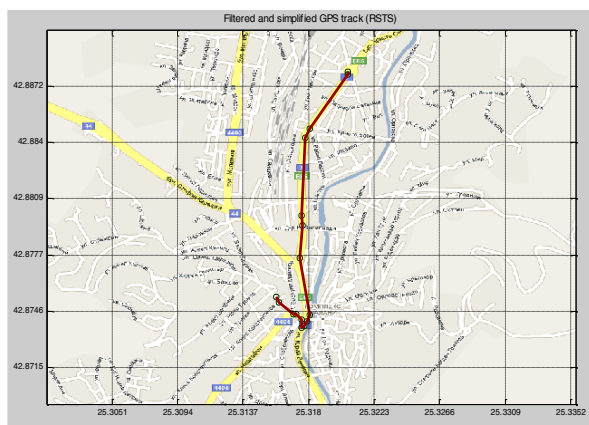
Алгоритмите за опростяване RSTS и DP първо се сравняват визуално. Получените резултати са показани на Фиг. 2.12. Установено е, че алгоритмите RSTS и DP осигуряват сходно представяне на формата на пистите и сходно намаляване на броя на точките.



а)



б)



в)

Фиг. 2.12 Визуално сравнение на следите: а) Траектория 1 (RSTS вляво, DP вдясно), б) Траектория 2 (RSTS вляво, DP вдясно), в) Траектория 3 (RSTS вляво, DP вдясно)

2.3.3.2.2 Предварителна обработка и филтриране на GPS данни на базата на знания

Този етап има за цел главно да филтрира системните грешки в GPS данните. Задачата не е да се премахне максимален брой точки, а само онези, които не могат да бъдат компенсирани в етапа „Адаптивно филтриране на Калман“. Получените резултати са показани в Табл. 2.4. Намалението на броя точки е от 3,4% до 6%.

Табл. 2.4 Резултати след филтриране на GPS данни въз основа на знания

Траектория	Сурови данни		Филтрирани данни	
	Брой точки	Дължина на траекторията, m	Брой точки	Дължина на траекторията, m
1	1234	1192	1230	1087
2	1056	1409	1019	1280
3	352	1804	331	1762
4	1395	14123	1348	13140

2.3.3.2.3. Размита оценка на случайния шум в GPS данните

Представен е анализ на важността на параметъра *OUT*, който дава оценка за шума в GPS. Табл. 2.5 показва отклонението от оптималния брой точки, когато неясното оценяване на шума в GPS данните е активирано и деактивирано.

Табл. 2.5 Значение на параметъра *OUT*

Траектория	Отклонение от оптималния брой точки, %	
	Fuzzy <i>OUT</i>	<i>OUT</i> =const
1	7.1	28
2	23.8	42.9
3	10	40
4	3.6	23.8

При използване на размита логика точността при определяне на броя на критичните точки се подобрява 4 пъти.

2.3.3.2.4. Адаптивно филтриране на Калман

Тъй като алгоритъмът за опростяване на траекторията RSTS е от локален тип, той е чувствителен към грешки в GPS позициите. Анализира се значението на етапа „Адаптивно филтриране на Калман“ за качеството на алгоритъма RSTS. За тази цел

се сравняват два параметъра, „брой критични точки“ и „позиционна грешка след опростяване на траекторията“ (RMS грешка), когато AKF е активен и неактивен. Резултатите са показани в Табл. 2.6.

Табл. 2.6 Значение на етапа „Адаптивно филтриране на Калман“

Траектория	Параметри	Стойности на параметрите			
		AKF=ON		AKF=OFF	
		RSTS	DP	RSTS	DP
1	Критични точки (намаление, %)	14 (98.87)	14 (98.87)	25 (97.97)	16 (98.70)
	Дължина на траекторията, m (грешка, m)	1067 (32)	1069 (30)	1141 (42)	1151 (52)
	RMS грешка, m	0.82	0.712	1.84	2.36
2	Критични точки (намаление, %)	21 (98.01)	20 (98.11)	31 (97.06)	25 (97.63)
	Дължина на траекторията, m (грешка, m)	1259 (8)	1258 (9)	1262 (5)	1341 (74)
	RMS грешка, m	0.958	0.962	0.97	0.99
3	Критични точки (намаление, %)	10 (97.16)	14 (96.02)	18 (94.89)	23 (93.47)
	Дължина на траекторията, m (грешка, m)	1647 (6)	1650 (9)	1675 (34)	1755 (114)
	RMS грешка, m	1.16	0.94	0.88	1.71
4	Критични точки (намаление, %)	108 (92.26)	104 (92.54)	246 (82.37)	132 (90.54)
	Дължина на траекторията, m (грешка, m)	13032 (109)	13080 (61)	13851 (710)	14012 (871)
	RMS грешка, m	1.10	1.15	1.26	1.41

Използваме мярка, базирана на разстоянието, за да получим позиционна грешка след опростяване на траекторията. Изчисляваме средноквадратичното отклонение (RMS) между оригиналната траектория и нейните опростени версии като сума от нормализирани перпендикулярни разстояния за всички точки, описващи оригиналната траектория.

Експериментите потвърждават, че алгоритъмът RSTS е по-чувствителен към шума в GPS позициите в сравнение с алгоритъма DP. Ако AKF е активен, резултатите, получени от алгоритмите RSTS и DP, са сравними. Броят на точките се намалява средно с 96%, без да се променя формата на траекториите.

2.3.3.2.5 Анализ на използваната флаш памет

Предложеният алгоритъм оптимизира използването както на DRAM, така и на флаш памет. Табл. 2.7 показва степента на намаляване на използваната флаш памет.

Табл. 2.7 Намаляване на флаш паметта

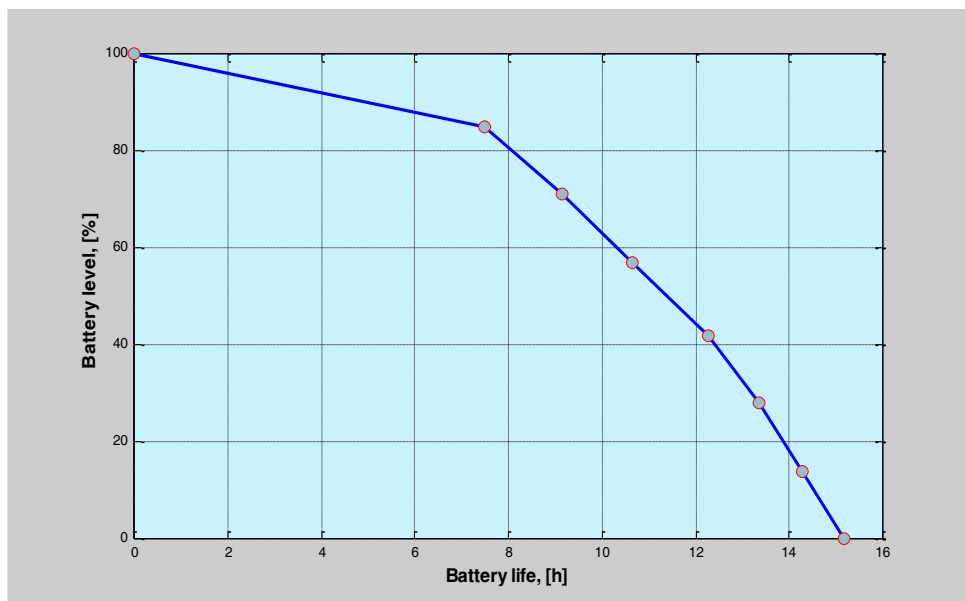
Траектория	Размер на файла с необработени данни, KB			Размер на файла с изходните данни, KB	
	NMEA	GPX	CSV	GPX	CSV
1	464	317	70	7	2
2	391	402	89	10	2
3	138	154	34	5	1
4	506	570	126	47	8

Флаш паметта се намалява 32,11 пъти, когато се използва формат GPX за входни и изходни файлове, и 32,31 пъти за формат CSV.

2.3.3.2.6 Анализ на живота на батерията

За мобилните приложения е важно колко дълго могат да се използват без презареждане на батерията. За да се удължи времето за проследяване, се предприемат следните мерки: Използва се външен GPS приемник с Bluetooth интерфейс; Аритметиката с плаваща запетая е сведена до минимум; Приложението работи на заден план; и интензивно използване на софтуерни нишки, които се активират само когато настъпят очаквани събития.

Фиг. 2.13 показва как се променя нивото на енергията на батерията с течение на времето. По време на теста за сравнителна оценка не се извършват входящи и изходящи повиквания. Нивото на енергията на батерията се измерва на интервали от 1 минута, докато батерията се разрежи напълно.



Фиг. 2.13 Сравнителен анализ на нивото на батерията

Тестът за живота на батерията показва, че когато се използва външен Bluetooth GPS приемник, времето за проследяване е до 15 часа.

2.3.4 Заключение

Разработен и тестван е алгоритъм за опростяване на GPS траектории, наречен RSTS. Алгоритъмът позволяват онлайн реализация на платформи с ограничени ресурси и са адаптивни към точността на GPS приемника, шума в GPS данните и начина на транспорт.

Алгоритъмът е анализиран използвайки 4 типа траектории, събрани в продължение на 6 месеца. Резултатите от теста показват, че алгоритъмът RSTS превъзхожда или е с подобна ефективност на алгоритъма на Дъглас-Пюкер, който се счита за най-добрия за опростяване на траектории.

Алгоритъмът е проектиран да намалява излишните точки по траекторията и да намира критични точки в процеса на навигация. Максималното време за проследяване е до 15 часа. Намаляването на броя точки, които описват траекторията във всички видове транспорт, е повече от 90%, а размерът на изходния файл е намален над 30 пъти.

ГЛАВА 3: МОДЕЛИРАНЕ НА СГРАДИ С ЦЕЛ ИНТЕРИОРНА НАВИГАЦИЯ

3.1 Въведение

В тази глава е представен информационен модел на сгради (BIM), който описва топологията, геометрията и семантиката на сградите; предпочитанията на потребителите и състоянието на сензорите, разположени в сградата. Целта е да се предложи иновативен подход за разработване на система за навигация на закрито с отчитане на контекста, която подпомага слепи и хора с увредено зрение при непознати за тях големи обществени сгради със сложна хоризонтална и вертикална свързаност. Предложеният модел има за цел да преодолее недостатъците на съществуващите системи за вътрешна навигация, базирани на BIM.

Иновационните аспекти на предложения модел на данни могат да бъдат обобщени, както следва: 1) Абстрактно описание на йерархията на сградните елементи; 2) BIM е фокусиран върху навигацията на закрито. Предлаганият модел позволява да се извлече информация за топологията на определена част от сградата, която се използва от алгоритъм за намиране на маршрут от грубо към фино описание ([Zlatanova и колектив, 2013](#)); 3) Предоставя богата семантична информация за всички елементи на сградата, обекти и потребители, намиращи се в сградата ([Kang и Li, 2017](#)); и 4) Предоставя цялата необходима информация за избягване на препятствия (неподвижни и подвижни) ([Giudice и колектив, 2019](#)).

Най-големите ограничения за хората с увредено зрение в ежедневието им зависят главно от фактори на средата, като например придвижването в големи обществени сгради. Навигацията е сложна задача, която включва: 1) локализиране на потребителя; 2) намиране на път от началната до крайната точка; и 3) физическо придвижване до крайната точка, включително избягване на препятствия ([Real и Araujo, 2019](#)).

Възможният път може да се състои от елементи за преминаване, които се намират извън сградите, във вътрешността на сградите или както във вътрешността, така и извън тях. В много случаи намирането на най-краткия, най-бързия или най-безопасния път в големи сгради предполага възможността пътят да съдържа преодолими елементи извън сградите. Поради това навигацията в рамките на града трябва да съчетава външна и вътрешна навигация. Тъй като застроената среда нараства по размер и сложност, е необходимо да се разработят подходящи 3D

модели на града. Тези модели трябва да описват триизмерната инфраструктура на урбанизираните територии, включително йерархията на сградните елементи и тяхната хоризонтална и вертикална свързаност. При разработване на системи за навигация на закрито за слепи и слабо виждащи хора проектантите трябва да вземат предвид техните предпочитания и възможности за навигация в непознати за тях сгради ([Papadopoulos и колектив, 2011](#)).

Една от основните пречки пред успешното интегриране на населението в социалния, икономическия и културния живот е невъзможността за безопасно и надеждно да се движите в непознати пространства, независимо дали са на открито или на закрито. Проблемите, характерни за навигацията на открито, вече са частично решени. Въпреки това системите за навигация на закрито, особено тези, предназначени за хора с увреждания, нямат реално приложение ([Fernies и колектив, 2019](#)). Основната причина за недостатъците на настоящите INS е използването на *фрагментарни* подходи при проектирането на такива системи. Целта е да се проектира и реализира информационен модел на сградата, който е насочен към навигацията на потребителите и който описва йерархията, геометрията и семантиката на сградата, предпочитанията на потребителите, сензорите, разположени в сградата, и свързаността между елементите на сградата.

3.2 BIM-базирани навигационни системи

Технологията за моделиране на сгради получава все по-голямо внимание през последните няколко години. Информацията, интегрирана в BIM, описва целия жизнен цикъл на сградата.

Добре известни BIM, които могат да се използват за навигация в закрити помещения, са:

- Модел CityGML. CityGML е отворен модел за описание на 3D градски обекти. Той дефинира класовете и взаимоотношенията за най-релевантните топографски обекти в градовете и регионалните модели по отношение на техните геометрични, топологични и семантични свойства.
- Модел Industry Foundation Classes (IFC). IFC е отворен и независим от доставчици BIM хранилище за семантична информация за строителни обекти, включително геометрия, свойства и взаимоотношения. Регистриран като ISO 16739, IFC е обектно-ориентирана база данни с информация, която позволява споделяне на данни.

- Модел IndoorGML. IndoorGML е OGC стандарт, който определя отворен модел на данни и XML схема на пространствена информация за вътрешни помещения. За разлика от стандарти като CityGML и IFC, IndoorGML е фокусиран върху моделирането на вътрешни пространства за навигационни цели.
- BIM-ориентиран информационен модел на данни (BO-IDM). Този модел е опростена версия на модела IFC. Моделът използва стандарта ISO 19107 за описание на елементите на сградите и геометрията на обектите. Пространствената семантика на BO-IDM е фокусирана върху навигацията в закрити помещения. Моделът може да представя функционалното състояние на елементите на сградите, например затворени или отворени врати.
- Подобрен геометричен мрежов модел (IGNM). Този модел премахва някои от ограниченията на моделите на сгради, които се срещат в настоящата литература. Той не пренебрегва архитектурните характеристики на сградата или динамичните промени в средата на сградата (препятствия могат да бъдат добавяни и премаховани в реално време).
- Модел на вътрешно пространство за спешни случаи (IESM). Моделът IESM се базира на IFC. Моделът интегрира 3D архитектурна и семантична информация за вътрешното пространство, необходима на екипите за спешна помощ по време на извънредна ситуация.
- Модел на пространство за навигация във вътрешни помещения (INSM). Този модел предоставя разширена категоризация на вътрешните пространства въз основа на семантиката на сградите. Той има подобро представяне на графиките на свързаност в сравнение с подобни работи, извършени по-рано от авторите на настоящото проучване. Свързаността между сградните компоненти може да бъде изградена автоматично въз основа на внедряването на модела на сградата.

BIM стандарти, като CityGML и IFC, са създадени, за да опишат 3D геометрията и семантиката на сградите, но им липсват важни характеристики, които са необходими за надеждна навигация в закрити помещения. CityGML има ограничена поддръжка както за фиксирани, така и за подвижни препятствия. Нито един от стандартите не поддържа потребителски ограничения. IndoorGML използва граф модел за представяне на свързаността на сградите, но намиране на пътя обикновено използва само центроида на пространствата и информацията за връзките между пространствените обекти при изчисляване на възможен път. За да

се намери възможният път, трябва да се вземе предвид действителната геометрия на възлите и препятствията в закритото пространство. Въпреки че BO-IDM описва пространствената семантика, този модел не е оптимизиран за навигация в закрити помещения в големи сгради с комплексна свързаност. Моделът INSM може да се използва за определяне на възлите и ръбовете на графа на свързаността, но е проектиран да поддържа грубо ниво на намиране на пътя. Този модел поддържа само два критерия (степен на абстракция и брой отвори) за изчисляване на възможен път. Геометричните данни, необходими за изчисляване на физическия път, не са част от този модел, а INSM не е валидиран. Други BIM, като IESM, са проектирани така, че да отговарят на изискванията за бърза и надеждна евакуация в случай на авария. Евакуацията на хора обаче е само специален случай на навигация в сградите. Този модел не взема предвид семантиката на сензорите и потребителите. Авторите на IGNM съобщават за някои недостатъци на този модел, например проблеми с неконтекстни обекти. За врати, разположени в ъглите на стаите, алгоритъмът за планиране на пътя може да генерира невалиден път.

Хората с увреждания не могат да използват съществуващите BIM без модификации за навигация в закрити помещения. За да могат те да се ориентират надеждно в големи обществени сгради, които не са им познати, е необходим BIM, който комбинира обектно-ориентиран модел на данни с цялата необходима информация.

3.3 Предлаган подход

На Фиг. 3.1 са показани основните модули, които трябва да бъдат част от система за навигация в закрити помещения. Тези модули са:

- пространствен модел;
- модул за локализация;
- потребителски модел;
- алгоритъм за намиране на път с избягване на препятствия;
- планиране на евакуация и събиране;
- филтриране и обработка на данни от сензори.

Предлаганият модел на сгради е проектиран да представя и комбинира човешки, технически и инфраструктурни фактори, от които зависи придвижването на хора с увреждания на зрението в непознати обществени сгради.

всеки модул зависи от дизайна и функционалността на другите модули. Например, алгоритъмът за намиране на маршрут зависи от топологията и геометрията на сградата и обектите, намиращи се в нея, модула за локализация на потребители и обекти в сградата, текущото състояние на сензорите, разположени в сградата, и предпочитанията на потребителите.

3.4 Описание на предлагания модел

Предлаганият BIM не описва архитектурни компоненти на сградата. Той се фокусира върху възможните елементи на маршрута, като стаи и коридори, и свързаността между тези елементи. Като се имат предвид недостатъците на настоящите BIM, са определени следните основни изисквания:

- Мащабируемост (възможност за моделиране както на малки, така и на големи сгради).
- Йерархичност (йерархичното описание на топологията на сградите намалява използването на паметта и подобрява времето, необходимо за намиране на оптимален път).
- Контекстна осведоменост. Системата трябва да се адаптира към динамичните промени в контекста, като анализира данните от сензорите, разположени в сградата. Моделът трябва да улеснява описанието на предпочитанията на потребителя и възможностите за движение и да позволява автоматично преизчисляване на пътя в динамични среди (когато целевата точка променя местоположението си или потребителят се отклони от пътя).
- Намиране на пътя в движение. Алгоритъмът за намиране на пътя да може да работи, когато само част от данните за свързаността на сградата са заредени в паметта.
- Лесно и бързо извличане на топологията на сградата от нейния модел на данни.

Използват се няколко нива на абстракция за моделиране на йерархията на сградата. Йерархичният дизайн е лесно използваем, може да представя големи сгради със сложна свързаност, обикновено се мащабира добре и оптимизира използването на паметта и процесора. Предполага се, че сградите имат крила. Всяко крило има едно или няколко нива (етажа). Стаите и коридорите са част от нивата. Големите стаи и коридори могат да бъдат разделени на подчасти, за да представят по-точно вътрешното пространство.

На Фиг. 3.2 са показани под формата на UML диаграма връзките между класовете, които отговарят за функционалността на предложения модел. Следва описание на по-важните класове:

Сграда (Building)

Базовият клас в горната част на пространствената йерархия е Building. Всяка сграда има свойства като име, тип (университет, училище, болница и т.н.), брой крила, адрес и т.н.

Крило на сграда (Wing)

Второто ниво на пространствената йерархия е класът Wing, който съдържа едно или няколко нива. На всяко крило на сградата се присвоява име, идентификационен код, списък с нива, свързаност с други крила и списък с всички възможни аварийни изходи (*emergencyExits*). С помощта на методите на този клас е възможно да се получи името на крилото, броят на нивата, списък с аварийните изходи и др.

Ниво (Level)

Клас Level съставлява третата част от пространствената йерархия. Всяко ниво на сградата може да съдържа елементи на пътя. Елементът на пътя е абстрактна единица на сградата, която може да бъде част от възможния път. За да се ускори търсенето на елементи на пътя, се използва йерархия на елементи на пътя (атрибут *pathElementGroups*), например стаи, врати, асансьори и др.

Елемент от маршрут (PathElement)

Свързаността на сградата е моделирана въз основа на елементи на пътя. В предлагания модел елементът на пътя е абстрактен компонент, който представлява всяка част от пътя (възел или връзка). Например, стаите, коридорите и асансьорите са възли. Вратите, отворените пространства, стълбите, ескалаторите и хоризонталните рампи са връзки. Атрибутът *nodes* съдържа списък на всички връзки, чрез които потребителят може да влезе в даден възел. Атрибутът *links* описва възлите, които дадена връзка свързва. Чрез стойността на тези два атрибута се генерира граф на свързаността, който е необходим за алгоритъма за намиране на пътя.

Ограничения (Constraints)

Този клас описва ограниченията, които могат да бъдат наложени на всеки елемент на пътя. Ограниченията могат да бъдат пространствени, временни или свързани с предпочитания на потребителя. Например, някои стаи могат да бъдат достъпни само за персонала. Части от стаите могат да бъдат зони, през които не може да се преминава. Достъпът до другите помещения може да бъде ограничен във времето. В случай на авария достъпът до елементи на сградата, като асансьори и коридори, засегнати от пожар или дим, трябва да бъде блокиран. Някои врати могат да осигуряват достъп само в една посока (аварийни изходи) или да забраняват достъпа на определена група потребители (зони с определено ниво на сигурност).

Връзка (Link)

Връзката описва свързаността между възлите. Връзките се разделят на хоризонтални и вертикални. Например, с атрибутите на класа `Link` може да се идентифицира кои възли свързва връзката и дали може да се използва за евакуация.

Врати и прозорци (Door, Window)

Класът `Door` описва всички видове врати. Този клас описва типа на вратата (реална или виртуална, еднопосочна или двупосочна, външна или не), посоката на отваряне, състоянието, позицията на дръжката и т.н. Класът `Window` има сходни атрибути с класа `Door`. Ако прозорецът свързва сградата с външната среда, той може евентуално да се използва за евакуация на потребителите, ако всички други възможни пътища към външната среда са блокирани.

Възел (Node)

Възелът е клас, който описва пространствата в сградата, през които потребителят може да премине, ако има съответните права. Възлите се разделят на хоризонтални (клас *HorizontalNode*) и вертикални (клас *VerticalNode*) възли. Хоризонталните възли са свързани с хоризонталното движение на потребителите, например стаи и коридори. Асансьорите са специален тип възел и могат да се използват за свързване на много етажи в сградата. Въпреки че асансьорът е част от вертикалната свързаност на сградата, в предлагания модел той не е свързващ елемент, защото асансьорът има врати и в него могат да се намират някои потребители и обекти. Методите на този клас позволяват да се получи списък с обектите, намиращи се във възела, да се търси конкретен обект и т.н.

Стая и коридор (Room, Corridor)

Тези класове представляват основни хоризонтални възли. Възможно е да се разбият пространствата в сградата, като дълги коридори и големи стаи, на по-малки части (клас Part), за да се опишат по-точно вътрешните пространства. Дългите коридори се разделят автоматично на по-малки части в зависимост от пресичанията на коридора и местоположението на пожарните и газовите сензори. Това е необходимо, защото алгоритъмът за намиране на маршрут връща пътища покрай стените на коридорите и е възможно да се получи дълъг обход, когато коридорът има много пресечни точки. Освен това, в случай на пожар и дим, е възможно само определени части от коридора да бъдат блокирани. Конкретни места в големи стаи могат да бъдат описани чрез класа Zone, например рецепция в лобито на хотела.

Обект (Object)

Този клас описва всички физически обекти, намиращи се в сградата. Обектите се класифицират като: 1) Ориентири (обекти, които помагат на човек да се ориентира), 2) Препятствия и 3) Опасности (препятствия, които са трудни за откриване). Основните атрибути, които описват характеристиките на обекта, са следните: име и идентификационен номер; възел, към който принадлежи обектът; списък със сензори; свързани с обекта, дали обектът е неподвижен или подвижен и т.н.

Сензор (Sensor)

Класът Sensor описва всеки сензор, намиращ се в сградата. Класът SensorInfo се използва за описание на характеристиките на сензора, като идентификационен номер, тип, местоположение, а класът SensorContext представлява контекста, получен от суровите данни на сензора. За да получат данни от сензорите, класът на програмата имплементира интерфейса SensorListener. Функциите за обратно извикване *newContext* и *newEvent* трябва да бъдат декларирани в класа, който имплементира този интерфейс.

Хора (People)

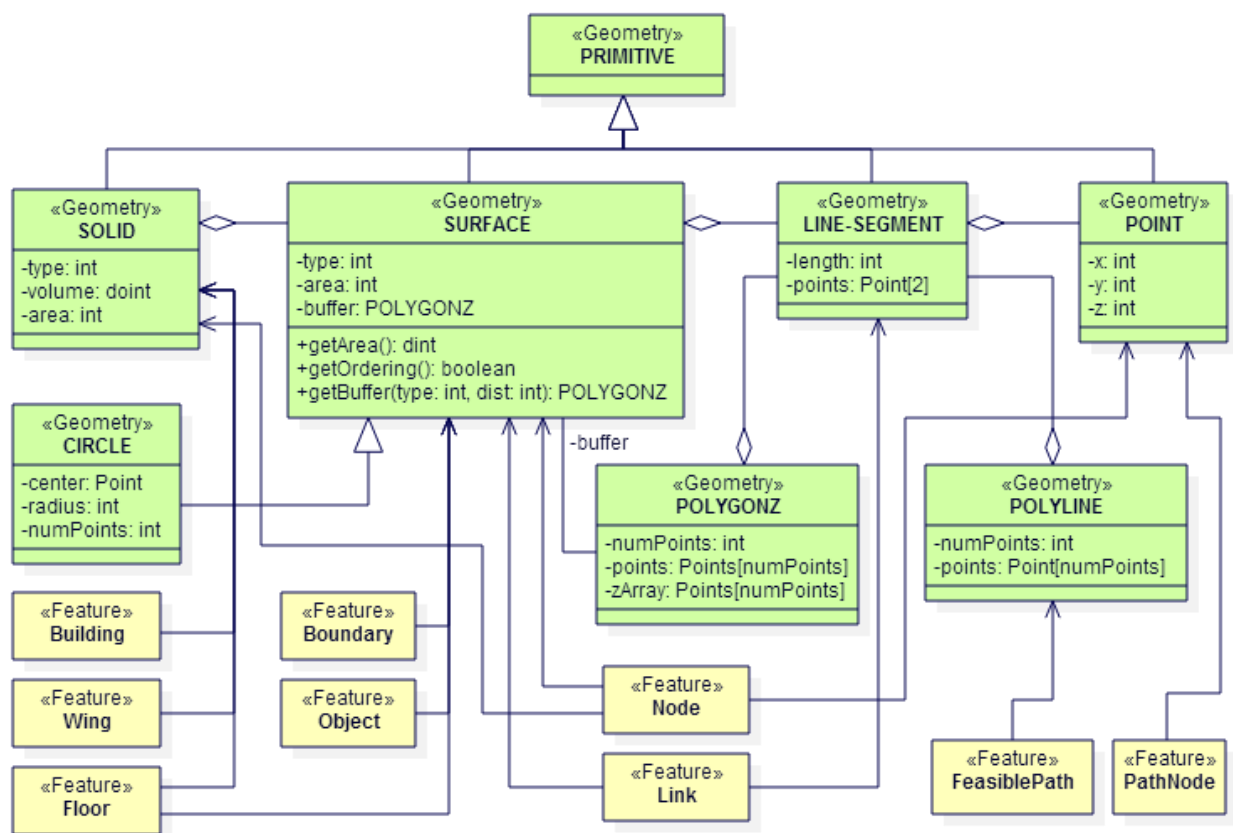
Класът People позволява достъп до модела на потребителя. Хората, намиращи се в сградата, се разделят на персонал и посетители. Атрибутите предоставят семантична информация, като демографски данни за потребителя (име и пол), вид увреждане, предпочитания за навигация, предпочитания за потребителски интерфейс, история на движението и др.

Други класове

Класът Boundary се използва за представяне на подовете и стените на всички възли и връзки. Класът Wall представя вертикалното разделение между стаи и коридори.

3.4.1 Геометричен модел

Геометричният модел позволява точното локализиране на обекти и фино описание на маршрутите. Използва се опростен геометричен модел (виж Фиг. 3.3), за да се извличат необходимите геометрични данни в реално време.



Фиг. 3.3 Геометрична част от предлагания BIM

Основните използвани геометрични обекти са точка, отсечка, полилиния и 3D полигон. За представяне на геометрията на елементите на пътя и обектите се използват евклидови или WSG 84 пространства. Полигоните се използват за представяне на геометрията на сградата и обектите. Административните граници и зоните от интерес се моделират с 2D полигони.

Основата на сградата, крилата и обектите се моделират с 3D полигони (клас POLYGONZ). 3D-полигоните съхраняват z-координати, вградени в геометрията на

обекта. Класът CIRCLE описва обекти с кръгло напречно сечение. Всъщност всеки кръг е полигон, чиито точки се определят автоматично в зависимост от радиуса на кръга. Класът SOLID представлява цялата сграда, крилата, етажите, възлите и обектите. В момента атрибутите на класа SOLID представляват типа, обема и площта на стаите и коридорите. Класът SURFACE представлява границите на възлите, връзките и обектите.

Слепите хора предпочитат да следват маршрути покрай коридорите и стените на стаите, използвайки бяла бастун. Полигонното буфериране се използва за представяне на маршрута близо до стените. За стаите и коридорите се изчислява вътрешен буфер. За да се заобиколят обектите, намиращи се в сградата, се изчислява външен буфер.

3.4.2 Семантичен модел

Семантичният модел представя три типа семантика:

- Пространствена семантика, свързана със сградата и нейните елементи.
- Потребителска семантика.
- Семантика, свързана с обектите, намиращи се в сградата.

Табл. 3.1 показва важни семантични данни във формат ключ-стойност. Семантичната информация е важен фактор, от който зависи времето за придвижване до целевата точка. За потребителите с увреждания използването на семантична информация е още по-важно. Чрез нея потребителите могат да си изградят по-точна представа за пътя, по който трябва да вървят, например характеристиките на обектите по пътя, материала на настилка и стените, наличието на ориентири и препятствия по пътя и т.н.

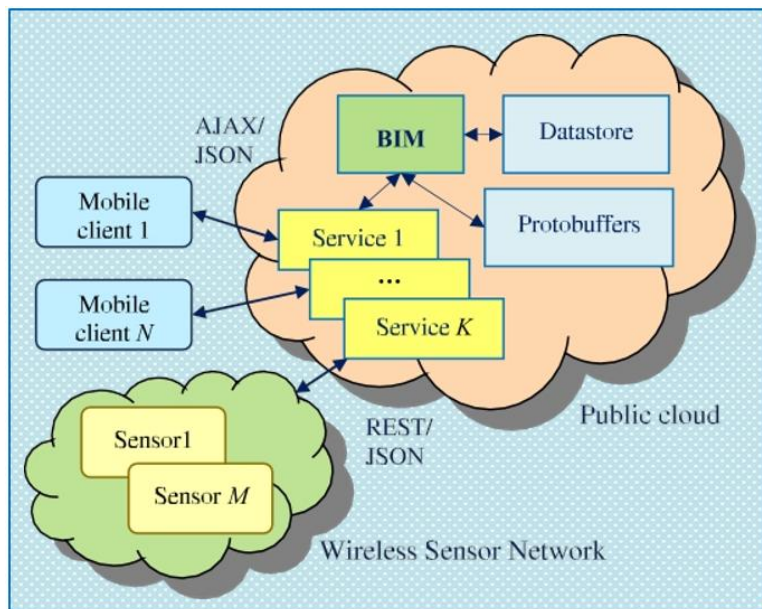
Навигацията, която се основава само на геометрични размери и разстояния, в повечето случаи не е достатъчно информативна за хората с увреждания. Например, изречението „Следвайте стената отлясно, докато стигнете до врата“ е по-информативно за лицата с увреждания на зрението, отколкото изречението „Вървете направо 15 метра“.

Табл. 3.1 Семантичен модел (опростен)

		User
name	usernames	
gender	male, female, undefined	
type	staff, visitor	
location	user location (path element id)	
position	3D user position	
preferred-user-interface-modality	text (1), audio (2), voice (4), vibro-tactile (8)	
preferred-spatial-representation	allocentric, egocentric	
preferred-path	min-path-length (1), min-number-of-turn-points (2), max-number-of-landmarks (4), min-number-of-obstacles (8), min-number-of-doors (16)	
activity-status	stay, walk, run	
impairment-type	none, blind, visually impaired, mobility	
deviation-history	normal, often, too-often	
		Building
type	university, hospital, mall, etc.	
metrics	Metric system, for example SI	
address	country, city, street, GPS location	
		Room
area	area of the room	
parts и zones	list of parts и zones, which a room contains	
		Door
virtual	true, false	
width, height	width и height of the door	
material	Material of the door	
is-emergency-exit	true, false	
function-type	fire-exit, emergency-exit	
access	automatic, manual	
scope	room, floor, wing, building	
status	open, closed, locked, blocked	
type	sliding, hinged, folding, revolving	
direction	entrance, exit, bidirectional	
opening-direction	left, right, bottom, top, sliding	
		Window
width, height	width and height of the window	
material	material of the window	
external	true, false	
status	open, closed, locked, blocked	
type	sliding, hinged	
opening-direction	left, right, bottom-top, sliding	
		Stairs
handrail-position	left, right, both	
material	Material of the stairs	
number-of-steps	number of steps	
incline	incline in degree	
direction	up, down, up-down	
		Elevator
accessible	true, false	
type	passenger, safety lift, luggage lift	
capacity	maximum number of persons	
status	in operation, not operating	
		Object
type	landmark, obstacle, hazard	
movable	true, false	
material	textiles, wood, metal, leather, etc.	
material-type	flammable, fire-resistant, water-resistant	
		Wall
id	Wall identification code. Each room can have four or more walls. In this case, a room can have three walls if there is an open space on the site of the fourth wall.	
material	plaster, stucco, wallpaper, etc.	
is-external	true, false	

3.5 Внедряване на BIM

В тази секция се разглежда как да се внедри предложения BIM като хранилище на данни за сгради. Всеки BIM модул се специфицира чрез XML файл и се дефинира в индивидуално и глобално уникално XML целево пространство от имена. Разработена е услуга в облака за навигация в закрити помещения, за да се валидира предложеният модел на данни за сгради (виж Фиг. 3.4). За внедряването на услугата се използва Google App Engine (GAE).



Фиг. 3.4 Имплементиране на предлагания BIM

За достъп до услугата, отговаряща за събирането и управлението на данни от сензорите, се използват REST заявки. RESTful протоколът предоставя интерфейс на ниво обект (сензори и актуатори) към съответните BIM обекти. Имената на тези обекти се идентифицират директно чрез URI. Когато сензор промени състоянието си, към услугата се изпраща REST заявка, съдържаща информация за тази промяна. Сензорите могат да бъдат програмирани да изпращат данни, когато:

- Стойността на даден параметър се е променила с определена абсолютна стойност или процент;
- Стойността на даден параметър е над или под определено прагово ниво;
- Засечено е събитие от интелигентен сензор.

Мобилните клиенти използват AJAX, за да позволят бърз асинхронен трансфер на информация от клиентите към съответната услуга.

3.5.1 Контекстен модел

Контекстният модел описва контекста, който е достъпен от сензорите и потребителите. За управление на контекстните данни се използва обектно-ориентиран модел, а за семантичните данни – модел ключ-стойност. Това решение гарантира повторно използване и капсулиране на данните. За да се изгради интелигентна среда, трябва да се разположат голям брой сензори, които да реализират функционалността на услугата. Тези сензори ще генерират големи масиви от данни, от които трябва да се извлича контекстна информация. Контекстният мидълуеър предоставя на приложенията контекстна информация. Основните предимства на това решение са:

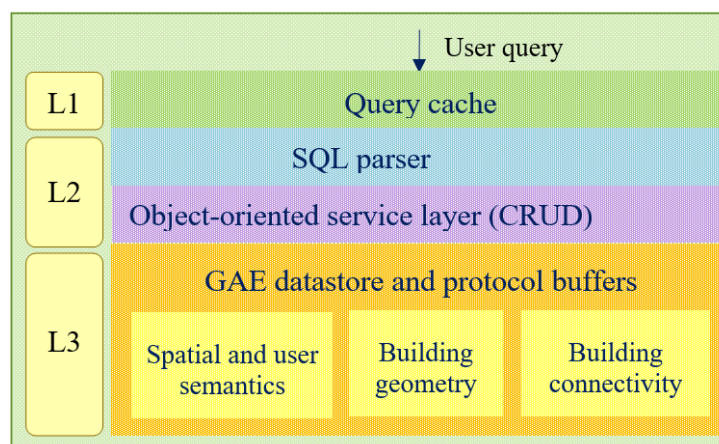
- Приложенията получават (обикновено чрез push известия) необходимата контекстна информация, без да се интересуват от това как тя се извлича от суровите данни на сензорите.
- Трафикът между клиентите и сървъра се намалява.
- По-лесно е да се гарантира поверителността и сигурността на данните.

Използват се три нива на контекстна осведоменост:

- 1) Персонализация: лична информация и предпочитания на потребителя, например пол, местоположение, вид увреждания, предпочитано пространствено представяне, текуща активност на потребителя (престой, ходене, отклонение) и история на отклоненията;
- 2) Пасивна контекстна осведоменост. Системата постоянно наблюдава околната среда и предлага подходяща услуга на потребителите. Например, когато потребител влезе в сградата, той получава информация за сградата; и
- 3) Активно осъзнаване на контекста. Например, системата постоянно наблюдава околната среда за извънредни ситуации. Предложеният контекстен модел може да съхранява данни от сензори и събития за бъдеща употреба (история на контекста). Историята на отклоненията на потребителя се използва, за да се гарантира по-прецизно насочване по пътя.

3.5.2. Модел за достъп до данни

Приложен е подход за достъп до данни на слоеве, за да се гарантира мащабируем дизайн на модела за достъп до данни (виж фиг. 3.5). Всяко заявка на потребител, базирана на местоположението, се преобразува в заявка в стил SQL.



Фиг. 3.5 Модел за достъп до данните

Специално проектираният SQL слой (L1) преобразува SQL заявките в заявка към втория слой L2. Кешът на заявките използва SELECT клаузата като ключ и набор от резултати като стойност. Ако текущата SELECT заявка се намира в кеша, не е необходимо SQL анализиране и изпълнение. Вторият слой (L2) изпълнява основни CRUD операции (създаване, четене, актуализиране, изтриване).

Служебният слой е обектно-ориентиран модел за достъп до данни, който съдържа част или всички данни на сградата.

Тази архитектура е оптимизирана за ускоряване на операциите четене. Когато вторият слой не може да върне отговор, защото необходимите данни липсват в паметта, тези данни се извличат от протоколните буфери или хранилището на данни (L3). В сравнение с XML описанието на данни, прото-буферите на Google са до 10 пъти по-малки и до 100 пъти по-бързи.

3.5.3. Модел за запитвания, базирани на местоположението

Навигационната система трябва да предоставя механизъм за обслужване на запитвания, базирани на местоположението. Тази задача става по-трудна, когато:

- 1) навигацията се извършва в големи обществени сгради със сложна свързаност; и
- 2) някои целеви точки могат да променят местоположението си, докато потребителят преминава по маршрута.

Предлаганият ВІМ поддържа непрекъснати запитвания, базирани на местоположението – системата е в състояние да коригира отговора си, ако бъде засечена промяна в позицията на крайната точка. Непрекъснатите запитвания, зависещи от местоположението, могат да се считат за ключови елементи за разработването на услуги, съобразени с контекста. Програмният обект, който обслужва запитването на потребителя, остава

активен, докато потребителят достигне целевата точка. Обектът се сериализира и става персистентен. Ако има push данни от сензори (например, целевата точка променя местоположението си), обектът се десериализира.

3.5.4 Алгоритъм за намиране на път

Алгоритъмът за намиране на път трябва да намери най-подходящия път за даден потребител от началната до крайната точка. Използва се специално разработен алгоритъм за планиране на маршрути, който е описан подробно в Глава 4.

Алгоритъмът намира оптималния път в три стъпки:

- Първо, получават се K на брой най-кратки макро пътища от началната до крайната точка;
- Второ, изчисляват се микро маршрути в рамките на всяка стая през която преминава макро маршрута с помощта на геометричен модел;
- Накрая се връща оптималният път с най-висок общ ранг, като се използва техника за многоцелева оптимизация.

Въз основа на проведено анкетирание се решава, че изборът на оптималния път ще зависи от шест параметъра: 1) Дължина на пътя (мин.); 2) Брой обекти, които са препятствия или опасности за потребителите с увреждания на зрението (мин.); 3) Брой обекти, които са ориентири и указания за потребителите с увреждания на зрението (макс.); 4) Брой врати, през които потребителят трябва да мине (мин.); 5) Брой промени в посоката на движение (мин) и 6) Брой пресечни точки в коридорите (мин). Началните стойности на важността на всеки критерий се получават с помощта на теорията за корелация на ранговете.

Алгоритъмът взема предвид текущата позиция на потребителя. Той е независим от броя на препятствията, тяхната форма, ориентация и позиция в стаята и работи с циклични или ациклични, насочени или ненасочени графи.

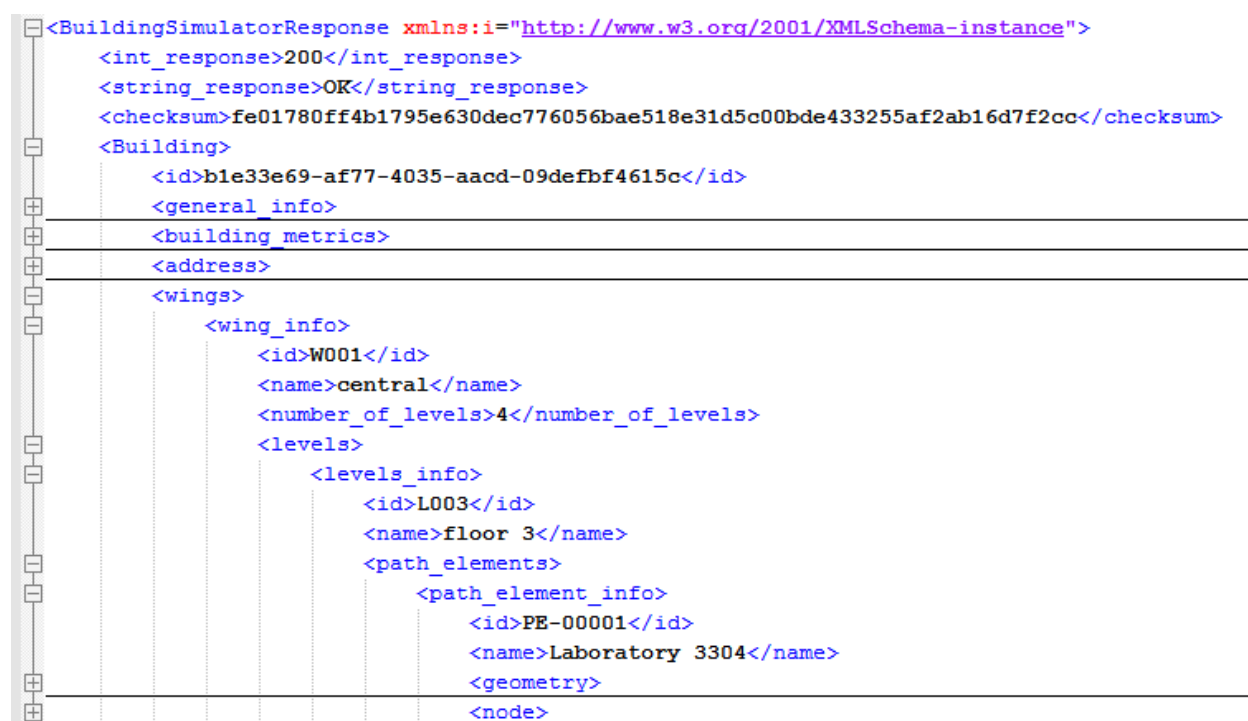
3.5.5. Евакуация

Когато се активира евакуационен план, се изпълняват следните действия: 1) Всички асансьори, стаи или части от коридори, които вече са засегнати от пожара или газа, се изключват, 2) Изпраща се push съобщение до клиентите, което ги предупреждава да изтеглят нова карта на свързаността, 3) Прозорците могат да се използват като евакуационни изходи, ако не може да се намери път за бягство; 4) Получава се информация за материала и дебелината на стените и вратите, за да се предвиди пътят на пожара и скоростта и посоката на разпространението му и 5) Модул за

намиране на пътя с много крайни точки (евакуационни изходи) се адаптира към активирания евакуационен план.

3.6 Експерименти

На този етап услугите, които реализират навигационна система на базата на BIM, са разгърнати в GAE. Използва се Web услуга (Building Builder), която позволява 2D/3D моделиране и визуализация на сградата и нейните елементи с помощта на WebGL. Пространствената информация може да бъде генерирана с помощта на примитивни графики или IFC файлове. Информацията, необходима за изграждане на BIM, може да бъде експортирана в XML формат (виж Фиг. 3.6).



Фиг. 3.6 XML скрипт, описващ топологията на сграда

Услугите е тествана чрез XML скриптове, които описват промените в контекстните данни във времето (състояние на сензори, обекти и клиенти). Тези скриптове се генерират от Web услуга наречена Building Simulator. Потребителите на услугата могат да филтрират информацията, която желаят да използват. За тази цел информацията за сградата е разделена на слоеве: топология; сензори и актуатори; обекти; персонал и посетители; ограничения; суров контекст; и събития. Услугата позволява на потребителите да променят топологията; да добавят и премахват обекти, сензори и актуатори; да генерират определен брой посетители и членове на персонала с определено или случайно местоположение; да задават ограничения

за достъп (пространствени, времеви или свързани с типа на потребителските предпочитания и увреждания); и да симулират промяна на контекста.

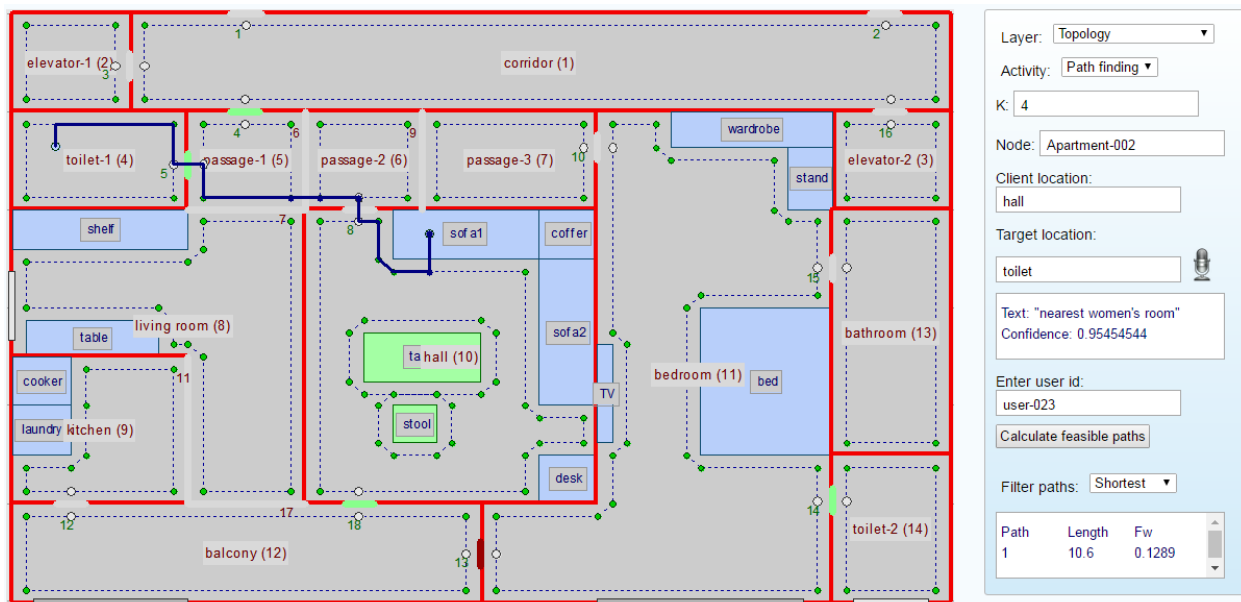
Всеки клиент на услугата получава NFC идентификационна карта. Картата се използва за: а) автоматично инсталиране на приложението на клиентската страна при първото докосване на мобилния телефон до картата; б) автоматично стартиране на приложението, ако то вече е инсталирано; в) оторизиране на клиента за достъп до услугата. За да се гарантира сигурността на клиента, се използва авторизация и достъп до ресурсите на услугата чрез токени.

Потребителят достига до желаната сграда, използвайки GPS-базирана система за навигация на открито. За да информира клиентите, че са достигнали сградата, системата използва BLE маяци. Когато клиентът се приближи до един от входовете на сградата (15-20 м), той получава push уведомление с името и предназначението на сградата. Клиентът може да получи допълнителна информация за топологията на сградата, включително информация за етажите, входовете, изходите и дали сградата е достъпна за хора с увреждания.

Системата предоставя на клиента подсказки чрез синтезиран глас (TTS), за да избере крайна точка. Това може да бъде номер на етаж, име на стая, аварийен изход, име на член на персонала или посетител, или друга важна уникална информация. Описанието на големи обществени сгради с помощта на BIM включва работа с големи масиви от данни. Затова е необходим гъвкав, бърз и интелигентен начин за извличане на данни от BIM. Един от начините да се постигне това е да се позволи използването на естествени гласови заявки.

Предложената система използва естествени гласови заявки за извличане на данни от BIM, например: „информация“, „ортопедична спешна помощ“ или „д-р Иванов“. Системата използва модул за разпознаване на реч с разпределена архитектура. Целта е по-висока вероятност за разпознаване на речта и надеждност на услугата. За внедряването на автоматично разпознаване на речта (ASR) на този етап системата използва две облачни услуги, които са достъпни чрез следните API: Google Speech API и Bing Speech API. И двете услуги изискват регистрация за получаване на ключ за достъп. И двете API използват POST заявки за изпращане на гласови данни (Google Speech API изисква FLAC формат с честота на дискретизация 16 KHz, докато Bing Speech API изисква WAV формат с честота на дискретизация 8 или 16 KHz). Резултатът се предоставя в JSON формат и съдържа статус, разпознат текст и стойност на достоверност.

Нека да предположим, че клиентът е жена, която търси най-близката дамска тоалетна. Клиентът се намира в една от стаите на апартамента (виж Фиг. 3.7), който има 12 възли (стаи) и 18 връзки (врати и отворени пространства).



Фиг. 3.7 Тестване на услугата в симулирана 2D среда

За изречението „Къде е най-близката дамска тоалетна?“ Google Speech API връща следния резултат:

```
Result:{"result":[{"alternative":[{"transcript":"where is the nearest women's room", "confidence": 0.82209402}, {"transcript":"where is the nearest woman's home"}, {"transcript":"where is the nearest woman's womb"}, {"transcript":"where is the nearest women's home"}, {"transcript":"where is the nearest woman's room"}], "final": true}], "result_index": 0}
```

Най-вероятният резултат (доверие 0,82209402) според тази услуга е „къде е най-близката дамска тоалетна“. Microsoft API връща същия резултат (доверие 0,9442599) и предоставя допълнителна информация, като лексикална интерпретация, маркери и произношение.

След като услугата получи заявката като текст, тя започва лингвистичен анализ в следната последователност: а) разделяне на изречения (разделяне на текста на изречения, намиране на маркери за края на изреченията); б) токенизация (разделяне на последователността от символи на думи); в) маркиране (идентифициране на категорията на всяка дума – съществително, глагол, прилагателно и т.н.); г) семантичен анализ (намиране на връзки между ключовите

думи от заявката на потребителя и имената на обектите и техните свойства); д) картографиране (намиране на ключови думи, заменянето им със синоними, ако е необходимо, и картографиране на крайните ключови думи към имената на BIM обектите или имената на техните свойства).

За реализацията на останалата част от обработката на естествен език, предлаганата система използва Microsoft Linguistic Analysis API. Тази услуга, базирана в облака, използва POST заявки за изпращане на резултата, получен от ASR. Резултатът се предоставя в JSON формат и съдържа информация за типовете фрази. Средното време за отговор е 250 ms. Ако гласовото запитване на клиента е „Къде е най-близката дамска тоалетна?“, се получава следният резултат от лингвистичния анализ:

```
"(TOP (SBARQ (WHADVP (WRB where)) (SQ (VP (VBZ is) (NP (NP (DT the) (JJ nearest) (NNS women)) (VBZ 's)))) (NN room) (. ?)))"
```

където:

- SBARQ е пряк въпрос, въведен с въпросителна дума или фраза
- WHADVP е въпросителна адвербиална фраза;
- WRB е въпросителна адвербиална дума (къде);
- SQ е обърнат въпрос с отговор „да“/„не“ или главно изречение на въпросителен въпрос;
- VP е глаголна фраза;
- VBZ е глагол, сегашно време, 3-то лице единствено число (e);
- NP е съществително;
- DT е определител (the);
- JJ е прилагателно (nearest);
- NNS е съществително, общо, множествено число (women);
- VBZ е глагол, сегашно време, 3-то лице единствено число ('s);
- NN е съществително, общо, единствено число или множествено число (room).

В някои случаи системата може да изисква от клиента да потвърди или уточни заявката си. Това е необходимо, ако вероятността за правилна идентификация е под предварително определено ниво или ако заявката предполага двусмислен отговор, например: „Моля, потвърдете, че търсите ортопедичен кабинет“, „В тази сграда работят трима лекари на име Иванов (следват пълните им имена и специалности). Кой от тях търсите?“

В нашия случай е ясно, че това е запитване за място (къде е). Клиентът търси „тоалетна за жени“. Има допълнително изискване: търси се най-близката до позицията на клиента стая. Запитването, което се генерира, за да се извлече информация за обектите, които съответстват на това запитване, е:

```
SELECT * FROM Node WHERE name='toilet' AND scope='apartment';
```

След като началната и крайната точка са известни, се стартира алгоритъмът за планиране на маршрута (стартовата точка съвпада с текущото местоположение на клиента).

Първо, алгоритъмът извлича информация от BIM за свързаността на елементите на сградата, които са в обхвата на заявката (възел, апартамент, етаж, крило или сграда). Използвайки тези данни, алгоритъмът намира K възможни пътя (макроскопичен етап). По подразбиране $K = 4$.

Второ, използвайки данни от геометричната част на BIM, микроскопичната фаза на алгоритъма изчислява оптимален път за даден клиент. Тя взема предвид предпочитанията на клиента, които се извличат от неговия потребителски модел, и състоянието на сензорите, разположени в сградата. Системата използва многопараметрична оптимизация, за да изчисли оптимален път. Изчисленият най-добър възможен път се описва в XML формат като поредица от междинни точки. В конкретния случай системата връща всички обекти от клас „тоалетна“, които попадат в определения обхват „Апартамент-002“: тоалетна-1 (възел 4) и тоалетна-2 (възел 14). От частта „Ограничения“ на BIM става ясно, че и двете тоалетни са достъпни за посетители. Системата проверява дали потребителят има достъп до намерените тоалетни. Тази информация се получава от профила на потребителя, който е част от предложената BIM. Броят на всички възможни маршрути от текущата позиция на потребителя до всяка една от тоалетните е шест. Макроскопичната част от алгоритъма за намиране на маршрути позволява намирането на няколко подобни целеви точки в едно търсене. Алгоритъмът връща първите 4 най-кратки възможни маршрути до целевата точка (тоалетна), подредени според броя на връзките, през които потребителят трябва да мине:

- 1: hall |d8| passage-2 |d6| passage-1 |d5| toilet
- 2: hall |d18| balcony |d13| bedroom |d14| toilet
- 3: hall |d18| balcony |d17| living room |d7| passage-1 |d5| toilet
- 4: hall |d8| passage-2 |d9| passage-3 |d10| bedroom |d14| toilet

За да се намери оптимален път за конкретен потребител, трябва да се изчислят стойностите на целевата функция Fw за първите четири най-добри възможни пътя. Теглата на използваните критерии за конкретен потребител (получени от модела на потребителя) са $W = [0,2640, 0,2360, 0,2267, 0,1507, 0,0853, 0,0373]$. При тези тегла, оптималният маршрут е този с номер 1. Тъй като клиентът търси най-близката дамска тоалетна, системата предоставя информация за маршрута с най-кратка дължина. В настоящия случай този маршрут е и оптимален и за клиента. Намирането на маршрут, съобразено с контекста, отнема максимум 65 ms. Средното време за отговор на заявка, базирана на местоположението, е 1850 ms.

3.7 Заключение

На този етап няма навигационна система, която да осигурява необходимата надеждност за навигацията на слепи хора, независимо от дизайна и архитектурата. Използването на решения, които са независими от инфраструктурата на сградата, се основава на разпознаването на сцени/изображения и инерциална навигация. Въпреки много добрите резултати в областта на разпознаването на модели чрез дълбоко обучение на невронни мрежи, навигационните системи, използващи този подход, не са приложими за хора с увреждания по няколко причини. Тези системи са силно зависими от осветлението в сградата и стават неизползваеми при липса на електрозахранване или при пожар или дим в сградата; а вероятността за правилно разпознаване в реални условия не е достатъчна за надеждна навигация за хора с увреждания.

Настоящото проучване представя подход за разработване на система за навигация в сгради за хора с увреждания на зрението. Този подход използва специално проектиран BIM, който комбинира в един модел цялата информация, необходима за навигация в закрити помещения на големи обществени сгради. Предлаганият подход за създаване на системи за навигация в закрити помещения, който се основава на обединяването на цялата информация, необходима за надеждна навигация, в единен модел на данни, включва използването на интелигентни сгради със специфична инфраструктура (сензори и актуатори). Изискванията за използване на BIM за проектиране и строителство на сгради в целия свят, както и тенденциите за строителство на интелигентни сгради, сочат, че предложеният подход би бил лесен за прилагане в близко бъдеще. Този подход се основава на технологии, които започват да показват своя потенциал, например: интернет на нещата (IoT), микроконтролери с ултра ниска консумация на енергия, персонализирани приложения и облачни услуги.

ГЛАВА 4: АЛГОРИТЪМ ЗА НАМИРАНЕ НА K НАЙ-КРАТКИ ПЪТИЩА В МНОГОЕТАЖНИ СГРАДИ

4.1 Въведение

В тази глава се представя подход за изчисляване на K най-къси маршрути в големи сгради с комплексна хоризонтална и вертикална свързаност. Топологията на сградата се получава от BIM и се реализира с помощта на насочени мултиграфи. Йерархичният дизайн позволява изчисляването на възможните пътища, без да е необходимо да се зарежда в паметта топологията на цялата сграда. Възможно е разширяване на графа с нова свързаност в реално време. Изчислените пътища могат да се състоят от компоненти на сградата през които може да се преминава. Тези компоненти може да се намират както извън, така и вътре в сградата. Производителността (време за изчисляване на маршрутите и размер на използваната памет) се оптимизира чрез използване на подходящи колекции (карти, списъци и множества). Предложеният алгоритъм е оценен в няколко сценария на използване – пълни графи и при реални сградна инфраструктура. Във всички тестови сценарии предложеният алгоритъм за намиране на маршрути е по-бърз и използва по-малко памет в сравнение с бързата версия на алгоритъма KSP на Yen.

4.2 Анализ на проблема

Навигацията е сложна задача, която включва:

- локализиране на потребителя;
- ориентация на потребителя в средата на сградата;
- намиране на път от началната точка до целевата точка;
- физическо придвижване до целевата точка, включително избягване на препятствия по пътя.

Алгоритмите за намиране на маршрут зависят от няколко фактора, като топология и геометрия на сградата; пространствена и потребителска семантика; местоположение на обекти и хора в сградата; състояние на сензорите, разположени в сградата; потребителски предпочитания; и наличие на ориентири, препятствия и опасности по пътя. Тези алгоритми трябва да намерят оптималния път за всеки потребител, а не най-краткия или най-бързия път.

Пътят от началната до крайната точка може да се състои от елементи, които се намират вътре в сградите, извън сградите или както вътре, така и извън тях. Следователно навигацията в града трябва да комбинира навигация на открито и на закрито. С разрастването и усложняването на застроената среда е необходимо да се разработят подходящи 3D модели на града. Тези модели трябва да описват 3D интелигентна градска инфраструктура, включително йерархията на елементите на сградите и тяхната хоризонтална и вертикална свързаност (Tashakkori и колектив 2015, Abdoulaye и Zlatanova 2017). Цифровото описание на сградите предполага работа с големи масиви от данни. Ако обработката на тези данни трябва да се осъществява в реално време, едно възможно решение е тяхното йерархично моделиране. Например, когато се придвижваме между две сгради, не ни интересува свързаността на ниво стаи, а само техните точки за достъп. Когато се придвижваме между два етажа, важните аспекти са свързаността на тези етажи и съответната вертикална свързаност. Следователно ефективното йерархично моделиране на топологията е ключов елемент в реализацията на намирането на пътя в големи обществени сгради. Най-използваните алгоритми за намиране на маршрут са проектирани да намират най-краткия или най-бързия маршрут при придвижване на открито. При навигация в сгради параметрите, които определят кой път е оптимален за даден потребител, са много повече от два. Например, такива параметри могат да бъдат броят на препятствията и опасностите по пътя, броят на ориентири по пътя, броят и състоянието на вратите, през които потребителят трябва да мине, броят на промените в посоката на движение и броят на пресечните точки в коридорите. Всеки от тези параметри може да има различна тежест в зависимост от конкретния потребител. Освен това някои параметри трябва да бъдат минимизирани, докато други трябва да бъдат максимизирани. Това е типична задача за оптимизация с много параметри и не може да бъде решена само чрез промяна на крайната цена в топологичния модел на данни. Освен това е възможно да се получат отрицателни тегловни коефициенти, но малък брой алгоритми за намиране на пътя позволяват това. Следователно, използването на многостепенни алгоритми за намиране на пътя предлага по-голяма перспектива (Xiong и колектив 2015). Обикновено на първото (грубо) ниво се изчислява най-краткият път между стартовото и целевото помещение, а на второто (фино) ниво се изчисляват физическите пътища във всяко от помещенията, през които минава маршрута. При използване на евристични алгоритъм за намиране на пътя е възможно да се получи близо до оптималния груб път. По-добро решение е да се намерят първите K възможни груби пътища и след това за всеки от тях да се получи

подходящият физически път. Изборът на оптималния път от всички K физически пътища може да се реализира чрез многопараметрична оптимизация.

Често за моделиране на свързаността между компонентите на сградите се използват графи. Обикновено възлите описват компонентите на сградата, а ръбовете описват свързаността между тези възли. Основният недостатък на тези модели е, че те не отчитат геометричните размери на възлите и обектите в тях. Тази информация е много важна за надеждността на навигацията, особено за хората с увреждания. Надеждна навигация в сградите е невъзможна без богата семантична информация за компонентите на сградата, обектите в сградата и хората, които се намират в сградата. За тази цел се използват различни семантични модели. Най-добрият вариант за получаване на модели, които могат да се използват многократно, е да се приложат чрез онтология ([Kun и колектив 2017](#), [Lee и колектив 2017](#)). Тези модели описват по-абстрактно вътрешното пространство, но предоставят важна информация за навигацията, като свойствата на компонентите на сградата, през които минават пътеките (стаи, врати, препятствия, стълби, асансьори), както и семантиката на потребителите. С помощта на тази информация е възможно да се вземат предвид личните предпочитания на всеки потребител и по този начин да се реализира оптимално намиране на пътя и персонализирана навигация. Семантичните модели не могат да се използват самостоятелно поради невъзможността за достатъчно точно описание на маршрутите.

Тъй като всеки подход за моделиране на данни за вътрешни пространства има своите силни и слаби страни, може да се препоръча интегрирането на силните страни на няколко подхода в единен модел за пространствени данни за вътрешни пространства, за да се компенсират слабите страни. На практика се използват хибридни модели, които комбинират няколко от анализирания модели. Например, [Lin и колектив \(2017\)](#) комбинират топологични и мрежови модели, за да подобрят времето за намиране на най-краткия път. [Ху и колектив \(2018\)](#) представят хибриден мрежови модел, който съдържа геометрична, топологична и семантична информация за вътрешното пространство.

Алгоритмите за намиране на маршрути могат да бъдат класифицирани като класически, евристични и хибридни. Класическите методи за намиране на пътя могат да бъдат класифицирани като методи, базирани на пътна карта (графика на видимостта и диаграми на Вороной), методи за разлагане на клетки, като точно, приблизително или вероятно разлагане на клетки, и методи за изкуствено

потенциално поле. Тези алгоритми се използват главно с геометрични пространствени модели. Евристичните алгоритми обикновено намират почти оптимално решение.

Основните мета-евристични техники, които могат да се използват за планиране на пътя в закрити помещения, са следните: оптимизация на мравуняка, оптимизация на роя от частици, табу търсене, генетични алгоритми, изкуствени имунни мрежи и др. Евристичните методи не гарантират оптимално решение (те могат да намерят локално оптимално решение). Хибридните алгоритми съчетават предимствата на класическите и евристични техники. Част от алгоритмите за намиране на пътя, като пътни карти, са предназначени за планиране на маршрути на открито. Други, като потенциалните полета, са подходящи за навигация на роботи и хора без увреждания. Методите за клетъчна декомпозиция не са приложими за много големи сгради с комплексна свързаност между стаи, етажи и крила поради невъзможността за йерархично описание на сградите и голямото количество памет, което ще бъде необходимо за тяхната реализация. Оптимизацията на мравуняка и генетичните алгоритми не предоставят оптимално решение и за много големи сгради може да не дадат отговор в реално време.

Повечето системи за навигация в закрити помещения използват описание на свързаността на елементите на сградата чрез графи. Нека $G(V, E)$ представлява граф, където V е набор от n възли, а E е набор от m ребра. Най-често се използва намиране на път от един източник до една дестинация. В този случай се използва алгоритъм търсене на най-кратък път (SSP). Целта е да се намерят най-краткия или най-бързия път от началната точка до целевата точка. За тази цел най-често се използва алгоритъмът на Dijkstra (Dijkstra, 1959) или някоя от неговите модификации (Lee и колектив, 2017). В прости насочени графи този алгоритъм намира оптимален път до целевата точка, но изисква прекалено много изчислително време. Сложността на най-добрия вариант на алгоритъма на Dijkstra, който използва хийп на Фибоначи като приоритетна опашка, е $O(n \log(n))$. Основният недостатък на алгоритъма на Dijkstra е експоненциалното увеличение на изискванията за памет, когато е необходимо увеличаване на размера на картата на сградата. Алгоритъмът на Dijkstra може да се използва само в графи, които не съдържат отрицателни тегла на ребрата. За графи с отрицателни тегла проблемът за най-краткия път може да се реши с алгоритъма на Bellman-Ford (Goldberg и Radzik, 1993). Сложността на изпълнението му е $O(nm)$. Когато местоположението на целевата точка е известно, алгоритъмът за намиране на път може да използва хеуристична функция, за да определи посоката на намиране на пътя към това местоположение. Евристичните

алгоритми са подходящи за използване както с модели, базирани на мрежа, така и с модели, базирани на графи. Най-често използваният евристичен алгоритъм за намиране на път е A^* (Hart и колектив 1968). Този алгоритъм използва търсене по принципа „най-доброто първо“, за да намери пътя с най-ниска цена. Поради използвания евристичен подход, алгоритъмът A^* не гарантира оптимално намиране на път при избор на неподходяща хеуристична функция. За евклидови графи A^* алгоритъмът може да намери най-краткия път със сложност $O(n)$. Основният недостатък на A^* алгоритъма е количеството памет, необходимо за работата му. A^* е изключително ограничен по отношение на пространството в практиката и бързо изчерпва наличната памет. Точността на евристичната функция определя времевата сложност на е алгоритъм за търсене. Следователно, A^* работи с линейно време, ако евристичната функция за оценка е точен оценител. Алгоритъмът Fringe-Retrieving A^* (FRA*) (Sun и колектив 2010) е по-бърза версия на алгоритъма A^* . Той позволява преобразуване на предишното дърво за търсене в актуализирано дърво въз основа на новите местоположения на потребителя и целевия обект. FRA* не взема предвид йерархичната структура на интериорната среда.

С изключение на възлите на сградата, потребителят трябва да може да зададе като целеви точки обект или човек, който се намира в сградата. В този случай алгоритъмът, който търси път от текущата позиция на потребителя до избраната целева точка, трябва да се адаптира към променящия се контекст в сградата. Много малко системи за навигация в закрити помещения и свързани с тях изследвания вземат предвид промените в контекста на сградата (Merkel и колектив 2014, Ji и колектив 2015). За хората с увреждания разстоянието и времето за пътуване не са единствените критерии, които определят оптималния път. Няколко допълнителни параметри, като сложност на пътя (брой завои, брой и тип на пресечките на коридора), оценка на риска по пътя (брой препятствия и опасности по пътя), достъпност на пътя, класификация на евакуацията (аварийни изходи, ширина на коридора) и др. трябва да се вземат предвид при използването на подходяща функция на разходите за ребрата на графа. Значението на всеки елемент от пътя може да бъде представено чрез теглото на ребрата.

Ако значимостта на елементите на пътя зависи от много параметри и навигацията се реализира в големи сгради със сложна свързаност, по-доброто решение е да се намери първият K -най-къс път (KSP). Повечето KSP алгоритми имат експоненциална сложност на изпълнение и пространство. Топологията на сградата обикновено се представя с прости насочени графи. Най-бързите алгоритми за KSP,

в случай на претеглен насочен прост граф, са предложени от Yen (1971). Когато се използва алгоритъмът на Dijkstra като SSP функция, сложността на алгоритъма на Yen е $O(kn(m + \log n))$. Hershberger и колектив (2007) представят KSP алгоритъм за ориентирани прости графи с времева сложност $O(kc(n, m))$. Това е подобрение $O(n)$ спрямо алгоритъма на Yen, но алгоритъмът е податлив на грешки в някои редки случаи. Ако важността на реброто зависи от множество параметри, съставната цена на реброто може да има отрицателни стойности. В този случай SSP функцията може да се използва алгоритъмът на Bellman-Ford.

За ненасочени графи най-доброто KSP решение дава алгоритъмът на Katoh (Katoh, и колектив 1982). Той има сложност $O(k(m + n \log n))$, когато проблемът за най-краткия път е реализиран чрез бързия алгоритъм на Dijkstra. Най-известният KSP алгоритъм, базиран на евристика, за графи с цикли е алгоритъмът K^* (Aljazzar и Leue 2011). Този алгоритъм комбинира характеристики на търсене с еднаква цена и чисто евристично търсене, за да намери оптималното решение по-ефективно. K^* прилага A^* търсене към графа, за да изчисли дървото за търсене. Следователно недостатъците на алгоритъма A^* са и недостатъците на алгоритъма K^* . K^* работи в реално време, което означава, че не изисква графът да е изрично наличен и съхранен в основната памет. Части от графа ще бъдат генерирани при необходимост. Неговата сложност е $O(k + m + n \log n)$.

Понякога е необходимо да се използват графи с цикли. Да предположим, че в дълъг коридор има пожар. Ако евакуираният може да заобиколи пожара, като мине през съседните стаи, е необходимо той да пресече коридора два пъти. Ако алгоритъмът за намиране на път работи само с прости графи, този важен заобиколен път няма да бъде изчислен. Най-бързият алгоритъм за намиране на KSP в претеглени графи с насочени цикли е алгоритъмът на Епщайн (Eppstein, 1999). Той има сложност на изпълнение $O(m + n \log n + k(n + \log k))$.

Сравнение на сложността на най-използваните SSP и KSP алгоритми, които могат да се използват за намиране на маршрути в сгради, е представено в Табл. 4.1.

За надеждна навигация в големи обществени сгради е необходимо да се получи оптимален маршрут за всеки потребител. Този маршрут трябва да описва точно и подробно движението от началната до крайната точка, като се вземат предвид препятствията по пътя. В същото време алгоритъмът трябва да се реализира в реално време. Това са противоречиви изисквания, които не могат да бъдат удовлетворени поотделно от нито един от анализираните модели на пространствени данни и алгоритми за намиране на маршрути.

Табл. 4.1 Сравнение на най-използваните алгоритми SSP и KSP.

Algorithm	Type	Graph type	Worst-case complexity	Comments
Dijkstra's	Single-source SSP	Directed simple graphs	$O(m + n \log n)$	Only for static environments
Bellman-Ford	Single-source SSP	Graphs with negative weights	$O(mn)$	Only for graphs with negative weights
A*	Single-pair SSP	Directed simple graphs	$O(n)$	Heuristic-based search, can be not optimal
FRA*	Single-pair SSP	Directed simple graphs	$O(n)$	Can be used in dynamic environments
Yen's	KSP	Directed simple graphs	$O(kn(m + n \log n))$	Easiest to implement, but slower than some of the other KSP algorithms
Katoh's	KSP	Undirected simple graphs	$O(k(m + n \log n))$	Best worst-case is complexity и fast runtime but only works on undirected graphs.
Eppstein's	KSP	Cyclic graphs	$O(k(n + \log k) + m + n \log n)$	Only for cycled graphs
K*	KSP	Cyclic directed graphs	$O(k + m + n \log n)$	Heuristic-based On-the-fly pathfinding

4.3 Предлаган подход за намиране на маршрут

Предлаганият подход за намиране на маршрут е част от услугата за намиране на маршрути, базирана на BIM. Оптималният път се изчислява в три стъпки.

- Първо, намират се K -най-късите пътища до целевата точка, преминавайки през мултиграф на свързаност, извлечен от топологичния модел на BIM (макроскопично намиране на маршрут).
- Второ, намират се фините пътища, като се използва геометричният модел на BIM, базиран на граници (микроскопично намиране на пътя).
- Накрая, изчислява се оптималният за потребителя маршрут с най-висок общ ранг, като се използва техника за многоцелева оптимизация.

За целта се използва BIM, описан в Глава 3. Този модел трябва да представя топологията и геометрията на сградата, както и контекстуалната и семантичната информация, необходима за оптимално намиране на пътя. Информацията, необходима за намиране на пътя, ще бъде извлечена директно от BIM.

Предлаганият модел се фокусира върху възможните елементи на пътя и свързаността между тези елементи. Целта е да се предостави възможност за описание на свързаността на различни нива на йерархията, включително моделиране на административната структура на сградата. Този BIM представя 3D измеренията на елементите на сградата и обектите, намиращи се в сградата (препятствия, опасности и ориентири), състоянието на сензорите, разположени в сградата, предпочитанията на потребителите и вида на увреждането, семантиката на сградата и ограниченията (пространствени, времеви и специфични за потребителя). Ограниченията помагат да се получи информация за това кои компоненти на сградата са недостъпни за конкретен потребител.

Причината за използването на този BIM, а не на някой от съществуващите стандартни BIM, е невъзможността на този етап BIM да съдържа цялата информация, необходима за надеждното намиране на пътя в големи обществени сгради. BIM стандарти като IFC и 3D модели на сгради като CityGML са публикувани, за да опишат 3D геометрията и семантиката на сградите, но им липсват важни характеристики, необходими за надеждна навигация в закрити помещения. CityGML има ограничена поддръжка както за фиксирани, така и за подвижни препятствия. Нито един от стандартите BIM не поддържа потребителски ограничения и семантика.

IndoorGML е стандарт за моделиране на вътрешни пространства. Той дефинира следната информация за вътрешните пространства: контекст и ограничения на навигацията, подразделения на пространствата и видове свързаност между пространствата, геометрични и семантични свойства на пространствата и навигационни мрежи. IndoorGML може да представя препятствия като пространства, които не са достъпни за навигация. Този стандарт не може да представи необходимата йерархия на сградите, например административната структура на дадена сграда.

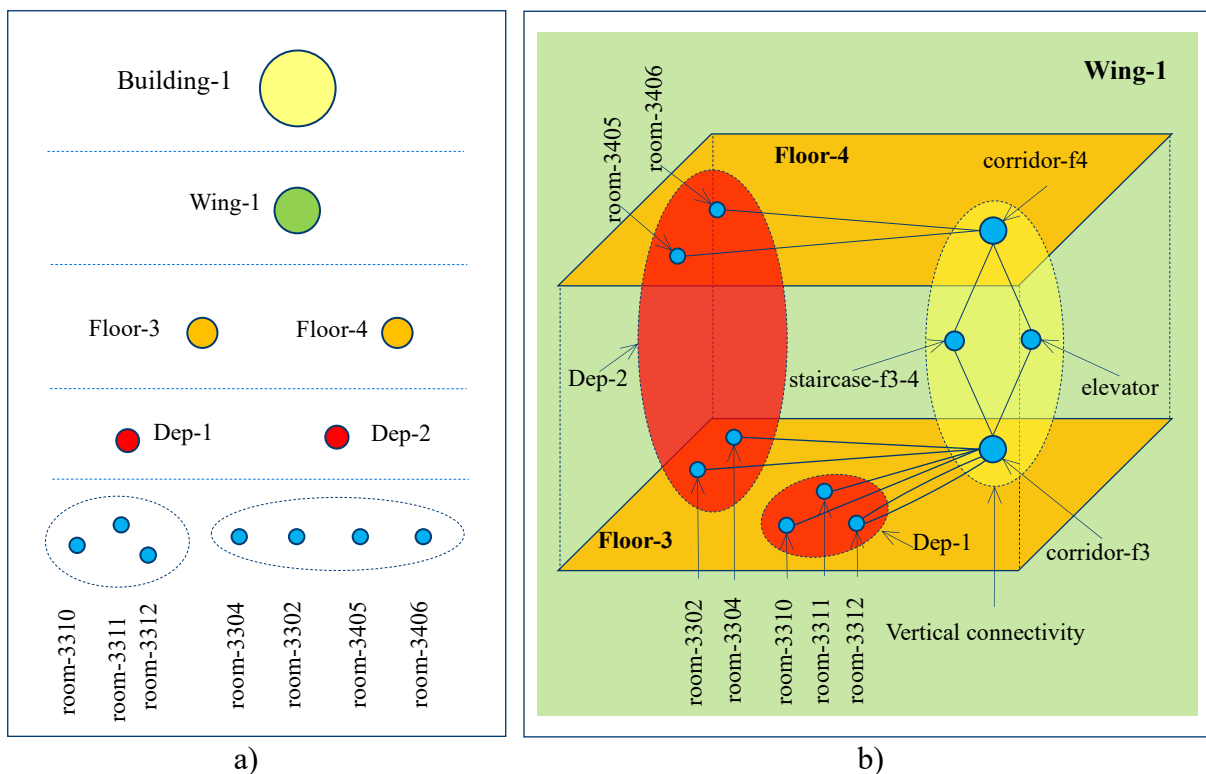
Никой от тези BIM и 3D модели на сгради не позволява достатъчно богато семантично описание на компонентите на сградите, както и на семантиката на потребителя.

4.3.1 Моделиране на свързаността

Използват се няколко нива на абстракция за моделиране на йерархията на сградата. Йерархичният дизайн е лесно приложим, може да представя големи сгради със сложна свързаност, обикновено се мащабира добре, намалява използването на памет и подобрява времето, необходимо за намиране на оптимален път.

Предполага се, че сградите имат крила. Всяко крило има един или няколко етажа. Свързаността на сградата е моделирана въз основа на елементи на пътя. В предлагания модел елементът на пътя е абстрактен компонент, който представя всяка част от пътя (възел или връзка). Възелът е клас, който описва пространствата в сградата, през които потребителят може да премине, ако има съответните права. Възлите се разделят на хоризонтални възли (стаи, коридори) и вертикални възли (стълбища, асансьори, ескалатори). По този начин хоризонталната и вертикалната свързаност могат да бъдат бързо извлечени за всяко ниво на йерархията. Стаите и коридорите са част от етажите. Големите стаи и коридори могат да бъдат разделени на подчасти (зони), за да представят по-точно вътрешното пространство. Връзките описват свързаността между възлите. Връзките са реални и виртуални врати, прозорци и отвори.

Използва се насочен мултиграф, за да се представи всяка свързаност между компонентите на сградата под формата на възли. Всяка връзка свързва два възела. Връзките могат да бъдат насочени или ненасочени. Между два възла може да има произволен брой насочени и ненасочени връзки. Свързаността на компонентите на сградата може да бъде описана на няколко йерархични нива: сграда, крила, етажи, вертикална свързаност и административни структури (виж Фиг. 4.1).



Фиг. 4.1 Модел на свързаност: (а) нива на абстракция; (б) нива на йерархия

Най-долното ниво на йерархията съдържа всички възли и връзки и се описва с помощта на еднослоен мултиграф. На най-високото ниво на йерархията графиката съдържа само един възел – цялата сграда е моделирана като един възел (Сграда-1). Това ниво се използва за намиране на свързаност между две сгради и при търсене на входи и изходи на сграда. В предлагания модел се използва специален възел, наречен „на открито“, който представлява пространството около сградите. Свързаността на компонентите на по-ниските йерархични нива (крило, етаж, вертикална свързаност) се описва с помощта на подграфи, чиито възли са подмножество от възлите на цялата графика. Благодарение на йерархичното и обектно-ориентирано описание на свързаността в BIM, получаването на подграф на различни йерархични нива е лесно и бързо.

Фиг 4.1б показва йерархичните нива, които се използват за описание на свързаността в многоетажна сграда с едно крило. За по-голяма простота са показани само два етажа на сградата с имена „Етаж-3“ и „Етаж-4“. Свързаността между двата етажа е моделирана чрез йерархично ниво „вертикална свързаност“. Това ниво трябва да съдържа всички възли от вертикален тип за съответното йерархично ниво, както и хоризонталните възли, чрез които е възможен достъп до тях. В конкретния случай вертикалните възли са тези, които описват асансьора и стълбището между двата етажа („elevator“, „staircase-f3-4“). Хоризонталните възли са коридорите на двата етажа („коридор-етаж 3“ и „коридор-етаж 4“). На йерархичното ниво на етажите подграфът съдържа всички хоризонтални възли от съответния етаж. Благодарение на класа `AdministrativeStructure` е възможно да се генерира подграф, който описва свързаността в административна единица, например отдел или болнично отделение. В този пример има два отдела, наречени „Dep-1“ и „Dep-2“. „Dep 1“ съдържа възли, разположени на 3-тия етаж, а „Dep 2“ съдържа възли, разположени на 3-тия и 4-тия етаж.

4.3.2 Изграждане на модел на свързаност

Големите обществени сгради се описват с разредени графи, тъй като броят на връзките между възлите е много по-малък от този, получен за пълни графи. Моделът, който описва свързаността, трябва да отговаря на следните по-важни условия:

- Да бъде ефективен при разредени графи;
- Да позволява бързо добавяне на възли и връзки за обединяване на два или няколко подграфи;

- Да позволява моделиране на всякаква свързаност между елементите на сградата. За разредени графи е подходящо да се опише свързаността чрез списък от съседни възли.

При предлагания модел, за да се получи по-бърз достъп до информацията, свързаността се моделира с помощта на две хеш карти. Първата карта, наречена *nodes*, описва съседните възли за всеки възел. Втората карта, наречена *links*, описва връзките, които свързват възела с неговите съседни възли. За хеш картите основните операции като *get*, *put*, *contains key* и *size* имат сложност $O(1)$. Този модел на свързаност отговаря на всички споменати изисквания.

Възлите и връзките могат да бъдат реални или виртуални. Виртуалните възли се използват за моделиране на конкретни зони, като например рецепция в лобито на хотел, и за реализиране на сценарии за намиране на пътя „възел към възли“ и „възли към възел“. Виртуалните връзки се използват за представяне на отворени пространства между стаи и за описание на виртуални връзки между реални възли, например „стълбище-коридор“.

Нека да опишем сценария за намиране на път, в който началната точка е възел, наречен „стая-3312“, а целевият възел е „стая-3405“ (фигура 4.1б). Възелът „стая-3312“ е част от „Дер-1“, а възелът „стая-3405“ е част от „Дер-2“. Можем да получим възможни пътища между тези възли, като използваме различни графи:

- граф на най-долното ниво на йерархията (свързаност за цялата сграда);
- граф, който обединява подграфи за нивата „Етаж-3“, „Етаж-4“ и „Вертикална свързаност“ на йерархията;
- граф, който обединява подграфи за нивата „Дер-1“, „Дер-2“ и „Вертикална свързаност“ на йерархията.

Фиг. 4.2 показва модели на свързаност за йерархичните нива „Дер-1“, „Дер-2“ и „Вертикална свързаност“, реализирани с помощта на хеш карти.

Има четири възможни пътя между началната и крайната точка, защото „Стая-3312“ има две точки за достъп („врата-1-R-3312“ и „врата-2-R-3312“). Всеки възможен път се състои от поредица от имена на възли и връзки. Възможността за обединяване на няколко модела на свързаност в един ни позволява да работим само с част от свързаността на сградата – тази, която е необходима за намиране на конкретен път.

Department 1 (Dep-1) Nodes: {room-3310=[corridor-f3], room-3312=[corridor-f3], room-3311=[corridor-f3], corridor-f3=[room-3310, room-3311, room-3312, room-3312]} Links: {room-3310=[door-R-3310], room-3312=[door-1-R-3312, door-2-R-3312], room-3311=[door-R-3311], corridor-f3=[door-R-3310, door-R-3311, door-1-R-3312, door-2-R-3312]}	Department 2 (Dep-2) Nodes: {room-3302=[corridor-f3], room-3304=[corridor-f3], room-3406=[corridor-f4], room-3405=[corridor-f4], corridor-f3=[room-3302, room-3304], corridor-f4=[room-3405, room-3406]} Links: {room-3302=[room-R-3304], room-3304=[room-R-3304], room-3406=[room-R-3406], room-3405=[room-R-3405], corridor-f3=[door-R-3302, door-R-3304], corridor-f4=[door-R-3405, door-R-3406]}
Vertical connectivity Nodes: {staircase-f3-4=[corridor-f3, corridor-f4], elevator=[corridor-f3, corridor-f4], corridor-f3=[elevator, staircase-f3-4], corridor-f4=[elevator, staircase-f3-4]} Links: {staircase-f3-4=[vd, vd], elevator=[elevator-d3, elevator-d4], corridor-f3=[elevator-d3, vd], corridor-f4=[elevator-d4, vd]}	Feasible paths (from room-3312 to room-3405) Path 1: room-3312 room-1-R-3312 corridor-f3 elevator-d3 elevator elevator-d4 corridor-f4 door-R-3405 room-3405 Path 2: room-3312 room-2-R-3312 corridor-f3 elevator-d3 elevator elevator-d4 corridor-f4 door-R-3405 room-3405 Path 3: room-3312 room-1-R-3312 corridor-f3 staircase-f3-4 corridor-f4 door-R-3405 room-3405 Path 4: room-3312 room-2-R-3312 corridor-f3 staircase-f3-4 corridor-f4 door-R-3405 room-3405

Фиг. 4.2 Намиране на възможните пътища, обединяващи подграфи за различни нива на йерархията: „Dep-1“, „Dep-2“ и „Вертикална свързаност“ (виж Фиг. 4.1б)

4.3.2.1 Инициализиране и разширяване на мултиграф

За изграждане на модела на свързаност се използват следните два алгоритъма:

- *addNode* – вмъква нов ключ в хеш картата на възлите или актуализира стойността за този ключ.
- *addLink* – вмъква нов ключ в хеш картата на връзките или актуализира стойността за този ключ.

Чрез тези алгоритми е възможно да се инициализират модели на свързаност или да се обединят няколко модели на свързаност (подграфи) в един.

Алгоритъмът *addNode* (Фиг. 4.3а) се използва за моделиране на връзката между съседни възли (аргумент *newAdjacentNodes*) и възел, определен като ключ (аргумент *node*). Ако ключът съществува (ред 1), новият списък (реди 2 и 3) се добавя към текущия списък на съседни възли. Ако ключът не съществува, се създава нова асоциация (ред 5). Алгоритъмът има времева сложност $O(1)$.

Algorithm *addNode*

Associates specified key (*node*) with the specified value (*newAdjacentNodes*). If the map contains the key, the new value is added to the current value.

Input: Key name: *node* (*String*)

List of adjacent nodes: *newAdjacentNodes* (*ArrayList*)

```

1 if nodes contains the key node then           O(1)
2   adjacentNodes = get adjacent nodes for key node O(1)

3   add newAdjacentNodes to adjacentNodes         O(1)
4 else
5   make a new association node-newAdjacentNodes O(1)
6 end if

```

а)

Algorithm *addLink*

Associates specified key (*node*) with the specified value (*newAdjacentLinks*). If the map contains the key, the new value is added to the current value.

Input: Key name: *node* (*String*)

List of adjacent links: *newAdjacentLinks* (*ArrayList*)

```

1 if (links contains the key node) then           O(1)
2   adjacentLinks = get adjacent links for key node O(1)

3   add newAdjacentLinks to adjacentLinks         O(1)
4 else
5   make a new association node-newAdjacentLinks O(1)
6 end if

```

б)

Фиг. 4.3 Инициализиране или разширяване на графика: (а) алгоритъм *addNode*; (б) алгоритъм *addLink*

Алгоритъмът *addLinks* (Фиг. 4.3б) се използва за описание на връзките (аргумент *newAdjacentLinks*), чрез които се достига до съседните възли за даден възел (аргумент *node*). Ако този възел (ключ) съществува (ред 1), новият списък (редици 2 и 3) се добавя към текущия списък с връзки. Ако ключът не съществува, се създава нова асоциация (ред 5). Алгоритъмът има времева сложност $O(1)$.

4.3.2.2 Премахване на Възел или Връзка от мултиграф

Два алгоритъма, наречени *removeNode* и *removeLink*, се използват за моделиране на достъпността на възлите и връзките за всеки конкретен потребител. Алгоритъмът за намиране на път взема предвид ограниченията, свързани с извънредни ситуации, сигурност, време и специфични за потребителя ограничения от BIM класа *Constraints*. Този клас описва ограниченията, които могат да бъдат наложени на всеки елемент от пътя – възел или връзка. Вместо да проверяваме многократно достъпността на всеки възел и връзка по време на изчисляването на възможните пътища, ние премахваме от модела на свързаност всички възли и връзки, които не са достъпни за конкретен потребител в определен момент. Това решение позволява намирането на пътя да се основава само на данните в модела на свързаност. Ако навигационната услуга има разпределена архитектура, планирането на пътя може да се реализира от страна на клиента. В този случай сървърът трябва да върне само модела на свързаност, тъй като информацията за достъпността на възлите и връзките не е необходима.

За да се премахне възел от алгоритъма на модела на свързаност, се използва *removeNode* (Фигура 4.4а). Времевата сложност на алгоритъма е $O(1)$. Алгоритъмът

removeLink (Фиг. 4.4б) се използва за блокиране на достъпа до възлите, които са недостъпни чрез връзката, посочена като аргумент.

Algorithm *removeNode*

Removes specified key (*node*) from hash maps *nodes* и *links*

Input: Node name: *node* (*String*)

```
1  remove key node from nodes                O(1)
2  remove key node from links                 O(1)
```

a)

Algorithm *removeLink*

Blocks specified link.

Input: Link name: *link* (*String*)

```
1  for ∀ key in links do                        O(1)
2    adjacentLinks = get adjacent links for this key  O(1)
3    id = get position of link in adjacentLinks      O(b)
4    if id ≠ -1 then
5      adjacentNodes = get adjacent nodes for this key  O(1)
6      secondNode = set name of node with index id to "blocked" O(1)
7      break for
8    end if
9  end for
10 if ∃ secondNode then
11   adjacentLinks = get adjacent links for key secondNode  O(1)
12   id = get position of link in adjacentLinks          O(b)
13   if id ≠ -1 then
14     adjacentNodes = get adjacent nodes for key secondNode O(1)
15     set name of node with index id to "blocked"        O(1)
16   end if
17 end if
```

б)

Фиг. 4.4 Премахване на възел или връзка от модела на свързаност: (а) алгоритъм *removeNode*; (б) алгоритъм *removeLink*

Това изисква премахването на всички връзки с това име от връзките на хеш картата, както и съответните им възли в възлите на хеш картата. Тъй като съседните възли и връзки се описват чрез списъци от тип *ArrayList*, операцията *remove* има сложност $O(n)$. По тази причина, вместо операцията *remove*, се използва операция *set*, която има сложност $O(1)$ (редове 6 и 15). Тя променя само името на възлите, до които потребителят може да стигне, преминавайки през посочената връзка. Критичните по отношение на времето операции са тези, които проверяват дали списъкът с връзки съдържа връзка с посоченото име (редове 3 и 12). Използва се методът *indexOf*, който има сложност $O(b)$, където b е броят на връзките към съседни възли. Методът връща (-1), ако няма съвпадение. Сложността на алгоритъма е $O(nb)$, където n е броят на ключовете в хеш картите. При търсене на пътища в големи сгради $b \ll n$.

4.3.2.3 Вмъкване на Виртуални Възли В мултиграф

Най-често навигацията е към определен целеви възел. В някои случаи потребителите искат да достигнат до конкретен възел, който отговаря на определено условие. Например, ако потребителят търси „магазин за мъжки обувки“ в търговския център. Най-вероятно в търговския център има няколко магазина за мъжки обувки и всеки от тях може да бъде целеви възел. Това изисква намирането на всички целеви възли, които отговарят на условието. От всички възможни възли този, който ще бъде избран, се определя от алгоритъма за намиране на оптималния път за конкретен потребител. За да се реализира възможността за намиране на път „от възел до възли“, без да се променя програмният код на алгоритъма за намиране на път, се използва виртуален целеви възел. Този виртуален възел се свързва с виртуални връзки към всички реални целеви възли. След като бъдат намерени K възможни пътя до нов целеви възел, виртуалният възел се премахва. За да бъде възможно условното търсене, името на всеки възел трябва да съдържа ключови думи, които го описват, например „мъжки-обувки-магазин-f2“. За да се провери дали името на възела отговаря на конкретно условие, се използва търсене по шаблон. Търсенето по шаблон се реализира с помощта на регулярни изрази.

Фиг. 4.5 показва псевдокода на алгоритъма *insertTargetVirtualNode*. Входните аргументи са два: името на виртуалния целеви възел (*vnnode*) и масив, който съдържа имената на целевите възли (*target_nodes*). Всеки елемент в този масив съдържа името на възела или регулярен израз. Ако името на възела започва с низ „regex“, то това е регулярен израз, който описва шаблона за търсене на възела, например:

```
target_nodes = {"room – 3304", "room-3306", "room-3312"},
```

```
target_nodes = {"regex(^WC – men.* f (3|4).* $)"}
```

В първия пример целевите възли са три стаи. Във втория пример целевите възли са всички мъжки тоалетни на 3-ти или 4-ти етаж. Алгоритъмът започва с извличане на броя на елементите в *target_nodes* (ред 2). За всеки елемент в *target_nodes* (ред 3) се проверяват всички ключове в картата *nodes* (ред 5), за да се види дали името на възела е равно на *targetNode* (ред 11) или дали името на възела съвпада с шаблона (ред 9). Ако има съвпадение (ред 13), виртуалният възел се добавя към съседните възли (ред 16) и името на връзката между тях се задава като „vd“ (ред 17).

Algorithm *insertTargetVirtualNode*

Inserts target virtual node which name should contains word virtual.

Input: Virtual node name: *vnode* (*String*)

List of target nodes: *target_nodes* (*String* array).

If target node name contains a word “regex” then method finds all nodes that match regular expression.

```
1  set nodeList и linkList as empty ArrayLists
2  n = number of elements in target_nodes                                O(1)
3  for i=0 to n-1 do
4      targetNode = get i-th element in target_nodes                    O(1)
5      foreach node in nodes do
6          if targetNode contains regular expression then
7              template = get regular expression template
8              make pattern и matcher objects
9              searchFlag = node matches template                        O(k)
10         else
11             searchFlag = node equals targetNode                      O(1)
12         end if
13         if searchFlag = true then
14             adjacentNodes = get adjacent nodes for key node          O(1)
15             adjacentLinks = get adjacent links for key node          O(1)
16             add vnode to adjacentNodes                                O(1)
17             add “vd” to adjacentLinks                                  O(1)
18             add adjacentNodes to nodeList                             O(1)
19             add adjacentLinks to linkList                             O(1)
20         end if
21     end foreach
22 end for
23 if nodeList is not empty then                                          O(1)
24     if nodes contains key vnode then                                  O(1)
25         add nodeList to the value for key vnode in nodes            O(1)
26         add linkList to the value for key vnode in links            O(1)
27     else
28         make a new association vnode-nodeList in nodes              O(1)
29         make new association vnode-linkList in links                O(1)
30     end if
31 end if
```

Фиг. 4.5 Алгоритъм *insertTargetVirtualNode*

Алгоритъмът продължава да добавя всички съседни възли и връзки към списъците *nodeList* и *linkList* (редове 18 и 19). След обработката на всички елементи в *target_nodes*, алгоритъмът проверява дали има съвпадения (ред 23). Ако виртуалният възел вече съществува (ред 24), списъците със съседни възли и връзки се добавят към текущите им стойности в картите *nodes* и *links* (редове 25 и 26). В противен случай, нова асоциация за виртуалния възел и неговата стойност се добавя към хеш картите *nodes* и *links* (редове 28 и 29). Алгоритъмът позволява масивът *target_nodes* да съдържа произволна комбинация от имена на възли и редовни изрази. В този алгоритъм най-бавната операция е търсенето с помощта на

шаблон (ред 9). Най-лошият случай на сложност на алгоритъма е $O(n)$, ако не се използва регулярен израз, и $O(nk)$ в противен случай, където n е броят на целевите възли. Параметърът k е максималният брой символи в имената на възлите.

4.3.3 Изчисляване на Възможните пътища

Нека да намерим K -най-късите пътища от началната точка s до крайната точка t . Пътят p_{st} се дефинира като поредица от възли и връзки между начален възел и крайни възли, $[p_{st} = \{n_s, l_{s i}^k, n_i, \dots, l_{j t}^k, n_t\}]$, където l_{ij}^k е k -тата връзка между възел n_i и възел n_j . Всеки два съседни възела могат да бъдат свързани с една или няколко насочени или ненасочени връзки. Приемаме, че на макроскопично ниво всички ребра имат еднаква тежест. Цената $d(s, t)$ за достигане от възел s до възел t е равна на броя връзки, през които трябва да мине потребителят (дължината на пътя). Целта е да се сведе до минимум броят връзки, през които минава потребителят. Генерираните пътища са подредени от най-късия към най-дългия.

Използваме специално проектирана бърза реализация на алгоритъма Breadth First Search (BFS). За крайни графи алгоритмите BSF са пълни и оптимални за връзки с единична стъпка на разходите. За да се ускори изчисляването на пътя, се използват подходящи типове данни, като колекции и множества. Алгоритъмът *calculatePaths* (Фиг. 4.6) може да намери K възможни маршрута в насочени, не насочен, цикличен и ацикличен мултиграф.

Algorithm	<i>calculatePaths</i>	
	Finds K feasible paths from the starting node to the target node.	
Input:	Connectivity graph: <i>graph</i> (<i>nodes</i> , <i>links</i>) Name of start point: <i>start</i> (<i>String</i>), Name of target point: <i>target</i> (<i>String</i>) Find simple paths flag: <i>simplePathsFlag</i> (<i>boolean</i>) Max length of feasible paths: <i>MAX_PATH_LENGTH</i> (<i>int</i>) Number of feasible paths to find: <i>K</i> (<i>int</i>)	
Output:	List of feasible paths: <i>FeasiblePaths.nodes</i> , <i>FeasiblePaths.links</i> (<i>ArrayList</i>)	
Objects:	Temporal paths: <i>visitedNodes</i> и <i>visited_Links</i> (<i>ArrayList<LinkedList></i>) The algorithm uses following objects: <i>current_nodes</i> , <i>current_links</i> , <i>clone_current_nodes</i> , <i>clone_current_links</i> , <i>paths_to_remove</i> (<i>LinkedList</i>); <i>visitedNodes</i> , <i>visitedLinks</i> (<i>ArrayList</i>) The object <i>paths_to_remove</i> holds indexes of infeasible paths, which should to be removed.	
Initialization:	<i>iterations</i> = 0, set <i>currentLinks</i> = $\emptyset$, add <i>start</i> node to <i>current_nodes</i> , add <i>current_nodes</i> to <i>visited_nodes</i> , add <i>current_links</i> to <i>visited_links</i>	
1	while $N = \text{size of } \textit{visited_nodes} \neq 0$ do	$O(1)$
2	for $i = 0$ to $N - 1$; $i++$ do	
3	$\textit{current_nodes} = i\text{-th element in } \textit{visited_nodes}$	$O(1)$
4	$\textit{current_links} = i\text{-th element in } \textit{visited_links}$	$O(1)$
5	$\textit{last_node} = \text{get и remove last node from } \textit{current_nodes}$	$O(1)$
6	$\textit{targetFlag} = \textit{last_node} \text{ equals } \textit{start node}$	$O(1)$
7	$\textit{loopFlag} = \textit{current_nodes.contains last_node}$	$O(m)$
8	add $\textit{last_node}$ to $\textit{current_nodes}$	$O(1)$

```

9      removeFlag = false
10     if targetFlag = true then
11         removeFlag = true
12         add current_nodes to FeasiblePaths.nodes                                O(1)
13         add current_links to FeasiblePaths.links                            O(1)
14         if number of feasible paths = K then
15             stop (return)
16         end if
17     else if loopFlag = true then
18         if simplePathsFlag = true then removeFlag = true
19         else
20             last_link = get last link in current_links                        O(1)
21             if current_links \ last_link  $\cap$  last_link  $\neq \emptyset$  then removeFlag = true    O(m)
22             end if
23         end if
24     end if
25     if removeFlag = true then add i to paths_to_remove и continue end if
26     next_nodes = get from graph adjacent nodes for last_node                O(1)
27     if next_nodes =  $\emptyset$  then add to graph on-the-fly missing last_node и reload next_nodes end if    O(1)
28     next_links = get from graph adjacent links for last_node                O(1)
29     possibleNewPaths = size of next_nodes                                    O(1)
30     for j = 0 to possibleNewPaths - 1 do
31         node = get j-th element in next_nodes                                O(1)
32         link = get j-th element in next_links                                O(1)
33         clone_current_nodes = current_nodes, clone_current_links = current_links
34         clone current_nodes, clone current_links                            O(b)
35         add node to clone_current_nodes, add link to clone_current_links      O(1)
36         if j > 1 then
37             add clone_current_nodes to visited_nodes                        O(1)
38             add clone_current_links to visited_links                        O(1)
39         end if
40     end for j
41 end for i
42 remove all infeasible paths with indexes in paths_to_remove                O(R)
43 if ++iterations > MAX_PATH_LENGTH then stop (return) end if
44 end while

```

Фиг. 4.6 Алгоритъм *calculatePaths*

Алгоритъмът *calculatePaths* може да намери *K* възможни пътя в насочени, ненасочени, циклични и ациклични мултиграфи. Той използва два списъка, *visited_nodes* и *visited_links*, от тип *ArrayList*, за да съхранява временни пътища като поредица от посетени възли и връзки (*LinkedList*). В етапа на инициализация *visited_nodes* съдържа само един път с един елемент – начален възел. Тези два списъка във всеки даден момент съдържат части от пътищата между началния и целевия възел. Някои от тези пътища ще бъдат премахнати, ако съдържат цикли във възлите или списъците с връзки. Друга част от временните пътища, описващи възможните пътища, ще бъде премахната от *visited_nodes* и *visited_links* и преместена в списъци, които описват възможните пътища (списъци с възли и връзки от клас *FeasiblePaths*). Основният цикъл (*while*) на алгоритъма (ред 1) проверява дали има пътища в *visited_nodes* и *visited_links*. Цикълът *while* се

повтаря D пъти, където D е максималното разстояние между началната и крайната точка. Всеки път се изпълнява програмният код от тялото на цикъла и се получават всички възможни пътища с равно разстояние d между началната и крайната точка, $d = [1 \div D]$. Следователно, получените пътища ще бъдат подредени от най-късия към най-дългия. Цикълът `while` може да бъде прекъснат в зависимост от стойностите на следните две константи:

- K - брой пътища, които трябва да бъдат търсени (ред 14). Ако $K = 1$, се получава най-късият път, а за $K \rightarrow \infty$ се изчисляват всички възможни пътища.
- `MAX_PATH_LENGTH` - този параметър определя максималната дължина на пътищата, които трябва да бъдат намерени. Търсенето на целевия възел може да бъде ограничено до определена зона на сградата (ред 43).

Статусът на временните пътища се проверява в тялото на цикъла `for` (ред 2): дали е достигнат целевият възел (ред 6) и дали пътят съдържа цикъл (ред 7). Ако целевият възел (ред 10) е достигнат, пътят се маркира за премахване (ред 11) и се добавя към възможните пътища (реди 12 и 13). Алгоритъмът се прекратява, ако текущият брой възможни пътища е достигнал стойността на K (ред 14). Ако временен маршрут съдържа цикъл (ред 17), алгоритъмът проверява дали се търсят само прости маршрути (ред 18). В противен случай (търсене на маршрути с цикли), пътя се маркира за премахване само ако съдържа повтарящи се връзки (ред 21). Ако текущият път трябва да бъде премахнат (ред 25), неговият индекс се запазва в списъка `paths_to_remove` (LinkedList). Ако стойността на `removeFlag` е `false`, се извличат съседните възли за `last_node` (ред 26) и връзките за `reached` (ред 28). Ако списъкът `next_node` е празен (ред 27), това означава, че липсва информация за свързаността за `last_node`. В този случай липсващата информация за свързаността се добавя към графика в движение. Броят на съседните възли определя колко нови пътеки (`possibleNewPaths`) са възможни за всяка временен маршрут (ред 29). Съдържанието на временните маршрути се актуализира от тялото на цикъла (ред 30). Съседен възел (ред 35) се добавя в края на временния маршрут и се генерира броят (`possibleNewPaths-1`) нови временни пътища в `visited_nodes` и `visited_links` (редове 37 и 38).

След актуализиране на съдържанието на всички временни пътища, всички R пътища трябва да бъдат премахнати от `visited_nodes` и `visited_links`, чиито индекси са съхранени в списъка `paths_to_remove`. Анализирани са два варианта на алгоритъма за премахване на невъзможни пътища. В първия вариант се използва

методът `remove (ArrayList)` за изтриване на елемент от *visited_nodes* и *visited_links*. Той има сложност $O(M)$, където M е броят на временните пътища. Следователно сложността на алгоритъма за премахване на всички R пътища е $O(RM)$. Вторият вариант на алгоритъма може значително да намали времето за премахване на невъзможните пътища. Той получава и едновременно премахва стойността на последния елемент в *visited_nodes* и *visited_links*. При условие че последният път не трябва да бъде изтрят, стойността на елемента, който трябва да бъде изтрят, се замества със стойността на последния елемент. За масиви от списъци сложността на операциите „получаване и изтриване на последния елемент“ и „задаване на стойност на елемент чрез неговия идентификатор“ е $O(1)$. Следователно вторият вариант има сложност $O(R)$ в най-лошия случай.

Сложността на предложения алгоритъм за намиране на път зависи от това колко разреден е графа. Ако графът е гъсто свързан, крайният възел не може да бъде далеч от изходния възел. Ако графът е разреден, крайният възел може да бъде много отдалечен. Сложността на изпълнението зависи от максималния брой връзки, съседни на всеки възел b . Има най-много b^d възли на разстояние d от началния възел. Ако разстоянието от началния възел до целевия възел е D , тогава сложността на изпълнението ще бъде $O(b^D)$. В предложения подход за намиране на път не е необходимо целия граф да се зарежда в паметта благодарение на йерархичното описание на свързаността на сградата. Необходимият модел на свързаност се получава чрез обединяване на няколко подграфи, които описват необходимата хоризонтална и вертикална свързаност. Това намалява броя на възлите, участващи в намирането на пътя многократно. Това също намалява стойността на параметъра b . Това прави алгоритъма приложим при търсене на пътища в големи сгради с комплексна хоризонтална и вертикална свързаност.

4.4 Експериментални резултати

Проведени са тестове на настолен компютър с процесор AMD Pro A8-8650B (4 ядра) с честота 3,2 GHz, 256 KB L1 кеш, 4 MB L2 кеш и 32 GB оперативна памет, работещ под Windows 10 Enterprise (64 бита). Времето за изпълнение се измерва с помощта на NetBeans IDE profiler. Алгоритъмът за намиране на пътя се тества в симулирана и реална среда на сграда.

Програмното осигуряване за намиране на маршрути в сгради е достъпно за сваляне от следния адрес:

<https://github.com/rs-soft-bg/KSP>

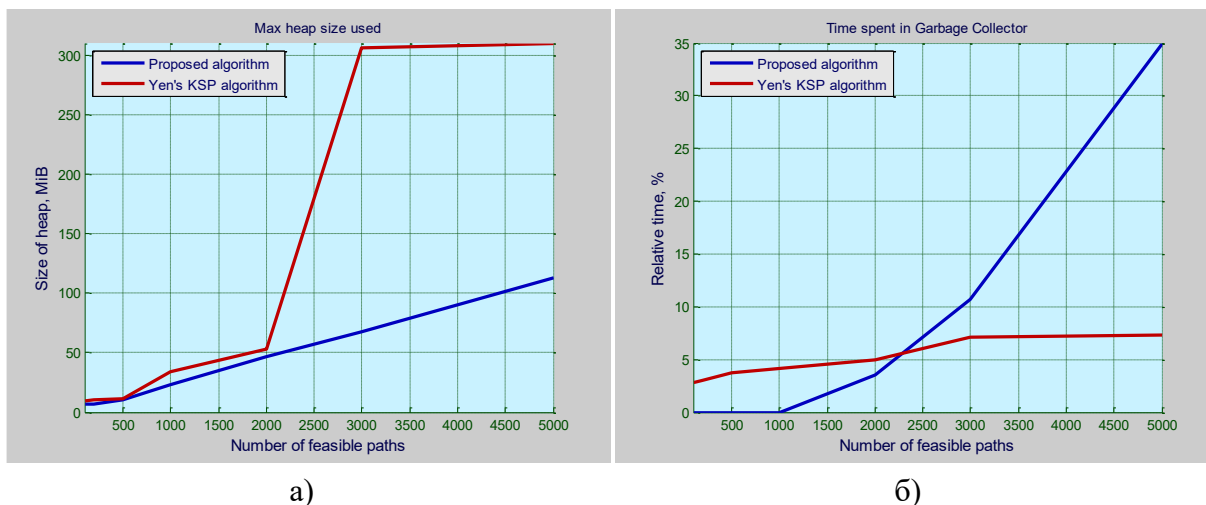
4.4.1 Намиране на път в пълни ненасочени графи

За да анализираме сложността и използваната памет на предложения алгоритъм, тествахме алгоритъма за намиране на маршрути в пълни ненасочени графи. Ако n е броя на възлите, броят на връзките m е $n(n - 1)/2$. Максималният брой пътища N от начален до целеви възел зависи от броя на възможните комбинации от пътища с различна дължина. Анализирахме времето за изпълнение на алгоритъма, необходимо за изчисляване на всички възможни пътища, когато броят на възлите n е между 4 и 11. Получените резултати са показани в Табл. 4.2.

Табл. 4.2 Средно време за изпълнение на търсени на маршрути при пълни графики за предложените алгоритми и алгоритъма на Йен.

n	m	Feasible paths	Proposed algorithm		Yen's algorithm
			Total runtime, ms	Remove the path time, ms (% of total runtime)	Total runtime, ms
4	6	5	0.34	0.026 (7.65%)	4.87
5	10	16	0.94	0.099 (10.53%)	8.16
6	15	65	2.42	0.299 (12.36%)	19.80
7	21	326	7.87	0.963 (11.89%)	65.07
8	28	1,957	29.86	3.594 (12.04%)	620.49
9	36	13,700	128.29	13.421 (10.46%)	19,698
10	45	109,601	4,659.04	79.144 (1.70%)	1,197,469
11	55	986,410	76,891.17	639.01 (0.83%)	110,170,175

Резултатите показват, че предложеният алгоритъм има по-добри резултати в сравнение с алгоритъма KSP на Yen във всички тестови случаи. Например, когато $n = 4$ (5 възможни пътя), предложият алгоритъм е 14,32 пъти по-бърз, а когато $n = 11$ (986,410 възможни пътя), той е 1433 пъти по-бърз. Най-бавната част от програмния код е алгоритъмът, който премахва цикличните временни маршрути и възможните маршрути от списъците *visited_nodes* и *visited_links*. Използвайки избрани колекции и методи, това време не надвишава 12,5% от общото изчислително време. Експериментите показват, че предложият алгоритъм използва по-ефективна динамичната памет в сравнение с алгоритъма на Yen за всички възможни стойности на K (виж Фиг. 4.7a). При алгоритъма на Yen за брой възможни пътища над 2000 размерът на необходимата динамичната памет нараства експоненциално. Предложият алгоритъм не извиква Garbage Collector (GC) за брой пътища под 1000 (виж Фиг 4.7б), но времето, прекарано в GC, ако броят на възможните пътища е над 1000, нараства експоненциално.



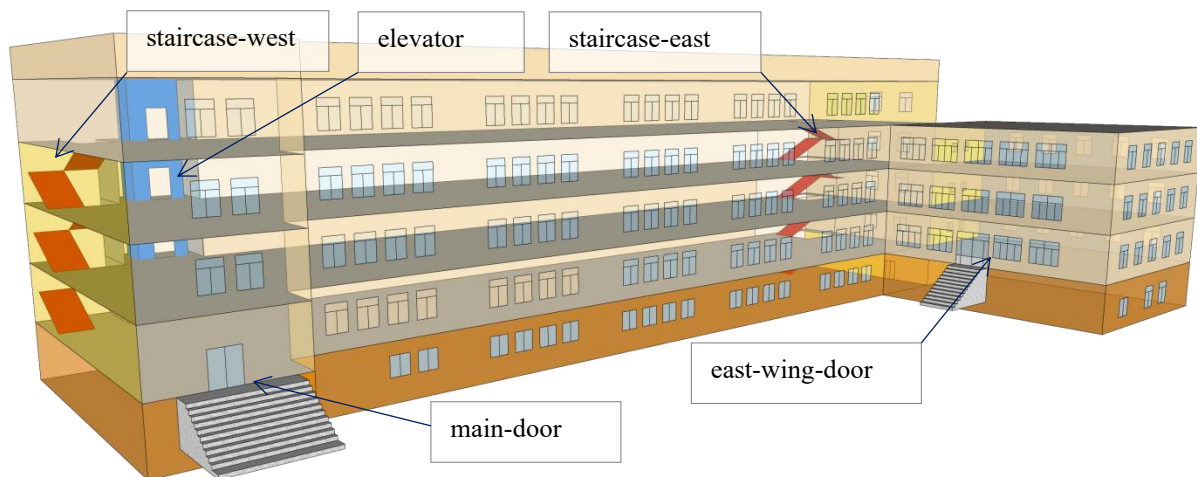
Фиг. 4.7 Максимален размер на използваната динамична памет (а) и относително време, прекарано в GC; (б) за пълен граф с $n=11$ и K от 100 до 5000

Времето, прекарано в GC при използване на алгоритъма на Yen, зависи малко от броя на възможните пътища и не надвишава 8% от общото изчислително време. Тъй като намирането на пътища в сгради не изисква K да е по-голямо от 50, предложеният алгоритъм превъзхожда алгоритъма на Yen както по отношение на производителността, така и по отношение на ефективността на използване на динамичната памет и GC.

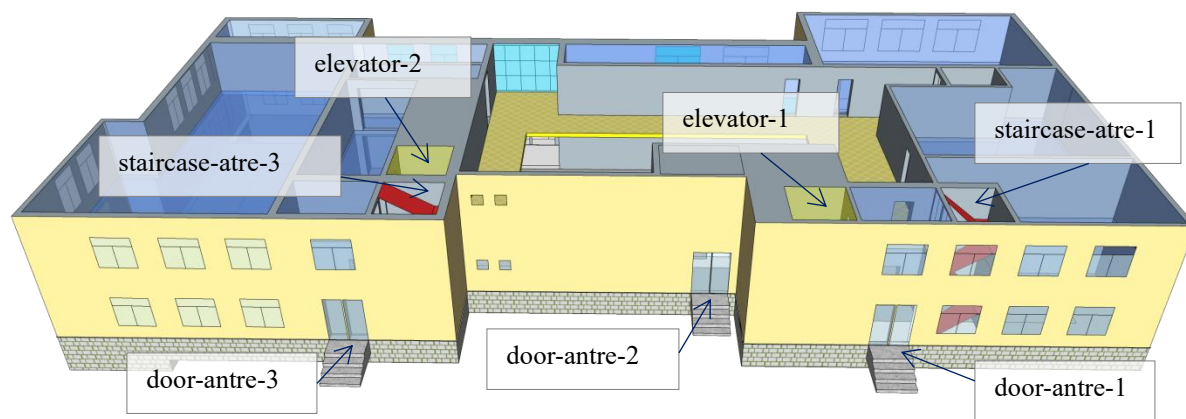
4.4.2 Намиране на маршрут в многоетажни сгради

Проведени са експерименти в две многоетажни сгради – 5-етажната сграда „Ректорат“ (Фиг. 4.8а) и 3-етажната сграда „Библиотека“ (Фиг. 4.8б) на Технически Университет – Габрово.

Моделът на свързаност се получава от BIM за различни нива на йерархията. Броят на възлите и връзките, описващи свързаността между елементите на сградата на различни нива на йерархията, е показан в Табл. 4.3.



a)



б)

Фиг. 4.8 Многоетажни тестови среди: а) сграда „Ректорат“; б) сграда „Библиотека“

Табл. 4.3 Брой възли и връзки за различни нива на йерархията

Hierarchy level	Number of nodes / links	
	“Rectorate” building	“Library” building
building	1 / 3	1 / 5
vertical connectivity	16/13	19/15
B1F	12 / 11	-
1F	25 / 29	13 / 11
2F	9 / 8	12 / 11
3F	22 / 21	7 / 6
4F	19 / 18	-
base	94 / 97	41 / 44

Целта на експериментите е да се анализират възможностите на предложения алгоритъм в различни сценарии за намиране на маршрути. Предложеният алгоритъм е тестван в три сценария за намиране на маршрути:

- намиране на маршрути в рамките на един етаж;
- намиране на маршрути между етажи;
- намиране на маршрути между сгради.

За всеки сценарий намирането на маршрут се реализира по три различни начина:

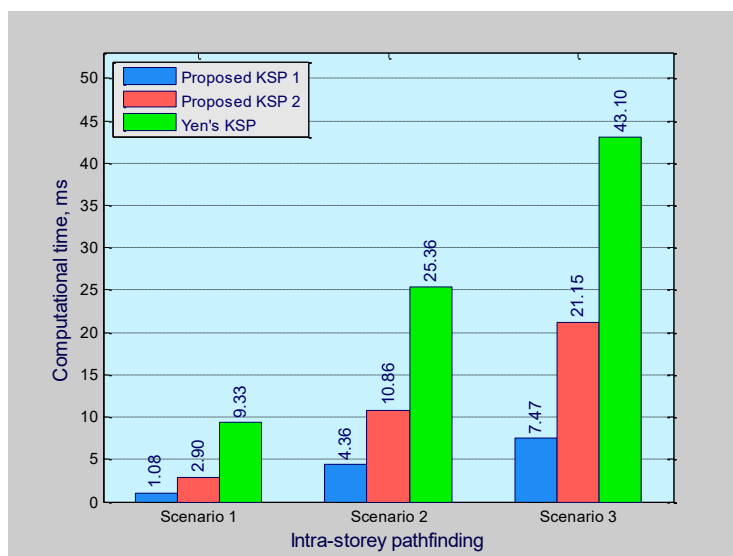
- чрез използване на предложения алгоритъм и обединяване на подграфи на различни йерархични нива (алгоритъм „предложен KSP 1“);
- чрез използване на предложения алгоритъм на базовото ниво на йерархията (алгоритъм „предложен KSP 2“);
- чрез използване на бързата версия на алгоритъма на Yen.

4.4.2.1 Намиране на път в рамките на един етаж

В сградата „Ректорат“ бяха проведени три експеримента:

- намиране на път от възел „стая-3304“ до възел „стая-3316“ (1 възможен път),
- намиране на път от възел „стая-3304“ до възел „асансьор“ (15 възможни пътя) и намиране на път от възел „стая-3312“ до всички тоалетни на 3-тия етаж (4 възможни пътя).

За да се опишат всички тоалетни, се използва виртуален целеви възел, наречен „regex(^WC.*f3)“. Получените резултати са показани на Фиг. 4.9.



Фиг. 4.9 Сравнение на времето за намиране на маршрути при сценарий за намиране на път в рамките на един етаж

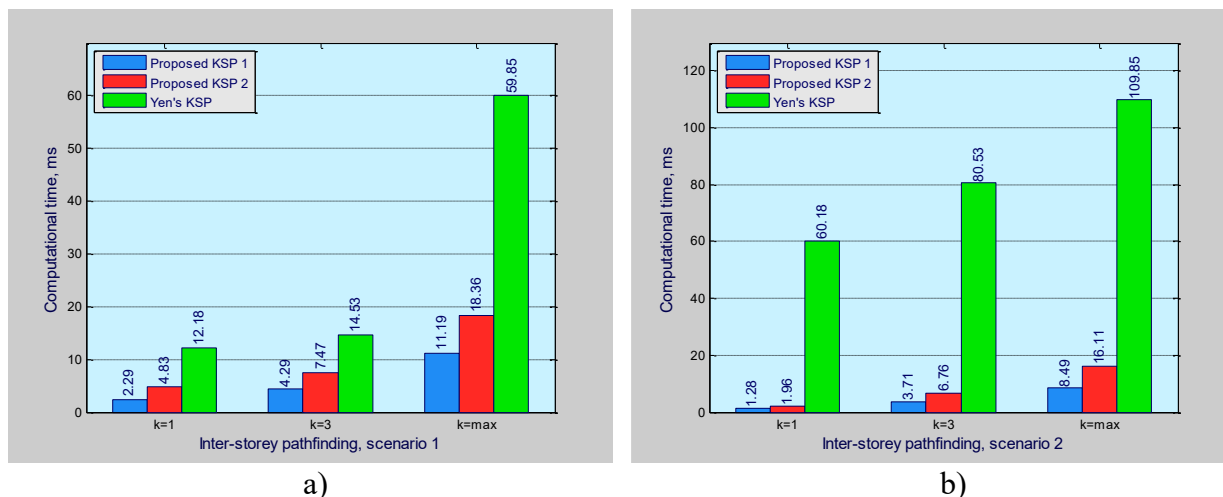
Във всички тестови сценарии използването на алгоритъма „предложен KSP 1“, който обединява подграфа за 3-тия етаж и подграфа за вертикална свързаност, изчислява възможните пътища за най-кратко време. Този алгоритъм е средно 7,53 пъти по-бърз от алгоритъма на Yen и средно 2,67 пъти по-бърз от „предложения KSP 2“, който използва граф за базовото ниво на йерархията.

4.4.2.2 Намиране на път между етажите на сграда

В сградата „Ректорат“ са проведени два експеримента:

- намиране на пътя от възел „стая-3414“ (ниво 4F) до възел „печатница“ (ниво 1BF) (75 възможни пътя) и
- намиране на пътя от възел „стая-3304“ до всички възможни мъжки тоалетни на 3-ти и 4-ти етаж (10 възможни пътя).

В този случай, за да се опишат всички целеви възли, се използва виртуален целеви възел с име „regex(^WC-men.*f(3|4).*\$)“. Анализирано е времето за намиране на различен брой възможни пътя ($K = 1, 3, MAX$). Получените резултати са показани на Фиг. 4.10. В този случай алгоритъмът „предложен KSP 1“ дава най-бързото изчислително време за всички стойности на параметъра K . При търсене на възможни пътища между пет етажа (експеримент 1) алгоритъмът „предложен KSP 1“ е средно 4,69 пъти по-бърз от алгоритъма на Yen. При търсене на възможни пътища между два етажа (експеримент 2) алгоритъмът „предложен KSP 1“ е средно 27,22 пъти по-бърз от алгоритъма на Yen.



Фиг. 4.10 Сравнение на време за намиране на път между етажите: а) тестов сценарий 1; б) тестов сценарий 2

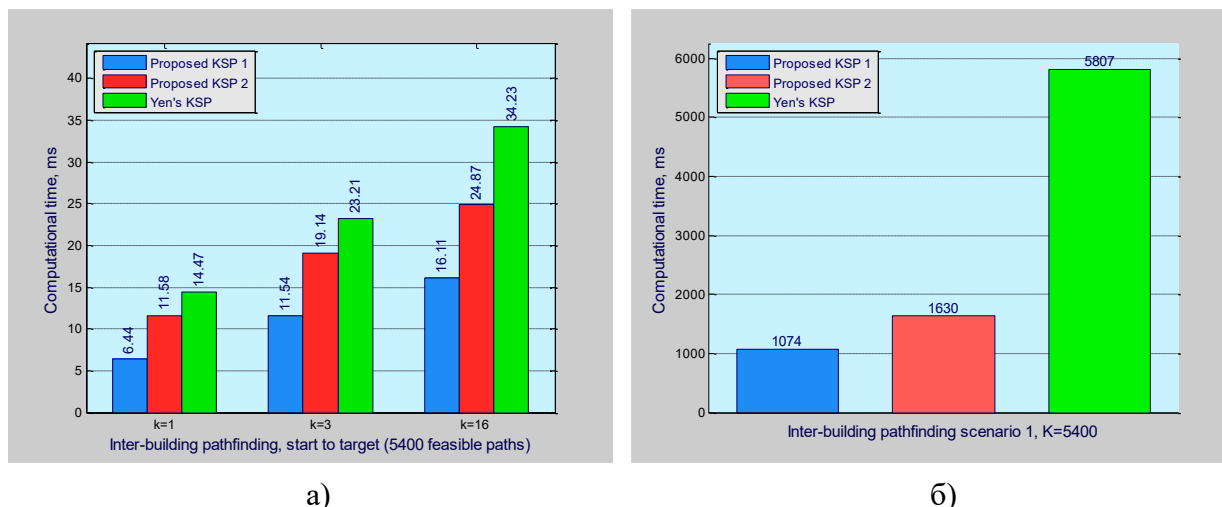
4.4.2.3 Търсене на маршрути между сгради

Предлаганият алгоритъм за търсене на маршрути позволява обединяване на моделите за свързаност на две или повече сгради, ако е необходимо. Това дава възможност търсене на маршрути, при които началната и крайната точка се намират в различни сгради.

Нека намерим възможните маршрути от „стая-3414“ (сграда „Ректорат“, етаж 4F) до „хранилище-4-f3“ (сграда „Библиотека“, етаж 3F). В този случай броят на възможните маршрути е 5400. За да намерим тези маршрути, използваме два тестови сценария:

- намиране на пътища от началната точка „стая-3414“ до крайната точка „хранилище-4-f3“;
- комбиниране на всички възможни маршрути, получени чрез търсене на маршрути от началната точка „стая-3414“ до открито пространство и от открито пространство до крайната точка „хранилище-4-f3“.

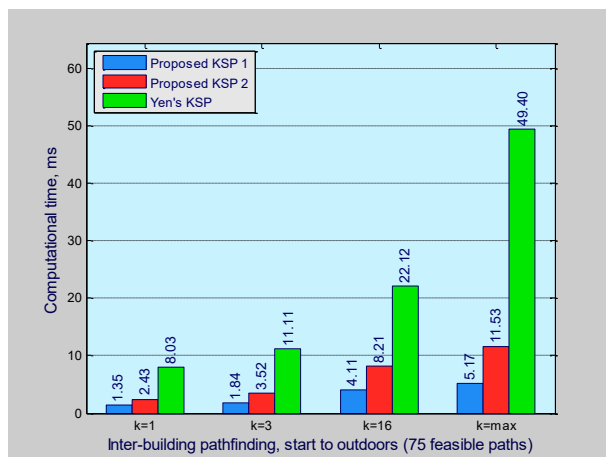
Резултатите, получени при прилагането на сценарий 1, са показани на Фиг. 4.11.



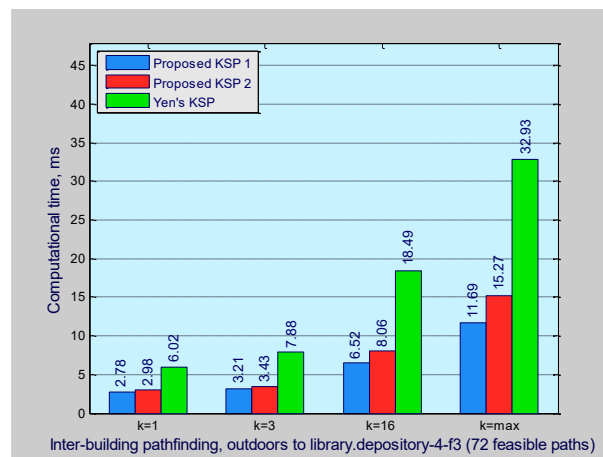
Фиг. 4.11 Сравнение на изчислителното време за сценарий 1 за намиране на път между сгради: а) $K = 1$, $K = 3$ и $K = 16$; б) $K = 5400$

Времето, необходимо за изчисляване на всички възможни 5400 пътя (Фиг. 4.11б) с помощта на „предложения KSP 1“, „предложения KSP 2“ и алгоритмите на Йен, е съответно 1074 ms, 1630 ms и 5807 ms. Когато броят на възможните пътища K се ограничи до 1, 3 и 16, времето за тяхното изчисляване се намалява значително. Например, ако $K = 16$, намалението е 66,7 пъти за алгоритъма „предложен KSP 1“, 65,5 пъти за алгоритъма „предложен KSP 2“ и 169 пъти за алгоритъма на Йен. Това означава, че алгоритъмът на Йен е много по-чувствителен към броя на търсените възможни пътища в сравнение с предложения алгоритъм.

Значително по-малко процесорно време е необходимо за изчисляване на всички възможни пътища при прилагане на сценарий 2. В този случай броят на възможните пътища от началната точка до външната среда е 75, а от външната среда до целевата точка е 72. Времето за изчисляване на всички възможни пътища се намалява до 16,86 ms за алгоритъма „предложен KSP 1“, 26,8 ms за алгоритъма „предложен KSP 2“ и 82,33 ms за алгоритъма KSP на Йен (виж Фиг. 4.12).



а)



б)

Фиг. 4.12 Сравнение на време за намиране на път между сгради при сценарий 2: а) намиране на пътища от възел „стая-3414“ до възел „на открито“; б) намиране на пътища от възел „на открито“ до възел „хранилище-4-f3“

4.5 Заключение

Разработен и тестван е алгоритъм за намиране на K възможни маршрути в големи обществени сгради. Алгоритъмът използва BIM, специално проектиран за навигация в многоетажни сгради със сложна свързаност. Този BIM описва йерархично всички компоненти на сградата и позволява извличане на свързаност на различни йерархични нива: сграда, крило, етаж, вертикална свързаност и административна структура. Това позволява намиране на пътеки, използвайки моделите на свързаност на отделни части от сградата, а не модела на свързаност за цялата сграда.

Предложеният модел на свързаност позволява описание на всякаква свързаност на възлите. Могат да се получат както ациклични маршрути, така и маршрути с цикли. Възможно е да се обединява свързаността от различни йерархични нива в един модел. Ако в паметта липсва свързаност за някой възел, алгоритъмът може да я добави към модела на свързаност в реално време. Възможно е едновременно търсене на няколко целеви възли.

Предложеният алгоритъм за намиране на пътя е оптимизиран по производителност чрез използване на подходящи програмни структури и методи. Алгоритъмът позволява да се получат K възможни пътя, които са сортирани по дължина – от най-късия до най-дългия. Намирането на пътя може да бъде спряно, ако са изчислени всички K възможни пътя или ако са изчислени всички пътя с предварително определена максимална дължина. Благодарение на йерархичното

описание на топологията на сградата, е възможно да се реализират различни видове намиране на пътя: намиране на път в рамките на етаж и между етажите; и намиране на път вътре в сградата и между сградите.

Експериментите, проведени с пълни ненасочени графики, показват, че предложеният алгоритъм е от 14 до 1433 пъти по-бърз от алгоритъма Yen. Той също така използва паметта по-ефективно.

Основният недостатък на предложения алгоритъм е, че времето, прекарано в Garbage Collector, се увеличава при търсене на повече от 1000 пътя. В конкретната реализация на алгоритъма това не е проблем, тъй като при търсене на маршрути в сгради стойността на K най-често е по-малка от 50.

Разработени са няколко сценария за тестване, за да се валидира предложеният алгоритъм в различни реални сградни среди. Алгоритъмът гарантира, че всички възможни маршрути ще бъдат намерени във всички тестови сценарии. В сравнение с алгоритъма на Yen, времето за неговата реализация е няколко пъти по-бързо във всички тестови сценарии. Експерименталните резултати показват, че предложеният алгоритъм може да се използва успешно като първо ниво на услуги за търсене на маршрути в големи обществени сгради.

ГЛАВА 5: АЛГОРИТЪМ ЗА ЗАОБИКАЛЯНЕ НА ПРЕПЯТСТВИЯ

5.1 Въведение

Едно от основните препятствия пред успешната интеграция на хората с увреждания на зрението в социалния, икономическия и културния живот е невъзможността им да се придвижват безопасно и надеждно в непознати закрити пространства. Повечето системи за навигация в закрити пространства използват граф-базирани модели, за да представят свързаността на сградите. Базираното на графи (макроскопично) намиране на път обикновено използва само информацията за връзките между пространствените обекти при изчисляване на възможен път. Тези стратегии за намиране на път не могат да се използват директно в сгради, особено за хора с увреждания на зрението. Те реализират навигация от врата до врата, но не дават информация за това как потребителят трябва да премине през стаята, докато стигне до следващата врата, и не поддържат избягване на препятствия. В тази глава се представя подход за бързо микроскопично (физическо) намиране на маршрут, което е част от тристепенна (от грубо към фино към оптимално) стратегия за планиране на пътя.

Основните геометрични модели, които се използват за намиране на път в сгради, са моделите за представяне на граници и моделите на базата на мрежа. Моделите, базирани на мрежи, използват регулярна (обикновено триъгълна или квадратна) или нерегулярна (диаграми на Вороной) мрежа, за да разделят вътрешното пространство на клетки. Размерът на клетките на мрежата обикновено е между 0,5 и 1 метър, за да се позволи по-точно описание на възможните пътища. Двата основни недостатъка на мрежовите модели са, че (1) изискват много памет при моделиране на големи сгради и (2) след получаване на възможния път, той трябва да бъде изгладен, тъй като може да съдържа зигзагообразни части и части, „залепени“ за стените. В този случай процесът на намиране на пътя изисква значително количество памет и процесорно време. За да се намали това време, най-често се използват евристични алгоритми за намиране на пътя. В многоетажни сгради с комплексна хоризонтална и вертикална свързаност изборът на подходяща евристична функция е истински проблем. Използването на евклидово разстояние за оценка на разстоянието между съседни възли и целевия възел дава неточни резултати. В този случай вероятността евристичните алгоритми да не върнат

оптималния път е висока. Необходимо е да се използват други метрики за изчисляване на хоризонталните и вертикалните разстояния в закрити помещения.

За да се намери възможният път, трябва да се вземе предвид действителната геометрия на стаите и препятствията в закритото пространство. Геометричният модел е част от пространствения модел на сградата, който трябва да съдържа подробна топологична и семантична информация за елементите и обектите на сградата. Повечето навигационни системи на закрито за слепи и хора с увреждания на зрението използват пространствени модели, които не вземат предвид наличието на ориентири, препятствия и опасности в стаите и коридорите. Тези модели не са подходящи за хора с увреждания, тъй като не предоставят необходимата информация за безопасно придвижване.

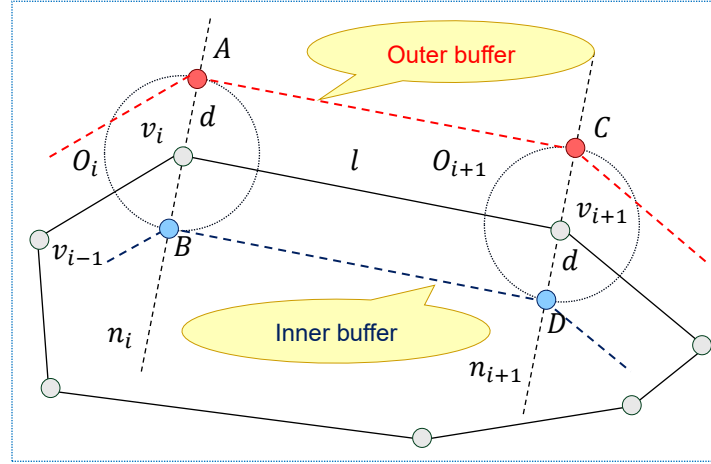
5.2 Описание на предлагания подход

Предлагания подход за микроскопично намиране на маршрути се базира на предварително получаване на K -най-късите пътища от началната точка s до целевата точка t . Тези пътища се описват като поредица от възли и ребра. Генерираните пътища са подредени от най-късия към най-дългия. След като се намери необходимият брой пътища от началната до крайната точка, за всеки макроскопичен път се изчислява съответен микроскопичен път. Микроскопичният път описва координатите на всяка междинна точка, включително точките, които заобикалят препятствия по пътя. За да се генерират такива детайлни пътища, е необходимо да се разполага с точен геометричен модел на сградата и обектите, намиращи се в нея. Слепите и хората с увредено зрение предпочитат да следват маршрути по коридорите и стените на стаите, като използват белия бастун. Предлаганият алгоритъм използва полигонно буфериране, за да опише път в близост до стените. Буферирането създава нов обект, подобен на даден обект, но на определено разстояние от него, което се нарича буферно разстояние. За стаите и коридорите се изчислява вътрешен буфер. За да се заобиколят обектите, намиращи се в сградата, се изчислява външен буфер.

5.2.1 Полигонно буфериране

Буферирането на полигон е основна операция в планирането на движението на роботи и при CAD/GIS системите. Предлаганият алгоритъм използва опростен математически апарат с цел да реализира буфериране без да се използват алгоритми като диаграми на Вороной, суми на Минковски или някои хибридни подходи.

За целта стаите, коридорите и обектите се описват с прости полигони. Ако $P: \{x_0, y_0; x_1, y_1; \dots, x_n, y_n\}$ представлява прост полигон, чиито начална и крайна точки съвпадат $\{x_0, y_0\} \equiv \{x_n, y_n\}$, тогава предложеният алгоритъм намира вътрешните и външните буфери (Фиг. 5.1), независимо коя е първата точка на полигона или как са подредени точките.



Фиг. 5.1 Буфериране на полигон

Алгоритъмът се изпълнява в следната последователност:

1. Определяне дали точките на полигона са подредени по часовниковата стрелка или обратно на часовниковата стрелка, като използвате оценка за площ на полигона:

$$\hat{A} = \sum_{i=0}^{N-1} (x_i y_{i+1} - x_{i+1} y_i), \quad (5.1)$$

където x_i и y_i са координатите на върха v_i . Ако $\hat{A} < 0$, тогава полигонът е с отрицателна ориентация (по часовниковата стрелка), а ако и ако $\hat{A} > 0$, полигонът е с положителна ориентация (против часовниковата стрелка).

2. Намиране на координатите на точките от буфера, които са на разстояние d от върховете и страните на полигона. За всеки връх на полигона $v_i(x_i, y_i)$, се изгражда окръжност $O_i(v_i, d)$. Намират се координатите на точките на пресичане на линията n_i , която е перпендикулярна на линията $l: \{v_i, v_{i+1}\}$, с окръжността O_i , както е показано на Фиг. 5.1. Линията l се описва с помощта на следното уравнение:

$$y = y_i + m(x - x_i), \quad (5.2)$$

където m е наклонът на правата, $m = (y_{i+1} - y_i)/(x_{i+1} - x_i)$. Правата n_i се описва с помощта на следното уравнение (нейният наклон е $m_{\perp} = -1/m$):

$$y = y_i - \frac{1}{m}(x - x_i). \quad (5.3)$$

Нека пресечните точки между n_i и O_i бъдат точки A и B . За да намерим техните координати, решаваме следната система от уравнения:

$$\begin{aligned} y &= y_i - \frac{1}{m}(x - x_i) \\ (x - x_i)^2 + (y - y_i)^2 &= d^2 \end{aligned} \quad (5.4)$$

Координатите на пресечните точки A и B са:

$$\begin{aligned} x &= x_i \pm \frac{md}{\sqrt{1+m^2}} \\ y &= y_i - \frac{1}{m}(x - x_i) \end{aligned} \quad (5.5)$$

По същия начин намираме координатите на точките C и D , които са точките на пресичане между линията n_{i+1} и окръжността $O_{i+1}(v_{i+1}, d)$. Трябва да определим кои две от следните четири точки $\{A, B, C, D\}$ принадлежат към буфера. Вместо да използваме алгоритъм от типа „точка в многоъгълник“, анализираме позицията на тези точки спрямо линията l . Ако потребителят се премести от връх v_i към връх v_{i+1} и полигонът е ориентиран отрицателно (по часовниковата стрелка), тогава се избират точките от дясната страна на линията l . Ако многоъгълникът е ориентиран положително (против часовниковата стрелка), тогава се избират точките от лявата страна на линията l .

Намирането на външния буфер се реализира по подобен начин. Ако потребителят се премести от връх v_i към връх v_{i+1} и многоъгълникът е ориентиран отрицателно (по часовниковата стрелка), тогава се избират точките от лявата страна на линията l . Ако многоъгълникът е ориентиран положително (против часовниковата стрелка), тогава се избират точките от дясната страна на линията l .

Този начин за получаване на точки от вътрешния и външния буфер е по-бърз в сравнение с методите, базирани на броя на навивките. За да изградим множеството LP , което съдържа левите точки, и множеството RP , което съдържа десните точки, намираме следните два коефициента:

$$\begin{aligned} k_1 &= (y_{i+1} - y_A) / (x_{i+1} - x_A) \\ k_2 &= (y_{i+1} - y_i) / (x_{i+1} - x_i) \end{aligned} \quad (5.6)$$

Ако $k_1 < k_2$, то $LP = \{A, C\}$, $RP = \{B, D\}$. Ако $k_1 > k_2$, и $LP = \{B, D\}$, $RP = \{A, C\}$.

3. Преобразуване на получения буфер (точки в множеството LP или RP) в прост полигон. След като получим всички десни или леви точки, трябва да проверим дали те образуват прост многоъгълник. Ако връх v_i е вдлъбнат и се опитваме да намерим външния буфер, или ако връх v_i е изпъкнал и се опитваме да намерим вътрешния буфер, е възможно буферът да не е прост полигон (да съдържа сегменти на буфера, които се пресичат). Анализират се всеки три последователни сегмента s_{i-1} , s_i и s_{i+1} (точки p_i, p_{i+1}, p_{i+2} , и p_{i+3} от буфера). Ако $(s_{i-1} \cap s_{i+1}) \neq \emptyset$, тогава точките p_{i+1} и p_{i+2} трябва да бъдат премахнати и трябва да бъде въведена нова точка, която е точка на пресичане между сегментите s_{i-1} и s_{i+1} . Нека s_{i-1} и s_{i+1} лежат на линии P_1 и P_2 :

$$\begin{aligned} P_1 &= P_i + k_1(P_{i+1} - P_i) \\ P_2 &= P_{i+2} + k_2(P_{i+3} - P_{i+2}) \end{aligned} \quad (5.7)$$

Условието, при което тези две прави имат точка на пресичане, е $P_1 = P_2$:

$$\begin{aligned} x_i + k_1(x_{i+1} - x_i) &= x_{i+2} + k_2(x_{i+3} - x_{i+2}) \\ y_i + k_1(y_{i+1} - y_i) &= y_{i+2} + k_2(y_{i+3} - y_{i+2}) \end{aligned} \quad (5.8)$$

От (5.8), за k_1 и k_2 получаваме:

$$\begin{aligned} k_1 &= [(x_{i+3} - x_{i+2})(y_i - y_{i+2}) - (y_{i+3} - y_{i+2})(x_i - x_{i+2})] / cof \\ k_2 &= [(x_{i+1} - x_i)(y_i - y_{i+2}) - (y_{i+1} - y_i)(x_i - x_{i+2})] / cof \end{aligned} \quad (5.9)$$

където

$$cof = (y_{i+3} - y_{i+2})(x_{i+1} - x_i) - (x_{i+3} - x_{i+2})(y_{i+1} - y_i). \quad (5.10)$$

Сегменти s_{i-1} и s_{i+1} имат пресечна точка ако

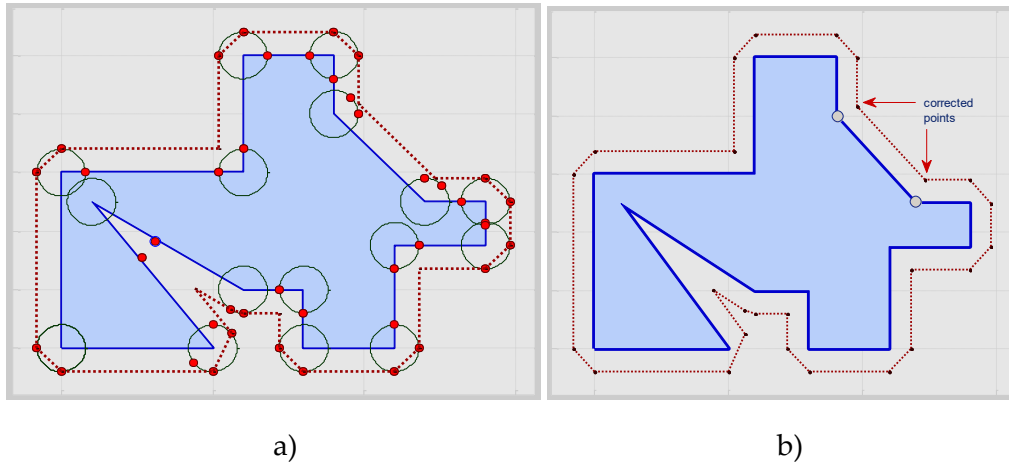
$$cof \neq 0, k_1 \neq 0, k_2 \neq 0, \{k_1, k_2\} \in [0, 1] \quad (5.11)$$

Координатите на пресечната точка са следните:

$$\begin{aligned} x &= x_i + k_1(x_{i+1} - x_i) \\ y &= y_i + k_1(y_{i+1} - y_i) \end{aligned} \quad (5.12)$$

Времевата сложност за изграждане на буфера е $O(n)$, където n е броят на върховете в полигона. При използване на хеш масиви времевата сложност за премахване на

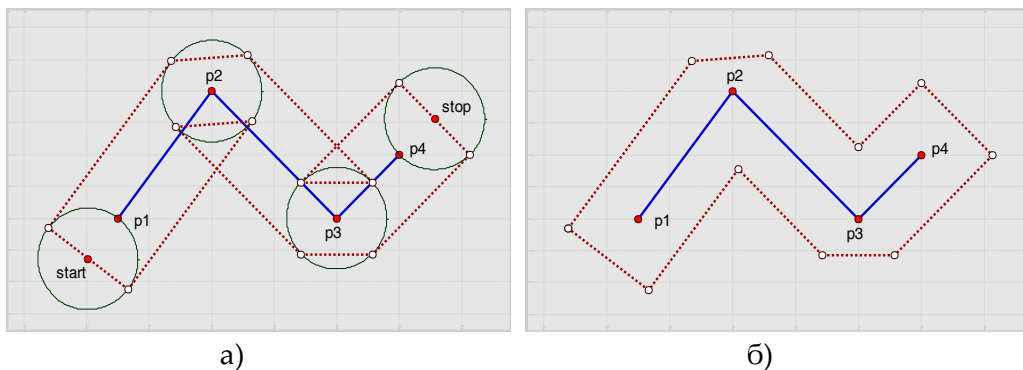
невалидните цикли е $O(n + k)$, където k е броят на самопресичанията. Фиг. 5.2 показва многоъгълник със сложна форма и два вдлъбнати върха. Всички точки на външния буфер са показани на Фиг. 5.2а. Те образуват форма, която не е прост многоъгълник. И за двата вдлъбнати върха трябва да се открият и премахнат отсечките, които се пресичат (виж Фиг. 3.5б).



Фиг. 5.2 Пример за външно буфериране на полигон: а) алгоритъм в действие; б) краен резултат

5.2.2. Буфериране на полилинии

Полилиниите се използват с цел моделиране на тънки стени и обекти с много малка ширина (w) спрямо дължината им (l), $l \gg w < d$. Намирането на буфера на полилинията се осъществява по начин, подобен на буферирането на полигони. За всеки сегмент от полилинията намираме две успоредни линии на разстояние d от него. Останалите сегменти се обработват по подобен начин. След това намираме буфера. След това се идентифицира наличието на пресичания на сегментите, които формират буфер. Крайният буфер трябва да бъде прост полигон (виж Фиг. 5.3).



Фиг. 5.3 Пример за буфериране на полилинии: а) алгоритъм в действие; б) краен резултат

5.2.3 Алгоритъм за заобикаляне на препятствия

Нека RC е множеството от точки, които описват вътрешния буфер на стаята, а OC е множеството от точки, които описват външния буфер на препятствието:

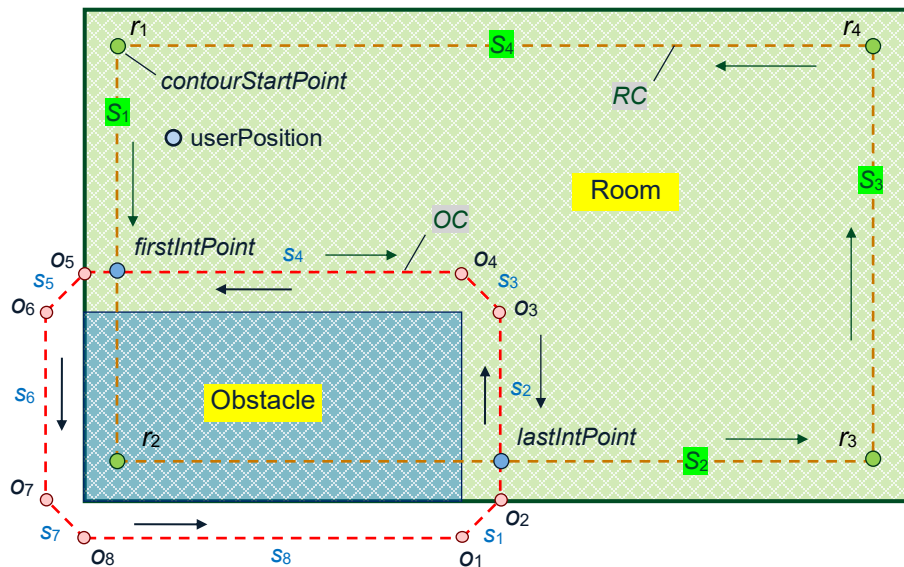
$$RC = \{r_1, r_2, r_3, \dots, r_N, r_1\},$$

$$OC = \{o_1, o_2, o_3, \dots, o_M, o_1\},$$

където r_i е i -та точка от буфера, описващ стаята, $i \in [1, N]$, и o_j е j -та точка от буфера, описващ препятствието, $j \in [1, M]$.

Алгоритъмът за заобикаляне на препятствия работи по следния начин:

1. Намира се началната точка (*contourStartPoint*) от вътрешния буфер RC на стаята. За да намерите тази точка, изчисляваме коя точка от буфера е най-близо до текущото местоположение на потребителя. В примера, показан на Фиг. 5.4, местоположението на потребителя е представено от точката *userPosition*.



Фиг. 5.4 Пример за заобикаляне на препятствие

2. Намират се точките на пресичане между контурите RC и OC , $\{p_{int}\} = RC \cap OC$. От всички точки в множеството p_{int} алгоритъмът използва само първата и последната точка на пресичане.

3. Намира се идентификационния номер на сегментите от OC и от RC , върху които лежат точките *firstIntPoint* и *lastIntPoint*:

$$firstIntPoint \in s_4: \{o_4, o_5\} \text{ and } S_1: \{r_1, r_2\},$$

$$lastIntPoint \in s_2: \{o_2, o_3\} \text{ and } S_2: \{r_2, r_3\}.$$

4. Намира се множеството OD , което съдържа точка за заобикаляне на препятствие. В този случай точките, които принадлежат към лявата и дясната пътека за заобикаляне, са:

$$OD_R = \{o_5, o_6, o_7, o_8, o_1, o_2\}, \quad OD_L = \{o_4, o_3\}.$$

Алгоритъмът трябва да определи как да насочи потребителя около препятствието – отляво или отдясно спрямо точката $firstIntPoint$. Ако ориентацията на стаята и буферите на препятствията е известна, не е необходимо да се изчислява коя точка от сегментите се намира вляво или вдясно спрямо точките на пресичане $firstIntPoint$ и $lastIntPoint$:

- Ако буферите на стаята имат същата ориентация като препятствието (виж Фиг. 5.4), първата точка на отклонение е първата точка на сегмента, на която се намира точката $firstIntPoint$, а последната точка на отклонение е втората точка на сегмента, на която се намира точката $lastIntPoint$.
- Ако буферите на стаята и препятствието имат обратна ориентация, първата точка на отклонение е втората точка на сегмента, на която се намира точката $firstIntPoint$, а последната точка на отклонение е първата точка на сегмента, на която се намира точката $lastIntPoint$.

В настоящия случай пътят на отклонението трябва да включва точките o_4 и o_3 ,

$$OD = \{firstIntPoint, \{o_4, o_3\}, lastIntPoint\}.$$

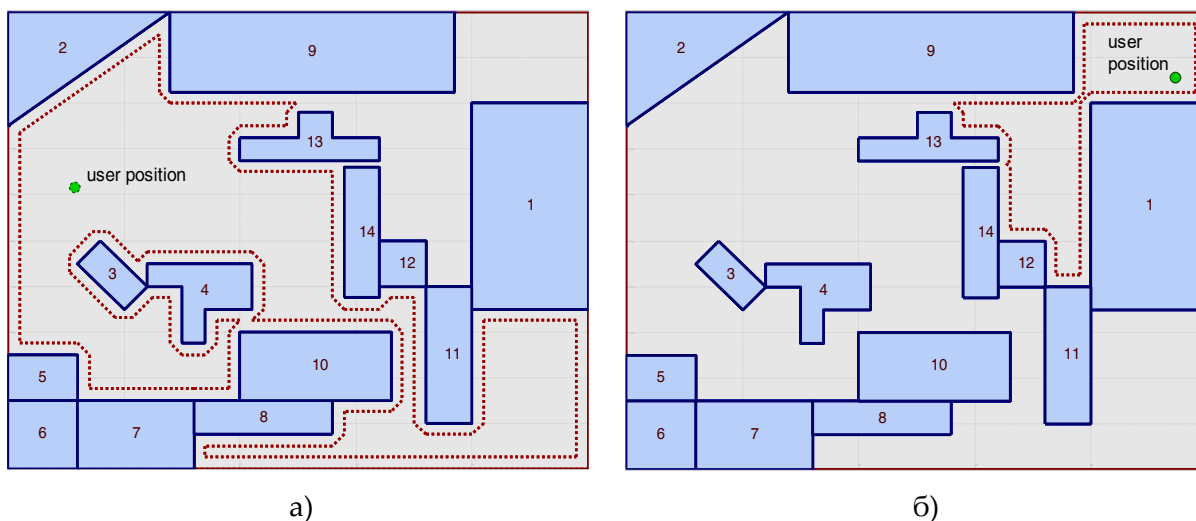
5. Намира се нов буфер на стаята RC_{new} . От множество RC премахнете всички точки, които се намират между точките $firstIntPoint$ и $lastIntPoint$. На тяхно място вмъкнете точки от обходния път OD . Ако ориентацията на буферите е еднаква (виж Фиг. 5.4), първата точка, която трябва да бъде премахната, е втората точка (r_2) на сегмента (S_1), на който се намира точката на пресичане $firstIntPoint$. Следващата точка, която трябва да бъде премахната, е първата точка (r_2) на сегмента (S_2), на който се намира точката на пресичане $lastIntPoint$. Следователно:

$$RC_{new} = \{r_1, OD, r_3, r_4, r_1\}.$$

Всички стъпки на алгоритъма за избягване на препятствия се повтарят, докато в стаята има обекти, чийто контур пресича текущия контур на стаята.

5.3 Експерименти

Предлаганият алгоритъм за търсене на маршрут при заобикаляне на препятствия е тестван в различни стимулационни среди. На Фиг. 5.5 е показан интериор с 14 обекта, които се явяват препятствия за незрящите.



Фиг. 5.5 Буфериране на стая: а) потребителят е близо до обекти 2 и 3; б) потребителят е близо до обекти 1 и 9

Обектите имат различни форми, ориентация и позиции. За правилната работа на алгоритъма трябва да е известна позицията на потребителя (зелена точка). В зависимост от тази позиция е възможно определени части от стаята да бъдат недостъпни за потребителя, тъй като някои препятствия блокират тези области. Когато потребителят е близо до обекти 2 и 3 (виж Фиг. 5.5а), алгоритъмът за заобикаляне се прилага пет пъти, за да се намери успешно път, който избягва всички препятствия. Когато потребителят е близо до обекти 1 и 9 (виж Фиг. 5.5б), алгоритъмът за заобикаляне се прилага три пъти.

На Фиг. 5.6 е показана последователността на работа на алгоритъм за случая от Фиг. 5.5а. Необходими са пет итерации до намиране на крайния маршрут:

Итерация 1: 1 1 0 0 1 1 1 0 1 0 0 0 0 0

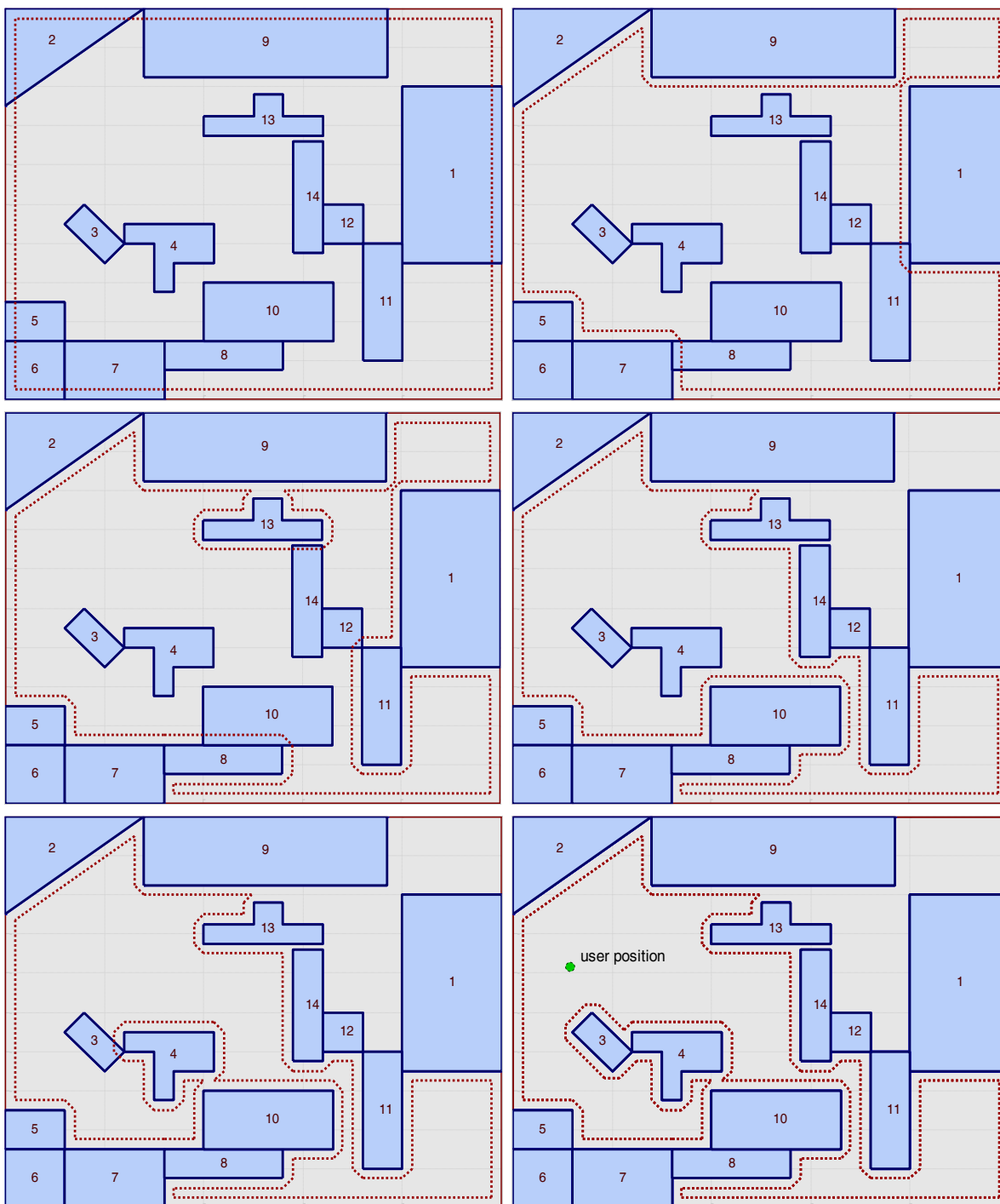
Итерация 2: 1 1 0 0 1 1 1 1 1 0 1 0 1 0

Итерация 3: 1 1 0 0 1 1 1 1 1 1 1 1 1 1

Итерация 4: 1 1 0 1 1 1 1 1 1 1 1 1 1 1

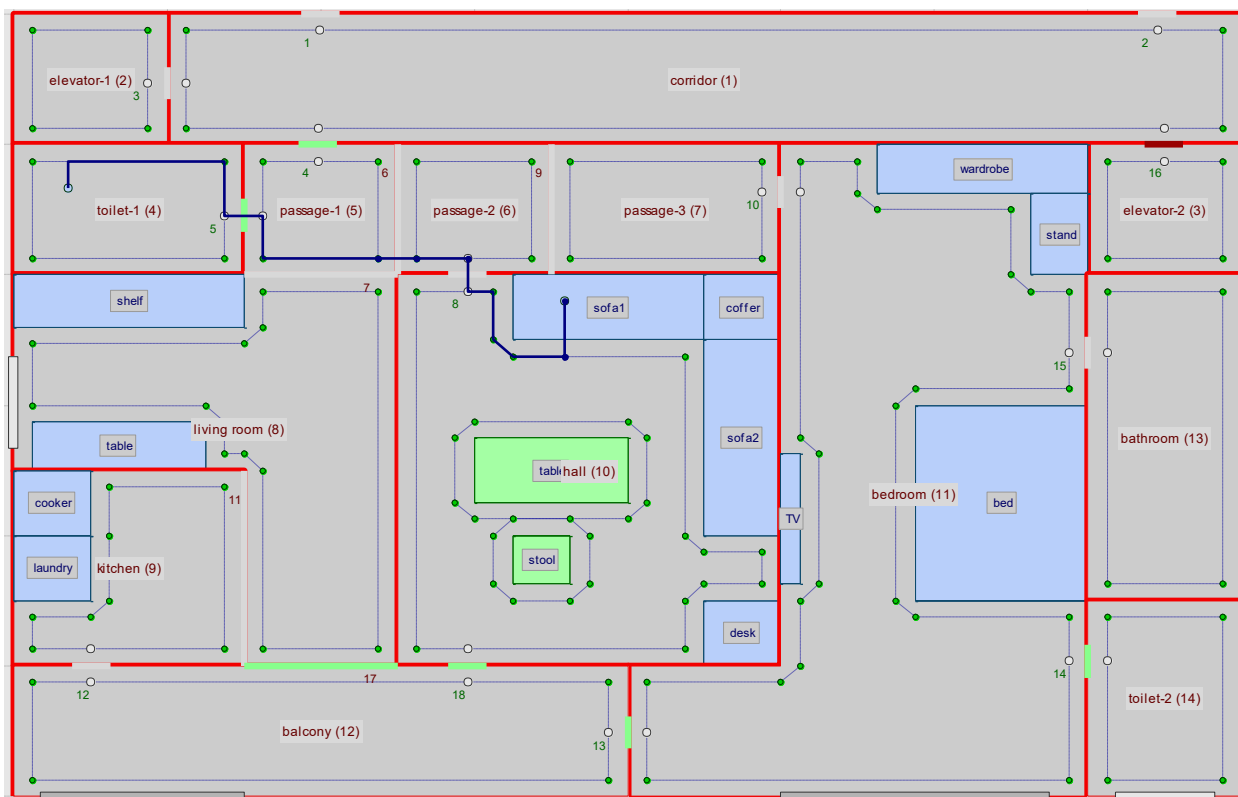
Итерация 5: 1 1 1 1 1 1 1 1 1 1 1 1 1 1

При всяка итерация се изчислява с кои обекти маршрутът има пресечна точка (тези обекти са отразени с 1). Например, при итерация 1 маршрутът има пресечни точки с обекти с номера 1, 2, 5, 6, 7 и 9.



Фиг. 5.6 Последователност на работа на алгоритъма за заобикаляне на препятствия

За да се генерира маршрут в реална сграда трябва да се обединят оптималните маршрути за всяка стая и/или коридор през които крайният маршрут трябва да премине. На Фиг. 5.7 е показана показва топологията на интериор с 12 стаи, 18 врати и отворени пространства.



Фиг. 5.7 Търсене на оптимален микроскопичен път в сграда

Да приемем, че потребителят се намира в хола (стая 10) и иска да намери тоалетна. Алгоритъмът намира две тоалетни в близост до потребителя: тоалетна-1 (стая 4) и тоалетна-2 (стая 14). Броят на всички възможни маршрути от текущата позиция на потребителя до целевите точки е дванадесет. В Табл. 5.1 са описани първите 4 възможни маршрута и техните параметри на базата на които се избира оптимален маршрут.

Табл. 5.1 Избор на оптимален възможен път

Път	Параметри за оптимизация						F_w^o
	Дължина, m	Препятствия, брой	Забележителности, брой	Врати, брой	Смяна на посоката, брой	Пресичания, брой	
1	10.60	0	0	2	11	0	0.1289
2	25.35	0	0	3	15	0	0.3667
3	22.19	0	0	4	11	0	0.3645
4	28.08	1	2	3	22	0	0.4732

Целевата функция $F_w^0(x)$ има минимална стойност (0.1289) за маршрут 1, който се приема за оптимален:

хол | d8 | коридор-2 | d6 | коридор-1 | d5 | тоалетна.

ГЛАВА 6: ИНТЕРИОРНИ НАВИГАЦИОННИ СИСТЕМИ, БАЗИРАНИ НА RFID И NFC ТЕХНОЛОГИИ

6.1 Технологии RFID и NFC

6.1.1 Принципи на работа и технически характеристики

Радиочестотната идентификация (RFID) и технологията за комуникация в близо поле (NFC) представляват основополагащи технологии за безжична комуникация на малко разстояние, които намират широко приложение в областта на интериорната навигация и локализация. Тези технологии се базират на принципа на електромагнитната индукция и обратното разсейване на модулацията за предаване на данни между четец и етикет.

RFID системите работят в различни честотни диапазони, като най-широко използваните са ниска честота (LF: 125 - 134 kHz), висока честота (HF: 13.56 MHz) и ултра висока честота (UHF: 860 - 960 MHz). Всеки честотен диапазон предлага специфични характеристики по отношение на обхват на четене, скорост на предаване на данни и устойчивост на смущения. Според анализа на [Тан и колектив \(2022\)](#), HF RFID етикетите осигуряват оптимален баланс между точност на локализация (± 0.5 - 2 метра) и енергийна ефективност за интериорни приложения, като системата с една антена демонстрира подобрена производителност чрез използване на една антена с множество лъчи.

NFC технологията, която е подмножество на HF RFID стандарта ISO/IEC 18092, работи на честота 13.56 MHz с максимален обхват на комуникация до 10 сантиметра. Тази технология поддържа три режима на работа: четене/запис, равнопращане комуникация и емуляция на карта, като всеки режим предлага различни възможности за навигационни приложения.

Основното разграничение между пасивни и активни RFID етикети има критично значение за проектирането на навигационни системи. Пасивните етикети, които не притежават собствен източник на енергия, получават необходимата мощност чрез електромагнитната индукция от четеща. Активните етикети, снабдени с вътрешна батерия, могат да предават сигнали на по-големи разстояния (до 100 метра) и да поддържат по-сложни функционалности, включително сензорни възможности за наблюдение на околната среда.

6.1.2 Предимства при интериорна навигация

RFID и NFC технологиите предлагат редица предимства за интериорните навигационни системи, които ги правят особено подходящи за прилагане в сложни архитектурни среди. Основното предимство е устойчивостта на ефектите от многопътното разпространение, които представляват основен проблем за GPS и други радиочестотни технологии в затворени пространства.

Прецизността на локализация при RFID базираните системи може да достигне точност под 1 метър чрез използване на алгоритми за обработка на сигнала. Изследването на [Yang и колектив \(2014\)](#) показва, че може да се постигне реално време проследяване на мобилни RFID етикети с висока прецизност, използвайки стандартни търговски устройства.

Енергийната ефективност представлява друго ключово предимство, особено при пасивните RFID етикети, които не изискват поддръжка на батерии и могат да функционират десетилетия без обслужване. Това прави технологията изключително подходяща за мащабни инсталации в обществени сгради, търговски центрове и индустриални обекти.

Мащабируемостта на RFID системите позволява поддържането на хиляди етикети в една мрежа без значително влошаване на производителността. Това е особено важно за големи комплекси като летища, болници и университети, където могат да се инсталират стотици референтни точки за навигация.

Възможностите за интеграция с мобилни устройства представляват допълнително предимство, особено за NFC технологията. Съвременните смартфони са снабдени с NFC четци, което позволява директна интеграция на навигационните услуги без необходимост от допълнителен хардуер за крайните потребители. Смартфоните с ОС Android могат да четат както NFC, така и RFID етикети. Смартфоните с ОС iOS могат да четат само NFC етикети. [Ozdenizci и колектив \(2015\)](#) представят NFC система, която позволява на потребителите да се ориентират в сгради чрез просто докосване на NFC етикети, разпръснати в околната среда.

6.1.3 Ограничения и предизвикателства

Въпреки значителните предимства, RFID и NFC технологиите се сблъскват с редица технически и практически ограничения при приложение в интериорни навигационни системи. Ограниченият обхват на комуникация представлява основно предизвикателство, особено при пасивните RFID етикети, чийто ефективен обхват рядко надвишава 2-4 сантиметра при HF честоти.

Смущенията от метални обекти и течности създават значителни проблеми за стабилността на сигнала. Металните повърхности предизвикват отражение и поглъщане на радиочестотните сигнали, което може да доведе до непредсказуеми промени в стойностите на силата на получения сигнал и грешки в локализацията. Според анализа на [Straka и колектив \(2022\)](#), металните конструкции оказват значително влияние на разпространението на UHF RFID сигналите в интериорни среди.

Проблемът с поверителността и сигурността представлява друго критично предизвикателство, особено в контекста на персоналната навигация. RFID етикетите могат да бъдат четени от неоторизирани устройства, което създава потенциални рискове за проследяване на движението на хората без тяхно знание. Съвременните криптографски решения, като AES шифроване и протоколи за взаимна автентификация, предлагат частично решение, но за сметка на увеличена сложност и разходи.

Околната среда също оказва значително влияние върху навигационните системи. Температурните колебания, влажността и електромагнитните смущения от други електронни устройства могат да предизвикат временни или постоянни нарушения в комуникацията. Това е особено проблематично в индустриални среди с високи температури или силни електромагнитни полета.

Проблем е и наличието на множество етикети в полето на четеща. Макар че съществуват протоколи против този проблем като ALOHA, тези решения могат значително да забавят процеса на четене при висока плътност на етикети. [Peng и колектив \(2023\)](#) предлагат динамичен протокол за дървовидни заявки базиран на контрол на достъпа за бързо идентифициране на движещи се RFID етикети, докато [Yan и колектив \(2023\)](#) разработват ефективни алгоритми за търсене на етикети при сложни RFID системи.

Инфраструктурните разходи представляват допълнително ограничение, особено за големи инсталации. Въпреки че отделните RFID етикети са относително евтини (под 1 до няколко лева), общата стойност на системата, включваща четци, мрежова инфраструктура и софтуер за управление, може да бъде значителна. Анализът на [Menanno и колектив \(2023\)](#) за дигитализация на веригите за доставки чрез RFID показва как технологията може да предостави значителни ползи в управлението на запасите и проследяването, но изисква внимателна оценка на разходите и ползите.

Ограничената изчислителна способност на традиционните RFID етикети ограничава възможностите за локална обработка на данни и изпълнение на сложни алгоритми. Въпреки че новите поколения чипове предлагат подобрени изчислителни възможности, те остават значително ограничени в сравнение с други безжични технологии като BLE.

Стандартизацията и взаимодействието между различни производители остават предизвикателство, въпреки съществуването на международни стандарти като серията ISO 18000. Различните имплементации често водят до проблеми със съвместимостта, които усложняват интеграцията на разнородни системи.

Тези ограничения налагат внимателно проектиране и планиране при разработването на RFID/NFC базирани навигационни системи, като се вземат предвид специфичните изисквания на приложението и характеристиките на операционната среда. Преодоляването на тези предизвикателства изисква мултидисциплинарен подход, комбиниращ алгоритми за обработка на сигнали, устойчиво системно проектиране и иновативни интеграционни решения.

6.2 Система RSNAVI за интериорна навигация на слепи хора

6.2.1 Въведение

Повечето от съществуващите системи за навигация в закрити помещения за слепи хора не отчитат техните специфични нужди, свързани с ориентацията и придвижването в непозната сграда. В тази глава се представя система за навигация в закрити помещения за слепи хора, наречена RSNAVI, която има следните предимства: *Първо*, системата използва 4D моделиране на сгради – 3D геометрия на сгради и обекти и състояние на всички сензори във времето. За да подобрим процеса на навигация, използваме семантично богат модел, който описва не само позицията и формата на всички обекти, но и техните характеристики. *Второ*, алгоритъмът за планиране на маршрута използва мултипараметрична оптимизация, за да получи оптималния маршрут за незрящите потребители. Системата позволява автоматично пренасочване, когато незрящият потребител се отклони от маршрута или ако засече промяна в състоянието на сензорите, например блокиране на достъпа до стаята поради пожарна аларма например. Накрая, се използват само алгебрични изрази за изчисляване на оптимални точки на маршрута и за заобикаляне на препятствията.

През последните години навигацията на открито за слепите хора се осъществява сравнително успешно чрез приложения, които използват глобални навигационни

спътникови системи като GPS, GLONASS, диференциален GPS (DGPS), спътникова система за подсилване (SBAS) и GPS карти с висока точност и голям брой ориентири. Навигацията на слепите хора в непозната за тях интериорна среда обаче остава проблематично. Причините за това са комплексни: 1) Използването на GNSS позициониране в сгради понастоящем не е пряко приложимо; 2) Голяма част от системите за навигация в закрити помещения за слепи хора не са в съответствие със специфичния начин на ориентация и движение на слепите хора в непозната среда; 3) Част от системите изискват модификация на инфраструктурата на сградите, което в повечето случаи е твърде скъпо; 4) Някои системи изискват непрекъснато обучение на незрящите, преди да могат да бъдат използвани на практика; 5) Използването на стерео сигнали в някои навигационни системи може да бъде опасно за незрящите, защото те губят способността да чуват звуците от обкръжаващата ги среда.

При проектирането на навигационни системи за слепи хора трябва да се вземе предвид предпочитаният от тях начин на ориентация и придвижване в непозната среда. За тази цел трябва да се използва дизайн, ориентиран към потребителя (UCD). Такива системи трябва да са лесни за използване, надеждни и с цена, която съответства на покупателната способност на слепите хора.

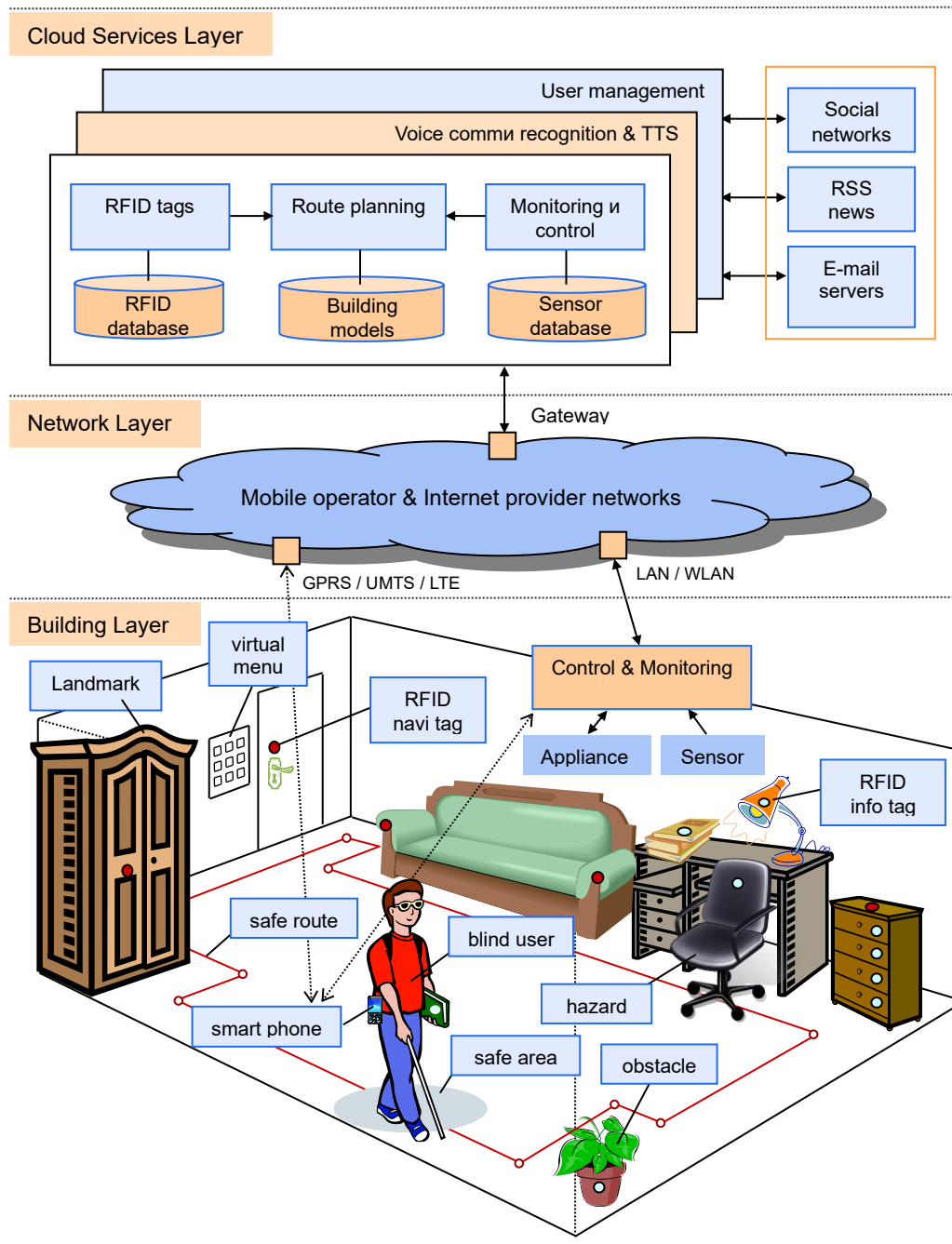
6.2.2 Архитектура и компоненти на предлаганата система

Архитектурата на предлаганата система за навигация в закрити помещения за слепи хора е избрана след интервю с контролна група от 20 слепи лица. Тя е разделена на три слоя (виж Фиг. 6.1):

- Слой „Сграда“. Отговаря за проследяване на слепите потребители, идентифициране на предмети и мониторинг на състоянието на сензорите (смартфони, пасивни RFID етикети и шлюз за контрол и мониторинг);
- Мрежов слой. Основната му функция е да доставя информацията, получена от слоя „Сграда“, до услугите, намиращи се в интернет.
- Приложен слой. Този слой реализира функционалността на навигационната система чрез уеб услуги, базирани на Java. Когато не е възможно да се реализира мрежова връзка, се използва SMS комуникация чрез 4G/5G модем.

Предложената навигационна система използва следните технологии: 1) Пасивни RFID етикети (Mifare Classic 1K и Mifare Ultralight). Те се използват за идентифициране и проследяване на незрящи потребители, локализиране на обекти и идентифициране на предмети; 2) Мобилни телефони с вграден интерфейс

за близка комуникация (NFC), електронен акселерометър и магнитометър (компас); 3) Гласова навигация - използва се Text-To-Speech (TTS) програмен модул; 4) Beck's SC13 System on Chip (SoC) се използва за имплементиране на Control & Monitoring Gateway.



Фиг. 6.1 Архитектура на RSNAVI

Системата за навигация в закрити помещения RSNAVI позволява оптимално планиране на маршрута, проследяване и проучване на потребителя, гласова навигация по маршрута, евакуация в реално време и обучение преди пътуването (възможност за проучване и изучаване на маршрута виртуално). Всички режими на навигационната система трябва да се реализират в реално време при използване на мобилна платформа. За тази цел се използват само алгебрични изрази за изчисляване на оптимални маршрутни точки и за заобикаляне на препятствия.

Навигационната система RSNAVI работи в два режима:

1. Локален режим: този режим не изисква достъп до услуги, намиращи се в интернет. Всички алгоритми се реализират от страна на клиента (мобилен потребител). Порталът за контрол и мониторинг чрез Bluetooth интерфейс уведомява клиента за състоянието на сензорите. Този режим е подходящ за навигация в малки жилища.
2. Глобален режим: този режим изисква потребителите да имат постоянна интернет връзка. За да се намали броят на заявките към услугите, намиращи се в интернет, необходимата информация (план на етажите на сградата, изчислени маршрути, информация за интелигентни обекти и т.н.) се съхранява на флаш диска на мобилния терминал. Ако потребителят не може да достигне до крайната точка, например поради пожар в стая, услугата за планиране на маршрута автоматично изчислява нов маршрут. В този случай услугата за навигация изпраща push съобщение, за да уведоми клиента. Използва се Simple Object Access Protocol (SOAP) за прехвърляне на XML-кодирани данни между мобилни клиенти и услугите, намиращи се в интернет. Глобалният режим е подходящ за навигация в големи обществени сгради (болници, училища, университети, хотели, държавни учреждения и др.).

Навигационната система RSNAVI използва няколко вида пасивни RFID етикети: 1) навигационни етикети - позволяват проследяване на потребителите; 2) етикети пощенски кутии - позволяват на потребителите да оставят аудио съобщения; 3) етикети за проследяване на обекти - локализиране на обекти в сградата; 4) етикети за информация за обекти - идентифициране на интелигентни обекти и 5) URI етикети - съдържат унифициран идентификатор на ресурс (URI) за услуги, намиращи се в интернет.

За да се създаде виртуално меню (група RFID етикети, подредени в редове и колони, виж Фиг. 6.1), се използват URI етикети. Когато незрящият човек доближи

мобилния си телефон до етикета на менюто, той чува името на услугата, до която етикетът предоставя достъп, например: получаване на модел на сградата, достъп до портал за контрол и мониторинг, търсачка (търсене на хора, стаи и обекти, намиращи се в сградата) и др.

Пример за сценарий за използване на предложената навигационна система е следният:

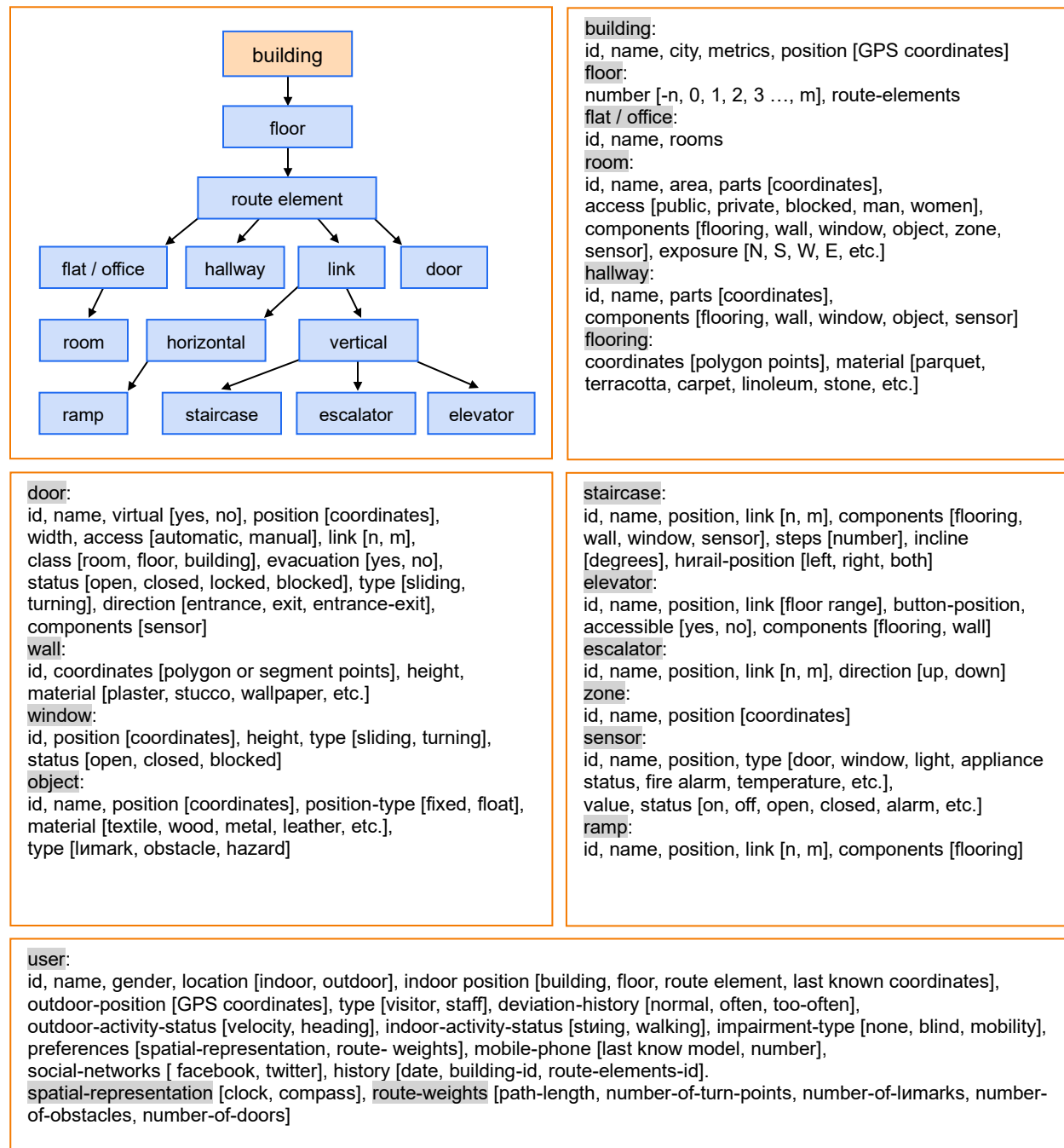
- (1) Незрящият потребител достига желаната сграда, например чрез GPS-базирана навигационна система. След това потребителят трябва да намери URI етикет, разположен на входната врата, преди да влезе в сградата. Активираната услуга (уеб услуга в глобален режим или портал за контрол и мониторинг в локален режим) намира в базата данни модела на сградата и генерира информация за клиента в XML формат (етажи, коридори, врати, ескалатори, офиси, стаи и т.н.).
- (2) Потребителят избира крайната точка (етаж, апартамент или офис, стая, зона и т.н.). Потребителят може да избере предпочитана дестинация, ако има няколко възможни дестинации (например тоалетна). След това активира алгоритъма за планиране на маршрута. Той връща оптималния маршрут от текущата позиция на потребителя до избраната дестинация.
- (3) Навигационната услуга генерира поредица от инструкции (в зависимост от състоянието на околната среда и сензорите), за да ориентира и насочи потребителя към дестинацията.

6.2.3 Моделиране на сградите

Най-често се използват геометрични модели за описание на сградите, например модели за компютърно проектиране (CAD), 3D модели на данни от географска информационна система (GIS) и обектно-ориентирано 3D моделиране на информация за сгради (BIM). Последните модели са разработени, за да обхванат всички етапи от жизнения цикъл на сградата и да опишат както геометричната, така и семантичната информация. При навигацията за незрящи за представяне на инфраструктурата на сградите обикновено се използват графични или мрежови модели. Връзката „стая-врата“ се преобразува във връзка „възел-ребро“. Тези модели не са подходящи за много голям брой възли.

Едно от предимствата на предложената система за навигация в закрити помещения, в сравнение с други подобни системи, е използването на 4D мрежови модел на сградата. В този случай не е необходимо преобразуване на

геометрията на сградата в графично или мрежово представяне. В допълнение към 3D геометричните размери на сградата, се моделира състоянието на всички сензори във времето. Тази информация е необходима, тъй като изборът на оптимален маршрут зависи от състоянието на вратите (отворени, затворени, заключени, блокирани) и състоянието на сензорите.



Фиг. 6.2 Моделиране на сградата и потребителите

Използва се богато семантично обектно-ориентирано моделиране на сградите (виж Фиг. 6.2). Моделът съдържа информация за всички елементи на маршрута (стаи, коридори, връзки и врати) и вътрешни обекти (тяхното положение и атрибути).

6.2.4 Планиране на маршрути

При планирането на маршрути се използват различни групи алгоритми, например: базирани на потенциални полета, грид-базирани, алгоритми за маршрутизиране в компютърни мрежи, диаграми на Вороной, динамично програмиране и др. Повечето системи за навигация в закрити помещения използват алгоритмите на Дийкстра или A^* за най-кратък път, за да получат оптимален маршрут от всички възможни маршрути. За разлика от общия случай за виждащите, разстоянието и времето за придвижване не са единствените критерии, които определят оптималния маршрут. Общият брой ориентири и забележителности са допълнителни критерии, които трябва да повлияят на избора на оптималния маршрут.

В предложената система използваме много-параметрична оптимизация, за да получим оптималния маршрут. Навигационната система анализира пет параметъра:

- брой ориентири по маршрута - максимум;
- брой препятствия и опасности по маршрута - минимум;
- брой врати, през които потребителят трябва да мине - минимум;
- брой завои (потребителят променя посоката си) - минимум;
- дължина на маршрута - минимум.

Тежестта на всички параметри е определена след интервю с контролна група от незрящи лица. Потребителите имат възможност да зададат тежестта на всеки от петте параметри. Значението на всеки критерий се получава чрез многоцелева оптимизация, базирана на идеите на теорията за ранговата корелация. Всеки интервюиран попълва въпросник, който съдържа списък с критерии в произволен ред и поле за значението на всеки критерий. Ако броят на критериите е M , тогава най-високата оценка е 1, а най-ниската оценка е M . Интервюираните могат да поставят една и съща оценка за два или повече критерия, ако не могат да разграничат тяхната значимост. За да намерим оптималния път, използваме оптимизация, базирана на целевата функция от следния тип:

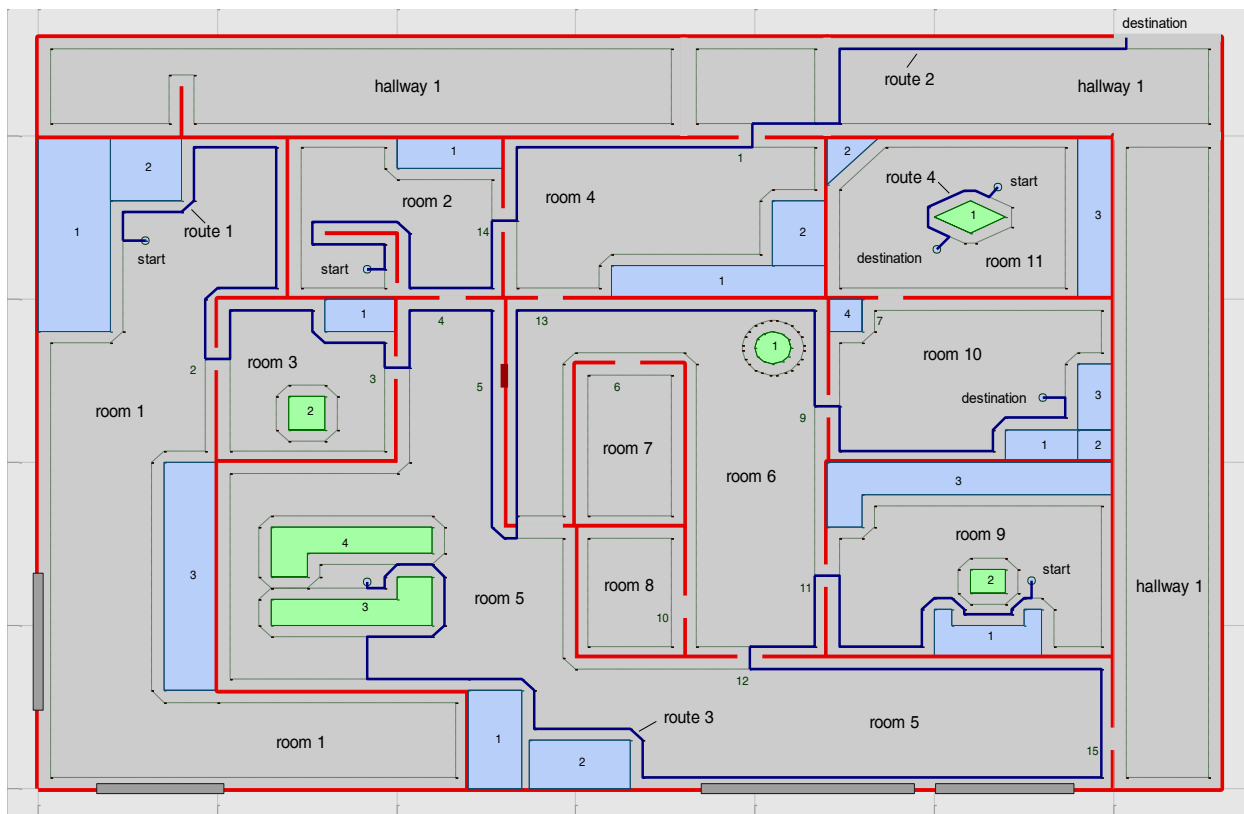
$$F_w^0(x) = \frac{1}{M} \sum_{j=1}^M \left| \frac{y_j^* - y_{j(x)}}{y_j^{max} - y_j^{min}} \right| (1 - W_j), \quad (6.1)$$

където W_j е тежестта (значимостта) на критерий j , y_j^* е най-добрата стойност за критерий j , а y_j^{min} и y_j^{max} са гранични стойности за критерий j . За да се намери максималната стойност на критерий j , тогава $y_j^* = y_j^{max}$, но за да се намери минималната стойност, формулата е $y_j^* = y_j^{min}$. За всеки път се изчислява стойността на целевата функция $y_j^* = y_j^{min}$. Оптималният път е този, който минимизира уравнение (6.1).

Намирането на оптималния маршрут се осъществява в няколко стъпки:

- 1) Груба оценка на оптималния маршрут. Намираме всички K възможни маршрути от началната стая на до крайната стая. Маршрутите се описват като поредица от идентификационни номера на стаи и врати, през които трябва да мине незрящият потребител. Например, 5-3-1-4-10 се интерпретира по следния начин: от началната стая 5 се влиза в стая 1 през врата 3. До крайната стая 10 се стига през врата 4.
- 2) Точна оценка на оптималния маршрут. Изчисляваме оптималните маршрути във всички стаи и коридори, които принадлежат към маршрутите, изчислени в стъпка 1. Алгоритъмът, който намира оптималния маршрут в стаите и коридорите, реализира заобикаляне на препятствия. Предполага се, че позицията на препятствията е известна. Изчисляването на маршрута се основава на факта, че незрящите потребители предпочитат да се придвижват в непознати сгради покрай стените и перилата на стълбите. Алгоритъмът за планиране на маршрута намира маршрут, чиито точки са на разстояние 0,5 м от стени и препятствия.
- 3) Крайният маршрут се получава като поредица от всички оптимални маршрути в стаите, през които незрящите трябва да минат.

Фиг. 6.3 показва резултатите, получени след планиране на маршрут в апартамент. С пунктирана линия са показани предварително изчислени безопасни за незрящите маршрути покрай стените на стаите и неподвижните обекти.



Фиг. 6.3 Пример за планиране на маршрут

Маршрут 1: Движение от началната точка, намираща се в стая 1, до крайната точка, намираща се в стая 10. Тъй като врата 5 е блокирана, маршрутът минава през отвореното пространство към стая 6.

Маршрут 2: Придвижване от точка, намираща се в стая 2, до стълбището в коридор 1.

Маршрут 3: Придвижване от началната точка, намираща се в стая 9, до крайната точка, намираща се в стая 5. Това е пример за заобикаляне на неподвижни препятствия.

Маршрут 4: Началната и крайната точка се намират в стая 11. В този случай разстоянието между началната и крайната точка е твърде късо и изчисленият маршрут заобикаля препятствие 1.

6.2.5 Навигация

За да бъдат възможни режимите на проучване на терена и навигация, навигационната система трябва да знае текущото местоположение на потребителя. За тази цел в RFID базата данни се записват местоположенията на всички

навигационни етикети. Те се използват като референтни точки в сградите. Навигационните етикети са поставени на всяка врата, обект и стена. За да се получи позицията на незрящ потребител във времето (проследяване на потребителя), той трябва да докосне мобилния си телефон до навигационния етикет. Когато незрящият се движи между две референтни точки, се използва техниката за приблизително изчисляване на местоположението, за да се актуализира текущата позиция на потребителя. Когато незрящият потребител се отклони от текущия маршрут и докосне навигационен етикет, навигационната система изчислява нов маршрут (автоматично пренасочване).

В режим на проучване незрящият потребител получава информация за характеристиките на текущата стая и обектите в околната среда. Незрящите предпочитат два основни метода за пространствено представяне:

- Алоцентрично. Предполага се, че референтната посока е една от четирите основни посоки (север, юг, изток и запад) или часовниковата стрелка (12 часа). Пример за алоцентрично ориентиране е „завийте на изток, вратата е на 9 часа“.
- Егоцентрично. В този случай информацията за текущата посока на потребителя се получава от сензор. Системата за навигация в закрити помещения RSNAVI поддържа и двата метода за пространствено представяне. В режим на проучване незрящият получава информация за: 1) Апартамент или офис: име, площ, брой стаи, имена на стаите; 2) Стая: име, площ, експозиция, препятствия и опасности. 3) Асансьор: обхват на етажите, достъпност, положение на бутоните; 4) Врата: автоматична или не, плъзгаща се или въртяща се, състояние (отворена, затворена, заключена, блокирана); 5) Описание на всички обекти (имена и разстояния) в текущата посока на потребителя; 6) Имената на най-близките обекти (отпред, отзад, отляво, отдясно).

След планирането на маршрута се получава поредица от точки, които описват маршрута от началната до крайната точка. В режим на навигация незрящият потребител получава команди как да стигне до всяка точка, например „следвайте стената от лявата ви страна, докато стигнете до врата“, „след пет стъпки завийте наляво“. Използва се навигация, базирана на ориентири, в дълги коридори и комбинация от навигация, базирана на ориентири и метрични данни, в стаи. По време на навигацията от началната до крайната точка незрящият потребител получава информация за материала на подовите и стените, покрай които се движи,

например: настилка – паркет, движи се покрай стена с латексно покритие. Системата генерира информация и за обектите, покрай които незрящият потребител минава, например: кожен диван, дървен гардероб, перална машина.

6.2.6 Заключение

В тази глава е представена архитектурата на RFID-базирана система за навигация в закрити помещения за слепи хора, наречена RSNAVI. Основните характеристики на предложената навигационна система са следните. *Първо*, 4D контекстно-ориентираният модел на сградите позволява адаптиране на режимите на проучване и навигация към състоянието на сензорите в сградата (състояние на врати, прозорци, пожарни сензори и др.). *Второ*, алгоритъмът за планиране на маршрута взема предвид важната за незрящите потребители информация: ориентири, препятствия и опасности в сградата, врати, стени и положението на парапетите. За да се получи оптималният маршрут за незрящите лица, се използва алгоритъм за много-параметрична оптимизация. Системата позволява автоматично пренасочване, когато състоянието на сензорите се промени и когато незрящият потребител се отклони от текущия маршрут. Трето, използва се алгоритъм за заобикаляне на препятствия. Алгоритъмът е независим от броя и позициите на препятствията и техните геометрични форми. *Накрая*, незрящият потребител може да търси лица и обекти, които се намират в сградата.

ГЛАВА 7: ЛОКАЛИЗАЦИЯ С BLUETOOTH LOW ENERGY МАЯЦИ

7.1 Въведение

Bluetooth Low Energy (BLE) технологията представлява революционен подход към безжичната комуникация с ниско енергопотребление, който от своето въвеждане с Bluetooth 4.0 спецификацията през 2010 година се е наложил като предпочитана платформа за приложения на Интернет на нещата и услуги за локализация. Технологията предлага уникална комбинация от ефективност на енергопотреблението и възможности за позициониране, което я прави изключително подходяща за създаване на системи за локализация в закрити пространства.

7.1.1 Технически характеристики на BLE маяците

BLE маяците функционират в 2.4 GHz ISM обхвата и използват 40 честотни канала, от които три канала (37, 38 и 39) са специално предназначени за рекламиране на устройства и откриване на услуги (Киула и колектив, 2024). Тези първични рекламни канали са разположени на честоти 2402 MHz, 2426 MHz и 2480 MHz съответно, като са стратегически разпределени за минимизиране на интерференциите с Wi-Fi каналите в същия обхват. Приложенията за мониторинг на устройства, определяне на разстояние и проследяване на активи използват индикатор за силата на получения BLE сигнал (RSSI) за оценка на близостта на устройството до приемника. Останалите 37 канала се използват за двупосочна комуникация след установяване на връзка между устройствата.

Съвременните BLE маяци внедряват Bluetooth стандарти версия 5.0 и по-нови, които осигуряват съществени подобрения по отношение на комуникационния обхват, пропускателната способност и енергийната ефективност на предаването (Рут и колектив, 2023). В типични експлоатационни условия BLE маяк постига работен обхват между 2 и 5 метра, докато максималният теоретичен радиус на действие може да достигне до приблизително 100 метра в открита среда без наличието на физически препятствия или електромагнитни смущения. Мощността на излъчвателя на устройствата може да се конфигурира до 8 dBm, а интервалът на рекламните пакети е програмируем в диапазон от 20 ms до 10.24 s, с дискретна стъпка на настройка от 0.625 ms, което позволява оптимизация спрямо конкретните

изисквания за енергийна консумация и честота на предаване (Киула и колектив, 2024).

Архитектурата на данните в пакетите от BLE маяци е базирана на стандартизирани спецификации, като iBeacon (Apple), Eddystone (Google) и AltBeacon, които дефинират формата и съдържанието на предаваните рекламни рамки. Например, протоколът Eddystone-UID структурира пакета по следния начин: 1 байт за идентификация на типа рамка, 1 байт за мощност на излъчвателя (TX Power), 16 байта за уникалния идентификатор на маяка (включващ 10-байтов Namespace ID и 6-байтов Instance ID), както и 2 байта, резервирани за бъдещи разширения на стандарта. Тази унифицирана и модулна структура осигурява висока ефективност при откриване, разграничаване и класификация на голям брой BLE маяци в споделена радио среда, като същевременно гарантира съвместимост между различни устройства и платформи.

7.1.2 Методи за измерване на разстояние чрез оценка на RSSI

Показателят за сила на получения сигнал (RSSI) представлява основния параметър, използван за оценка на разстоянието в BLE-базирани системи за локализация (Guo и колектив, 2023). RSSI стойностите се измерват в децибели-миливат (dBm) и служат като индикатор за мощността на приетия радиосигнал. Теоретичната основа на този метод стъпва върху модела на затихване на електромагнитните вълни в условия на свободно пространство, който описва зависимостта между излъчваната мощност, разстоянието и получената мощност при приемане.

Математически, зависимостта на RSSI може да бъде представена чрез уравнението на Friis, което в логаритмична форма се изписва като:

$$RSSI(d) = P_{TX} + PG_{TX} + G_{RX} - 10n\log_{10}\left(\frac{d}{d_0}\right) - L, \quad (7.1)$$

където:

- $RSSI(d)$ –стойност на RSSI, измерена на разстояние d от маяка;
- P_{TX} –мощност на излъчвателя на маяка [dBm];
- G_{TX} и G_{RX} – печалби на антените на предавателя и приемника [dBi];
- n – коефициент на затихване на средата,
- d – разстояние между предавател и приемник [m],
- d_0 – референтно разстояние (обикновено 1 m),

- L – допълнителни загуби (например поради препятствия или многопътно отражение) [dB].

В практически приложения за оценка на разстояние при BLE локализация широко се използва емпиричният модел на затихване, формулиран като:

$$RSSI(d) = RSSI(d_0) - 10n \log_{10} \left(\frac{d}{d_0} \right), \quad (7.2)$$

където:

- $RSSI(d_0)$ е референтната стойност на RSSI, измерена на разстояние d_0 ;
- n е коефициент на затихване по пътя (експонента на затихване), зависещ от физическите характеристики на средата на разпространение (например наличието на препятствия, многопътни отражения или поглъщане на сигнала от материали).

Съвременните научни изследвания демонстрират, че интегрирането на алгоритми за машинно обучение съществено повишава прецизността на локализация, базирана на RSSI измервания ([Niculescu и колектив, 2022](#)). Сред тези подходи особено ефективни се оказват алгоритмите от типа weighted k-nearest neighbors (wkNN), които оптимално обработват данни, генерирани BLE маяци. Методът се основава на обединяване на множество последователни RSSI измервания и прилагане на статистически филтри за потискане на шум и стохастични вариации в радио каналите. По този начин се постига значително подобрене в надеждността и стабилността на изчислената позиция, което го прави предпочитано решение за BLE системи за локализация с висока точност.

7.1.3 Шум в радио каналите

Радио каналите в закрити пространства представляват изключително сложна среда за разпространение на сигналите, характеризираща се с предизвикателства, които значително влияят върху точността на RSSI измерванията. Както отбелязват [Киула и колектив \(2024\)](#), в сложна интериорна среда RSSI-базираните решения може да не предоставят точна оценка. Основните източници на шум в BLE системите за локализация включват многопътното разпространение, сенчестото затихване и интерференциите от други безжични устройства.

Основните фактори, водещи до шум и интерференции, включват:

- Многопътно разпространение: едновременно приемане на директни и отразени сигнали, което води до конструктивни и деструктивни интерференции и флуктуации в RSSI.
- Сенчесто затихване (shadow fading): частично или пълно затихване на сигнала поради наличието на физически препятствия (стени, мебели, хора), които екранират или поглъщат радиовълните.
- Интерференции от други безжични системи: съвместната работа с Wi-Fi мрежи, Zigbee устройства и други източници в 2.4 GHz ISM обхвата води до допълнителни смущения и нестабилност на RSSI стойностите.

Тези ефекти правят необходима употребата на усъвършенствани методи за обработка на сигнала или алгоритми за машинно обучение, които да компенсират нелинейностите и статистическите вариации в реалната радио среда.

Сенчестото затихване представлява дългосрочните вариации в получената мощност, причинени от големи препятствия като стени, мебели или човешки тела. Особено проблематично е влиянието на човешкото тяло, което може да причини затихване от 10 до 20 dB, когато блокира пряката видимост между маяка и приемника. Това затихване не е постоянно и може да варира значително в зависимост от ориентацията и позицията на тялото спрямо сигналния път.

Интерференциите в обхвата 2.4 GHz представляват сериозно предизвикателство поради високата плътност на безжични устройства в съвременните среди. Wi-Fi мрежи, микровълнови печки, Bluetooth устройства и други ISM обхват приложения могат да причинят значителни интерференции. За справяне с този проблем, BLE протоколът използва честотно прескачане (frequency hopping) и адаптивни техники за избор на канали.

За минимизиране на влиянието на шума в радио каналите се прилагат различни техники за обработка на сигналите. Филтрирането с подвижна средна стойност и филтрите на Калман показват ефективност при намаляване на кратко-временните флуктуации в RSSI данните. Пространствено-времените методи за обработка на данни комбинират измервания от множество локации за получаване на по-надеждни оценки на посоката на пристигане (Direction of Arrival - DoA).

7.2 Алгоритъм за автоматично разполагане на маяци в сгради

Локализирането на посетителите в обществени сгради е ключова технология за създаване на достъпни среди с помощта на услуги, базирани на местоположението. Предлага се цялостно решение за създаване на системи за локализация в закрити помещения чрез BLE маяци, което включва: 1) Алгоритъм за бързо разделяне на компонентите на сградата на неприпокриващи се правоъгълници; 2) Алгоритъм за разполагане на сензорни възли въз основа на геометрията на получените правоъгълници; и 3) Алгоритъм за оптимизиране на разположението на сензорните възли, като се взема предвид свързаността между компонентите на сградата (стаи и коридори), както и статичните обекти, които представляват препятствия за радиосигналите.

7.2.1 Анализ на проблема

Много технологии могат да бъдат използвани за създаване на системи за позициониране в закрити помещения (Zafari и колектив, 2019). Повечето системи за позициониране използват сензорна мрежа, за да изчислят приблизителната позиция на посетителя. Сензорните мрежи могат да използват различни видове сензорни възли, например: инфрачервени, видима светлина (Li-Fi), ултразвук и радиочестоти (RF) (Mendoza-Silva и колектив, 2019). На този етап най-често използваните RF технологии за локализация в закрити помещения са Wi-Fi, Ultra-Wideband (UWB), Radio Frequency Identification (RFID) и Bluetooth Low Energy (BLE). Една от най-обещаващите технологии за локализиране на посетители в сгради е BLE (Spachos и Plataniotis, 2020). Тази технология гарантира ниска до средна точност на локализация при ниски системни разходи и средна до висока сигурност, в зависимост от версията на стандарта Bluetooth.

Разполагането на сензорите с цел локализация се състои в поставянето на сензори, които обхващат средата на сградата и осигуряват желаната точност на локализация във всеки компонент на сградата (стаи и коридори) (He и колектив, 2017). Оптималното разполагане на маяци в сграда зависи от много фактори, например: геометричните размери на компонентите на сградата (стаи и коридори); видът на материала на вратите и стените; броят на посетителите в различните части на сградата, както и използваният алгоритъм за позициониране. Невъзможно е да се моделират в реално време всички тези фактори, особено при локализиране на посетители в големи обществени сгради. Въпреки това, има важни правила за разполагане на сензорни възли, които трябва да се спазват, независимо от използвания алгоритъм за позициониране:

- Желателно е позицията на посетителя да бъде вътре в фигурата (най-често триъгълник), която образуват референтните възли (маяците), избрани за изчисляване на позицията. Локализирането е най-точно, когато посетителят се намира в геометричния център на тази фигура. За да се сведе до минимум броят на маяците, тази фигура най-често е триъгълник. В този случай най-висока точност се получава, ако триъгълникът е равнобедрен.
- Използването на само два маяка за определяне на позицията на посетителя води до двусмислено решение. В този случай има две точки, които са еднакво вероятни като позиция на посетителя и затова е необходимо да се използва допълнителна геометрична и семантична информация за сградата, за да се отхвърли като невъзможна една от двете позиции. Това решение се използва най-често, когато не се изисква локализация с висока точност (< 1 м).
- Желателно е маяците да се поставят близо до вратите на стаите, за да се гарантира висока точност на локализация в точките, които свързват две пространства. Стратегията за разполагане на маяците трябва да вземе предвид и дали посетителите са хора с увреждания. За да се гарантира надеждна навигация, е необходимо да се локализируют посетителите с увреждания много по-точно, отколкото посетителите без увреждания. Например, хората с увреждания на зрението предпочитат да се движат близо до стените на коридорите и стаите. В този случай е по-добре да се използва правоъгълна форма за разполагане на маяците. При тази форма е по-вероятно посетителят да се намира в триъгълника, образуван от маяците с най-силен сигнал. От друга страна, хората без увреждания предпочитат да вървят по средата на коридорите. В този случай се получават по-добри резултати, ако маяците са разположени в триъгълна форма.

Поради тези ограничения, разполагането на сензори в сгради е сложна задача, за която в реална интериорна среда няма оптимално решение. Оптимизирането на разположението на сензорните възли въз основа на анализа на разпространението на сигнала от всеки възел, като се отчита ефектът на многократното отражение и затихването, не води до значително повишаване на точността на локализирането, защото:

- Моделирането на разпространението на радиосигнала е неточно, тъй като се използват опростени модели (Alfaresi и колектив, 2020), като моделът на загуба в свободното пространство и моделът на загуба по логаритмично разстояние. Тези модели не отчитат наличието на препятствия за радио вълните в помещенията. Модели като модела на

Motley-Keenan ([Kar и колектив, 2016](#)) отчитат наличието на препятствия, но не са достатъчно точни.

- Всички алгоритми за оптимизиране на разполагането на маяци въз основа на анализ на разпространението на радиосигнала са точни само за конкретна статична среда. В реална динамична среда, където има случайно движение на посетители, те не дават оптимално решение.

Точността на локализирането на посетителите в обществени сгради силно зависи от разполагането на сензорните възли ([Tian и колектив, 2020](#)). Точността зависи както от плътността на сензорите (брой възли на m^2), така и от формата, която сензорите образуват. Точността на локализирането се увеличава с намаляването на разстоянието между маяците. В големи обществени сгради разходите за разполагане се увеличават с намаляването на разстоянието между маяците. По тази причина трябва да се направи компромис между разходите за разполагане и точността на локализация. Експерименти показват, че освен от гъстотата на маяците, точността на локализация зависи и от формата, която образуват маяците. [Rezazadeh и колектив \(2018\)](#) доказват, че най-доброто разположение на маяците е, когато всеки три от тях образуват равнобедрен триъгълник. Тази форма се нарича Crystal-shape iBeacon Placement (CiP) и дава по-добри резултати в сравнение със стратегията за разполагане Z-curve, описана в ([Rezazadeh и колектив, 2014](#)). В реална сградна среда тези форми не могат да бъдат гарантирани поради специфичните размери на стаите и коридорите, както и поради необходимостта от различна точност на локализация в различните стаи и коридори. Все още съществуват критични предизвикателства, включително оптимизиране на разположението на сензорните възли за точна локализация, както и оптимизиране на местоположението на маяците в съществуваща безжична мрежа, които по същество са NP-пълни. В тази област са проведени многобройни научни изследвания и са предложени много методи за оптимизиране на разположението на сензорните възли ([Farsi и колектив, 2019](#)). Техниките, базирани на груба сила, техниките, базирани на мрежи, и техниките, базирани на изчислителна геометрия, принадлежат към класическите методи за разполагане. Техниките, базирани на груба сила измерват целевата среда, за да получат параметри, които се използват за намиране на оптималното разположение на сензорите. Тези методи отнемат много време и не са подходящи за динамични среди като обществени сгради. Техниките за разполагане на сензори, базирани на мрежи, се използват много често. Обикновено сензорните възли се разполагат във върховете на мрежата, като триъгълна мрежа, правоъгълна мрежа и др. На практика е трудно да се приложи

един и същ мрежов модел за всички части на сградата поради техните специфични геометрични размери. Третият тип техники за разполагане на сензори се основава на стратегията за изчислителна геометрия. При тази стратегия разполагането на сензорите се основава на анализа на геометрични обекти като полигони, сегменти и точки. Това изисква получаване на контурите на основните компоненти на сградата (стаи и коридори), както и на обектите в тях. Това предполага наличието на информационен модел на сградата.

Намирането на глобално оптимално разположение на сензорите в големи обществени сгради е задача, която изисква висока изчислителна мощност. По тази причина повечето методи за разполагане на сензори предоставят локално оптимално решение. Някои мета-евристични алгоритми ([Dokeroglu и колектив, 2019](#)) дават глобално оптимално решение. Основните недостатъци на такива алгоритми (генетични алгоритми, изкуствена пчелна колония, оптимизатори на рояк от частици, хармонично търсене и др.) са висока консумация на енергия, дълго време за обработка и повишени разходи.

Целта на предлагания алгоритъм е да се даде решение за автоматично разполагане на сензорни възли при създаването на IPS. Предложеното решение се основава на прилагането на експертните правила, описани в тази секция. Целта е да се предостави на дизайнерите на IPS възможно разполагане на сензорните възли. Това решение може лесно да бъде коригирано чрез добавяне или изтриване на възли, ако дизайнерът счете това за необходимо.

7.2.2 Описание на предлагания алгоритъм

Автоматичното разполагане на маяците в сградата преминава през три етапа:

- Разделяне на контурите на стаите и коридорите на непресичащи се правоъгълници;
- Автоматично разполагане на маяците в зависимост от ширината и височината на всеки правоъгълник; типа на върховете на правоъгълника (вдълбнати или изпъкнали), както и минималното и максималното прагово разстояние между маяците;
- Оптимизиране на разполагането на маяците, като се вземат предвид връзките между компонентите на сградата, както и позицията на препятствията за радиосигналите.

7.2.2.1. Разделяне на полигон на неприпокриващи се правоъгълници

Стаите и коридорите се моделират като прости полигони без отвори. Всеки полигон се описва като поредица от точки в посока обратна на часовниковата стрелка. Началната точка е тази, която има най-малките стойности на координатите x и y . Да приемем, че многоъгълникът се описва с масив от N точки, $P: [p_1, p_2, \dots, p_N, p_1]$. Разделянето на многоъгълника на прости фигури (най-често правоъгълници) е основен проблем в изчислителната геометрия. Намирането на минимален брой непресичащи се правоъгълници е NP -труден проблем. Намаляването на сложността на алгоритмите за разделяне на многоъгълници на непресичащи се правоъгълници най-често се основава на ограничения върху формата на многоъгълника (Piva и. de Souza, 2017).

Предлаганият алгоритъм приема, че стаите и коридорите могат да бъдат описани с прости, праволинейни полигони. Праволинейните полигони са прости свързани едноциклични графи, така че всяка от техните страни е перпендикулярна или успоредна на съседните страни. В повечето алгоритми за разделяне на полигони на правоъгълници първоначално се получава максималният възможен брой правоъгълници. След това броят на правоъгълниците се намалява до минимум чрез обединяване на някои от тях. Предлаганият алгоритъм използва няколко хеш карти и множества, за да ускори процеса на разделяне. Всички операции (получаване, добавяне, съдържание) с хеш карти имат сложност $O(1)$. За всеки връх (точка), сегмент и правоъгълник се получава хеш стойност. Това позволява сравнението на две точки, сегменти и правоъгълници да има сложност $O(1)$. Хеш стойността H_P за всяка точка се запазва в хеш множеството и се изчислява по следната формула:

$$H_P = BIG_INT_P * x + y, \quad (7.3)$$

където x и y са координатите на точката, а BIG_INT_P е голяма цялочислена стойност. Хеш стойността H_S за сегмент $\overline{AB}$ се изчислява по подобен начин:

$$H_S = BIG_INT_S * x_A + x_B, \quad (7.4)$$

където x_A и x_B са x координатите на точките A и B , които образуват сегмента. Координатите на точките на сегмента се запазват в хеш карта. По този начин сегментите с еднакви x координати могат да бъдат групирани много бързо. Всеки два последователни сегмента с еднакъв хеш образуват правоъгълник. Координатите на върховете на всеки правоъгълник също се запазват в хеш карта. Целта е да се

групираат всички правоъгълници с еднаква максимална y координата. За тази цел стойността на y координатата се задава за хеш стойността:

$$H_R = y_{max}. \quad (7.5)$$

Алгоритъмът за разделяне на многоъгълника на неприпокриващи се правоъгълници е описан на Фиг. 7.1.

Input: An array with polygon (room, corridor) points P ; hash set that contains hash value for each polygon point HP ; an empty linked hash map for line segments S ; an empty array for unprocessed segments UPS .

Output: Linked hash map with rectangles R и array UPS

```

1:  $YPOS$  = get set of unique  $y$  coordinates of points in  $P$ 
2: foreach  $y$  in  $YPOS$  do
3:    $intersections = P \cap l: y$ 
4:    $segments$  = get all segments that  $intersections$  form
5:   remove segments in  $segments$  that are not a part of  $P$ 
6:   foreach  $segment$  in  $segments$  do
7:      $H_s$  = calculate hash value for  $segment$ 
8:     add  $segment$  to hash map  $S$  for  $H_s$  value
9:   end
10:  foreach  $key$  in  $S$  do
11:     $points$  = get data from map  $S$  for  $key$ 
12:    if  $points$  form a valid rectangle  $r$  then
13:       $H_R$  = calculate hash for rectangle  $r$ 
14:      add rectangle  $r$  to hash map  $R$  for  $H_R$  value
15:    end
16:    remove invalid segments in  $S$  for hash =  $key$ 
17:    memorize unprocessed segments to  $UPS$ 
18:  end
19: end
20: return  $R, UPS$ 

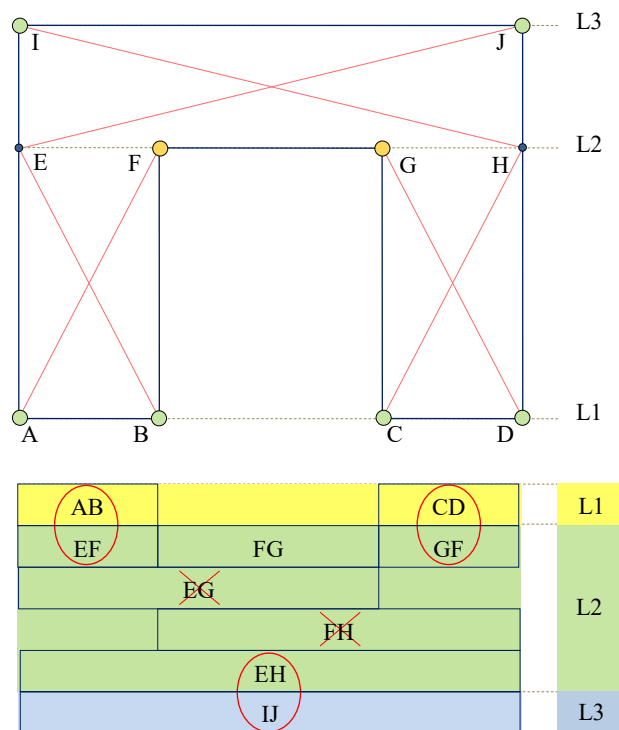
```

Фиг. 7.1 Алгоритъм *polygonPartitioning* – просто праволинейно разделяне на многоъгълник на неприпокриващи се правоъгълници

Първо, намираме всички сортирани, неповтарящи се y координати на точките на многоъгълника, $YPOS$ (ред 1). Основният цикъл (редове 2-19) се повтаря за всяка y координата в $YPOS$. След това се изчисляват пресечните точки на многоъгълника с линията $l: y$ (ред 3). Получават се всички сегменти, които тези точки образуват (ред 4). Сегментите, които не са част от полигона, се премахват (ред 5). За всеки валиден сегмент се изчислява хеш стойност и точките на сегмента се запазват в хеш карта S (редове 6-9). За всеки ключ в S (редове 10-18) проверяваме дали има два сегмента за дадена стойност на ключа и дали те образуват валиден правоъгълник r . Правоъгълникът е валиден, ако $r \subset P$. Този правоъгълник се запазва в хеш картата R (ред 14). Накрая, невалидните сегменти в хеш картата S се премахват (ред 16) и всички необработени, но валидни сегменти се запамятват (ред 17). Правилата за

невалидност на един сегмент са следните: (1) сегментът е страна на правоъгълник; и (2) страната на правоъгълника е част от сегмента.

На Фиг. 7.2 е показано как работи алгоритъм *polygonPartitioning*, при разделянето на коридор с П-образна форма.



Фиг. 7.2 Пример за разделяне на полигони (коридор)

Коридорът се описва като полигон $P: \{A, B, F, G, C, D, J, I, A\}$. Този многоъгълник има три уникални стойности на у координатите на върховете си. Следователно, за да се получат всички правоъгълници, е необходимо да се повтори три пъти основната верига на алгоритъма ($L1$, $L2$ и $L3$). Валидни сегменти за ниво $L1$ са $\overline{AB}$ и $\overline{CD}$. След изчисляване на техните хеш стойности, те се съхраняват в хеш карта S . Сегмент $\overline{BC}$ е невалиден, защото не принадлежи към многоъгълника. За ниво $L2$ се получават следните сегменти: $\overline{EF}$, $\overline{FG}$, $\overline{GH}$, $\overline{EG}$, $\overline{FH}$ и $\overline{EH}$. Всички тези сегменти се запазват в хеш карта S . Получават се два валидни правоъгълника: $\{A, B, E, F\}$ и $\{C, D, G, H\}$. На ниво $L3$ има само един валиден сегмент - $\overline{IJ}$. При записването на този сегмент в хеш карта S се получава нов правоъгълник $\{E, H, I, J\}$. Невалидните сегменти $\overline{EG}$ и $\overline{FH}$ се премахват. Сегментът $\overline{FG}$, който е валиден, но не принадлежи към правоъгълник, се запазва в масива UPS . В резултат се получават три правоъгълника, чиито диагонали са показани на Фиг. 7.2 с червен цвят.

7.2.2.2 Автоматично разполагане на маяците

Предлаганият алгоритъм за разполагане на маяци се основава на информацията, получена от алгоритъма за разделяне на многоъгълника на правоъгълници. Всеки правоъгълник може да описва стая, коридор или част от стая или коридор. Възможно е да се посочи в коя стая не трябва да се разполагат маяци, както и област (многоъгълник) от стая, в която не трябва да се поставят маяци. Алгоритъмът работи с два вида правила за разполагане на маяци. Основните правила, получени въз основа на експертна оценка, не могат да бъдат променяни. Проектантите на IPS имат възможност да интегрират допълнителни правила от типа if-else.

Алгоритъмът избира конкретна стратегия за разполагане на маяци във всеки правоъгълник. Стратегията зависи от множество параметри, например:

- Прагова стойност за максималното разстояние между маяците, *MAXD*. Алгоритъмът поставя и премахва маяци, така че разстоянието между два съседни маяка да не е по-голямо от *MAXD*.
- Прагова стойност за минималното разстояние между маяците, *MIND*. Алгоритъмът поставя и вмъква маяци така, че разстоянието между два съседни маяка да не е по-малко от *MIND*.
- Форма на разполагане. На този етап алгоритъмът позволява задаване на две форми на разполагане – правоъгълна и триъгълна. Ако е зададена триъгълна форма на разполагане, алгоритъмът формира колкото се може повече равностранны триъгълници. Ако това не е възможно, се формират равнобедрени триъгълници.
- Тип правоъгълник. Типът на правоъгълника зависи от неговата ширина и височина. На този етап се използват четири типа правоъгълници. За всеки тип правоъгълник има предварително дефиниран модел за разположението на маяците. Този модел се възпроизвежда в ширината, височината или и в двете.
- Позиция на вдлъбнатите върхове на многоъгълника, който описва стая или коридор. Тези точки са важни за алгоритмите за позициониране, тъй като ако маяците са разположени на тяхната позиция, е по-вероятно да се получи сценарий за локализация по линия на видимост (LOS). По тази причина маяците трябва да бъдат разположени на тези точки, освен ако разстоянието между тях е по-малко от *MIND*.

Алгоритъмът за разполагане на маяци е показан на Фиг. 7.3.

Input: Hash map with computed rectangles R ; Hash map with unprocessed segments for each y level UPS ; Deployment shape type DST ; minimum distance between beacons $MIND$; maximum distance between beacons $MAXD$; an empty linked hash set for beacon's position hash BH

Output: An array with position of beacons, BP

```

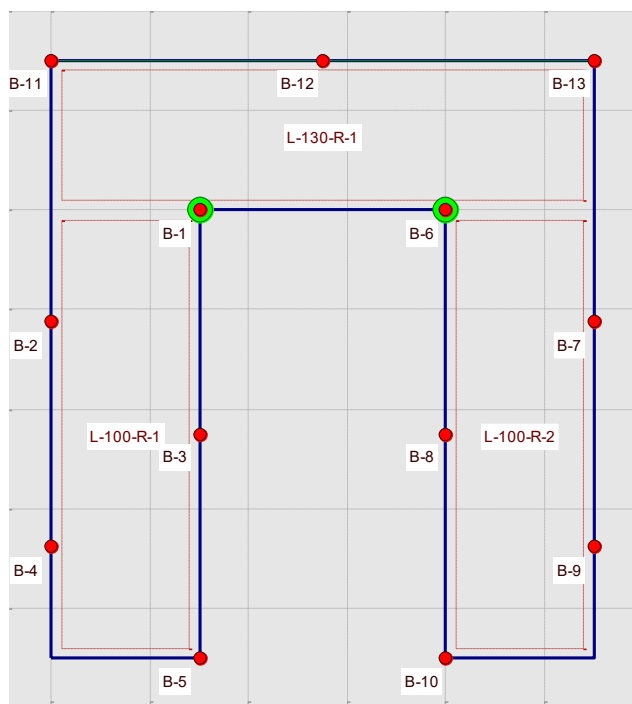
1: concave = getConcavePoints( $P$ )
2: foreach hash value in  $R$  do
3:    $y$  = hash
4:   foreach rectangle in  $R$  for level  $y$  do
5:      $rType$  = getRectangleType(rectangle)
6:     beaconP = deploy ( $rType$ , concave,  $MIND$ ,  $MAXD$ ,  $DST$ )
7:     foreach point in beaconP do
8:       hash = calculateHashForPoint (point)
9:       status = add hash to  $BH$ 
10:      if status == true do
11:        add beacon position (point) to array  $BP$ 
12:      end
13:    end
14:  end
15: foreach segment in  $UPS$  do
16:   beaconPoits = deployment (segment,  $MAXD$ )
17:   add beaconPoits to array  $BP$ 
18: end
19: end
20: return  $BP$ 

```

Фиг. 7.3 Алгоритъм *abd* – автоматично разполагане на маяци

Започва се с намиране на всички вдлъбнати върхове на многоъгълника (ред 1). След това следва основният цикъл (редове 2-19), през който се обхождат всички получени правоъгълници за всяко ниво y (ред 3). За всеки правоъгълник от ниво y (редове 4-14) се получава неговият тип (ред 5). Позициите на маяците в рамките на текущия правоъгълник се получават в ред 6. За всяка позиция на маяк се изчислява хеш стойност (ред 8). Тя се запазва в хеш множеството BH (ред 9). Хеш множеството се използва, за да се предотврати дублирането на позициите на маяците. Ако позицията на маяка е уникална (ред 10), тя се съхранява в масива BP (ред 11). Накрая, маяците се поставят на всички сегменти, които не са част от правоъгълник (редове 15-18). След изчисляване на позицията на тези маяци (ред 16), те се съхраняват в масива BP (ред 17).

Фиг. 7.4 показва как се разполагат маяците при използване на алгоритъм *abd* в многоъгълник, чиято форма описана на Фиг. 7.2. Площта на многоъгълника е 8,7 m². Правоъгълниците, получени чрез алгоритъм *polygonPartitioning*, са показани с червени пунктирани линии. Избрана е триъгълна форма за разполагане на маяците.



Фиг. 7.4 Пример за разполагане на маяци: триъгълна форма; максималното разстояние между маяците е 6 m; вдлъбнати ръбове (зелени кръгчета); и маяци (червени кръгчета)

Максималното разстояние между маяците е зададено да бъде 6 m. За вертикалните правоъгълници разполагането на маяците започва от вдлъбнатите върхове, маркирани с зелени кръгчета. Позицията на всеки маяк е показана с червено кръгче. Общият брой маяци за многоъгълника е 13. Плътността на разполагане на маяците е 1,5 маяка на m^2 ($13/8,7 m^2$). Ако максималното разстояние между маяците се намали на 5 m, броят на маяците се увеличава на 19. В този случай гъстотата на разполагане на маяците е 2,18 маяка на m^2 . В случай на правоъгълна форма на разполагане, броят на маяците се увеличава на 21, ако максималното разстояние между съседни маяци е 5 m, и на 15, ако това разстояние е 6 m.

7.2.2.3 Оптимизация на позициите на маяците

В реална сграда е необходимо да може да се зададе различна плътност на маяците за отделните компоненти на сградата (стаи и коридори). Точността на локализация зависи както от инфраструктурата на сградата, така и от типа посетители. Някои от посетителите могат да бъдат хора с увреждания (например зрителни и двигателни увреждания). В този случай е необходима по-висока точност на локализация, за да се гарантира надеждна навигация. В обществените сгради е важно да се осигури точно позициониране в близост до вратите, когато се предлагат специфични

услуги. В тези случаи е необходимо да се оптимизира позицията на маяците, получена с помощта на алгоритъм *abd*. Това се реализира чрез алгоритъм, описан като алгоритъм *obd* (виж Фиг. 7.5).

Input: Building Information Model, *BIM*; an array with type of building components (rooms, corridors, stairs, etc.) that should not contain beacons *beaconlessComponent*; deployment shape type *DST*; minimum и maximum distances between beacons, *MIND* и *MAXD*.

Output: Optimized positions of beacons for each building component.

```

1: components = getNodes(BIM)
2: doors = getLinks(BIM)
3: foreach component in components do
4:   if type of component is not in beaconlessComponent then
5:     P = getPolygon (component)
6:     component.beacons = abd(P, DST, MIND, MAXD)
7:   end
8: end
7: foreach door in doors do
8:   if type of door is not 'virtual' и door.withBeacon then
9:     componentIDs = getIdOfComponents(door)
10:    foreach id in componentIDs do
11:      component = components (id)
12:      beacons = component.beacons
13:      obstacles = getObstacles(component)
14:      TH = getDistThreshold(component.dimensions)
15:      foreach beacon in beacons do
16:        if distance(beacon, door.position) < TH then
17:          remove beacon from component.beacons
18:        end
19:        if beacon ∩ obstacles ≠ ∅ then
20:          correct beacon position
21:          update beacon in component.beacons
22:        end
23:      end
24:    end
25:  end

```

Фиг. 7.5 Алгоритъм *obd* – оптимизация на разполагането на маяци

Този алгоритъм позволява да се зададе различна плътност на маяците за различните компоненти на сградата (стаи и коридори), както и опция за задължително разполагане на маяци в близост до предварително избрани врати. Алгоритъмът позволява да се използва моделът на сградата (BIM), за да се получи информация за геометричните размери и позицията на всички компоненти, тяхната свързаност, както и наличната семантична информация (редове 1-2). Получава се местоположението на маяците за всеки компонент на сградата (редове 3-7). За всяка врата, която не е виртуална и нейният флаг *withBeacon* е *true*, позициите на маяците се оптимизират (редове 7-24). За всяка врата се получават

идентификаторите на компонентите, които тя свързва (ред 9). За всеки от тези компоненти (ред 11) се получава обект, който описва всички маяци, които принадлежат към него (ред 12). Всички обекти препятствия, намиращи се в компонента (ред 13), също се извличат. Въз основа на размерите на компонента се изчислява праговото разстояние TH (ред 14). Алгоритъмът премахва всеки маяк (ред 17), за който разстоянието между позицията на маяка и позицията на вратата е по-малко от TH (ред 16). След това се проверява дали някое от препятствията блокира сигнала от маяка (ред 19). Ако има колизия, позицията на маяка се актуализира, за да се избегне блокирането на сигнала (ред 20), и информацията за този маяк се актуализира (ред 21).

7.2.3 Експерименти

В тази секция са реализирани няколко тестови сценария, за да се анализира точността и приложимостта на предлаганите алгоритми. За тази цел е използвана тестова платформа, която включва:

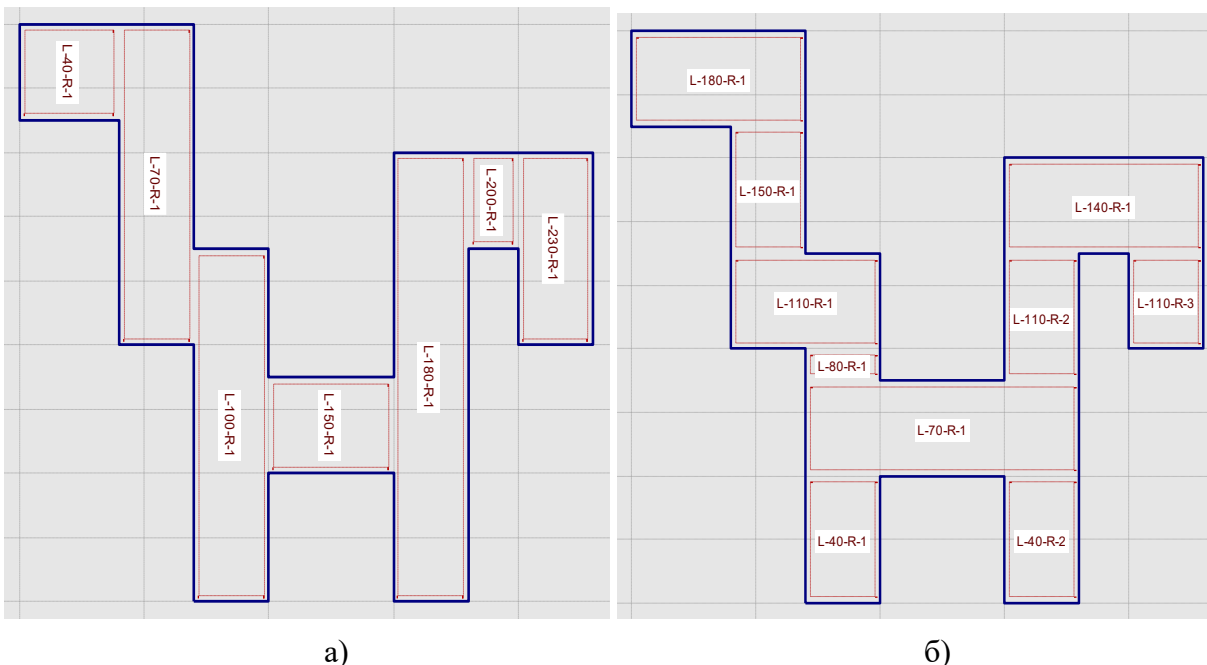
- Модел на сградата, описан в (Ivanov, 2017). Този модел предоставя информация за геометричните размери на стаите и коридорите, връзките между тях и размерите и местоположението на обектите (ориентири, препятствия и опасности);
- Софтуер на Matlab, който визуализира геометричните размери на елементите на сградата и връзките между тях;
- Софтуер на Matlab, който прилага предложения алгоритъм за разделяне на контурите на компонентите на сградата на неприпокриващи се правоъгълници;
- Софтуер на Matlab, който прилага предложения алгоритъм за разполагане на сензори;
- Софтуер на Matlab, който получава приблизителна оценка на местоположението на посетителя въз основа на анализ на местоположението на N брой сензори;
- Софтуер на Matlab за търсене на възможни маршрути в сгради, както и визуализация на движението на посетителите по даден маршрут.

7.2.3.1 Тестов сценарий 1

Целта на първият експеримент е да се анализира приложимостта на алгоритъма за разделяне на компонентите на сградата на неприпокриващи се правоъгълници.

Анализът на този алгоритъм се осъществява с помощта на база данни от 100 компонента на сграда (80 симулирани и 20 реални). Всеки компонент на сградата е описан като прост многоъгълник.

Приложението позволява да се зададе посоката на разлагане на многоъгълника (x или y). Възможно е да се избере автоматично посоката на разлагане, за да се получи минимален брой правоъгълници. Фиг. 7.6 показва резултата, получен след разделянето на коридор (номер в базата данни 25) в посока x и y .



Фиг. 7.6 Разделяне на коридора на неприпокриващи се правоъгълници (червени пунктирни линии): а) разлагане по ос x (7 правоъгълника); б) разлагане по ос y (10 правоъгълника)

Когато многоъгълникът се обработва в посока x , се получават 7 правоъгълника, а когато многоъгълникът се обработва в посока y , се получават 10 правоъгълника. Трябва да се има предвид, че минималният брой правоъгълници не винаги гарантира разполагането на минимален брой маяци. Техният брой зависи и от минималното разстояние между маяците и избраната форма на разполагане. Приложението позволява да се зададе посоката на разлагане на многоъгълника (x или y). Възможно е да се избере автоматично посоката на разлагане, за да се получи минимален брой правоъгълници.

7.2.3.2 Тестов сценарий 2

Целта на втория експеримент е да се потвърди приложимостта на алгоритъма за автоматично разполагане на маяци. Разработен е софтуер Matlab, който позволява

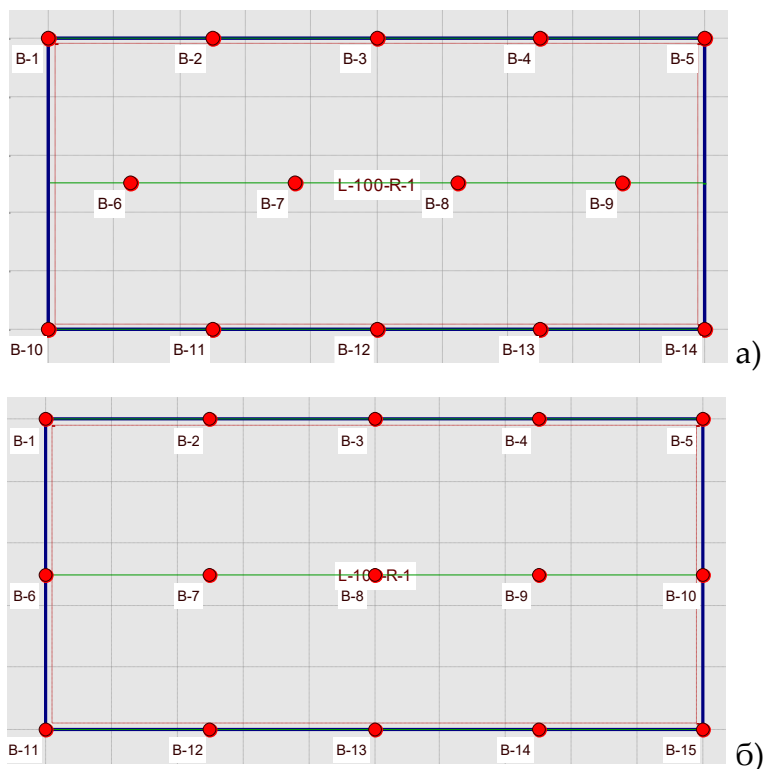
прилагането на алгоритъма.

Възможно е да се зададат следните входни параметри:

- Минимално и максимално разстояние между маяците. Алгоритъмът разполага маяците така, че разстоянието между всеки два съседни маяка да е между минималната и максималната стойност;
- Форма на разполагане на маяците.

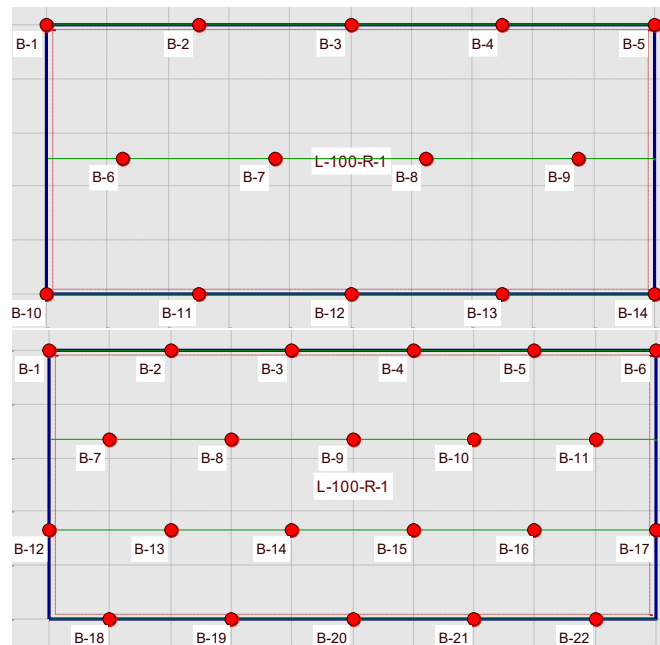
На този етап е възможно да се зададе триъгълна или правоъгълна форма на разполагане. При избор на триъгълна форма на разполагане алгоритъмът разполага маяците във върховете на равностранны или равнобедрени триъгълници. При избор на правоъгълна форма на разполагане алгоритъмът разполага маяците във върховете на колкото се може повече правоъгълници.

Фиг. 7.7 показва резултата, получен от алгоритъма за автоматично разполагане на маяци в стая с правоъгълна форма (20 x 10 m). Минималното разстояние между маяците е 5 m. За стаята, при избор на триъгълна форма (Фиг. 7.7а), броят на маяците е 14, а за правоъгълна форма (Фиг. 7.7б) – 15.

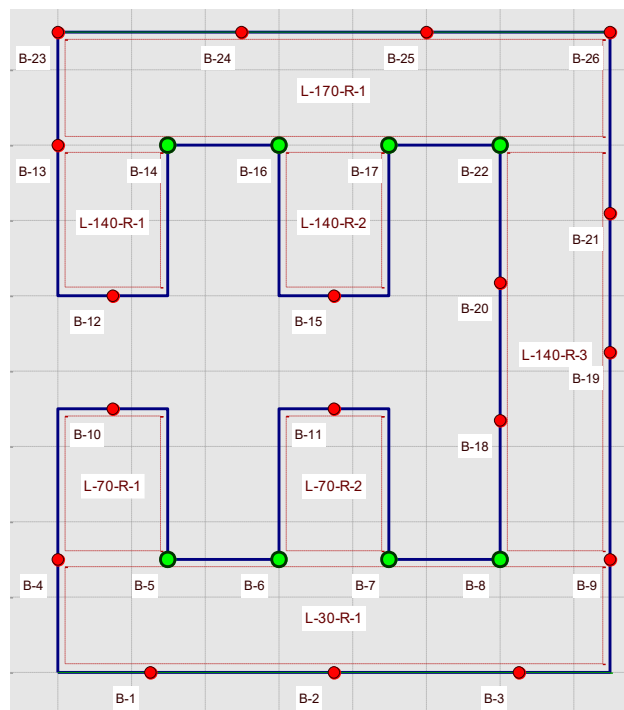


Фиг. 7.7 Автоматично разполагане на маяци: а) стая с правоъгълна форма, триъгълно разполагане; б) стая с правоъгълна форма, правоъгълно разполагане

На Фиг. 7.8 е показано разполагането на маяци в правоъгълна форма при различно максимално разстояние между маяците.



Фиг. 7.8 Разположение на маяците в правоъгълна зона (20 м x 10 м) за триъгълна форма и различна стойност за максималното разстояние между маяците

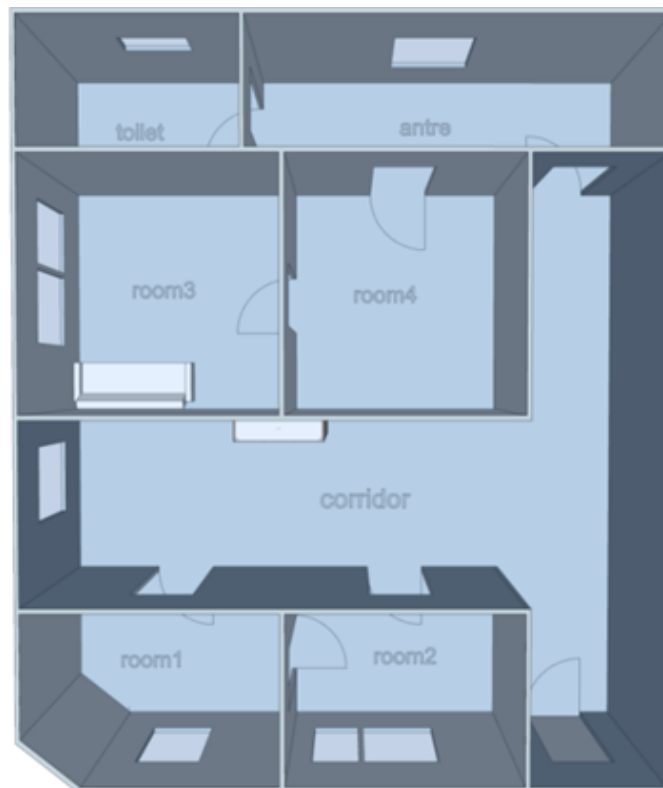


Фиг. 7.9 Разполагане на маяци в коридор със сложна форма, триъгълно разположение

На Фиг. 7.9 е показано разполагането на маяци за интериор със сложна геометрична форма. Избрано е триъгълно разполагане на маяците. Зададено е задължителното разполагане на маяци във всички вдлъбнати точки (зелени кръгчета). При тези входни данни алгоритъмът предлага да се използват 26 маяка.

7.2.3.3 Тестов сценарий 3

Този сценарий анализира приложимостта на алгоритъма за оптимизация на разполагането на маяци. За тази цел се анализира точността на процеса на разполагане в реална сградна среда, показана на Фиг. 7.10.

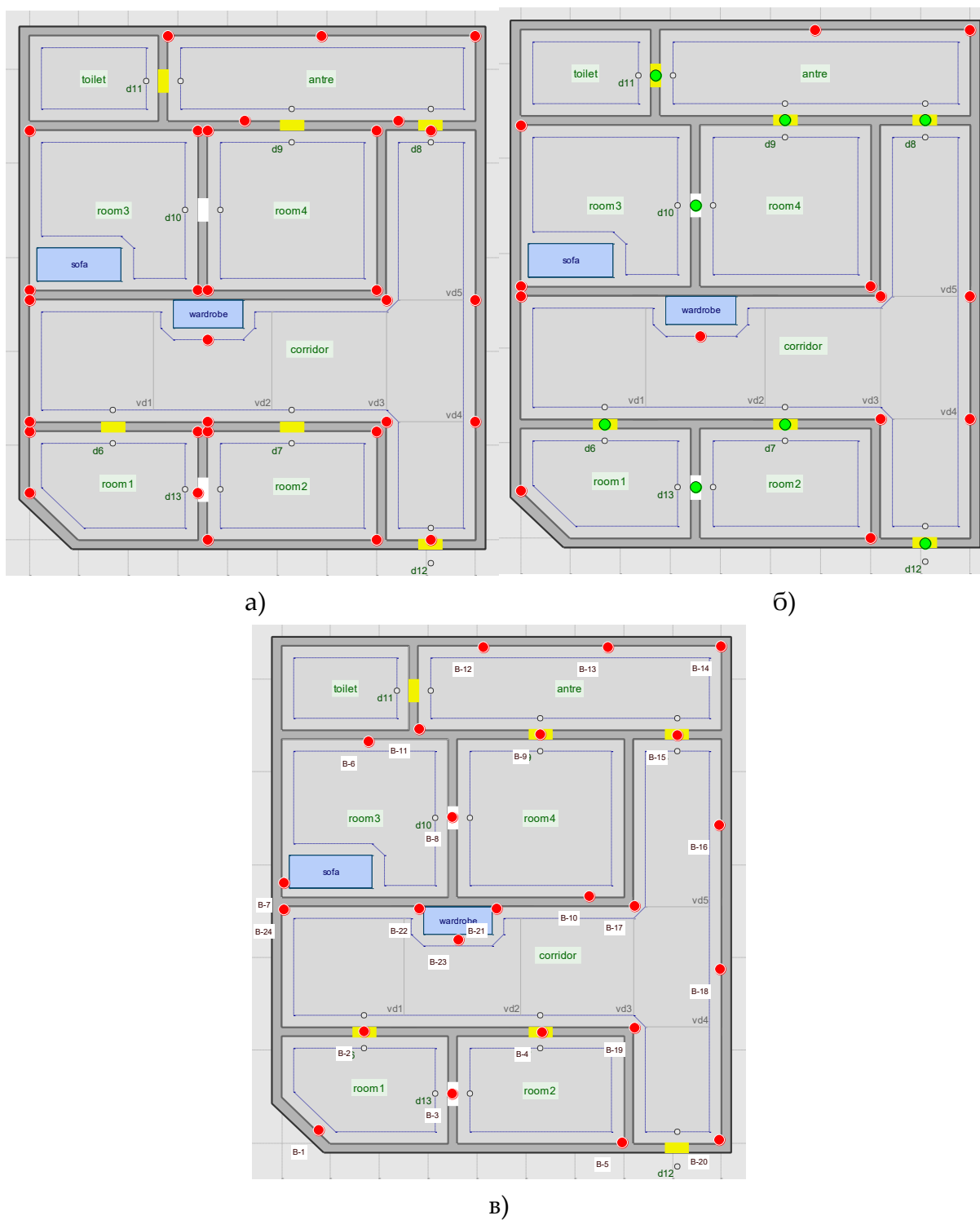


Фиг. 7.10 Тестов сценарий 3 – тестова среда (3D модел)

Маяците са разположени на един етаж, който се състои от пет стаи, коридор и входно фоайе. Общата площ на достъпните компоненти на сградата е приблизително 100 m². Правилата, които са зададени за алгоритъма за разполагане на маяци, са както следва:

- Не трябва да се поставят маяци в една от стаите (тоалетна);
- Минималното разстояние между маяците трябва да е 4 m, а максималното разстояние – 6 m;
- Предпочитаната форма на разполагане е триъгълна.

Фиг. 7.11 показва разположението на сензорите в три сценария.



Фиг. 7.11 Разположение на маяците: а) преди оптимизация (31 маяка); б) след оптимизация (21 маяка); и в) експертно разположение (24 маяка)

В първия сценарий се използва автоматично разполагане на сензорите (алгоритъм *abd*). За малките помещения е зададено правило за използване на четири маяка. В този случай (Фиг. 7.11а) броят на сензорите е 31, а гъстотата на разполагане е 0,34 маяка на m^2 .

Във втория сценарий броят на сензорите се оптимизира автоматично с помощта на алгоритъм *obd*. Зададено е правило за поставяне на маяк до всяка врата. В този случай (Фиг. 7.11б) броят на сензорите се намалява до 21, а плътността на разполагане е 0,23 маяка на m^2 .

В последния сценарий разполагането на сензорите се осъществява от експерт в областта на локализирането в закрити помещения. Целта на експерта е да сведе до минимум броя на използваните маяци, като се има предвид, че сред посетителите на сградата може да има хора с увреждания на зрението. Резултатът от разполагането на маяците е показан на Фиг. 7.11в. Експертът отбелязва наличието на препятствие за радиосигналите в коридора (гардероб). Той е разположил допълнителни маяци от двете страни на това препятствие. Освен това целта е била да се сведе до минимум броят на маяците, които са близо един до друг, но се намират в различни стаи. В този случай броят на сензорите е 24 (гъстотата на разположение е 0,27 маяка на m^2).

7.2.3.4 Сравнителен анализ на сценариите

За да анализираме приложимостта на всеки от сценариите от 1 до 3 за разполагане на сензори, се оценява точността на локализация, когато посетителите преминават по избрания маршрут, например от стая 3 до стая 1. Точността на локализация се получава като общо отклонение от маршрута (най-краткото разстояние между изчислената позиция и маршрута).

Програмното осигуряване за търсене на маршрут е настроено да се намират маршрути за хора с увредено зрение. Поради тази причина се изчислява възможен маршрут, който минава на разстояние 0,8 м по стените на стаите и коридора, тъй като хората със зрителни увреждания предпочитат да се придвижват по този начин в сградите. Има два възможни маршрута от стая 3 до стая 1. Дължината им е 25,3 m и 26,4 m. Избран е маршрутът, който не минава покрай препятствието (гардероб) (виж Фиг. 7.12).

Табл. 7.1 Обобщение на резултатите – минимални, средни и максимални стойности на грешката (отклонение от референтните точки) и стандартно отклонение

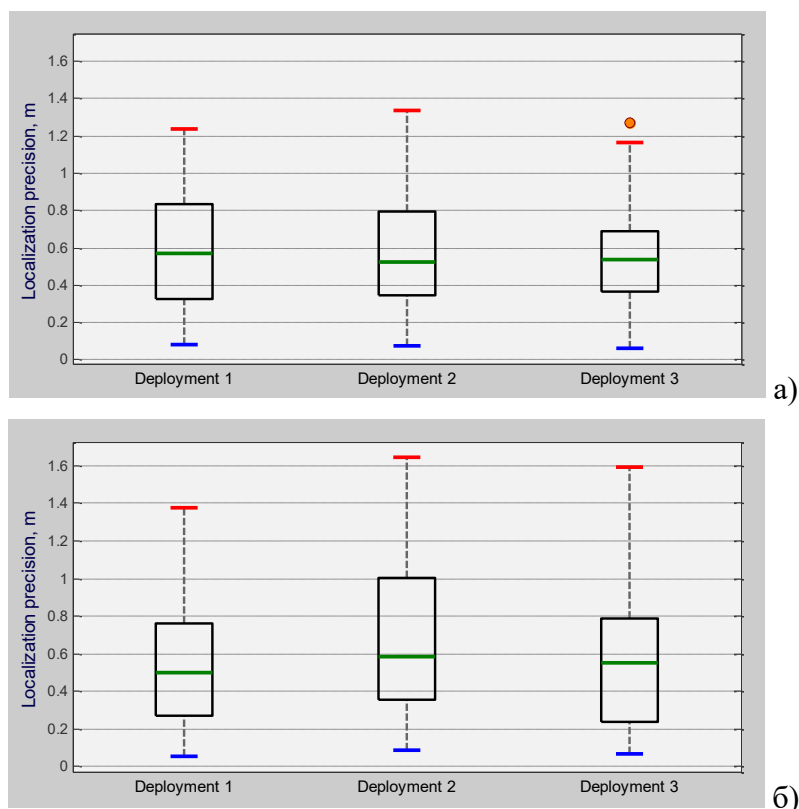
Deployment scenario	Localization technique	Error, m			Standard deviation, m
		min	mean	max	
1	NLLS	0.08	0.60	1.24	0.29
	Trilateration	0.07	0.55	1.37	0.30
2	NLLS	0.10	0.59	1.34	0.29
	Trilateration	0.07	0.64	1.79	0.41
3	NLLS	0.10	0.55	1.27	0.27
	Trilateration	0.08	0.60	1.61	0.36

Най-малката стойност на стандартното отклонение е 0,27 m. Тя се получава в сценарий 3 (разполагане от експерт) и алгоритъма за локализация NLLS. При използване на алгоритъма за локализация NLLS, оптимизираната версия на автоматично разполагане на сензори (сценарий 2) има подобна стойност на стандартното отклонение (0,29 m) като неоптимизираната версия (сценарий 1). Очакваната минимална стойност (1,24 m) за максималната стойност на грешката се получава в сценарий 1, където се използват най-много сензори (31).

Фиг. 7.13 показва сравнение на точността на локализация в различни сценарии за разположение на маяци, като се използва диаграма „box-и-whisper“. Фиг. 7.13а показва точността на локализация при използване на алгоритъма NLLS. Резултатите, получени при използване на алгоритъма Trilateration, са показани на Фиг. 7.13б. Алгоритъмът NLLS дава по-добри резултати във всички анализирани сценарии за разположение на маяци в сравнение с алгоритъма Trilateration. Алгоритъмът NLLS има две други предимства пред алгоритъма Trilateration:

- Данните от три или повече сензора могат да се използват за получаване на приблизителна оценка на позицията на целевия възел;
- Сензорите, участващи в изчисляването на позицията, могат да бъдат разположени на една линия.

При алгоритъма NLLS няма значителна разлика в точността на локализация в сценарий 1 и сценарий 2.



Фиг. 13 Сравнение на сценарии за разполагане на маяци с помощта на алгоритъм за локализация NLLS (а) и Trilateration (б): диаграма „box-и-whisper“ (средна стойност – зелена линия; горен и долен мустак – черна кутия; горна съседна стойност – червени лин

Грешката при локализация в 75% от контролните позиции е в диапазона 0,35-0,82 m. Този интервал в сценарий 3 е 0,38-0,7 m. Получените резултати показват, че при използване на оптимизирания вариант на алгоритъма за разполагане на сензорите средната стойност на грешката и стандартното отклонение са подобни на тези, получени при експертното позициониране на сензорите. Можем да заключим, че предложените алгоритми за разполагане на маяци могат да бъдат успешно използвани за внедряване на системи за позициониране в закрити помещения.

7.2.4 Заключение

Разработени са алгоритми и софтуер за автоматично разполагане на маяци в сгради, които могат да се използват от проектантите на системи за навигация в закрити помещения. Този софтуер прилага три нови алгоритъма:

- Алгоритъм за разделяне на компонентите на сградата на неприпокриващи се правоъгълници;

- Алгоритъм за разполагане на маяци, който се основава на предварително дефинирани експертни правила;
- Алгоритъм за оптимизиране на разполагането на маяци чрез анализ на формата, размерите и свързаността на компонентите на сградата.

Реализирани са 3 сценария за тестване на предложените алгоритми са реализирани чрез тестването им в симулирана и реална среда на сгради. Проведените експерименти показват, че предложените алгоритми могат да бъдат успешно използвани за навигация в реални сгради. Основният недостатък на алгоритъма за разделяне на компонентите на сградите е, че той работи само с многоъгълници без отвори.

7.3 Алгоритъм за избор на маяци с цел точна локализация

7.3.1 Анализ на проблема

С развитието на комуникационните технологии, безжичните сензорни мрежи и широката достъпност на смартфоните, значението на услугите, базирани на местоположението (LBS), става все по-важно както за индустрията, така и за обществения живот. Приложимостта на LBS зависи от точното изчисляване на местоположението на посетителите. Съществуват множество техники, технологии и алгоритми, които се използват за локализиране на посетителите в сградите ([Obeidat и колектив, 2021](#)). Всички те имат специфична точност, цена, предимства и недостатъци. Най-разпространените системи за позициониране в закрити помещения използват сензорна мрежа за изчисляване на приблизителната позиция на посетителя (целта). Сензорните мрежи могат да използват различни видове сензорни възли, например: инфрачервени, видима светлина, ултразвук, звук и радиочестота (RF).

Една от най-обещаващите RF технологии за локализиране на посетители в сгради е Bluetooth Low Energy (BLE) ([Spachos и Plataniotis, 2020](#)). Тази технология гарантира ниска до средна точност на локализация при ниски системни разходи и средна до висока сигурност, в зависимост от версията на Bluetooth стандарта. Повечето услуги за локализация, базирани на BLE, използват BLE маяци за реализиране на своите сензорни мрежи. Всеки BLE маяк изпраща рекламни пакети на определени интервали. Основният недостатък на системите за локализация в закрити помещения, използващи RF сензорни мрежи, са динамичните вариации в параметрите на радиоканалите, дължащи се на следните по-важни фактори: размери на стаите и коридорите, вид на строителните материали на стените и

таваните и положението на статични и динамични обекти в сградата. В повечето системи за локализация, базирани на RF, изчисляването на разстоянието между посетителя и сензорния възел се основава на стойността на индикатора за силата на получения сигнал (RSSI) ([Sadowski и Spachos, 2018](#)). Силата на предавания сигнал от BLE маяци намалява експоненциално, главно поради разстоянията, но има и други ефекти, които повече или по-малко влияят на силата на сигнала.

Точността на локализирането при използване на безжични сензорни мрежи зависи от редица фактори, като: алгоритъм за локализиране; стратегия за разполагане на маяци; избор на маяци, чрез които се получава приблизителна оценка на позицията на посетителя; наличието на статични (елементи на сградата и обекти) и динамични (посетители) препятствия за радиосигналите. Изборът на маяци е трудна задача. Най-често се избират маяците с най-висока стойност на RSSI. В реална интериорна среда стойностите на RSSI са нестабилни, главно поради ефекта на разпространение на радиовълните в много направления, поради тяхното отразяване. Следователно е необходимо да се вземат предвид както нивото на RSSI, така и качеството на комуникационните канали. Когато е необходимо да се сведе до минимум влиянието на динамичните смущения на радиосигналите, трябва да се използват нелинейни филтри, като Extended Kalman Filters ([Paterna и колектив, 2017](#)) или Particle Filters ([Zafari и Papapanagiotou, 2015](#)).

Друг подход за подобряване на точността на системите за локализация е да се избират само тези маяци, които имат пряка видимост с целевия възел - сценарий с пряка видимост (LOS). Съществуват много статистически техники за откриване на разпространение на радиосигнали без пряка видимост (NLOS) ([Yu и Guo, 2009](#)). Основният недостатък на тези техники е необходимостта от достатъчно статистически данни за точно идентифициране на NLOS. Тези техники са трудни за прилагане в реално време.

В реална среда е много вероятно да бъде избран правилно само маякът с най-висока стойност на RSSI. За другите два или повече маяка трябва да се вземе предвид каква форма образуват с най-близкия маяк. Изследванията показват, че най-висока точност се получава, ако формата на разположението на маяците е изпъкнал правилен многоъгълник ([Rezazadeh и колектив, 2014](#)).

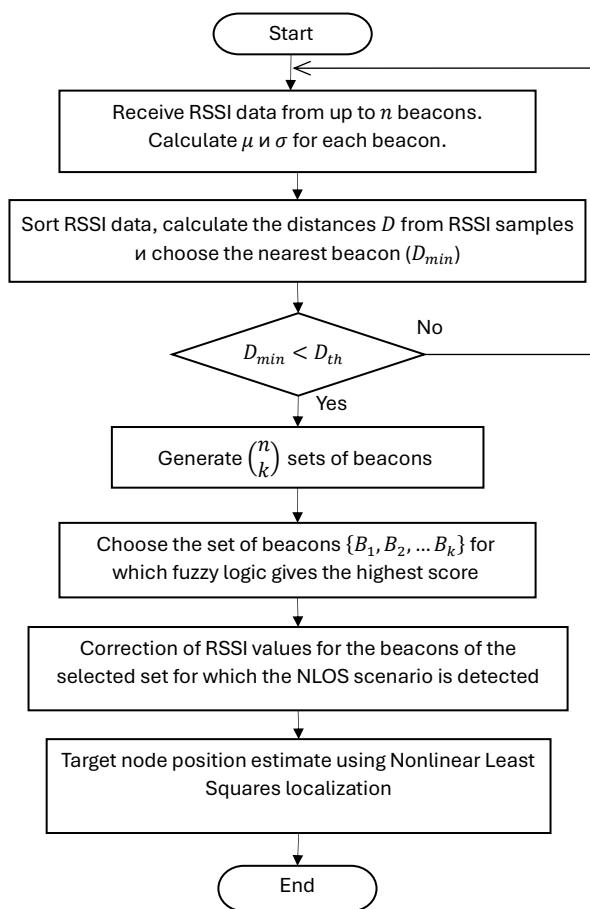
Точната локализация на посетителите в обществени сгради е ключова технология за създаване на достъпни среди с помощта на услуги, базирани на местоположението. За да се оптимизира съотношението цена-точност, локализацията най-често се осъществява чрез използване на безжични сензорни

мрежи. В този случай нивото на колебания на сигналите от сензорите е високо, главно поради наличието на статични и динамични препятствия за радиосигналите.

Целта на предлагания алгоритъм е автоматичен избор на маяци при създаването на точни системи за позициониране в закрити пространства. Предложеното решение се основава на прилагането на размита логика, за да се изчисли количествена оценка на качеството на всяка група от k на брой маяка, чрез които се получава приблизителна оценка на позицията на посетителя.

7.3.2 Описание на предлагания алгоритъм

Диаграмата на предложените алгоритми за автоматичен избор на маяк и оценка на позицията на целевия възел е илюстрирана на Фиг. 7.14.



Фиг. 7.14 Диаграма на предложения алгоритъм

Предполага се пряка видимост към маяка, най-близък до целевия възел. Ако μ и σ са средната стойност и стандартното отклонение на данните RSSI, то според принципа 3-сигма вероятността стойностите RSSI да са в интервала $(\mu - 3\sigma, \mu + 3\sigma)$

е 99,73%. В предложения алгоритъм се приема, че има пряка видимост между целевия възел и най-близкия маяк, ако разстоянието между тях D_{min} е по-малко от предварително определената прагова стойност D_{th} :

$$D_{th} = \frac{D_{bs}}{2} + 3\sigma, \quad (7.6)$$

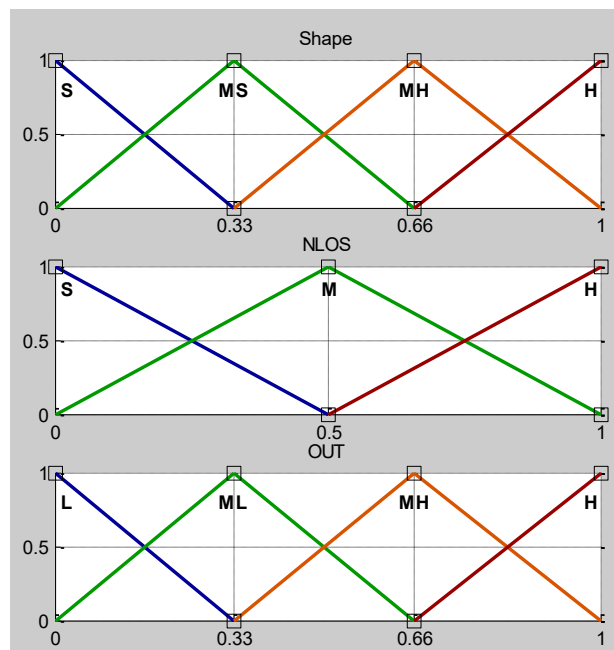
където D_{bs} е средното разстояние между маяците. Нека максималният брой маяци, които са близо до целевия възел, да бъде n . N_s набора от k маяци се формират, чрез които се оценява позицията на целевия възел (биномиален коефициент):

$$N_s = \binom{n}{k} = \frac{n!}{k!(n-k)!} \quad (7.7)$$

За всеки набор от маяци, използвайки размита логика, се получава оценка на важността на наборите. Оценката на позицията на целевия възел се реализира чрез използване на маяците от набора с най-висок резултат. След това проверяваме за пряка видимост между целевия възел и всеки от маяците в избрания набор. Когато се открие NLOS сценарий, се извършва корекция на RSSI стойността за съответния маяк.

7.3.2.1 Оценяване на качеството на Всеки набор от маяци с помощта на размита логика

Предлага се да се използва размита логика за оценяване на качеството на всеки възможен набор от маяци. На Фиг. 7.15 са показани размитите функции на принадлежност, които се използват.



Фиг. 7.15 Размити функции на принадлежност

Използват се две входни лингвистични променливи. Първата входна променлива, наречена *Shape*, отчита важността на формата на разположението на маяците. Висока важност се отдава на многоъгълници с приблизително равни ъгли. Използва се следното уравнение:

$$v_1(Shape) = \frac{1}{k+1} \sum_{i=1}^k \left| 1 - \frac{\alpha_i}{\alpha_{perfect}} \right|, \quad (7.8)$$

където k е броят на избраните маяци; α_i е ъгълът между маяците с индекси $i - 1$, i и $i + 1$; $\alpha_{perfect}$ е ъгълът, при който се получава правилен многоъгълник. Стойността на този параметър може да се получи чрез следната уравнение:

$$\alpha_{perfect} = \frac{\alpha_{sum}}{k}, \quad \alpha_{sum} = \begin{cases} 180^\circ, & k = 3 \\ 360^\circ, & k > 3 \end{cases}. \quad (7.9)$$

За правилен многоъгълник $v_1 = 0$. Максималната стойност на този параметър е 1. Използват се четири стойности за входната променлива *Shape*: S (малък), MS (средно малък), MH (средно голям) и H (голям). Втората входна лингвистична променлива, наречена NLOS, оценява вероятността за сценарий NLOS. Стойността на тази входна променлива се определя като съотношението между изчислената оценка за броя на NLOS сценариите M_{NLOS} и максималния възможен брой NLOS сценарии M :

$$v_2(NLOS) = M_{NLOS}/M. \quad (7.10)$$

Стойността на M_{NLOS} се получава чрез изчисляване дали има пряка видимост между маяка с най-висока стойност за RSSI и другите $M = N - 1$ маяка, участващи в изчисляването на позицията. За тази цел се анализира дали линиите между най-близкия маяк и другите M маяка имат пресечни точки с контура на компонент на сградата (стая, коридор) и статични обекти. Следователно оптималната стойност за v_2 е 0. В най-лошият случай стойността на този параметър е 1. Стойностите на входната променлива NLOS са следните: S (малка), M (средна) и H (висока).

Изходната лингвистична променлива, наречена *OUT*, оценява значимостта за всеки набор от маяци. Стойността *OUT* е 1 при оптимална комбинация от маяци – те образуват правилен многоъгълник и има пряка видимост между всички двойки маяци. Възможните стойности за изходната лингвистична променлива *OUT* са следните: L (ниска), ML (средно ниска), MH (средно висока) и H (висока). За да се получи изходната лингвистична променлива *OUT*, се използват 12 размити правила от следния тип:

if Shape=value И NLOS=value then OUT=value.

Фиг. 7.16 показва използваната матрица с размити правила.

<i>OUT</i>		<i>Shape</i>			
		S	MS	MH	H
<i>NLOS</i>	S	H	MH	ML	L
	M	MH	ML	L	L
	H	L	L	L	L

Фиг. 7.16 Матрица на размити правила

За извеждане на логическите произведения за всяко правило се използва методът Root-Sum-Square. Дефъзификацията на данните до ясно (crisp) изходно значение се реализира чрез комбиниране на резултатите от процеса на извеждане (оператор min) и изчисляване на размития центроида на областта.

7.3.2.2 Откриване на NLOS сценарий

В реална интериорна среда е възможно да има както статични, така и динамични обекти, които представляват препятствия за радиосигналите, излъчвани от маяците. Експериментите показват, че тези препятствия за радиосигналите могат да доведат до намаляване на стойностите на нивото RSSI от маяците с 3dBm до 9dBm, в зависимост от материала и размерите на обектите. Данните от блокираните маяци трябва да бъдат или игнорирани, или коригирани. По-доброто решение е да се коригират сигналите от блокираните маяци, тъй като в даден момент може да няма маяци, които да не са блокирани. Проверката за блокирани маяци от избраните k маяци се извършва в следната последователност. Първо се намира маяка B_i , за който разстоянието d_{ti} до целевия възел е най-малко. Второ, проверява се дали има блокиране на сигналите от другите $k - 1$ маяци. Нека тестваният маяк да е B_j . За очакваното разстояние d_{tj} между целевия възел и маяка B_j може да се напише:

$$d_{ij} - d_{ti} \leq d_{tj} \leq d_{ij} + d_{ti}. \quad (7.11)$$

където d_{ij} е разстоянието между маяците B_i и B_j . Предполага се, че ако условие (7.11) не е изпълнено, тогава има препятствие, което блокира сигнала от маяка B_j . В този случай се извършва корекция на стойността RSSI от маяка B_j , $RSSI_j$. Стойността на $RSSI_j$ намалява пропорционално на разстоянието $d_{tj} - \Delta d$, където $\Delta d = (d_{ij} - d_{ti})$. За коригираната стойност на $RSSI_j$ може да се напише:

$$\widehat{RSSI}_j = RSSI_j - \Delta RSSI + RSSI0_j, \quad (7.12)$$

където $RSSI0_j$ стойността на RSSI от маяка B_j на референтно разстояние 1 m. Стойността на $\Delta RSSI$ се получава чрез следната уравнение:

$$\Delta RSSI = RSSI0_j - 10\gamma \log_{10}(\Delta d), \quad (7.13)$$

където γ е експонентата на загубата по пътя.

7.2.2.3 Алгоритъм за локализация

Нелинейна локализация по метода на най-малките квадрати (NLSQ) се използва за получаване на позицията $\hat{p}_t$ на целевия възел. Нелинейната локализация по метода на най-малките квадрати е техника за оптимизация, която решава нелинейни проблеми за пригаждане на данни от следната двуизмерна форма:

$$\min_{x,y} \|f(x,y)\|_2^2 = \min_{x,y} (f_1(x,y)^2 + \dots + f_n(x,y)^2). \quad (7.14)$$

Когато позицията трябва да бъде изчислена, функцията, която трябва да бъде минимизирана, е както следва:

$$f = (x - x_i)^2 + (y - y_i)^2 - d_i^2. \quad (7.15)$$

където (x, y) е приблизителната позиция на целевия възел, (x_i, y_i) е позицията на i -тия маяк, а d_i е разстоянието между i -тия маяк и приблизителната позиция на целевия възел. Този алгоритъм започва от точката $p(x_0, y_0)$ и връща приблизителната позиция на целевия възел, когато функцията f падне под предварително определения праг. За да може да се оцени позицията на целевата точка, трябва да са налични две или повече оценки на разстоянията между посетителя и маяците (броят на наблюденията трябва да е равен или по-голям от броя на неизвестните параметри).

7.3.3 Експерименти

В тази секция се анализира как предлаганият алгоритъм за избор на маяци влияе върху точността на локализация. За да се валидира предложеният алгоритъм, се изчислява точността на локализация. Точността на локализация ε_t е мярка за коректността на получената оценка за позицията на целевия възел, която се изчислява като разликата между действителната позиция на целевия възел, $p_t = (x_t, y_t)$ и оценката за тази позиция, $\hat{p}_t = (\hat{x}_t, \hat{y}_t)$:

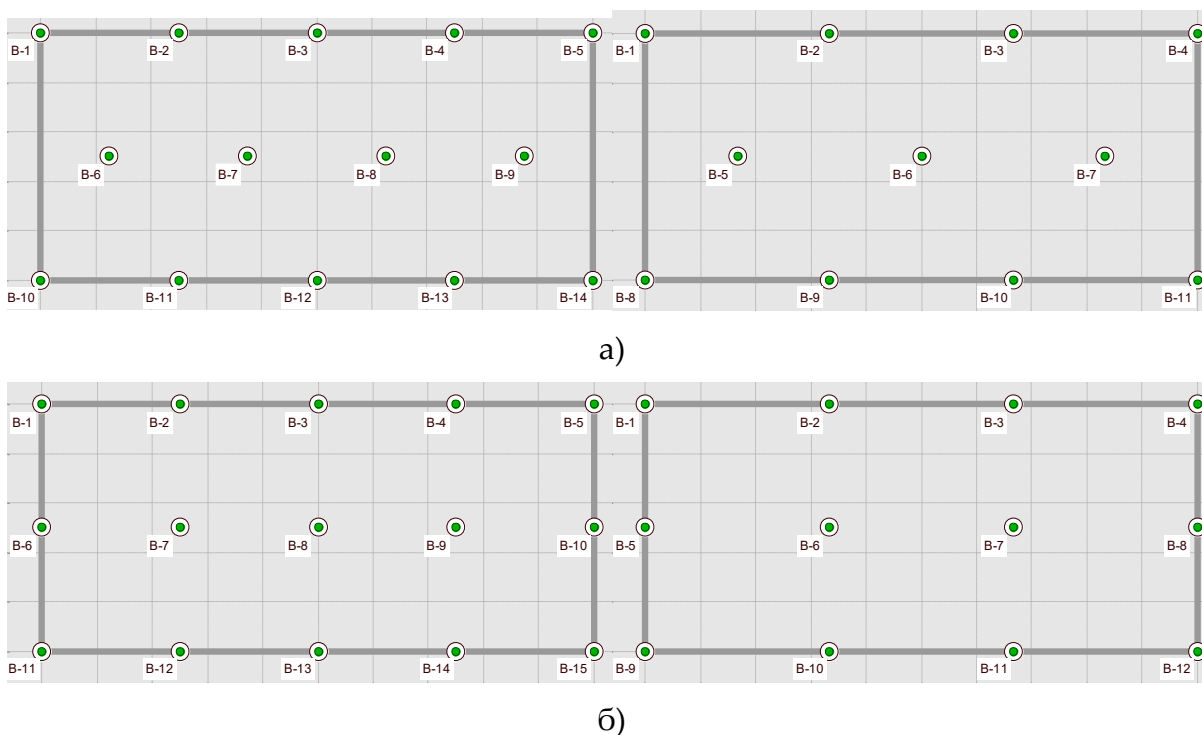
$$\varepsilon_t = \sqrt{(x_t - \hat{x}_t)^2 + (y_t - \hat{y}_t)^2}, \quad (7.16)$$

Тестването и оценката на предложените алгоритми се осъществява с помощта на специално разработен софтуер Matlab, чрез който е възможно:

1. Автоматично разполагане на маяци в сградна среда съгласно предварително определени експертни правила.
2. Автоматичен подбор на най-добрия набор от маяци.
3. Анализ на точността на оценката на позицията на целевия възел.

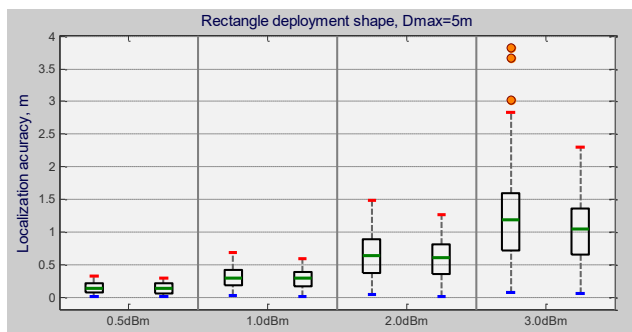
7.3.3.1 LOS сценарий

Ще анализираме точността на локализация в правоъгълна стая с размери 20 м x 10 м и различни форми на разположение на маяците, различни гъстоти на разположение на маяците (0,055 - 0,075 маяка/m²) и различни нива на шум в данните RSSI (от 0,5 dBm до 3 dBm). Фиг. 7.17 показва местоположението на маяците в различните тестови сценарии.

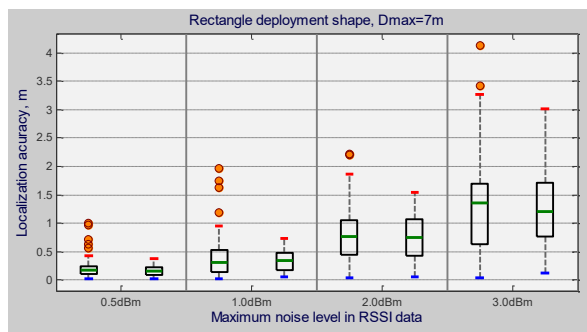


Фиг. 7.17 Сценарии за тестване на разполагането на маяци ($D_{max}=5\text{m}$ за левите изображения, $D_{max}=7\text{m}$ за десните изображения): а) разполагане във формата на триъгълник; б) разполагане във формата на правоъгълник

Точността на локализация беше анализирана с помощта на $k = 3$ маяка за изчисляване на позицията на целевия възел. Получените резултати за правоъгълно разположение на маяците са показани на Фиг. 7.18, а за триъгълно разположение на маяците – на Фиг. 7.19.

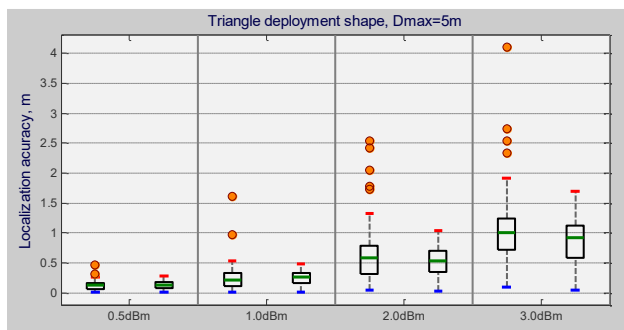


а)

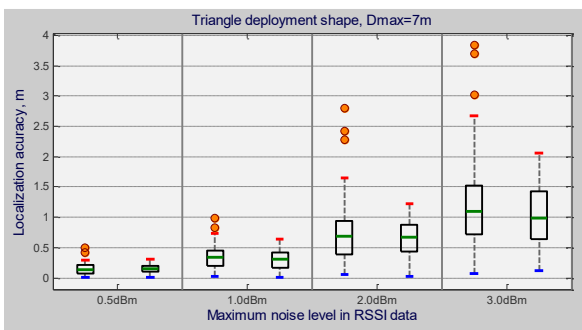


б)

Фиг. 7.18 Точност на локализация за правоъгълна форма на разположение (графика „box-i-whisper“): медианна стойност – зелена линия; горни (75%) и долни (25%) мустаци – сив правоъгълник; горна съседна стойност – червена; долна съседна стойност – синя; изключения – червени кръгчета



а)



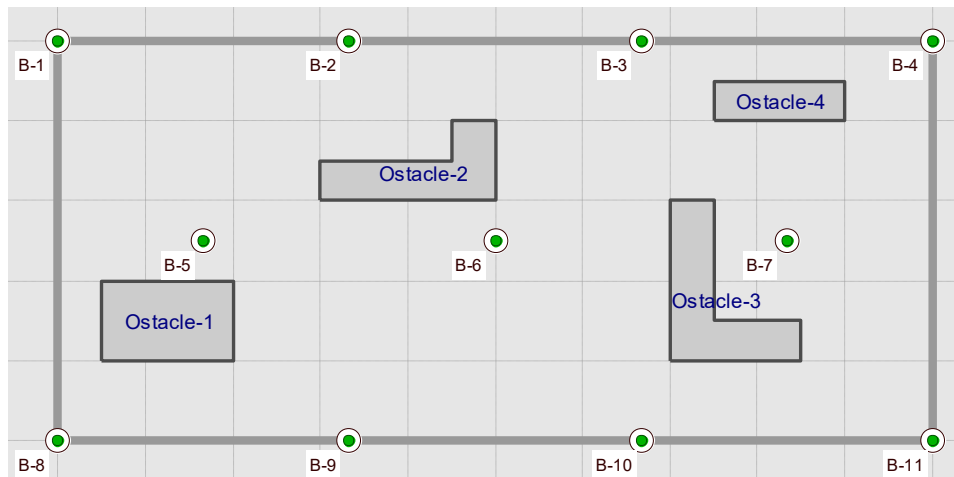
б)

Фиг. 7.19 Точност на локализация за триъгълна форма на разположение (диаграма „box-i-whisper“): а) $D_{max}=5m$ (14 маяка); б) $D_{max}=7m$ (11 маяка)

За всяка стойност на шума, вляво е точността на локализация, когато не се използва предложеният алгоритъм за избор на маяци, а вдясно – когато алгоритъмът се използва. Във всички тестови сценарии предложеният алгоритъм за избор на маяци води до повишаване на точността на локализация. При правоъгълно разположение на маяците и максимално разстояние между тях $D_{max}=5m$, максималната грешка при локализация се намалява с до 37% при ниво на шум 3dBm в данните RSSI. Стойностите за разстоянията между действителните и изчислените позиции на целевия възел, които попадат в интервала (25% - 75%), са с 10% по-малки, ако се използва предложеният алгоритъм. Значително по-забележим е ефектът от използването на предложеният алгоритъм при по-голямо разстояние между маяците, $D_{max}=7m$ (виж фигура 7.17б). Трябва да се отбележи, че предложеният алгоритъм елиминира всички изключения за позицията както при правоъгълно, така и при триъгълно разположение на маяците. При използване на триъгълно разположение (виж Фиг. 7.18), максималната грешка при

локализирането се намалява с до 60% при ниво на шум 3 dBm в данните RSSI. В този случай няма значителна разлика в точността на локализация, ако максималното разстояние между маяците е 5 m или 7 m. Причината за това е, че при $D_{max}=7$ m маяците образуват приблизително равностранни триъгълници, а при $D_{max}=5$ m образуват равностранни триъгълници. В този случай, размитата логика оценява само формата на многоъгълниците, образувани от маяците.

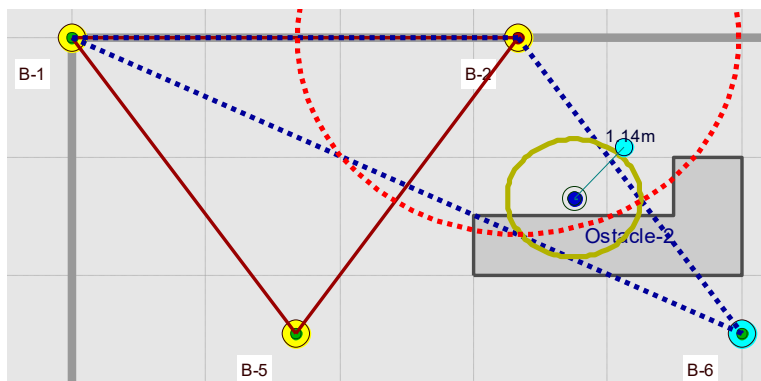
Фиг. 7.20 показва тестовата среда - стая с размери 20 m x 10 m.



Фиг. 7.20 Тестова среда: правоъгълна стая с размери 20 m x 10 m; триъгълна форма на разположение на маяците; $D_{max}=7$ m; 4 препятствия

Използват се 11 маяка, за да се получи приблизителна оценка на позицията на целевия възел. Формата на разположението на маяците е триъгълна. Максималното разстояние между маяците е 7 m. За да симулираме ефекта на затихване, към сигналите от всеки маяк беше добавен случаен шум. В стаята са разположени четири препятствия. Всяко от тези препятствия въвежда случайно затихване от 3dBm до 9dBm в сигналите от маяците, които попадат в радиовълновата сянка.

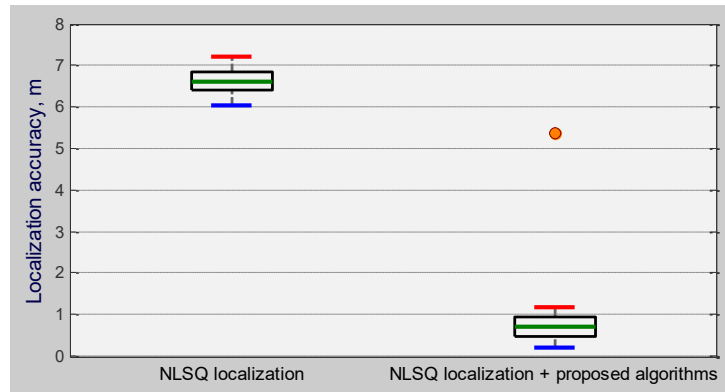
Фиг. 7.21 показва пример за локализация, когато няма пряка видимост между целевия възел и някои от маяците. В този случай целевият възел е в непосредствена близост до препятствие 2. Най-близкият маяк до целевия възел е B_2 (действително разстояние - 2,83 m, изчислено разстояние - 3,12 m). Анализират се сигналите от $n = 4$ маяка с най-високи RSSI стойности $\{B_2, B_1, B_6, B_5\}$.



Фиг. 7.21 NLOS сценарий: $n = 4$, $k = 3$; най-близък възел - B_2 ; избрани маяци $\{B_2, B_1, B_6, B_5\}$, реална позиция на целевия възел – син кръг; приблизително разстояние до най-близкия маяк – червен пунктиран кръг; приблизителна позиция на целевия възел след корекция на RSSI – цианов кръг

За да се получи приблизителна позиция, се използват $k = 3$ маяка. Фигура 7.21 показва два от триъгълниците, които образуват избраните маяци. Размитата логиката оценява триъгълник 2-1-5 по-високо (0,938) в сравнение с триъгълник 2-1-6 (0,469). Причината за това е, че първият триъгълник има почти равни страни ($v_1=0,062$). След това проверяваме дали маяците 1 и 5 са в радио сянка. Като се използва (7.11) се получава, че възможният интервал за разстоянието между целевия възел и маяка B_1 е (3,55 m, 9,78 m). Очакваното разстояние е 9,12 m и следователно този маяк не е в радиосянка. За маяка B_5 този интервал е (2,89 m, 9,13 m), а очакваното разстояние е 12,63 m. Налага се корекция на стойността на RSSI за маяка B_5 с помощта на (7.12) и (7.13) (от -76,87 dBm до -65,62 dBm). След тази корекция грешката в оценката на позицията на целевия възел се намалява до 1,14 m.

Точността на алгоритъма за откриване на NLOS сценарий и алгоритъма за корекция на RSSI данни за маяци, които се намират в радио сянка, се анализира чрез оценка на точността на локализация, когато целевият възел е в съседство с препятствие 2. Направени са 50 измервания. За да се симулира ефектът на затихване, към сигналите от всеки маяк е добавен случаен шум с амплитуда от 1dBm до 2dBm. За всеки маяк, който се намира в радиосянка, е зададено допълнително затихване на сигнала от 3dBm до 9dBm в зависимост от материала на препятствията. Получените резултати са показани на Фиг. 7.22.



Фиг. 7.22 Точност на локализация в сценарий NLOS: ляво – точност на локализация без използване на предложените алгоритми; дясно – точност на локализация с използване на предложените алгоритми

Средната стойност на точността на локализация без използване на предложените алгоритми е 6,6589 m. Тази стойност се намалява до 0,7860 m при използване на предложените алгоритми. В този случай едно от петдесетте измервания не е открило правилно наличието на NLOS сценарий. Най-добрият набор от маяци е $\{B_2, B_6, B_5\}$. Два от маяците (5 и 6) са в радиосянка, но алгоритъмът открива NLOS сценарий само за маяк B_5 .

При тестване на точността на локализация с произволно генерирани позиции на целеви възли в цялата стая, ефективността на алгоритъма за откриване на NLOS сценарий е 90%.

7.3.4 Заключение

Предложен е алгоритъм, който използва размита логика, за да избере най-добрия набор от k маяка от всички налични n маяка с цел точна локализация на посетителите на сгради. Експериментите показват, че този алгоритъм е ефективен както при наличие, така и при отсъствие на препятствия. Когато маяк, който се намира в радиосянка, е в непосредствена близост до целевия възел, използването на (7.11) води до грешно откриване на NLOS сценария.

Предварителните експерименти с три типа смартфони при интервал на сканиране на маяците $T_s = 1s$, показват, че за работа в реално време броят на анализирани маяци n не трябва да надвишава 6. В този случай ($k = 3$, $n = 6$ и $N_s = 20$) времето за оценка на позицията на клиента не надвишава 25% от интервала T_s .

7.4 Алгоритъм за микролокализация в големи обществени сгради

7.4.1 Анализ на проблема

Услугите, които зависят от местоположението на клиентите им, се наричат услуги, базирани на местоположението (LBS). Те са системи, чувствителни към контекста, защото местоположението е вид контекст. Концепцията за интелигентни домове и интелигентни обществени сгради се основава на услуги, чувствителни към контекста, като LBS. Приложимостта на тези услуги е пряко свързана с точността на определяне на местоположението на всеки клиент. Точното изчисляване на тази позиция би позволило създаването на разнообразни и висококачествени LBS.

Използването на решения, базирани на интернет на нещата, за изчисляване на позицията на посетителите в сградите би повишило точността на тези системи. В този случай ще бъде възможно да се разработят услуги, които се адаптират към промените в средата на сградата. На този етап цената на системите за локализация на сгради, базирани на IoT, е твърде висока. Причината за това е необходимостта от изграждане на сензорна мрежа, в която всеки възел трябва да може да обменя информация с всеки от останалите възли. Въз основа на тази информация такава услуга би могла да се адаптира към промените в средата и по този начин да сведе до минимум грешката при изчисляване на местоположението на своите клиенти.

Най-често решенията за позициониране в закрити помещения използват сензорни мрежи с възли, които работят независимо. Такива решения са по-малко точни и трудни за мащабиране, но имат по-ниска цена и не изискват значителни промени в инфраструктурата. Точността на такива услуги може да бъде подобрена чрез използване на по-точни алгоритми за локализация, както и чрез по-точна оценка на необходимата плътност на сензорите (брой маяци на квадратен метър) и тяхното разположение. Холистичният подход към проектирането на IPS е единственият начин да се гарантира разработването на IPS с оптимално съотношение между точност и цена.

Една от най-обещаващите технологии за локализиране на посетители в сгради е BLE (Jeon и колектив, 2018). Тази технология гарантира ниска до средна точност на локализация при ниски разходи за системата и средна до висока сигурност, в зависимост от версията на стандарта Bluetooth. BLE е отворен индустриален стандарт на Bluetooth SIG, който има за цел да подобри производителността на Bluetooth Classic. BLE устройствата работят в 2,4 GHz ISM лента при 2,400-2,483.5MHz. Предимствата са свързани с по-добра сигурност, намалена максимална

консумация, намалена латентност, увеличена скорост на трансфер и увеличено максимално разстояние между BLE устройствата.

Повечето услуги за локализация, базирани на BLE, използват BLE маяци за реализиране на своите сензорни мрежи. Всеки BLE маяк изпраща така наречените рекламни пакети на определен интервал. Данните се изпращат в пакети чрез 3 от 40-те радиоканала. Всеки пакет съдържа информация, която е специфична за всеки тип BLE маяк (iBeacon от Apple, Eddystone от Google или BLE маяци, специфични за производителя). Полето с данни в стандарта iBeacon включва: универсален уникален идентификатор (UUID), TxPower и стойности на полета Major и Minor.

BLE технологията има и някои недостатъци. BLE маяците могат да се използват за локализация на клиента, но оценката на разстоянието е доста неточна и варира в зависимост от околната среда и разстоянието между BLE възлите. Мобилните устройства, като смартфоните, се нуждаят от 2 до 4 секунди, за да открият маяк в обхвата си. Отворената ISM честотна лента, използвана от BLE устройствата, е запълнена с много други безжични протоколи, като Wi-Fi, както и с потенциални смущения от домакински уреди. Wi-Fi обаче има значително по-висока изходна мощност, до 23 dBm, в сравнение с максимално допустимите 10dBm за BLE стандарта. Това означава, че поставянето на маяк много близо до Wi-Fi източник вероятно ще наруши предаването на данните.

В повечето RF-базирани системи за локализация изчисляването на разстоянието между клиента и сензорния възел се основава на стойността на силата на получения сигнал (RSSI). Независимо от честотата, стойността на RSSI намалява приблизително логаритмично с увеличаване на разстоянието между референтния възел и целевия възел. Намаляването на вариациите в параметрите на радиоканалите е трудна задача, която не може да бъде решена в реално време при използване на мобилни устройства. Използването на математически модел на комуникационния радиоканал (модел на загуба в свободното пространство, логаритмично-нормален модел на засенчване и др.) за прогнозиране на затихването на стойностите на RSS не дава задоволителни резултати. Причината за това е, че тези модели не описват точно промените в комуникационния канал. Мощността на предавания сигнал от BLE маяци намалява експоненциално, главно поради разстоянията, но има и други ефекти, като избледняване и мултипътен ефект, които повече или по-малко влияят на мощността на сигнала. Избледняването е времевата вариация на получения сигнал. То зависи от околната среда и

движението на обектите. Мултипътният ефект причинява пристигането на сигнала по различни начини до устройството поради отражения, дифракция и разсейване.

Най-често използваният модел за оценка на RSS стойностите, измерени в точка x от BLE маяк, разположен в точка x_0 е следния:

$$RSS^k = RSS_0^k - 10\gamma^k \log_{10} \left(\frac{d}{d_0} \right) + N(x_0, x), \quad (7.17)$$

където стойността RSS^k е приблизителна оценка за силата на сигнала от k -тия маяк, измерен на разстояние d ; стойността RSS_0^k е оценка за силата на сигнала от k -тия маяк, измерен на разстояние d_0 (обикновено 1 m); γ^k е експонентата на загубата по пътя, а $N(x_0, x)$ представлява случайните колебания в RSS, дължащи се на различни видове затихване. Шумът $N(x_0, x)$ е зависимо от местоположението затихване на сигнала между маяка и позицията на целевия възел. Обикновено за шума се приема логаритмично-нормално разпределение:

$$N(x_0, x) \sim N(0, \sigma_\gamma^2), \quad (7.18)$$

където σ_γ^2 е вариацията в сигнала. За да локализираме посетителя, трябва да преобразуваме измерването на RSS (RSSI) в приблизително разстояние от маяка:

$$d = 10^{\frac{RSSI_0^k - RSSI^k - N(0, \sigma_\gamma^2)}{10\gamma^k}}. \quad (7.19)$$

Експонентата на загуба по пътя γ^k може да варира в зависимост от честотата на сигнала и средата на сградата, от 1,5 за сценарий с пряка видимост (LOS) до 3,5 за сценарий без пряка видимост (NLOS). В реална среда трябва да се моделира радиоканала между целевия възел и всеки референтен възел. На практика най-често се използва един модел за цялото помещение или коридор. Експериментите показват, че RSSI сигналите в динамична среда имат мултимодално, а не унимодално разпределение, а комуникационният канал в повечето случаи е нелинейна система. Следователно трябва да се използват нелинейни филтри, например разширени филтри на Калман. Когато тези алгоритми се прилагат за множество радиоканали, те изискват прекалено голяма изчислителна мощност, за да бъдат приложени в реално време.

Алгоритмите за локализация, базирани на BLE, могат да бъдат разделени на две основни категории ([Paterna и колектив, 2017](#)):

- алгоритми базирани на разстояние;
- алгоритми базирани на радиочестотен отпечатък (fingerprinting).

Алгоритмите, базирани на разстояние, оценяват позицията на целевия възел, като изчисляват (пряко или непряко) разстоянието до три или повече референтни възли. Тези алгоритми (Luo и колектив, 2015), могат да бъдат разделени на две категории:

- алгоритми, базирани на обхват;
- алгоритми, независими от обхват.

Алгоритмите, базирани на обхват (Mekelleche и Haffaf, 2017), трябва да измерват разстоянията между референтните възли и целевия възел. След това разстоянията се използват за изчисляване на позицията на целевия възел. Алгоритмите без обхват (Shahra и колектив, 2017) могат да получат косвено приблизителна оценка за разстоянията. Алгоритмите, базирани на обхват, имат по-добра точност на локализация в сравнение с алгоритмите без обхват.

Алгоритмите, базирани на радио-отпечатъци, създават така наречената референтна карта на радио-отпечатъците, използвайки вектор от RSSI измервания в известни позиции. Получаването на радио отпечатък е скъпа техника за локализация, която изисква повече усилия и време за внедряване. Всяка значителна промяна в инфраструктурата изисква всички измервания да бъдат повторени, за да се актуализира референтната карта на отпечатъците. Тази техника не е подходяща за използване в големи обществени сгради и динамични среди. Системите, базирани на отпечатъци, анализирани в (Paterna и колектив, 2017), имат средна точност на локализация от 2,5 m (от 0,8 m до 4,12 m). Подобна точност се получава с системи, базирани на разстояние – 2,1 m (от 0,75 m до 3 m). Причината за разликите между минималните и максималните стойности на точността на локализация са различните параметри, при които са проведени експериментите (размери на помещението и плътност на разположение). Например, Faragher и Harle (2015) описват локализация, базирана на BLE радио-отпечатъци, използвайки 19 маяка, разпределени на площ от 600 m². Те постигат точност на проследяване < 2,6 m в 95% от случаите, използвайки гъсто разпределение на маяците (1 маяк на 30 m²) и < 4,8 m, използвайки по-ниска плътност на разпределение (1 маяк на 100 m²). Grzechca и колектив (2016) използват локализация чрез радио-отпечатъци в зала с размери 7 m x 5 m, където максималното разстояние между целевите и референтните възли е по-малко от 9 m. Използвайки алгоритъм за трилатерация, средната грешка е 2,8 m.

Подобряването на точността на системите за локализация е възможно чрез използване на различни техники за филтриране. Park и S. Kang (2016) използват

Калман филтър (KF) за намаляване на грешката при локализация. Тестовата среда се състои от шест BLE маяка, разположени в хола (8,5 m × 13 m) на реална къща. Средната точност на локализация е намалена до 1,77 m (41%) в сравнение с точността на локализация чрез трилатерация. [Zafari и Papapanagiotou \(2015\)](#) използват метода на particle филтри, за да подобрят точността на локализация. Те отчитат грешки в проследяването от едва 0,27 m в по-малко пространство и 0,97 m в по-голямо пространство.

7.4.2 Описание на предлагания алгоритъм

7.4.2.1 Предварителна обработка на RSSI данните

Колебанията в RSSI данните, дължащи се на наличието на статични (мебели, архитектурни обекти) и динамични (посетители) препятствия, са причината за ниската точност на локализация. Точното моделиране на такива системи е трудно да се реализира в реално време с помощта на мобилни устройства. Едно възможно компромисно решение е да се филтрират RSSI данните, така че за кратки интервали комуникационният канал да бъде квазилинейна система. Най-често използваните техники за филтриране в реално време на наблюдавания набор от RSSI стойности са филтърът Moving Average (MA) и филтърът Калман. Филтърът на Калман е оптимален метод за оценка. Той може да бъде описан като двуетапен процес: етап на прогнозиране, който използва научената динамика на системата, за да предвиди следващата стъпка на системата, и етап на корекция (актуализация на измерването), който взема ново наблюдавано измерване и го използва, за да коригира динамиката и състоянието на системата. За IPS нивото на шума при измерването зависи от околната среда и движението на посетителите. В този случай филтърът на Калман не е оптимален и може да доведе до ненадеждни резултати. Едно от решенията на проблема е да се използва адаптивен филтър на Калман (AKF).

Предлага се комбинация от MA филтър и адаптивен филтър на Калман като нов алгоритъм за динамично намаляване на шума в сигналите от маяците. Алгоритъмът използва обработка на сигнала на ниво времеви прозорец. За всеки времеви прозорец (N брой RSSI стойности) се изчисляват средната (u_{avg}), максималната (u_{max}) и минималната (u_{min}) RSSI стойности. MA филтърът се използва за получаване на средната стойност на входния сигнал u :

$$u_{avg} = \frac{1}{N} \sum_{i=1}^N u_i, \quad (7.20)$$

Параметърът Δu се използва за оценка на колебанията на *RSSI* в рамките на времеви прозорец:

$$\Delta u = u_{max} - u_{min}. \quad (7.21)$$

Предложеният алгоритъм АКФ се използва за филтриране на стойностите u_{avg} , u_{max} и u_{min} . В този случай АКФ е едномерен. Ако данните *RSSI* се анализират на интервали от една секунда, матриците за преход във времето са една секунда. В този случай шума от процеса е същият като шума от измерването. По този начин уравненията АКФ се опростяват до следната форма:

$$\hat{x}_k^- = \hat{x}_{k-1}$$

$$P_k^- = P_{k-1} + q \quad (7.22a)$$

$$K_k = P_k^- / (P_k^- + r)$$

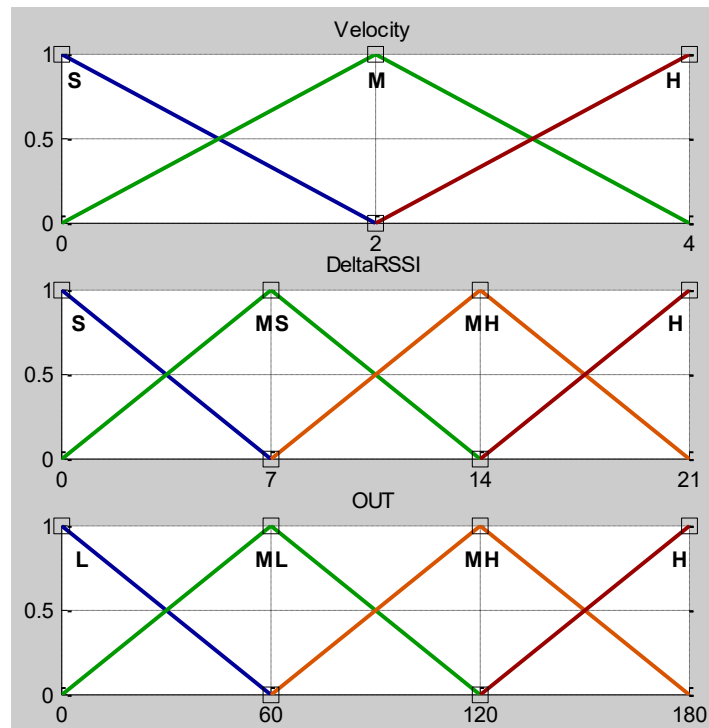
$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{x}_k^-) \quad (7.22b)$$

$$P_k = (1 - K_k)P_k^-$$

където k е дискретно време, $\hat{x}_k^-$ е априорна оценка на състоянието, $\hat{x}_k$ е апостериорна оценка на състоянието, P_k^- е априорна оценка за ковариацията на грешката, P_k е апостериорна оценка за ковариацията на грешката, K_k е матрицата на Калман, а z_k е вектора с наблюдаваните данни (*RSSI*). Приема се, че вариацията на шума в процеса q е постоянна ($q = 0,2$). Филтрираният сигнал $\hat{x}_k$ се изчислява въз основа на оценка на разликата между измерената стойност z_k и априорната оценка на състоянието $\hat{x}_k^-$.

За да се получи оценка за степента на филтриране на данните *RSSI*, се предлага да се използва размита логика за оценка на стойността на вариацията на шума при измерването r (6a). Използват се две входни лингвистични променливи (виж Фиг. 7.23).

Първата входна лингвистична променлива *DeltaRSSI* оценява грешката в данните *RSSI*. Използваме параметъра Δu , получен с (7.21), като стойност за тази входна променлива. Използват се четири стойности за *DeltaRSSI*: S (малка), MS (средно малка), МН (средно висока) и Н (висока). Предполага се, че колкото по-ниска е стойността на *DeltaRSSI*, толкова по-малко трябва да се филтрират данните *RSSI*. Когато посетителите влизат в сградата, използването на неадаптивни филтри на Калман често води до прекомерно филтриране. В този случай времето за конвергенция на филтъра може да достигне няколко секунди.



Фиг. 7.23 Размити функции на принадлежност

За да се предотврати това, скоростта на движение на посетителя се използва като втора входна променлива (*Velocity*). Стойностите на променливата *Velocity* са следните: S (малка), M (средна) и H (висока). Колкото по-бързо се движи посетителят, толкова по-малко се филтрират RSSI данните. Възможните стойности за изходната лингвистична променлива *OUT* са следните: L (ниска), ML (средно ниска), MH (средно висока) и H (висока). Колкото по-висока е стойността на *OUT*, толкова по-силно се филтрират входните данни. За да се получи изходната лингвистична променлива *OUT*, се използват 12 размити правила от следния тип:

if Velocity=value И DeltaRSSI=value then OUT=value.

На Фиг. 7.24 е показана използваната матрица с размити правила.

		<i>DeltaRSSI</i>			
		S	MS	MH	H
<i>Velocity</i>	S	L	ML	MH	H
	M	L	L	ML	MH
	H	L	L	L	ML

Фиг. 7.24 Матрица с размити правила

За да се изведат логически продукти за всяко правило, се използва методът Root-Sum-Square. Дефъзифицирането на данните в измерим резултат се осъществява чрез комбиниране на резултатите от процеса на извеждане и изчисляване на размития център на площта.

7.4.2.2 Алгоритми за локализация

Предлагат се два алгоритъма за точна локализация в сгради, които ще бъдат сравнени с добре известни алгоритми като трилатерация и нелинейни най-малки квадрати (NLLSQ).

7.4.2.2.1 Алгоритъм за локализация 1 (ALG-1)

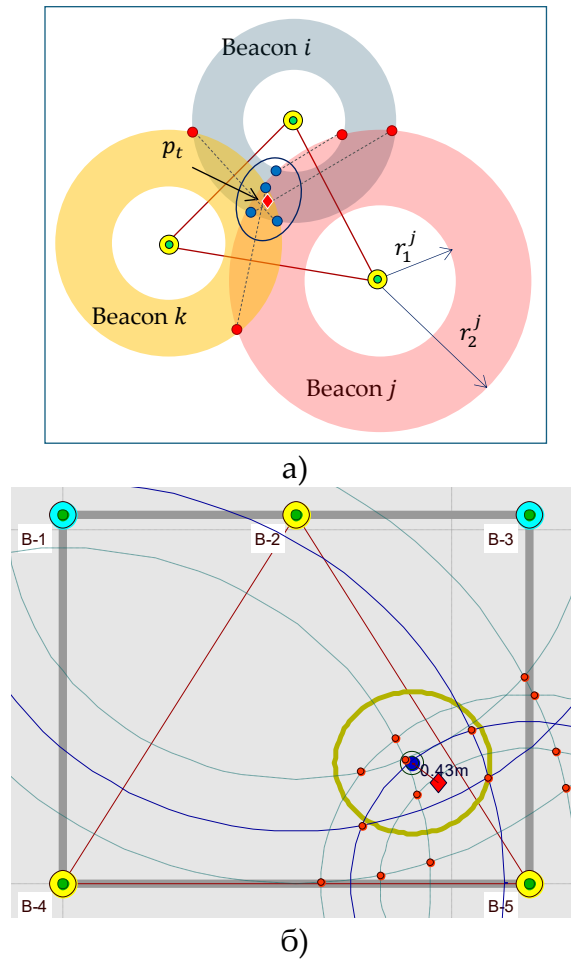
Предложеният алгоритъм ALG-1 принадлежи към алгоритмите за локализация на базата на обхват, но прилага някои идеи, използвани в алгоритъма за локализация без обхват, наречен Ring-Overlapping Circle RSSI (ROCRSSI) (Luo и колектив, 2011). Алгоритъмът ROCRSSI изисква маяците да могат не само да предават данни, но и да четат данни от други маяци. Това го прави неизползваем, ако се използват стандартни BLE маяци. Да приемем, че $RSSI_i$ е сигналът от i -тия маяк, а $RSSI_t$ е сигналът от целевия възел. Ако $RSSI_i < RSSI_t$, тогава i -тият маяк принадлежи към група 1. В противен случай той принадлежи към група 2. Максималната стойност на RSSI в група 1 се използва за изчисляване на радиус r_1 , а минималната стойност на RSSI в група 2 дава радиус r_2 . Приема се, че целевият възел се намира между два кръга (пръстен) с вътрешен радиус r_1 , външен радиус r_2 и център, съвпадащ с позицията на маяка. Повтаря се за всички маяци и се създава набор от припокриващи се двойки пръстени. Целевият възел се намира в центъра на тежестта на тези припокриващи се двойки пръстени.

Целта е да се получи приблизителна оценка за стойностите на радиусите r_1 и r_2 , без да е възможно всеки маяк да получава RSSI стойността от другите маяци. За тази цел колебанията на RSSI от всеки маяк се анализират и филтрират с помощта на предложения AKF. Затихването на RSSI сигнала е по-значително, когато между посетителя и тези възли има обекти или хора. В този сценарий целевият възел ще получи по-ниска стойност за мощността от BLE възела. Затихването на мощността може да се моделира, като се приеме, че позицията на целевия възел е в рамките на пръстен, чийто външен радиус е пропорционален на това затихване. От координатите на припокриващите се части на пръстените, получени за два или повече маяка, може да се получи приблизителна оценка на позицията на целевия възел. Използваме претеглен център на тежестта, за да изчислим окончателната приблизителна оценка за позицията. За да се получат вътрешният и външният

радиус на всеки пръстен, максималните и минималните RSSI стойности за всеки времеви прозорец се филтрират, като се използва комбинация от МА филтъра и АКФ. Въз основа на тези стойности ($\hat{u}_{max}$, $\hat{u}_{min}$) се изчисляват вътрешният и външният радиус на пръстена. Стойностите на r_1 и r_2 се получават, като се използва моделът за загуба на пътя (7.17):

$$r_1 = 10^{\frac{RSSI_0 - \hat{u}_{max}}{10\gamma}}, r_2 = 10^{\frac{RSSI_0 - \hat{u}_{min}}{10\gamma}}. \quad (7.23)$$

Изчисляването на приблизителната позиция на целевия възел се осъществява в следната последователност (виж Фиг. 7.25).



Фиг. 7.25 Локализация с помощта на алгоритъм ALG-1: (а) Как работи алгоритъмът: валидни пресечни точки (сини точки); невалидни пресечни точки (червени точки); приблизително местоположение на целевия възел p_t (червен ромб); (б) Алгоритъм в действие: $\Delta RSSI = \pm 6 \text{ dBm}$; целева точка (син кръг); кръг с $r = 1 \text{ m}$ (тъмножълт)

Първо, избираме маяците, участващи в получаването на позицията на целевия възел. За всеки маяк радиусите r_1 и r_2 се изчисляват с помощта на (7.23). Второ, получават се координатите на всички пресечни точки между пръстените. Ако броят на пресечните точки между всяка възможна двойка кръгове е два, тази, за която общото разстояние от точката до позицията на избраните маяци е по-малко, е печелившата точка (сини точки). Другата точка се счита за невалидна и се премахва (червени точки). Накрая се приема, че позицията на целевия възел p_t (червен ромб) съвпада с координатите на претегления център на площта, която образуват сините точки. По-високи тегла се задават за пресечни точки, които имат по-малко общо разстояние до избраните най-близки маяци:

$$w_i = 1/\sum_k \|p_i^{int}, p_k^{beacon}\|, \quad (7.24)$$

където p_i^{int} е позицията на точка на пресичане с индекс i ; p_k^{beacon} е позицията на маяк с индекс k ; $\|\cdot\|$ е разстоянието между две точки в декартова координатна система.

Оценката за позицията на целевия възел p_t се получава чрез следното уравнение:

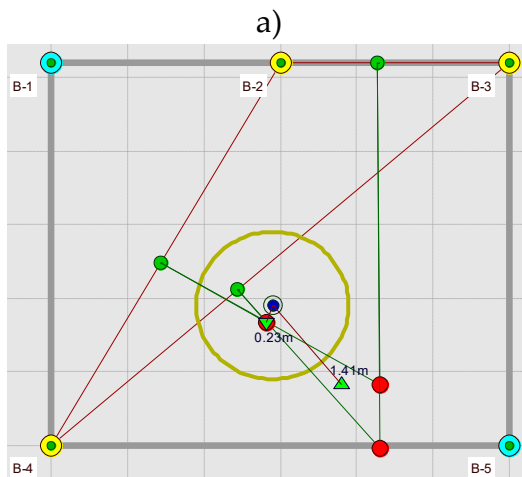
$$p_t = \sum_i (w_i p_i^{int}) / \sum_i w_i. \quad (7.25)$$

Фиг. 7.25б показва как работи ALG-1, когато са избрани три маяка за оценка на позицията на целевия възел. Добавен е случаен шум в RSSI данните с максимална стойност ± 6 dBm. Синият кръг е реалното местоположение на целевия възел, а червеният ромб – приблизителното местоположение. Точността на алгоритъма е най-висока, ако местоположението на целевия възел съвпада с центъра на тежестта на областта, която формират маяците (триъгълник $B_2B_4B_{25}$).

7.4.2.2.2 Алгоритъм за локализация 2 (ALG-2)

Основната причина за грешката при изчисляване на позицията на целевия възел в алгоритмите Trilateration и NLLSQ е неточното изчисляване на разстоянието между позицията на целевия възел и маяците. При алгоритъм ALG-2 позицията на целевия възел се изчислява въз основа на разликата между стойностите RSSI за всяка двойка маяци, участващи в процеса на определяне на позицията. Следователно всяка стойност на разликата RSSI е пропорционална на съотношението на разстоянията между целевия възел и всяка двойка маяци. Ако всички маяци са конфигурирани да предават с еднаква мощност, тогава този параметър ($TxPower$) няма да повлияе на точността на изчислението на позицията. В предложения алгоритъм, за да се опростят изчисленията, се приема, че позицията на целевия възел се намира в центъра на многоъгълник. Нека първо

Diagram illustrating the geometry of a triangle formed by three beacons (Beacon i, Beacon k, Beacon j) and a target point p_t . The diagram shows distances p_{int}^i , p_{int}^j , p_{int}^k from the beacons to the target, and distances d_i , d_j , d_k from the beacons to the target. The target point p_t is marked with a blue dot. The diagram also shows the perpendicular distances $l_{\perp}^{ik}$, $l_{\perp}^{ij}$, $l_{\perp}^{kj}$ from the target to the sides of the triangle.



В този случай позицията на целевия възел p_{ij} се получава чрез следните уравнения:

$$x = x_i + \frac{d_i}{d_i + d_j} (x_j - x_i) = x_i + \frac{x_j - x_i}{1 + cof'} \quad (7.26a)$$

$$y = y_i + \frac{d_i}{d_i + d_j} (y_j - y_i) = y_i + \frac{y_j - y_i}{1 + cof}. \quad (7.266)$$

188

$$cof = 10^{\frac{RSSI_i - RSSI_j}{10\gamma}} = 10^{\frac{RSSD_{ij}}{10\gamma}}. \quad (7.27)$$

Нека разгледаме случая, в който позицията на целевия възел е точка p_t , която попада в триъгълника, образуван от три маяка с индекси i , j и k (виж Фиг. 7.26а). Нека приемем, че проекциите на тази точка върху линиите между всяка двойка маяци са точките p_{ij} , p_{ik} , и p_{kj} . Ако в данните RSSI няма шум, тогава целевата позиция се получава като пресечна точка между линиите $l_{\perp}^{ij}$, $l_{\perp}^{ik}$, и $l_{\perp}^{kj}$. Линията $l_{\perp}^{ij}$ е перпендикулярна на линията l^{ij} , която свързва маяците i и j и минава през точката p_{ij} . За уравнението на линията $l_{\perp}^{ij}$ можем да напишем:

$$y = y_{p_{ij}} + m_{\perp}^{ij}(x - x_{p_{ij}}). \quad (7.28)$$

За да бъде права линия $l_{\perp}^{ij}$ перпендикулярна на права линия l^{ij} , трябва да е изпълнено следното условие:

$$m_{\perp}^{ij} = -\frac{1}{m^{ij}} = -\frac{x_j - x_{p_{ij}}}{y_j - y_{p_{ij}}}. \quad (7.29)$$

където m^{ij} е наклонът на правата l^{ij} . Уравненията за другите перпендикулярни прави $l_{\perp}^{ik}$ и $l_{\perp}^{kj}$ се намират по подобен начин. Остава да се намерят пресечните точки между всяка двойка прави линии $(l_{\perp}^{ij}, l_{\perp}^{ik})$, $(l_{\perp}^{ij}, l_{\perp}^{kj})$ и $(l_{\perp}^{ik}, l_{\perp}^{kj})$. Условието $l_{\perp}^{ij} \cap l_{\perp}^{ik} \neq \emptyset$ е изпълнено, ако $l_{\perp}^{ij} = l_{\perp}^{ik}$. В този случай за координатите на точката на пресичане p_{int}^i между линиите $l_{\perp}^{ij}$ и $l_{\perp}^{ik}$ се получават следните уравнения:

$$x_{int}^i = \frac{y_{p_{ik}} - y_{p_{ij}} + m_{\perp}^{ij}x_{p_{ij}} - m_{\perp}^{ik}x_{p_{ik}}}{m_{\perp}^{ij} - m_{\perp}^{ik}}, \quad (7.30a)$$

$$y_{int}^i = y_{p_{ij}} + m_{\perp}^{ij}(x - x_{p_{ij}}) \quad (7.30б)$$

Координатите на останалите пресечни точки се получават по подобен начин. Очакваната позиция на целевия възел съвпада с центъра на тежестта на областта, получена от всичките три пресечни точки:

$$p_t = \frac{p_{int}^i + p_{int}^j + p_{int}^k}{3}. \quad (7.31)$$

Ако n е броят на маяците, използвани за изчисляване на позицията на целевата точка, алгоритъмът може да се използва, ако $n \geq 2$.

Ще бъдат анализирани два варианта на предложения алгоритъм. В първия вариант (ALG 2A) позицията на целевия възел съвпада с позицията на точката, изчислена с помощта на (7.31). Във втория вариант на алгоритъма (ALG 2B) позицията на целевата точка съвпада с позицията на едно от пресечните точки, за което е изчислена минималната сума от разстоянията между пресечната точка и точките по страните на триъгълника, участващ в изчисляването на неговите координати:

$$p_t = \min_{\forall m} (\sum_{\forall q \setminus m} \|p_{int}^m, p_{mq}\|). \quad (7.32)$$

Точността на тези алгоритми зависи от геометричното разположение на маяците, участващи в изчисляването на позицията. Най-добрата точност на локализация се получава, ако целевият възел се намира в центъра на триъгълника, образуван от маяците. Когато целевият възел се намира в триъгълника, образуван от маяците, точността на алгоритмите ALG-2A и ALG-2B е приблизително еднаква. Най-лошата точност на локализация се получава, когато позицията на целевата точка е извън триъгълника. В този случай, както е показано на Фиг. 7.26б, алгоритъмът ALG 2B има по-добра точност от алгоритъма ALG 2A.

7.4.3 Експерименти

В тази секция са описани няколко сценария за използване на предлаганите алгоритми за локализация, за да се анализира точността им. Анализът на параметрите, от които зависи точността на локализация, се осъществява чрез тестова платформа, която включва:

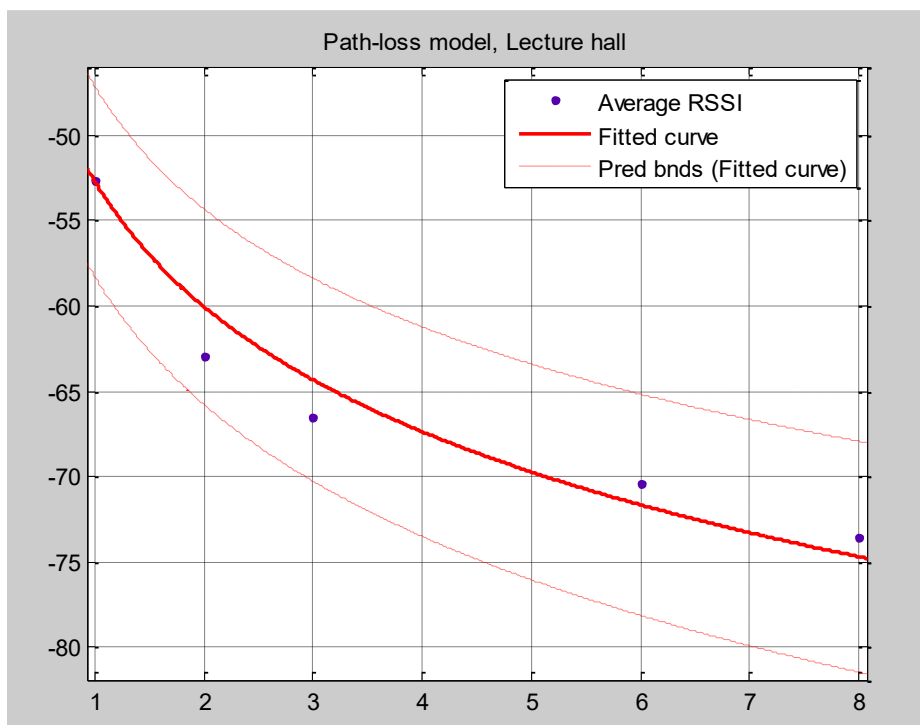
- Мобилно приложение, което записва в CSV файл информация от маяци с най-високи RSSI стойности и данни от вграден акселерометър.
- Texas Instruments (TI) CC2540 USB BLE дъннгъл и софтуер за пакетен снифър SmartRF BLE.
- Java приложение за предварителна обработка и извличане на сурови BLE пакетни данни.
- Програмно осигуряване на Matlab за автоматично разгръщане на маяци и сравнение на алгоритми за локализация.

Мобилните приложения за конфигуриране на BLE маяци най-често са специфични за всеки производител (Gimbal, Kontakt.io, iBKS и др.), за да се гарантира по-високо ниво на сигурност. Чрез тях маяците могат да бъдат конфигурирани чрез задаване на конкретна стойност за всеки параметър, например интервал на рекламиране, Major, Minor, RSSI₀ и TxPower. С помощта на TI CC2540 е възможно да се сканира всеки BLE канал за данни от BLE устройства. Данните от CC2540 се анализират с помощта на софтуера TI SmartRF BLE packet sniffer . Той позволява задаване на идентификатора на BLE канала, който трябва да бъде сканиран, както и начинът на визуализиране на съдържанието на пакетите с данни. Приложението позволява излъчване на BLE пакети към други приложения. За целта се използва протокол UDP. Разработено е Java приложение, което стартира UDP сървър за улавяне на UDP пакетите, предавани от SmartRF packet sniffer . Това приложение позволява записване в реално време на необработени и филтрирани данни като CSV файлове.

Впоследствие тези данни се обработват от Matlab приложение, което позволява статистически анализ на данните от CSV файлове.

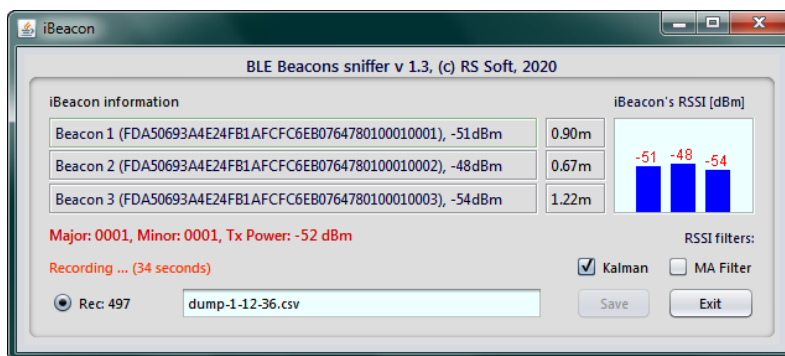
7.4.3.1 Тестов сценарий 1

Целта на първия експеримент е да се анализира приложимостта на алгоритъма за предварителна обработка на данните от маяците. Експериментът е проведен в лекционна зала за 50 студенти. Размерите на залата са $11\text{ m} \times 6\text{ m} \times 3,3\text{ m}$. Използвани шест BLE маяка, четири от които бяха разположени в ъглите на залата, а два – в средата на по-дългите страни на залата. Маяците са разположени на височина 1,5 m от пода на стаята. За всички маяци са зададени едни и същи конфигурационни параметри: интервал на рекламиране (200 ms), RSSI_0 (52 dBm) и TxPower (4 dBm). За да се получат параметрите на модела, описващ затихването на радиосигналите, се използва Matlab cftool (виж Фиг. 7.27).



Фиг. 7.27 Модел на загуба на сигнала в лекционна зала, $\gamma = 2,533$ (95% доверителни граници)

Софтуерът SmartRF BLE packet sniffer се използва за избор на конкретен BLE канал (37, 38 или 39) за четене на пакетите на маяка. Разработеното Java приложение стартира UDP сървър и улавя пакетите, предавани от TI SmartRF packet sniffer. Това приложение има графичен потребителски интерфейс (виж Фиг. 7.28).



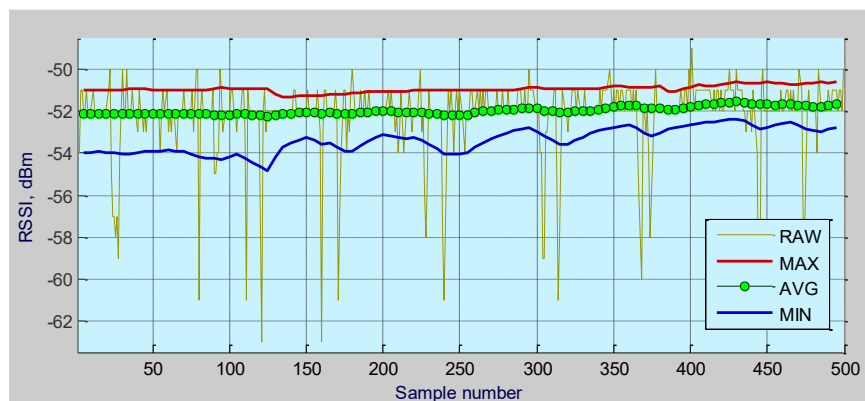
Фиг. 7.28 Java приложение за предварителна обработка и извличане на данни от маяци

Това приложение позволява:

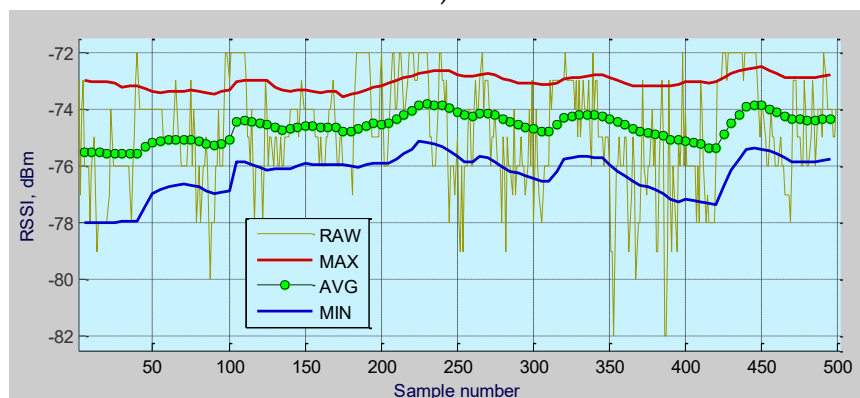
- Инициализиране на приложението, използвайки параметри от конфигурационен файл (минимална стойност на RSSI, под която данните от съответните маяци не се обработват; UUID на маяците, които ще се анализират; максимален брой маяци, които ще се анализират; параметри на филтъра и др.
- Визуализация в реално време на RSSI данни от трите маяка, най-близки до целевия възел;
- Избор на тип филтър: МА филтър, Калман филтър или комбинация от двата филтъра;
- Стартиране/спиране на записването на данни в CSV файл – приложението показва колко проби са записани и колко време е изминало от началото на записването.

Анализирани са сто записи, получени от различни маяци и канали, за конкретно разстояние между целевите и референтните възли, както и за шума в BLE каналите. За всяко от петте разстояния (1 m, 2 m, 3 m, 6 m и 8 m) са получени 20 записи. Всеки запис съдържа около 500 проби (100 секунди). За визуализиране на получените резултати се използва програмно осигуряване, написано на Matlab. Дължината на времевия прозорец N е зададена на 10 проби. АКФ връща резултат всяка секунда. За да бъде това възможно, се използва 50% припокриване на времевите прозорци. За да се анализира предложената АФК, са проведени два вида експерименти в реална среда. По време на всички тестове има движение на хора, които пресичат линията, която се образува при свързване на позициите на референтния (маяк) и целевия възел (посетител). Целта е да се симулира редуването на сценариите LOS и NLOS. В първия експеримент се анализират данните RSSI, получени от целеви възел,

който е неподвижен и се намира на определено разстояние (d_{ref}) от референтния възел. Фиг. 7.29 показва резултатите, получени след анализ на два записа на разстояние между възлите от 1 m и 8 m.



a)



b)

Фиг. 7.29 Предварителна обработка на данните от маяк – необработени данни (зелено), филтрирани данни (МА+АКФ): средни стойности (зелени кръгчета), максимална стойности (червено) и минимални стойности (синьо): а) 1 m разстояние между референтния и целевия възел; б) 8 m разстояние между възлите

Големи намаления в стойностите на RSSI се наблюдават, когато се загуби пряката видимост между двата маяка.

Обобщение на резултатите за всички разстояния е показано в Табл. 7.2. За всеки алгоритъм за филтриране и за всяко референтно разстояние между маяците са изчислени минималното, максималното и средното разстояние, както и стандартното отклонение. Ясно се вижда, че в динамична среда на сграда колебанията в данните за RSSI водят до големи грешки в изчисленото разстояние между възлите – от 0,24 - 1,72 m при $d_{ref}=1$ m до 3,3 - 7,3 m при $d_{ref}=8$ m.

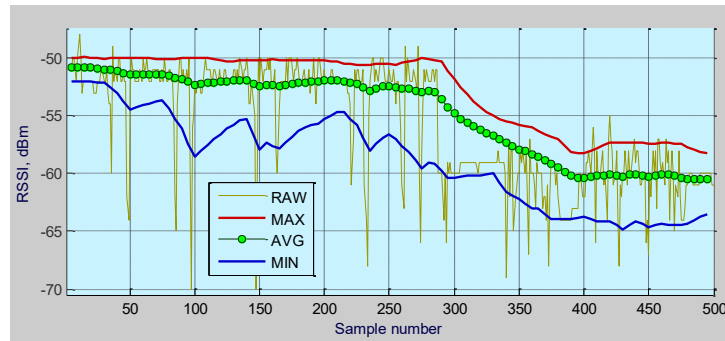
Табл. 7.2 Обобщение на резултатите, получени в динамична среда с използване на статичен целеви възел

Filtering algorithm Error, m		Distance, m				
		1	2	3	6	8
Raw	min	0.24	0.91	1.34	2.05	3.31
	max	1.72	4.16	8.22	8.10	7.29
	avg	0.04	0.13	0.23	0.28	0.35
	std	0.24	0.39	0.77	0.98	1.41
MA	min	0.10	0.22	0.70	0.89	1.84
	max	0.33	0.44	0.85	1.42	2.39
	avg	0.03	0.11	0.28	0.24	0.18
	std	0.08	0.12	0.38	0.47	0.95
MA+AKF	min	0.02	0.01	0.12	0.46	0.73
	max	0.04	0.20	0.54	0.78	1.53
	avg	0.003	0.10	0.18	0.23	0.25
	std	0.016	0.05	0.15	0.28	0.72

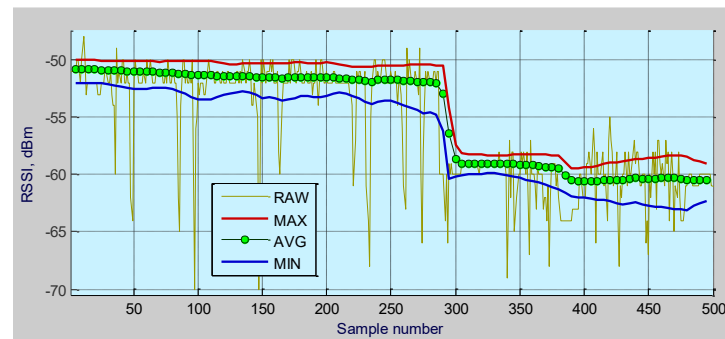
При използване на МА филтър грешката в изчисленото разстояние намалява, но за $d_{ref} > 3m$ тя е над 1m (0,9 - 1,42 m за $d_{ref} = 6m$ и 1,84 - 2,39 m за $d_{ref} = 8m$). При използване на комбинация от МА филтър и АКФ, максималната грешка за всички тествани референтни разстояния е по-малка от 1 m.

Във втория експеримент целевият възел (посетител) се движи в лекционната зала от едно референтно разстояние между възлите до друго референтно разстояние. Посетителят използва мобилно приложение, чрез което данните от шестте маяка и скоростта му на движение се записват в CSV файл. С помощта на приложение за Matlab се анализира колко време отнема реакцията на неадаптивния KF и предложението АКФ да се доближи до действителната стойност на RSSI. По време на всички записи в залата се движат хора.

На фиг. 7.30 е показан резултата след филтриране на стойностите на RSSI. Те са получени чрез преместване на целевия възел от референтния възел на начално разстояние 1 m (52 dBm) до крайно разстояние 2 m (-60 dBm).



а)



б)

Фиг. 7.30 Предварителна обработка на данните от маяка в динамична среда и движещ се целеви възел: а) използва се неадаптивен KF; б) използва се предложеният AKF

При използване на неадаптивен филтър на Калман (Фиг. 7.30а) времето за сближаване на реакцията на филтъра до действителната стойност за RSSI е над 100 извадки (20 секунди). При използване на предложения AFK (Фиг. 7.30б) това време се намалява до 15 извадки (3 секунди). Експериментите показват, че неадаптивният филтър на Калман не дава задоволителни резултати, ако целевият възел не е стационарен. В този случай конвергенцията на KF е недостатъчна, за да се отчете динамиката на целевия възел.

7.4.3.2 Тестов сценарий 2

Целта на втория експеримент е да се валидират алгоритмите за локализация в симулирана среда при различни позиции на целевите възли спрямо позицията на референтните възли, различни стойности на колебанията в стойностите на RSSI и различни разстояния между референтните възли (маяк). За да се валидират алгоритмите, се изчислява точността на локализацията. Точността на локализация ε_t е мярка за коректността на получената оценка за позицията на целевия възел, която се изчислява като разликата между действителната позиция на целевия възел, $p_t = (x_t, y_t)$ и оценката за тази позиция, $\hat{p}_t = (\hat{x}_t, \hat{y}_t)$:

$$\varepsilon_t = \sqrt{(x_t - \hat{x}_t)^2 + (y_t - \hat{y}_t)^2}. \quad (7.33)$$

Всички експерименти са проведени при предположението, че са анализирани сигналите от три референтни възли, които образуват равностраничен триъгълник. Анализирана е точността на локализация при три различни разстояния между маяците (3 m, 6 m и 9 m) и три различни позиции на целевия възел.

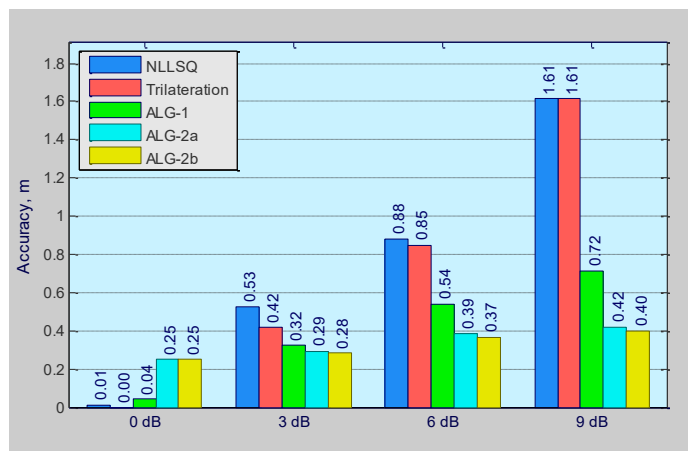
Точността на локализация се тества за следните позиции на целевия възел:

- Целевият възел е в центъра на равностраничен триъгълник;
- Позицията на целевия възел съвпада с позицията на референтен възел;
- Целевият възел е извън триъгълника.

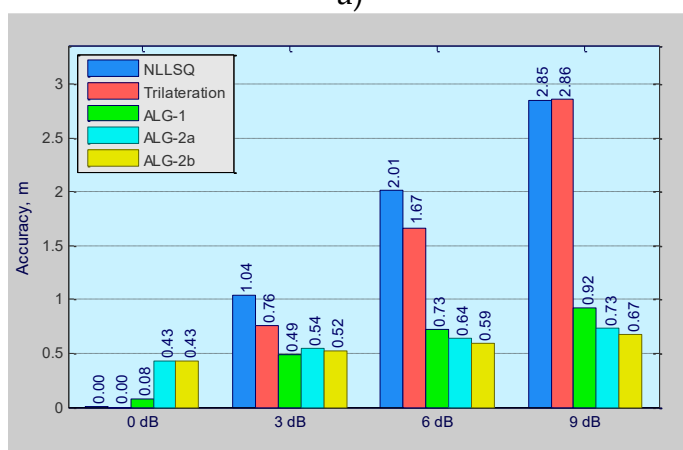
За всяка от тези три позиции са направени 50 оценки за позицията на целевия възел при четири различни максимални стойности за нивото на случаен шум в RSSI (0 dBm, ± 3 dBm, ± 6 dBm, и ± 9 dBm).

Фиг. 7.31 показва средните оценки за точността на локализация за всички възможни позиции на целевите възли при различни разстояния между референтните възли и шума в RSSI стойностите. Алгоритмите ALG-2A и ALG-2B имат почти постоянна точност за всички стойности на шума и разстоянията между референтните възли. Това е очаквано, тъй като алгоритмите ALG-2 не се влияят значително от шума в RSSI данните. По-точен е алгоритъм ALG-2B, който гарантира средна точност на локализация по-малка от 1 m за всички тестови сценарии. Алгоритъмът ALG-1 има приблизително същата точност като алгоритъм ALG-2 за шум в RSSI данни до ± 6 dBm.

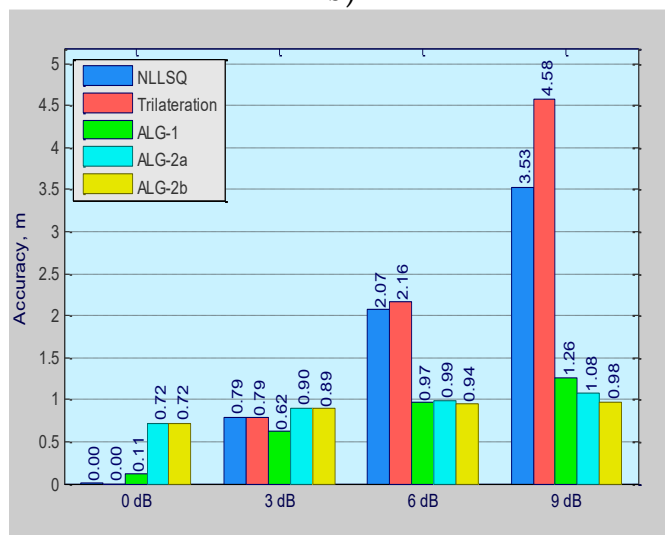
Алгоритмите Trilateration и NLLSQ могат да се използват за микролокализация само при малко разстояние между маяците и ниски нива на шум в комуникационните канали. Тези изисквания не могат да бъдат гарантирани в реална интериорна среда. Анализът на позицията на целевия възел показва, че всички алгоритми са най-точни, когато целевият възел се намира в центъра на тежестта на триъгълника, образуван от трите най-близки маяка. Въпреки това, при тази позиция на целевия възел, алгоритмите Trilateration и NLLSQ гарантират микролокализация само когато разстоянието между маяците е по-малко от 3 m. Алгоритмите ALG-1 и ALG-2 са най-неточни, когато позицията на целевия възел е извън триъгълника, образуван от референтните възли. В този случай тези алгоритми гарантират микролокализация само ако разстоянието между референтните възли е по-малко от 6 m.



a)



b)



c)

Фиг. 7.31 Сравнение на точността на локализация на алгоритмите: а) Разстоянието между маяците е 3 m; б) Разстоянието между маяците е 6 m; в) Разстоянието между маяците е 9 m

7.4.3.3 Тестов сценарий 3

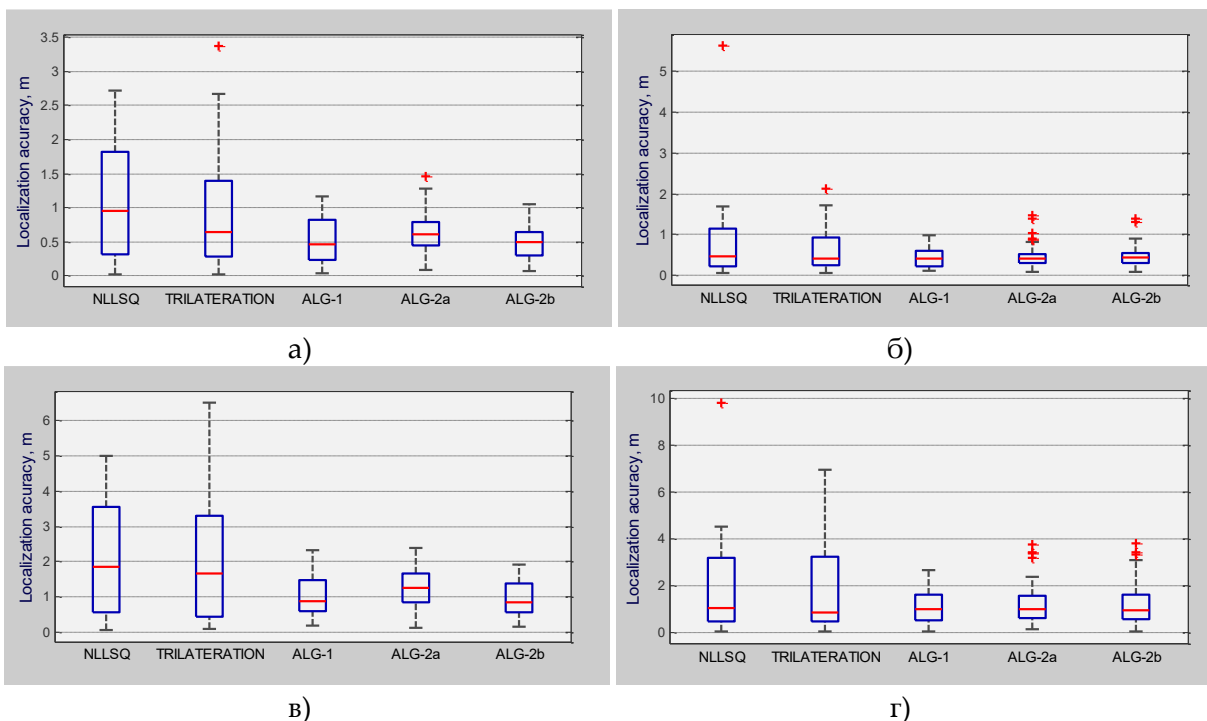
Този експеримент анализира как плътността и формата на разположение на маяците влияят върху точността на алгоритмите за локализация. Точността на локализацията се увеличава с намаляването на разстоянието между маяците. В големи обществени сгради разходите за разполагане се увеличават с намаляването на разстоянието между маяците. По тази причина трябва да се направи компромис между разходите за разполагане и точността на локализация. Експериментите показват, че освен от плътността на маяците, точността на локализация зависи и от формата, която образуват маяците. [Rezazadeh и колектив, \(2018\)](#) доказват, че най-доброто разположение на маяците е, когато всеки три от тях образуват равноностранен триъгълник. Тази форма се нарича Crystal-shape iBeacon Placement (CiP) и дава по-добри резултати в сравнение със стратегията за разполагане Z-curve, описана в ([Rezazadeh и колектив, 2014](#)). В реална сградна среда тези форми не могат да бъдат гарантирани поради специфичните размери на стаите и коридорите, както и поради необходимостта от различна точност на локализация в различните стаи и коридори.

Разработено е приложение на Matlab за автоматично разполагане на маяци, като се вземат предвид желаната форма и плътност на разполагане. Всички анализирани алгоритми за локализация са тествани за симулирано пространство от 200 m² (20 m × 10 m) в следните два сценария:

- CiP форма и разстояние между маяците 5 m (5 маяка) и 10 m (14 маяка).
- Правоъгълна форма и разстояние между маяците 5 m (6 маяка) и 10 m (15 маяка).

За всеки сценарий точността на локализация е анализирана за 50 тестови точки с псевдо-случайни координати. Към данните за RSSI е добавен различен случаен шум с максимална амплитуда от ± 5 dBm. Получените резултати са показани на Фиг. 7.32 като диаграма „box-and-whisper“.

Алгоритмите за трилатерация и NLLSQ имат по-добра точност на локализация, ако се използва форма CiP. От друга страна, алгоритъм ALG-2 има по-добра точност, ако се използва правоъгълна форма. Причината за това е, че алгоритъм ALG-2 имат по-малка точност, когато целевият възел е извън триъгълника, образуван от най-близките маяци. Такъв сценарий е възможен поради шума в радиоканалите.



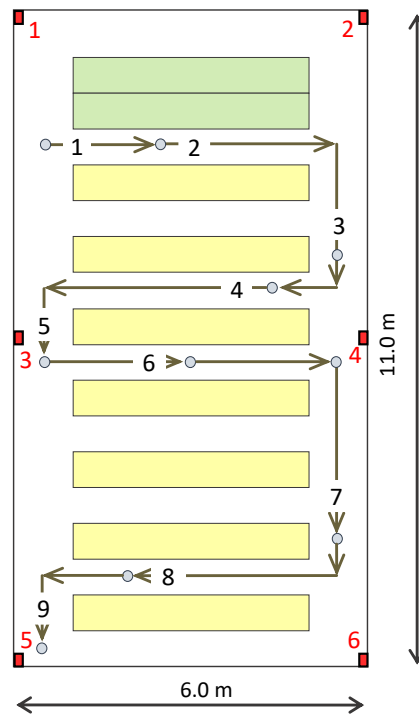
Фиг. 7.32 Сравнение на различни типове разположение на маяци – диаграма „box-and-whisker“ ($n = 50$, $\Delta RSSI = \pm 5 \text{ dBm}$): а) Правоъгълна форма, $d = 5 \text{ m}$; б) Триъгълна форма, $d = 5 \text{ m}$; в) Правоъгълна форма, $d = 10 \text{ m}$; г) Триъгълна форма, $d = 10 \text{ m}$

Точността на локализация на алгоритъма ALG-1 зависи в по-малка степен от формата на разположение и плътността на маяците в сравнение с други алгоритми. При разстояние между маяците от 5 m, само алгоритмите ALG-1 и ALG-2B могат да се използват за микролокализация. Например, за форма CiP, точността на алгоритъма ALG-1 е по-малка от 1 m за всички тестови точки. В този случай алгоритъмът ALG-2A има три тестови точки, а алгоритъмът ALG-2B има две тестови точки, за които точността на локализация е над 1 m, но по-малко от 1,5 m. При разстояние между маяците от 10 m най-добрата точност на локализация има алгоритъмът ALG-2B, когато се използва правоъгълна форма. В този случай нито един от алгоритмите за локализация не гарантира точност под 1 m за всички тестови точки.

7.4.3.4 Тестов сценарий 4

Целта на този сценарий е да валидира алгоритмите за предварителна обработка и локализация на данните от анализирания маяк в реална интериорна среда. За реализирането на този сценарий се използват специално разработени приложения за Android и Matlab. Експериментът е проведен в лекционна зала.

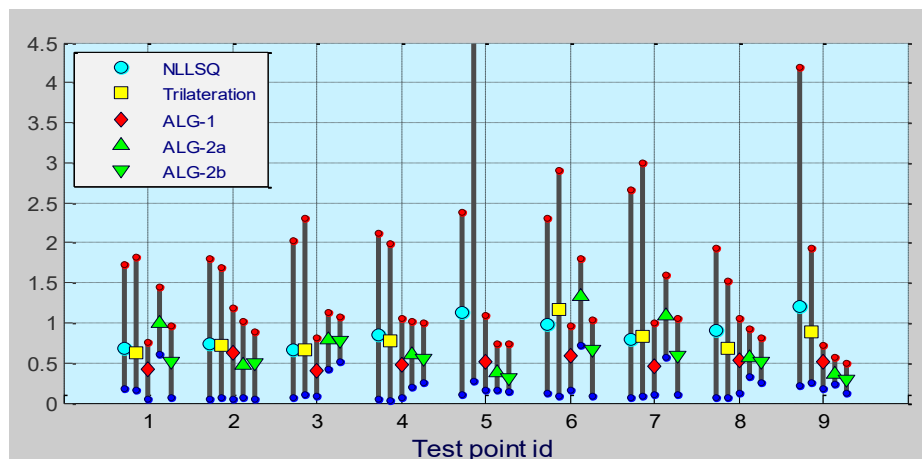
Проведени два експеримента. Първият експеримент анализира точността на алгоритмите за локализация в девет тестови точки в лекционната зала (виж Фиг. 7.33).



Фиг. 7.33 Тестова среда – лекционна зала: маяци (червени правоъгълници); тестови точки (сиви кръгчета); мебели (жълти и зелени правоъгълници)

Позициите на тези точки са избрани, за да симулират различни позиции на тестовата точка спрямо позициите на маяците. Когато участник в експеримента достигне тестова точка, той/тя натиска бутон, за да може мобилното приложение да получи информация за това събитие. Маршрутът с тестовите точки се преминава 20 пъти в един ден.

По време на експеримента пет души влизат в стаята. Софтуерът записва в CSV файл информацията от маяците и акселерометъра на всеки 200 ms. Софтуерът на Matlab изчислява точността на локализация за всеки анализиран алгоритъм във всички тестови точки (виж Фиг. 7.34).

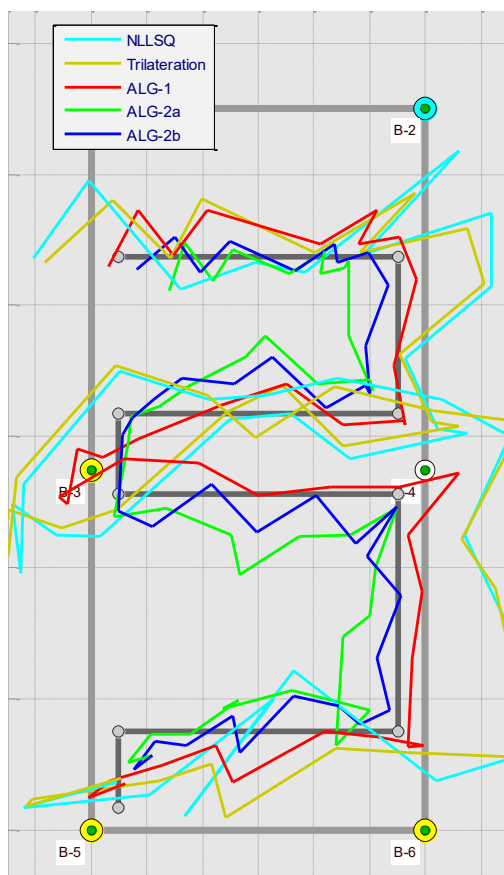


Фиг. 7.34 Точност на локализация за всички анализирани алгоритми за локализация във всички тестови точки

За всяка тестова точка се получава стойността на минималната (сини точки), максималната (червени точки) и средната (различен маркер) точност на локализация за всеки от алгоритмите за локализация. Средната точност на локализация на алгоритмите ALG-1 и ALG-2 е по-малка от 1 m за всички тестови точки. Средната точност на алгоритъма ALG-2A е над 1 m за тестовите точки 1, 6 и 7. Причината за това е, че този алгоритъм има ниска точност на локализация, когато целевата точка е близо до страната на триъгълника, образуван от маяците. Алгоритъмът за трилатерация дава висока грешка в локализацията за тестова точка 5, тъй като за някои от измерванията най-високата стойност на RSSI дава маяци 1, 3 и 5, които образуват права линия, а не триъгълник.

Алгоритмите за трилатерация и NLLSQ не могат да се използват за точно локализиране, тъй като има измервания, при които грешката е по-голяма от 1 m. Най-добри резултати се получават при използване на алгоритъма ALG-2B. Следващият по точност алгоритъм е ALG-1.

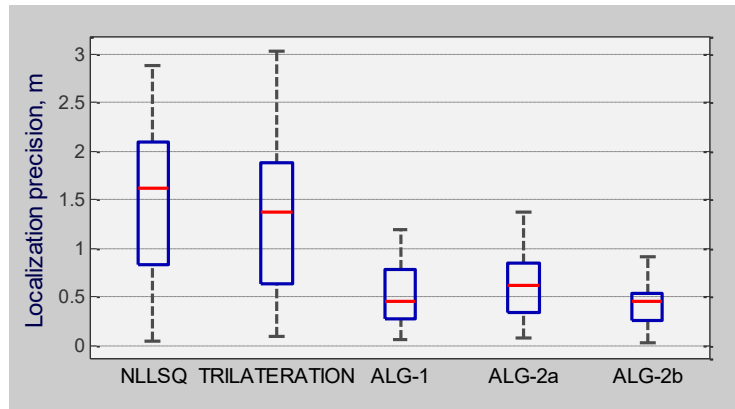
Вторият експеримент анализира точността на локализация на алгоритмите, когато участниците пресичат маршрута, който се формира от всички тестови точки. За да се сравни точността на локализация на различните алгоритми за локализация, се получава оценка за общото отклонение от пътя (най-краткото разстояние между изчислената позиция и маршрута). Фиг. 7.35 показва резултатите, получени при пресичане на маршрута от тестова точка 1 до тестова точка 9.



Фиг. 7.35 Предполагаеми маршрути спрямо идеалния маршрут (тъмно сиво)

Позицията на целевия възел се получава всяка секунда. Дължината на пътя е 30 m. Средното време за завършване на този експеримент е 40 секунди. Средната скорост на движение е 2,7 км/ч. Фиг. 7.35 показва идеалния маршрут (тъмно сиво) и маршрутите, получени с помощта на всеки от анализираните алгоритми за локализация.

Алгоритъмът ALG-2B има най-добрата точност на локализация (виж Фиг. 7.36). В този случай отклонението от маршрута е по-малко от 1 м за всички измервания. Следващият по точност алгоритъм е ALG-1, при който само едно измерване има отклонение по-голямо от 1 м, но по-малко от 1,2 м. Алгоритмите Trilateration, NLLSQ и ALG-2A не гарантират микролокализация.



Фиг. 7.36 Сравнение между различни алгоритми за локализация според точността на локализация

7.4.4 Заключение

Анализирана е възможността за създаване на IPS с ниска сложност с цел точно локализиране в интелигентни сгради с помощта на BLE маяци. Предложен е нов адаптивен алгоритъм за намаляване на шума в данните от BLE маяците. Алгоритъмът се базира на комбинация от филтър за пълзяща средна (MA) и адаптивен филтър на Калман (AKF). AKF използва размита логика за оценка на шума в данните от RSSI и за улавяне на динамиката на посетителите. Този алгоритъм е проектиран за изпълнение в реално време.

Предложените алгоритми за микролокализация в сгради са валидирани в различни симулирани и реални интериорни среди. Предимствата и ограниченията на всеки от алгоритмите са идентифицирани въз основа на анализ на това как тяхната точност зависи от фактори като ниво на шума в данните от маяците, плътност на разполагане на маяците, форма на разположение на маяците и позиция на целевия възел спрямо позицията на референтните възли.

Следва резюме на основните резултати, получени след експериментите:

- В реална интериорна среда (наличие на препятствия за радио вълните и много движещи се посетители) не е възможно да се получи микролокализация без използване на адаптивно филтриране на данните от маяците. Времето за оценка на позицията на целевия възел е фиксирано на една секунда, за да се опростят изчисленията на AKF.
- Нивото на колебания в данните RSSI е 2 - 9 dBm и зависи от редица фактори, като разстоянието между маяците и ефекта на многопътност на сигналите от

маяците. Най-зависими от нивото на шума са алгоритмите Trilateration и NLLSQ. Най-малко чувствителен към шума е алгоритъмът ALG 2B.

- Алгоритмите ALG-1 и ALG-2 са по-чувствителни от алгоритмите Trilateration и NLLSD към позицията на целевия възел само ако няма шум в данните RSSI. В реална среда, където има препятствия и движещи се хора, това е невъзможно. Всички алгоритми имат най-висока точност, когато целевият възел се намира в центъра на тежестта на фигурата, която образуват референтните възли.
- Плътността на маяците е основен фактор, който влияе на точността на всички алгоритми. В реална среда в сграда алгоритмите Trilateration и NLLSQ не могат да се използват, ако разстоянието между референтните възли d е по-голямо от 3 m. Другите алгоритми гарантират точна локализация, когато $d < 10$ m. Микролокализация се гарантира, когато $d < 6$ m.
- Формата на разположение на маяците влияе върху точността на локализацията, но по различен начин за различните алгоритми за локализация. Например, алгоритмите Trilateration и NLLSQ са най-точни, ако формата е равноностранен или равнобедрен триъгълник. От друга страна, алгоритмите ALG-1 и ALG-2 са по-точни, ако формата на разположението е правоъгълна.
- Алгоритмите NLLSQ и ALG-1 могат да изчисляват позицията на целевия възел, използвайки данни от два или повече маяка.
- Изборът на маяци за изчисляване на позицията на целевия възел не трябва да се основава единствено на RSSI стойности. В реална среда е гарантирано, че само маякът с най-висока стойност на RSSI е избран правилно. За останалите маяци трябва да се вземе предвид каква форма образуват с избрания маяк. Ако е необходимо, трябва да се избере следващата двойка маяци.
- При локализиране на движещи се обекти в реална сградна среда алгоритъмът ALG-2B има най-добра точност, следван от алгоритъма ALG-1.

Предложените алгоритми за предварителна обработка на данни от BLE маяци и прецизна локализация могат да се използват в услуги за локализация на посетителите на интелигентни сгради, например: доставка на персонализирано съдържание в зависимост от местоположението на всеки посетител; прецизна навигация на хора с увреждания на зрението в големи обществени сгради; откриване на социални взаимодействия между хора и др.

ГЛАВА 8: СИСТЕМА ЗА ПРЕВЕНЦИЯ НА ДЕМЕНЦИЯ ЧРЕЗ ТЕХНОЛОГИИ ЗА НАВИГАЦИЯ

8.1 Въведение

Думата деменция е латинска и буквално означава „без разум“. На съвременния етап с деменция се описват набор от симптоми, свързани с умствено разстройство, загуба на памет, трудности с ориентирането и мисловната дейност, необосновани промени в настроението, проблеми с ученето, езика и преценката ([Световна здравна организация, 2023](#)).

Според [McKhann и колектив \(2011\)](#) най-разпространени видове деменция са четири:

- Болест на Алцхаймер (AD). Това е най-често срещаният тип деменция, съставляващ 60 – 70% от случаите. При този тип деменция е характерно патологичното натрупване на амилоидни бета плаки и неврофибриларни сплитания, съдържащи хиперфосфорилиран тау протеин.
- Съдова деменция (VD). Този тип деменция заема второ място с 15 - 20% от случаите. Тя е в резултат от цереброваскуларна патология, която води до инфаркти или до широко разпространени промени в бялото вещество на мозъка.
- Деменция с телца на Леви (DLB). Този тип деменция се среща при 10 - 15% от случаите. Специфично за нея е наличието на алфа-синуклеинови протеинови агрегати, проявяващи се с характерни колебания в когнитивните функции, зрителни халюцинации и паркинсонизъм (болест на Паркинсон).
- Фронтално-темпоралната деменция (FTD). Този тип деменция засяга около 5 - 10% от пациентите, особено тези под 60-годишна възраст. Характеризира се с прогресивна дегенерация на фронталните и темпоралните кортикални области. Специфичното за този тип деменция са проблеми с поведението и езиковите разстройства.

Счита се, че деменцията е една от най-честите причини за увреждане при възрастните. С напредване на възрастта се увеличава риска от това заболяване. При възрастовата група 65 – 74 години деменцията се среща само при около 3% от хората. За хора от 75 до 84 години този процент се увеличава на 19. При възрастовата група над 85 години заболяването е характерно за над 50% от населението.

Деменцията е заболяване, което прогресира във времето. То преминава през няколко етапа:

- леко когнитивно увреждане;
- лека деменция;
- умерена деменция;
- тежка деменция.

При първият етап (леко когнитивно увреждане) най-характерно е загубата на памет (отделни думи). Възможно е този стадий да не прогресира до лека деменция, а да е свързан само с възрастта на пациента. При леката деменция за пациентите са характерни загуба на памет, трудно намиране на думи при съставяне на изречения (аномия), трудности при планиране на сложни задачи и загуба на ориентация. При умерената деменция пациентът изпитва симптоми, характерни за леката деменция, но те се проявяват по-често и се засилват с времето. За този етап е характерно и проблеми със съня, например събуждане около три часа през нощта. Този етап предполага необходимост от помощ за пациента при рутинни ежедневни дейности като ориентиране и обличане и събличане. Тежката деменция води до инвалидизация на пациентите. Те не могат да се справят самостоятелно с рутинни дейности.

При диагностирането на деменция се използват разнообразни подходи, повечето от които или са неточни, или инвазивни и много скъпи. За оценка на умствените нарушения на пациентите най-често се използват евтини невропсихологични оценки. Тези оценки се базират на резултатите от тестове, например Mini-Mental State Examination (MMSE) и Montreal Cognitive Assessment (MoCA). И двата теста използват 30-точкова скала. При MMSE теста, ако резултатът е под 24 се счита, че пациентът има когнитивни увреждания. Този тест не е особено подходящ за откриване на леки когнитивни нарушения (MCI). По-нов и по-чувствителен към MCI е MoCA теста. При него, при праг под 26 се счита, че пациентът има когнитивни нарушения. Проблемът и при двата теста е, че те са чувствителни към културните и образователни специфики на пациентите ([Nasreddine и колектив, 2005](#)).

Много по-точна оценка се получава при подходи, които използват различни видове биомаркери, например амилоидни Positron Emission Tomography (PET) изображения и анализ на цереброспиналната течност - Cerebrospinal Fluid (CSF). Проблемът е, че тези изследвания са инвазивни (инжектиране на радиофармацевтици при PET или лумбална пункция при CSF) и скъпи и не могат да се използват масово ([Jack и колектив, 2018](#)).

Образната диагностика чрез магнитно-резонансна томография (MRI) и функционална MRI (fMRI) предоставят важна информация за структурата на мозъка и неговата функционална активност. Поради високата цена образната диагностика не може да се използва за масов скрининг на населението.

Основен проблем при повечето методи за диагностика на деменция е, че те се прилагат след като е минал етапа на леките когнитивни нарушения ([Dubois и колектив, 2016](#)). Алтернатива на стандартните тестове като MoCA може да бъде цифровата биометрия - непрекъснато събиране на обективни данни за пациентите от смартфони, носими устройства и сензори. Изследванията на [Kaye и колектив \(2018\)](#) показват, че данните от акселерометъра (вграден в смартфон или носима електроника) могат да бъдат надежден индикатор за нарушения в походката и намалена физическа активност. Това са биомаркери, които са специфични за хора с ранна форма на деменция. Научните експерименти показват, че системите за локализация в сгради чрез BLE маяци и IR сензори, позволяват анализ на пространствените навигационни модели на пациентите. Това би позволило ранна диагностика на болестта на Алцхаймер ([Рое и колектив, 2021](#)). Чрез измерване на температурата на пациентите може да се анализират нарушения в циркадния ритъм, които са индикатор за невродегенеративни процеси ([Tamura и колектив, 2020](#)). Умните баджове на форма Kontakt.io интегрират тези функционалности – те комбинират сензори за движение, мониторинг на околната среда и идентификация на стаи чрез IR сензор в едно единствено, сравнително евтино устройство, подходящо за дългосрочно използване в домашни условия.

Необходими са устройства, които да се използват лесно и надеждно като холтер монитори – лекарят ги настройва и инсталира, а пациентът ги носи за 24-48 часа след което данните се анализират. Въпреки развитието на носимите технологии и комуникациите, все още има пропуски в областта на ранно откриване на деменция в домашни условия. Първо, липсват лесни за използване устройства чрез които да се получават необходимите биомаркери за дълъг период от време (минимум 1-2 дни). Второ, липсват стандартизирани рамки за получаване, съхранение и анализ на мултимодални сензорни данни, за да се получат клинично приложими заключения. Трето, повечето проучвания са валидирани в контролирана лабораторна среда, а не в домашна среда, което ги прави трудни за масов скрининг. Четвърто, трябва да се решат всички проблеми, свързани със защитата на личните данни на пациентите, съгласувано с националните и световни нормативните изисквания в тази област ([Ienca и колектив, 2018](#)).

Концепцията за ранна диагностика на деменцията в домашни условия би могла да преодолее много от ограниченията на традиционните диагностични подходи. Системите за диагностика в домашни условия се базират на това, че промените, свързани с деменцията, се проявяват по фин начин в ежедневните рутинни дейности, преди те да станат клинично очевидни.

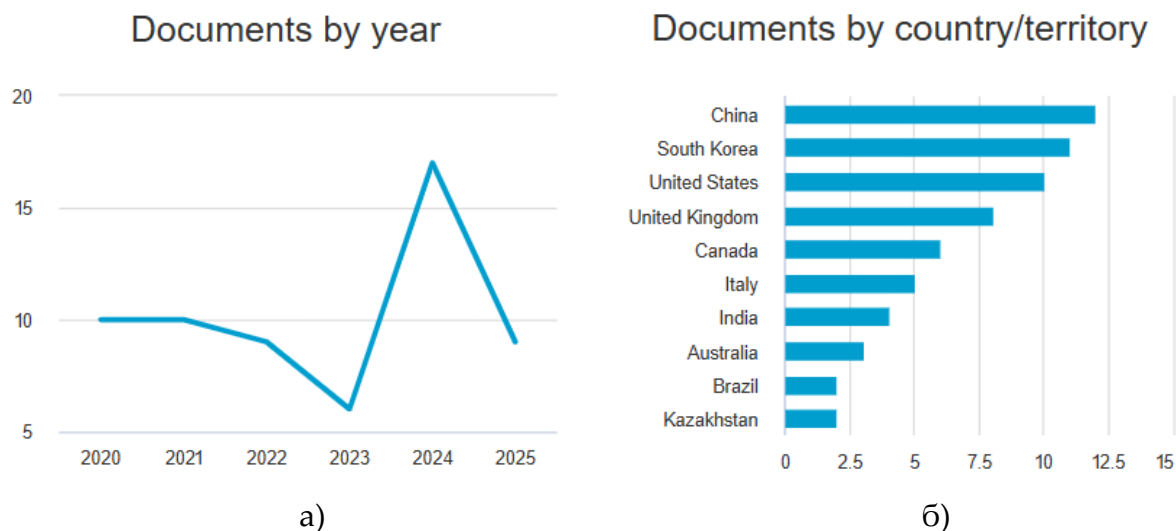
Основната цел на тази глава е да се разработи и валидира на нова, технологично ориентирана рамка за ранно откриване на деменция. Целта е да се проектира мащабируема и евтина система за ранно разпознаване на деменция в домашни условия.

8.2 Анализ на проблема

Анализирани са наличните в Scopus публикации за последните пет години, свързани тематично с превенцията на деменция чрез използване на сензори и биомаркери. Чрез низа за търсене се задава тематика „early dementia detection using sensors“ за последните пет години, публикувана в научни списания:

TITLE-ABS-KEY (early И dementia И detection И using И sensors) И PUBYEAR > 2019 И PUBYEAR < 2026 И (LIMIT-TO(DOCTYPE, "ar") OR LIMIT-TO(DOCTYPE, "re"))

Извлечени са 61 релевантни публикации. Статистическият анализ по година на публикуване (виж Фиг. 8.1а) показва, че тематиката е сравнително нова и има пик за 2024 г. На Фиг. 8.1б се вижда, че авторите на публикациите са от водещи в областта на превенцията на заболявания държави.



Фиг. 8.1 Статистика на анализирани документи по: а) година на публикуване и б) държава в която работят авторите

Както вече бе отбелязано настоящите методики за диагностициране на леко когнитивно увреждане (MCI) имат редица недостатъци. На този етап образната диагностика е прекалено скъпа, за да може да се използва за широкомащабен скрининг. Тези методи се използват основно при напредналите стадии на деменция, когато умствените увреждания са необратими (Ford и колектив, 2023).

Едно възможно решение на проблема са технологиите за отдалечен мониторинг на пациенти и носимите технологии. Експериментите потвърждават, че отклоненията в придвижването, ориентирането, съня и някои ежедневни дейности могат да бъдат използвани като много ранни индикатори за умствени проблеми. Например, при изследването на Rawtaer и колектив (2020) се използват пасивни IR сензори за движение, интелигентни контакти и сензори чрез които се анализират поведенческите модели на възрастни пациенти с и без MCI в домашна среда. Резултатите показват проблеми със съня за пациентите с MCI, както и по-слаба физическа активност. Анализът на приема на лекарства показва, че при пациентите с MCI е характерно често забравяне на приема на лекарства.

Shahzad и колектив (2022) анализират възможността за скрининг на MCI чрез биомаркери на базата на анализ на походката. За целта те използват инерциални сензори, прикрепени към пиццала на пациентите. На базата на тези биомаркери те разработват машинно обучени модели. Постигната е точност от 71,67% и чувствителност от 83,33% при разпознаване на MCI, което е доказателство за практическата приложимост на подхода за домашно проследяване на заболяването.

При разработката на Kim и колектив (2022) прогнозирането на деменция се базира на използването на пасивни IR сензори за мониторинг на физическата активност на пациентите в различни домашни условия. Системата демонстрира много добри резултати - площ под кривата (AUC) от 0.99 при работа с дълбоки невронни мрежи и анализ на главните компоненти (PCA). Това дава основание да се заключи, че мултимодалните подходи са перспективни, тъй като повишават точността на предсказване на MCI.

Jonell и колектив (2021) описват мултимодална система за ненаатрапчиво извличане на данни по време на реални клинични интервюта. За целта се използват множество сензорни устройства - смартфони, таблет, модул за проследяване на движението на очите, масив от микрофони и умна гривна. Системата се използва в болница Каролинска в Стокхолм и демонстрира осъществимостта на

предложената методология за подобряване на клиничните оценки на ранна деменция.

Значителен принос в областта представляват и изследванията на [Müller и колектив \(2020\)](#), които са разработили SENDA протокола. Този подход интегрира когнитивни, моторни, сензорни и неврофизиологични параметри, измервани чрез различни технологии. Проучването има за цел да идентифицира асоциации с когнитивните промени във времето чрез повторни измервания на интервали от 8 месеца в продължение на 3 години.

Особено внимание заслужават изследванията, фокусирани върху анализа на придвижване и моделите на движение като индикатори за прогресивна деменция. [Chaudhary и колектив \(2020\)](#) предлагат система за ранно откриване на деменция, базирана на класификация на моделите на придвижване чрез сигнали от околната среда чрез пасивни сензори. Системата сегментира движенията и ги класифицира чрез рекурентни невронни мрежи, като се справя с небалансираните класове.

Интересен аспект представлява и използването на баланса на пациентите като биомаркер за ранно откриване на когнитивни нарушения. [Jamshed и колектив \(2024\)](#) изследват чрез инерционни сензори постуралния баланс с цел предварителен скрининг на MCI в домашна среда. Тяхното проучване с 60 пациенти (30 когнитивно здрави и 30 с MCI) демонстрира, че значимите биомаркери за идентификация на MCI, свързани с баланс, показват специфични характеристики в четири различни състояния – отворени очи, затворени очи, вдигане на десен крак и вдигане на ляв крак. Системата, базирана на тези биомаркери, открива MCI с точност от 75,8%.

[Bijlani и колектив \(2024\)](#) предлагат нов подход, използвайки Graph Barlow Twins модел за откриване на неблагоприятни здравословни събития при хора с деменция в домашна среда. Експериментите показват, че техния модел превъзхожда известните алгоритми при откриване на падания в три пациентски групи. Средната точност на системата е 81%.

При ранната диагностика на деменция чрез използване на цифрови технологии предполага етични проблеми, свързани с обработката на лични данни. [Ford и колектив \(2023\)](#) изследват етичните въпроси при използването на цифрови биомаркери и изкуствен интелект за ранно откриване на деменция. Заключение е, че въпреки важността на заболяването, остават значителни предизвикателства,

включително липсата на налично лечение, както и риска от задълбочаване на здравните неравенства поради пристрастни данни за обучение.

[Hamedani и колектив \(2024\)](#) анализират необходимостта от количествена неврология и използването на сензори за оценка на моторните дефицити при деменция. Изследването им показва, че съвременните технологии могат да се използват с цел получаване на биомаркери, които да се използват прогнозиране на деменция.

Направеният литературен анализ показва, че носимите сензори и технологиите за дистанционен мониторинг са перспективни за ранно откриване и превенция на деменция. Въпреки обещаващите резултати, остават важни предизвикателства, свързани с възможността на подобни системи за масов скрининг. Проблеми има и с етичните въпроси и необходимостта от стандартизирани протоколи за клинично прилагане на подобни системи.

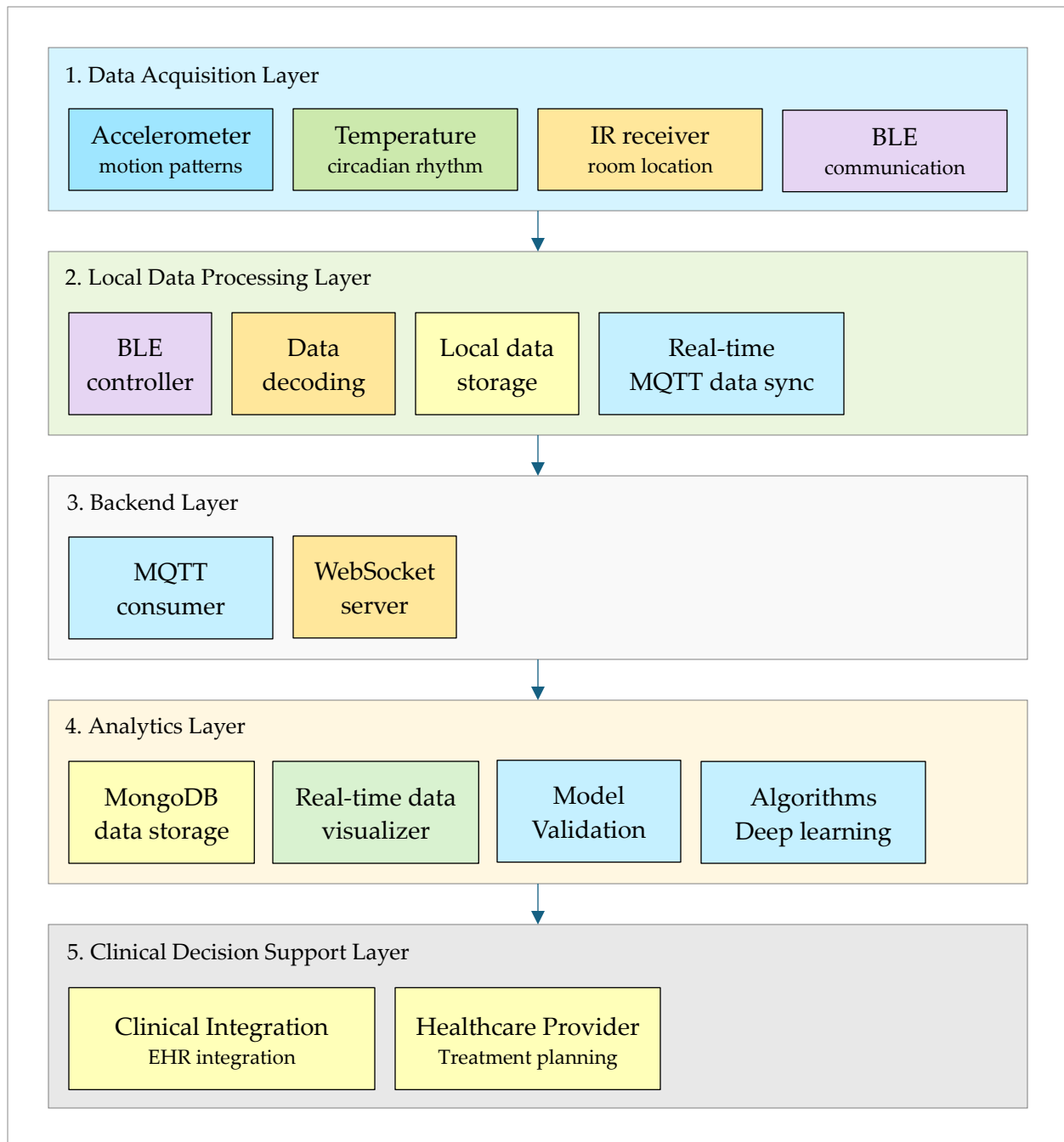
8.3 Концептуална рамка

Предлага се концептуалната рамка за ранно откриване на деменция чрез интелигентни баджове. Рамката интегрира мултимодално събиране на данни от сензори, отдалечен мониторинг на данните в реално време и запис на данните в база данни с цел последващ задълбочен анализ. Тази рамка се базира на теоретичното разбиране, че поведенческите и физиологичните промени, свързани с деменцията, се проявяват по фин начин в ежедневните дейности, преди клиничните симптоми да станат очевидни, като по този начин създава възможности за ранна превенция.

Рамката се основава на теорията за разпознаване на поведенчески модели – необратимите когнитивни промени се проявява чрез измерими промени в рутинните дейности, моделите на движение и циркадните ритми. Рамката се базира на теорията за цифровите биомаркери, която използва измерванията, получени от сензори, като обективни, количествено измерими физиологични и поведенчески индикатори за здравословното състояние. Анализът на данните от акселерометъра позволява откриването на аномалии в походката и двигателни нарушения, които предшестват клиничните симптоми на деменция с няколко години, докато пространствено-времевият анализ разкрива нарушения в рутинните модели, които са показателни за спад в изпълнителните функции. Температурните колебания, засечени чрез термични сензори, предоставят информация за нарушенията в циркадния ритъм, свързани с невродегенерацията,

докато проследяването на местоположението на ниво стая позволява количествено измерване на пространствената дезориентация.

На Фиг. 8.2 е показана структурата на предлаганата концептуална рамка. Тя е изградена от пет взаимно свързани слоя.



Фиг. 8.2 Структура на концептуалната рамка

Слой 1. Data Acquisition Layer

На най-ниско ниво е слой за извличане на данни (Data Acquisition Layer). На този етап този слой се реализира чрез използване на носими сензори, които предават данните в BLE пакети на разстояние до 30 метра. Това позволява проследяване на местоположението, мониторинг на температура, откриване на движение и дистанционна конфигурация чрез Over-the-Air (OTA) актуализации. Носимите сензори са снабдени с IR приемник, който позволява много точно определяне на местоположението в рамките на стая.

За целта носимите устройства приемат сигнали от IR модул, който е програмиран са излъчва идентификатора на стаята в която е инсталиран. Допълнително, акселерометъра отчита придвижването, а вграден температурен сензор следи температурата. Носимите сензори имат програмируеми бутони, които могат да бъдат използвани за подаване на сигнали – при инцидент или нужда от помощ.

Слой 2. Local Data Processing Layer

Слой 2 отговаря за обработка на данните от умните баджове на локално ниво. Този слой е реализиран чрез системата ESP32-S3-BOX-3 на фирма Espressif. Това е интелигентно, многофункционално устройство, базирано на мощния микроконтролер ESP32-S3. То представлява усъвършенстван умен контролен панел, проектиран за лесна интеграция в системи за домашна автоматизация, индустриално управление и IoT приложения. Устройството разполага с 4-инчов IPS сензорен LCD дисплей с резолюция 320x240 пиксела, което позволява създаване на графични потребителски интерфейси. Основният изчислителен модул – ESP32-S3 – включва двудрен процесор Tensilica LX7, поддържа AI ускорение (векторни инструкции), Wi-Fi 4 и Bluetooth 5 свързаност. ESP32-S3-BOX-3 използва външна SPI флаш памет с капацитет 16 MB и външна псевдостатична RAM (PSRAM) с капацитет 8 MB. PSRAM се използва като допълнителна оперативна памет (RAM) за изпълнение на по-тежки задачи като буфериране на данни и машинно обучение.

Суровите данни (пакети Location и Telemetry), получени от умните баджове, се декодират – извличат се данните, които представляват интерес за системата (стойности за ускорението по трите оси от акселерометъра, стойност на температурата, състояние на двата бутона, идентификатора на стаята в която се намира умния бадж).

ESP32-S3-BOX-3 поддържа протокол Message Queuing Telemetry Transport (MQTT). Това позволява изпращане на данните от баджовете в реално време към MQTT

сървър посредством Wi-Fi връзка. Преди да се изпратят данните, те се конвертират до JSON с вградена времева марка. За MQTT сървър се използва облачно разгърната инстанция на RabbitMQ. Ако в даден момент липсва Wi-Fi свързаност данните се буферират във флаш паметта. При възстановяване на връзката, буферираните данни се предават веднага към MQTT сървъра.

Слой 3 (Backend Layer) е междинен слой в предлаганата архитектура, който има за задача да чете данните от MQTT сървъра и да ги препраща за обработка в реално време към други програмни модули. За целта се стартира WebSocket сървър, който позволява извличане на данните от умните баджове. За да няма загуба на информация, сървърът буферира данните, генерирани през последните 10 минути.

Слой 4 (Analytics Layer) се състои от модули, които са свързани със съхранение и обработка на данните от умните баджове. Тези модули четат данните, изпращани от WebSocket сървъра, и ги обработват по специфичен начин. На този етап, данните могат да бъдат съхранявани като времеви серии в нерелационна база данни MongoDB (модул MongoDB data storage). Това позволява обработка на данните с мощни средства, например алгоритми за дълбоко машинно обучение, например LSTM и трансформаторни кодери. Този тип невронни мрежи са способни да улавят дългосрочни времеви зависимости в динамичните сензорни данни. Предвидена е възможност и за визуализиране на данните от баджовете в реално време (модул Real-time data visualizer).

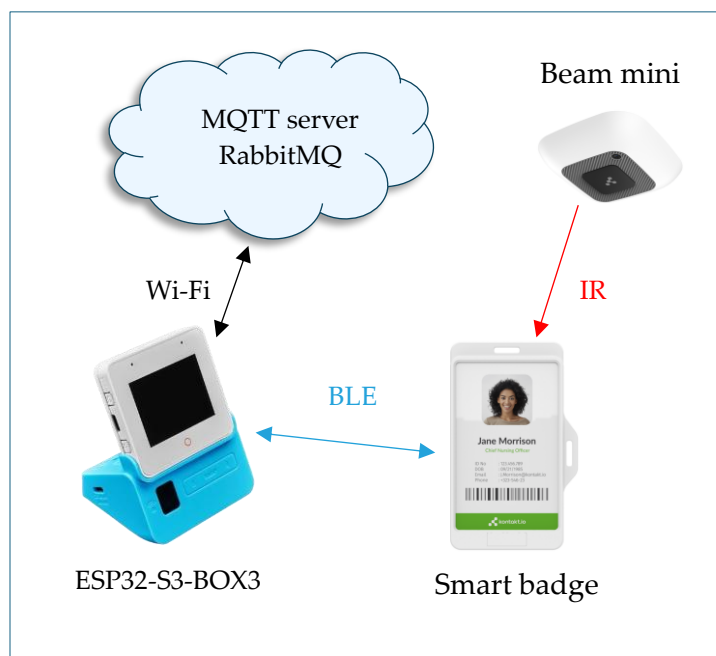
Последният, пети слой е предназначен за подкрепа при клинични решения. На този етап резултатите от аналитичния модел се трансформират в лесно интерпретируеми категории на риск за деменция: нисък (означен със зелен код), умерен (оранжев код) и висок (червен код). Тази цвятова система улеснява интерпретацията от страна на клиничния персонал с цел навременна и целенасочена интервенция. Слой 5 поддържа интеграция с електронни здравни досиета на пациентите, което позволява автоматизирано известяване за резултата от изследването. Крайната цел е да се подпомогнат здравните специалисти при вземането на информирани решения относно ранна диагностика и планиране на индивидуализирана терапия.

В своята цялост тази технологична рамка представлява съвременно решение, което комбинира носими технологии, изкуствен интелект и здравна информатика за системно и навременно управление на риска от деменция.

8.4 Архитектура на системата

Архитектурата на системата за ранно откриване на деменция е съобразена с това, че тя трябва да може да се използва в домашни условия, а не в контролирана среда. За целта тя трябва лесно да може да бъде разгърната и да не изисква сложна промяна на инфраструктурата.

На Фиг. 8.3 е показана опростената архитектура на системата. Пациентът получава умен бадж на фирма Kontakt.io, който трябва да бъде прикрепен към китката на ръката.



Фиг. 8.3 Архитектура на системата

Баджът е интелигентно устройство с формата на служебна карта. Корпусът на устройството е със защита IP65 срещу прах и пръски вода и е с размери приблизително 65 × 107 × 7.5 mm и тегло 29 грама. Работи при температури от 0 до 55 °C и относителна влажност на въздуха между 10 и 90%. Вградената батерия не се подменя, но издържа между 3 и 5 години в зависимост от режима на работа – около 3 години при активен инфрачервен датчик и до 5 години при стандартно BLE излъчване през 1 секунда.

Баджът получава информация за стаята в която се намира чрез анализ на сигналите от модул, наречен Beam Mini. Този модул изпраща на всеки около 4 секунди по инфрачервен (IR) канал идентификатора на стаята в която е поставен (най-често таван или стена). Умният бадж прочита този идентификатор чрез неговия IR сензор

и предава тази информация в т.нар. Location пакет. По този начин системата отчита много точно в коя стая се намира пациента във всеки един момент от време.

При конкретната разработка умните баджове се програмират да излъчват през 1 секунда телеметрични пакети (данни от акселерометъра, температурния сензор и статуса на бутоните) и през 2 секунди пакети за локация (идентификатор на стаята в която се намира умния бадж). Тази информация се получава от софтуера на ESP32-S3-BOX3. След декодиране на пакетите с данни те се предават като времеви серии през Wi-Fi канал към MQTT сървър.

8.5 Програмно осигуряване за ESP32-S3-BOX3

Представената система демонстрира ефективна интеграция на множество безжични протоколи в една архитектура, функционираща като IoT шлюз (изчисления на ръба). Хибридният дизайн комбинира локалното декодиране на BLE пакетите с глобална свързаност посредством Wi-Fi и MQTT, създавайки мащабируемо решение за разпределени сензорни мрежи.

Функционалността на системата ESP32-S3-BOX3 се основава на операционната система в реално време FreeRTOS, която осигурява многозадачно изпълнение и управление на събитията между отделните комуникационни подсистеми. Програмното осигуряване архитектурно реализира трислойна комуникационна йерархична система:

1. Wi-Fi свързаност.
2. BLE сканиране за разпознаване на близки устройства.
3. MQTT протокол за публикуване на данните от маяците.

Централната координация се реализира чрез механизма на групи събития (event groups) във FreeRTOS, използвайки бинарна семафорна сигнализация за състояния. Този подход осигурява атомарни операции за синхронизация на състоянията и предотвратява състезателни условия (race conditions) в многозадачната среда.

8.5.1 Wi-Fi мрежова свързаност

Управлението на Wi-Fi свързаността е реализирано чрез краен автомат (finite state machine), който автоматично възстановява връзката при прекъсване, използвайки криптографска защита чрез WPA2-PSK. Архитектурата на обработката на събития използва callback функции, регистрирани в рамките на ESP-IDF event loop системата. Състоянието WIFI_EVENT_STA_START инициира първоначалното свързване, WIFI_EVENT_STA_DISCONNECTED активира логика за повторно

свързване с експоненциално закъснение (exponential backoff), а състоянието IP_EVENT_STA_GOT_IP валидира успешно получената IP конфигурация чрез DHCP сървър. Системата автоматично конфигурира публичния DNS сървър на 8.8.8.8 за оптимизирана латентност при преобразуване на имена на домейни.

8.5.2 BLE сканиране

Използва се активен режим на сканиране на BLE каналите, оптимизиран за ниска латентност при откриване и висока енергийна ефективност. Параметрите за сканиране отговарят на спецификацията BLE 4.0+, със сканиращ интервал от 50 ms и времеви прозорец от 30 ms, което постига duty cycle от 60% с цел балансиране на производителността и консумацията на енергия. Архитектурата на обратните извиквания (callback) по профила Generic Access Profile (GAP) обработва събитията от жизнения цикъл на сканирането в реално време с минимална латентност.

8.5.2.1 Обработка на данните от баджовете

Обработката на данните от баджовете е реализирана чрез строго съответстващ на IEEE 802.11 Type-Length-Value (TLV) парсер. Алгоритъмът извлича идентификаторите на устройства за GAP AD типовете 0x08 (съкратено локално име) и 0x09 (пълно локално име), прилагайки защита срещу препълване на буфера чрез експлицитна проверка на дължината. Филтрирането на целевите устройства ограничава обхвата на обработката до устройства с идентификатор „RS2“, като по този начин се намаляват изчислителното натоварване и консумацията на енергия за нерелевантни устройства.

8.5.2.2 Декодиране на данните от баджовете

Декодирането на данните се свежда до анализ на три специализирани типа пакети, които баджовете генерират. Валидирането на пакетите включва проверка на заглавната част на пакетите (последователност 0x02 0x01 0x06) и консистентност на дължината преди разпределяне по тип към съответните декодиращи функции.

8.5.2.2.1 Телеметрични пакети (тип 0x03)

Телеметричните пакети съдържат данни от акселерометъра, сензора за температура и бутоните (син и червен) или части от тези данни в зависимост от версията на фирмения софтуер на баджа. Данните от акселерометъра (ID 0x06) предоставят триизмерни измервания, трансформирайки 8-битови цели стойности в SI единици (m/s^2). Температурните сензори включват модул с целочислена точност (ID 0x0B) и модул с висока резолюция (ID 0x13) с чувствителност $1/256^{\circ}C$. Поведенческото проследяване се реализира чрез записване на събития за движение (ID 0x16), включително брояч и времеви отпечатък на последната активност, а

взаимодействието с потребителя се следи чрез броене на натиснатите бутони (ID 0x19). Останалата информация, предавана в телеметричните пакети, не се използва.

На Фиг. 8.4 е показан примерен телеметричен пакет (Advertisement data) както сурови данни (зелено), така и като декодирани данни (синьо). В жълто са съобщенията от MQTT клиента.

```
RSSI: -60 dBm
MAC: 00:FA:B6:0B:11:17
Advertisement Data (29 bytes):
0000: 02 01 06 19 16 6A FE 03 05 06 10 01 FF BE 03 13 |.....j.....|
0010: 00 1A 04 16 97 04 00 05 19 15 0A FF FF |.....|
Telemetry Packet:
Acceleration: x=0.16, y=-0.16, z=-10.36 m/s2
Precise Temperature: 26.00 °C
Seconds Since Last Movement: 4 s
Movement Counter: 151
Blue Button Clicks: 21
Red Button Clicks: 10
Message published successfully, msg_id=10873
Scan Response Data (5 bytes):
0000: 04 09 52 53 32 |..RS2|
```

Фиг. 8.4 Примерен телеметричен пакет

8.5.2.2.2 Локационни пакети (тип 0x07)

В тези пакети се съдържа информация за идентификатора на стаята в която е баджа, времева марка на генериране на данните (UTC време), булева променлива за наличие на движение и използвания номер BLE комуникационен канал. На Фиг. 8.5 е показан примерен пакет с локационни данни.

```
RSSI: -69 dBm
MAC: 00:FA:B6:0B:11:5F
Advertisement Data (28 bytes):
0000: 02 01 06 18 16 6A FE 07 64 FC 26 1E 00 A7 9D 93 |.....j..d.&.....|
0010: 68 12 FF FF 31 32 6F 50 30 45 30 4A |h...120P0E0J|
Location Packet:
Battery Level: 100 %
TX Power: -4 dBm
Channel: 38
Device ID: 30
Movement: no
UTC Time: 1754504615
Beacon ID: 120P0E0J
Room ID: not used
Message published successfully, msg_id=8688
Scan Response Data (5 bytes):
0000: 04 09 52 53 32 |..RS2|
```

Фиг. 8.5 Примерен локационен пакет

8.5.3 MQTT клиент

MQTT клиентът е изграден върху TCP/IP стека и реализира пълноценен модел за публикация и абониране. Конфигурационните параметри включват 60-секунден интервал за поддържане на връзката (keepalive), в съответствие с спецификацията MQTT 3.1.1, както и 5-секунден таймаут за повторно свързване при прекъсване.

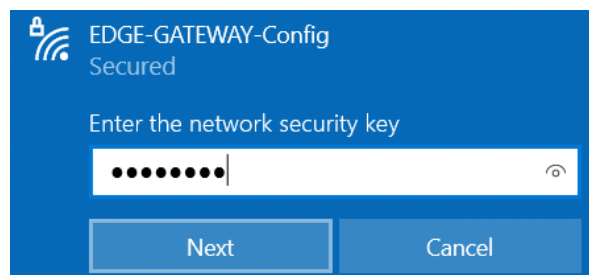
Управлението на качеството на обслужване (QoS) поддържа трите стандартни нива: QoS 0, QoS 1 и QoS 2, с автоматично управление на потвържденията за съобщенията. Осигурено е устойчиво съхранение на критични съобщения в паметта на брокера за по-късно присъединяващи се клиенти.

Сериализацията на данните трансформира декодираните измервания в JSON формат, съвместим със спецификацията RFC 7159, включващ ясни анотации на типовете и вградена информация за единиците. Схемата на JSON обектите включва класификация на типа пакет, идентификация по MAC адрес (IEEE 802), показатели за силата на сигнала (RSSI) и точен времеви отпечатък от таймерната подсистема на ESP32-S3. Публикуването чрез MQTT насочва всички BLE данни към йерархично организирана структура от теми („ble/badges“) с ниво на QoS 1, осигуряващо гарантирана доставка на съобщенията. Всяка операция по публикуване се предхожда от проверка на състоянието на връзката, като при нужда се активира механизъм за възстановяване при мрежови прекъсвания.

8.5.4 Дисплей

Дисплеят на ESP32-S3-BOX-3 е 2,4-инчов LCD панел с резолюция 320 x 240 пиксела (QVGA) и кондензаторен тъчскрийн. Той се управлява през SPI интерфейс и има собствен контролер с вградена памет, което позволява да се използва само малък буфер в ESP32-S3 вместо буфер за пълен кадър. Поддържа частично обновяване на изображението, така че отделни области на екрана могат да се променят без да се засяга целият дисплей, което пести памет, но може да доведе до леко размазване при движещи се елементи.

При конкретната разработка дисплеят се използва с цел конфигуриране на шлюза за комуникация в интернет през Wi-Fi и информироване за статуса на модула. Когато ESP32-S3-BOX-3 се стартира и няма запазени данни за Wi-Fi достъп (идентификатор на точката за достъп и парола), програмното осигуряване активира SoftAP (Soft Access Point) режим. В този режим шлюзът създава собствена безжична мрежа със програмно зададени име и парола (виж Фиг. 8.6).



Фиг. 8.6 Свързване с Wi-Fi AP на EDGE GATEWAY



а)

б)

Фиг. 8.7 Въвеждане на име и парола за достъп до домашната Wi-Fi мрежа: а) Помощна информация на дисплея на шлюза; б) Форма за задаване на името и паролата на домашната Wi-Fi мрежа през браузър

Информацията как да се свърже потребителя към безжичната мрежа на шлюза се показва на дисплея - име на мрежата, IP адрес и парола за достъп (виж Фиг. 8.7а). Потребителят се свързва към тази временна мрежа от телефон, таблет или компютър. За целта през браузър се достъпва точката за достъп на шлюза (IP адрес 192.168.4.1). На този адрес работи уеб сървър, който показва страница с форма за въвеждане на SSID и парола за желаната домашна мрежа (виж Фиг. 8.7б). След като потребителя въведе и потвърди данните, устройството ги записва в своята постоянна памет (NVS), прекратява режима SoftAP, превключва в режим Station и опитва да се свърже към домашната Wi-Fi мрежа. Ако връзката е успешна, шлюзът продължава работа в нормален режим вече като клиент на тази мрежа.

Дисплеят се използва и за визуализиране на статуса на шлюза за всеки един момент от време. Всички съобщения за проблеми или грешки се показват с червен цвят. При нормална работа, статусните съобщения се показват със зелен цвят. Освен статусна информация на дисплея се визуализира и информация за броя на получените от баджовете пакети по тип (телеметрични, локационни и профил на сигурността), както и общия брой получени пакети (виж Фиг. 8.8).



Фиг. 8.8 Информация, визуализирана на дисплея при нормална работа на шлюза

8.5.5 Диагностика и мониторинг

Системата включва изчерпателна диагностична подсистема с изобразяване на конзолата на данните от баджовете във формат от 16 байта на ред, както и ASCII представяне на данните. Конзолната визуализация използва ANSI escape последователности за цветово кодиране на различните типове данни и състояния на системата. Тази възможност за диагностика осигурява наблюдение в реално време на протоколните операции за оптимизация и отстраняване на грешки.

8.6 Бизнес логика

Бизнес логиката отговаря за събиране, буфериране и разпространение на данните от баджовете в реално време. Тя е изградена върху технологичен стек, базиран на Node.js, MQTT и WebSocket. Целта на системата е да осигури бърза комуникация между сензорните устройства и крайните клиенти, като приоритет се поставя върху надеждността, мащабируемостта и устойчивостта на системата.

8.6.1 Технологичен стек

Използваната сървърна платформа е Node.js, избрана поради своята асинхронна I/O архитектура и висока производителност при обработка на събитийно-базирани входи. MQTT протоколът е имплементиран чрез библиотеката `mqtt` – лек клиент за `publish/subscribe` комуникация, който е особено подходящ за системи с ограничени ресурси и нестабилна свързаност. За осъществяване на двупосочна комуникация с уеб-базирани клиенти се използва имплементация на WebSocket протокола чрез модул `ws`, който осигурява постоянна TCP връзка за предаване на данни в реално време. Данните се управляват чрез потребителски модул за буфериране, а конфигурацията се контролира чрез `.env` файл, който се чете чрез модул `dotenv`, позволявайки лесна адаптация към различни среди на изпълнение.

8.6.2 Реализация на системата

Системата е проектирана с цел да приеме, валидира и буферира IoT съобщения от множество устройства, свързани чрез MQTT брокер, и да ги предава в реално време към уеб клиенти посредством протокол WebSocket в реално време. Поддържат се два основни типа съобщения – телеметрични данни и статусна информация, което позволява интеграция със сензорни мрежи, системи за проследяване на обекти и индустриални приложения.

След зареждане на необходимите зависимости и конфигурационни параметри, системата инициализира WebSocket сървър. Всеки нов клиент получава буферирани данни в JSON формат, с което се осигурява синхронизация на състоянието още при установяване на връзката.

Паралелно се конфигурира MQTT клиент, който се свързва с MQTT брокера чрез идентификационни параметри и механизъм за автоматично повторно свързване. Използва се и т.нар. Last Will и Testament (LWT) съобщение, което служи за уведомяване на системата при неочаквано прекъсване на връзката с клиента – важна характеристика за устойчиви IoT решения.

След установяване на връзката с брокера, системата се абонира към предварително дефиниран списък от теми, чрез които се получават данни от устройствата. При всяко ново съобщение, се извършва декодиране и валидация. Поддържат се единствено JSON-съобщения с атрибут `type` със стойност „telemetry“ или „location“, като всяко валидно съобщение се буферира локално и едновременно се препраща към всички активни WebSocket клиенти.

На Фиг. 8.9 са показани данните, които бизнес логиката получава от ESP32-S3-BOX3.

a)

6)

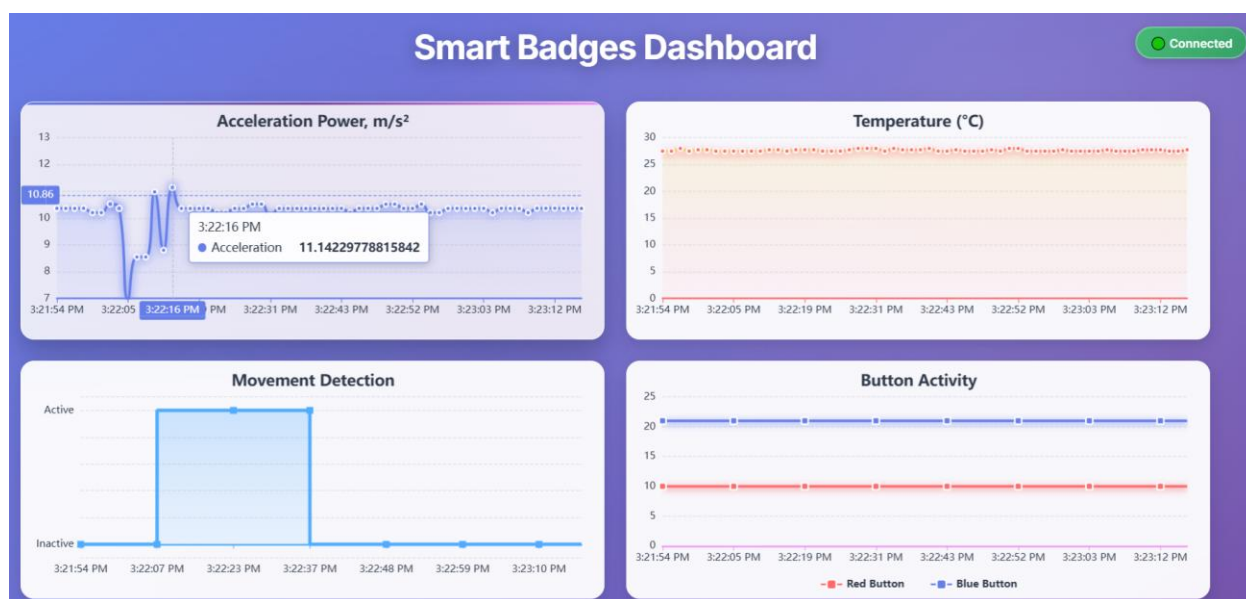
8.7 Визуализация на данните в реално време

Реализирана е функционалност за двупосочна комуникация чрез WebSocket, обработка на получени данни, динамична актуализация на графичния интерфейс и осигуряване на устойчивост при прекъсване на връзката. При успешно установяване на връзка с WebSocket сървър се инициира графичен индикатор, отразяващ състоянието на комуникацията, както и визуално активиране на диаграмите, които ще бъдат обновявани в реално време. В случай на прекъсване,

приложението съдържа механизъм за автоматично повторно свързване, при който броят на опитите и интервалите между тях се контролират адаптивно.

Централна роля във вътрешната логика играе структурата, която агрегира хетерогенни по характер измервания – ускорение, температура, натискания на бутони, детекция на движение и идентификатор на помещение – всички синхронизирани чрез унифицирани времеви марки. Така организираният данни се визуализират чрез библиотеката ECharts. От гледна точка на потребителския интерфейс, динамиката на обновяване на диаграмите се осъществява чрез директно манипулиране на DOM елементи, с което се осигурява мигновена реакция при постъпване на нови данни.

На Фиг. 8.10 е показано как се визуализира информацията от умните баджове.



Фиг. 8.10 Визуализиране на данните от баджовете в реално време

8.8 Запис на данните от баджовете в база данни

Записът на данните от баджовете в база данни се реализира от специално разработено програмно осигуряване. То работи като асинхронна система за събиране, обработка и съхранение на телеметрични и локационни данни в реално време. Изградено е с помощта на технологичен стек, включващ Node.js, комуникационния протокол WebSocket за двупосочна комуникация с отдалечен сървър и базата данни MongoDB като надеждно и мащабируемо хранилище на данни. Управлението на конфигурационни параметри е реализирано чрез dotenv,

което позволява сигурно и гъвкаво управление на чувствителна информация като URL адреси на сървъри и пароли.

Така разработеният софтуер получава, валидира и съхранява времеви потоци от сензорни данни, получавани чрез WebSocket комуникация. Данните се отнасят до физически устройства, идентифицирани чрез своя MAC адреси. След инициализация, системата изпълнява две базови операции:

1. Свързване към база данни MongoDB. Установява се асинхронна връзка с база данни (dementia) чрез MongoDB клиент.
2. Установяване на WebSocket клиентска връзка. WebSocket клиент се свързва към предварително зададен сървър. При успешно отваряне на връзката системата започва да приема съобщения, които се очаква да са във формат JSON с дефинирана структура – тип на съобщението (init или update) и съответните данни.

При получаване на съобщения, съдържащи данни от тип „telemetry“ или „location“, се задейства процес на валидация и съхранение. MAC адресът на устройството се използва като уникален идентификатор, от който се формира име на колекция в MongoDB. Този подход позволява логическо разделяне на данни по източник, което улеснява последващ анализ и извличане, както и анонимизиране на данните.

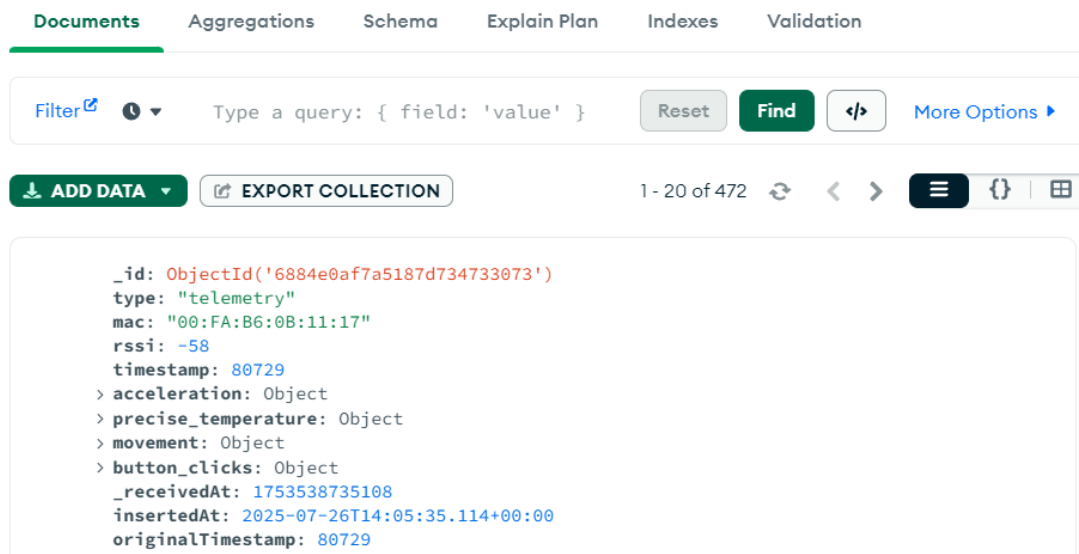
Всяка записана структура съдържа следните ключови елементи:

- оригиналните данни;
- времеви маркер на получаване;
- оригинален времеви маркер от източника, ако такъв е наличен.

В случай на невалидни или непълни данни (например липсващ MAC адрес), съобщението се игнорира и се логва предупреждение.

Системата е проектирана да бъде устойчива на прекъсвания. При загуба на WebSocket връзка автоматично се инициира повторен опит за свързване след кратка пауза. В допълнение, е имплементиран механизъм за „graceful shutdown“ – при прекратяване на процеса чрез сигнал SIGINT, връзката с MongoDB се затваря контролирано, осигурявайки целостта на данните.

На Фиг. 8.11 е показана чрез приложението Compass структурата на база данни с име dementia.



Фиг. 8.11 Съдържание на MongoDB колекция за конкретен бадж

8.9 Визуализация и експортиране на данните от базата данни

С цел визуализиране и експортиране на данните от база данни MongoDB е разработено приложение на Python. То реализира:

- зарежда данните за определен бадж;
- визуализира информацията от акселерометъра, температурата, статуса на бутоните и идентификаторите на стаите, които потребителя е посещавал;
- мащабира всеки един от визуализираните параметри;
- предварителна обработка на данните с цел машинно обучение на модели с за ранно предсказване на деменция.

8.9.1 Описание на програмното осигуряване

Разработено е уеб приложение за визуализация на данни от „умни баджове“, използвани за проследяване на пациенти с деменция. Сървърната част е изградена с Flask, който предоставя REST API за достъп до крайните точки на приложението. Данните се зареждат от база данни MongoDB. Върху тези данни се извършва обработка чрез Pandas и NumPy, като се извличат показатели като ускорение, температура, натискания на бутони и локация. За визуализация се използва библиотека Plotly чрез която се генерират интерактивни графики. Визуализациите се рендират директно в браузъра чрез Plotly.js, а интерфейсът позволява избор на бадж (MAC адрес), задаване на период от време и тестване на връзката с базата.

Целта на приложението е да подпомага наблюдението на пациенти с деменция чрез трансформиране на суровите данни от сензори в разбираема визуална форма и да експортира обработените данни с цел машинно обучение на модели за ранна диагностика на деменция.

Работният процес започва с инициализация на връзката с MongoDB базата данни с име dementia, като се извършва проверка на наличните колекции, които отговарят на различни носими устройства (умни баджове), идентифицирани чрез техните MAC адреси. Потребителят има възможност да избере конкретно устройство, както и времеви интервал, за който да се извлекат данни. Това позволява гъвкавост при анализа и осигурява прецизен контрол върху обхвата на визуализираната информация (виж Фиг. 8.12).

Smart Badge Data Visualization
Визуализация на данни от умни баджове за проследяване на пациенти с деменция

Филтри за данни

MAC адрес на бадж (колекция): 00fab60b1117

Начална дата: 07/26/2025 03:44 PM

Крайна дата: 07/27/2025 03:44 PM

Заредете данните Тест на връзката Експорт до HDF5 файл

Заредени са 174 записа

Фиг. 8.12 Филтриране на данните от базата данни

Отчитат се показанията от акселерометъра, които илюстрират физическата активност на пациента, температурните стойности, които дават информация за физиологичното състояние, броят на натисканията на бутоните като индикатор за взаимодействие с устройството, както и идентификаторите на помещения, чрез които се проследява местоположението на клиентите. Всяка графика е интерактивна, като позволява мащабиране, панорама и експорт, което улеснява задълбочения анализ.

Потребителският интерфейс е реализиран така, че да бъде едновременно интуитивен и функционален. Визуалните компоненти осигуряват ясна обратна връзка за състоянието на системата, като показват процеса на зареждане, броя на наличните записи и възможни грешки при връзката или обработката. Чрез това програмното осигуряване действа като надежден посредник между големите обеми сурови данни и крайния потребител, улеснявайки тяхната интерпретация и

подпомагайки процеса на вземане на решения в контекста на медицинското наблюдение.

На Фиг. 8.13 е показан начина на визуализиране на основните параметри, на базата на които се реализира предсказване на деменция.



Фиг. 8.13 Визуализиране на данните от акселерометъра, температурата и състоянието на бутоните

8.9.2 Предварителна обработка на данните и експортиране

Данните от умните баджове се обработват с цел да бъдат подходящи за машинно обучение.

На първо място, всички данни се синхронизират по време. За целта всеки пакет с данни има времеви печат, който позволява данните от различните сензори да се подравнят, за да се сравняват коректно.

Следва почистването на данните - необходимо е премахването на шума и аномалните стойности в данните, които могат да възникнат поради грешки в сензорите. При данните от акселерометъра се прилага филтриране с нискочестотен филтър, за да се изолират значимите движения и да се потисне високочестотният шум.

Третият етап е свързан с нормализацията на данните. Поради различните мащаби на измерваните величини, например ускорение в метри за секунда на квадрат и температура в градуси по Целзий, стойностите трябва да бъдат приведени в съпоставим вид. Това се реализира чрез нормализация в интервала от нула до едно или чрез стандартизация, при която данните се трансформират така, че да имат средна стойност нула и стандартно отклонение единица.

Данните за идентификаторите на стаите са кодирани в базата данни като цели числа, като всяко число отговаря на конкретна стая, в която пациентът е пребивавал в определен момент. Така се отразява последователността от стаи, които описват модела на движение на пациента. За целите на машинното обучение тази форма на данни не е директно приложима. Алгоритмите за предсказване и класификация не могат да работят с номинални категории, представени единствено като числа, тъй като числовата стойност не носи количествена интерпретация. Необходимо е трансформиране на идентификаторите в подходящо представяне, което да улесни моделирането на зависимости и закономерности. Един от често използваните подходи е да се използва едноразрядното кодиране (one-hot encoding), при което всяка стая се преобразува във вектор с дължина, равна на броя различни стаи, а позицията, съответстваща на конкретния идентификатор, приема стойност единица (всички останали остават нули). Това позволява на модела да възприема стаите като отделни категории без скрита числова връзка между тях.

В допълнение към основната обработка е препоръчително да се извърши и извличане на характеристики. От данните на акселерометъра се формират показатели като средни стойности за определен времеви прозорец, стандартно отклонение, енергия на сигнала, спектрални характеристики, както и стойността на ускорението. При данните за местоположението се анализира поредността на преминаване през различните стаи, времето на престой и честотата на преходите.

Всички данни могат да бъдат експортирани в CSV файл - редовете представляват отделни наблюдения, а колоните съдържат съответните характеристики. При дълбоко обучение и анализ на времеви редове, където е важно да се запази структурата на последователностите, приложението позволява експорт във формат Hierarchical Data Format version 5 (HDF5). HDF5 е бинарен файлов формат, който е подходящ за съхранение на големи обеми от данни. Този формат се характеризира се с йерархична структура, наподобяваща файловата система, в която „групи“ играят ролята на папки, а „дейтасети“ – на файлове със самите данни. Този формат позволява ефективно съхранение на масиви с различни размерности, както и добавяне на метаданни под формата на атрибути. Основните му предимства са високата компресия, бърз достъп до част от данните без зареждане на целия файл и добрата интеграция с Python чрез библиотеки h5py или pandas.

8.10 Заключение

Предложена е концептуална рамка за система за превенция на деменция в домашни условия. Проектирана, имплементирана и тествана е архитектурата на предлаганата система. Системата се характеризира с ниска цена, висока надеждност, а разпределената и архитектура позволява да бъде използвана при голям брой на клиентите. Данните, на базата на които се анализира наличието на деменция се получават от умни баджове на фирма Kontakt.io. Системата позволява както визуализиране на данните от умните баджове в реално време, така и записът има в база данни MongoDB с цел обработка в offline режим. По този начин се предоставя възможност за обучение на невронните модели, които да се използват в последствие за ранно разпознаване на деменция.

ГЛАВА 9: БЪДЕЩИ НАПРАВЛЕНИЯ И РАЗВИТИЕ

Системите за навигация и локализация се развиват поради новите технологични и научни разработки. Методите, алгоритмите и системите в монографията - RSTS алгоритъм за опростяване на GPS траектории, BIM-базирани интериорни навигационни решения, RFID/NFC системи, BLE подходи за локализация - са основа за нови навигационни технологии. При анализа на новите направления в областта на навигацията и локализацията се изисква разглеждане на технологичните възможности и научните предизвикателства пред общността.

9.1 Тенденции в технологичното развитие

Съвременната технологична еволюция в областта на навигационните системи се характеризира с радикални промени във всички компоненти на системната архитектура. Тези промени засягат не само отделните технологични елементи, но и начините на тяхната интеграция, създавайки качествено нови възможности за точност, ефективност и функционалност.

Навигационните системи се развиват с промени в компонентите на архитектурата. Развитието на съвременните сензорните технологии оказва влияние на навигационните системи. Микроелектромеханичните системи (MEMS) достигат миниатюризация и точност. Например, квантовите акселерометри и жирокопи се характеризират с точност от порядъка на нанометри (Templier и колектив, 2022; Salducci и колектив, 2024). Тези устройства използват квантовомеханични ефекти за измерване на ускорението и ъгловата скорост с точност, която превъзхожда MEMS сензорите с няколко порядъка. Прилагането на тези технологии ще позволи създаването на инерциални навигационни системи, които поддържат точност за дълги времеви интервали без корекция от външни референции.

Лидарните технологии се изменят с твърдотелни решения и фотонни интегрални схеми (Hu и колектив, 2024; Lukashchuk и колектив, 2024). Фотонните лидари установяват резолюция от милиметри при разстояния от стотици метри, като намаляват размера и енергопотреблението с порядъци (Zhang и колектив, 2022). Интеграцията на AI-ускорители в лидарните сензори осигурява обработка на облаци от точки в реално време, което води до възможности за автономна навигация в сложни среди.

Спектралните камери се развиват и разширяват възможностите за визуално базирана навигация отвъд видимия спектър. Хипер-спектралните камери осигуряват анализ на материалните свойства на обектите в околната среда, което обуславя навигация на база разпознаване на материали. Този подход се прилага в условия на ограничена видимост или при разграничаване на обекти с еднаква геометрия но различни материални характеристики.

Инерциалните измервателни устройства интегрират акселерометри и жирокопи, магнитометри с квантова точност, барометри с резолюция от паскали и термодатчици за компенсация на температурните дрейфове. Системите-на-чип (SoC) осигуряват интеграция на сензорните елементи, аналогово-цифрови преобразуватели, сигнални процесори и AI ускорители в една силициева подложка, което води до получаване на автономни сензори със способност за обработка на данни.

Искусственият интелект и машинното обучение променят подходите за навигация и локализация. Трансформаторните архитектури, разработени за обработка на естествен език, се прилагат в пространственото моделиране и прогнозирането на траектории. Самонаблюдаващите механизми дават възможност на системите да фокусират вниманието върху пространствените характеристики при вземане на навигационни решения.

Невросимволичните подходи съчетават възможностите на дълбокото обучение – разпознаване на образи – с логическото мислене от символния изкуствен интелект. Тази комбинация дава възможност за създаване на навигационни системи, които се учат от данни и могат да обясняват и обосновават решенията си пред потребителя. Каузалното обучение им помага да разбират връзките между събития и обекти в пространството, което подобрява способността им да се адаптират и прилагат знанията си в нови ситуации.

Федеративното обучение дава решение на важен проблем – как да се споделя знание между навигационни устройства без застрашаване на информацията за потребителите (Li и колектив, 2020; Dritsas и Trigka, 2025). При този подход локалните модели се обучават на устройствата, като към централния сървър се изпращат обобщени данни – параметри на модела – а не личните данни. Така се изграждат глобални навигационни модели, които се възползват от опита и поведението на потребители без нарушаване на поверителността им.

Мобилните технологии при 5G и 6G мрежи дават нови възможности на навигационните системи. Поради ниската латентност на 5G мрежите (под 1 милисекунда), системите реагират в реално време.

Мрежово нарязване (network slicing) при 5G мрежите позволява физическата мрежова инфраструктура да се раздели на виртуални „срезове“ (slices), оптимизирани за различни услуги. Технологията дава мрежови ресурси и свързаност за навигационни приложения като автономни превозни средства или спешни служби.

Технологиите за масивен Multiple-Input Multiple-Output (MIMO) в 5G и 6G мрежи дават възможност за определяне на местоположението чрез анализ на посоката и характеристиките на радиосигналите. Чрез техниките за насочено излъчване (beamforming), навигационните системи постигат точност до сантиметри в градски условия. Интегрирането на терагерцови (THz) честоти в 6G мрежите дава прецизност, тъй като високочестотните сигнали осигуряват пространствена информация за заобикалящата среда.

Интернетът на нещата се развива до Интернет на всичко (Internet of Everything), където физическите обекти участват в навигационния процес. Материалите, които имат вградени сензори и комуникационни технологии, превръщат сградите и инфраструктурата в навигационни среди. Тротоарите могат да осигуряват хаптична обратна връзка, която подпомага ориентацията на хора със зрителни увреждания. Стените в сгради могат да пренасочват Wi-Fi сигнали за подобряване на прецизността на локализацията в закрити пространства.

Дигиталните близнаци (Digital Twins) са виртуални модели на обекти, системи или процеси, които се актуализират с данни от сензори ([Grieves, 2014](#); [Tao и колектив, 2018](#)). Виртуалното копие отразява поведението, състоянието и средата на обекта, за да се симулират сценарии, предвиждат проблеми и оптимизират работата. Дигиталните близнаци на градове и сгради дават възможности за симулация и оптимизация на навигационни алгоритми в условия близо до реалността ([Shahat и колектив, 2021](#)). Виртуалните копия, запазвани с данни от сензори, дават възможност да се предвиждат навигационни условия и да се планират маршрути. Чрез интеграцията между физическия и цифровия свят – посредством кибер-физични системи – се създават начини за взаимодействие между човека и

навигационните технологии, като се повишава ефективността, адаптивността и персонализацията.

9.2 Подобрения на предложените методи и алгоритми

Системите и алгоритмите, представени в монографията, осигуряват основа за създаване на сложни и адаптивни навигационни решения. Разширенията на методите включват технологични подобрения и подходи за взаимодействие с потребителя и интеграция с нови технологии.

RSTS алгоритмът за опростяване на GPS траектории може да се разшири с използване на машинно обучение за адаптиране на параметрите според контекста на движение. Интеграцията с данни от инерциални сензори и камери може да подобри точността на опростяването в области с ограничено GPS покритие. Прогнозирането на траекторията чрез рекурентни невронни мрежи може да позволи планиране на ресурсите и зареждането на карти.

BIM-базираните навигационни системи могат да се подобрят чрез интеграция на данни в реално време от IoT сензори за създаване дигитални близнаци на сгради. Обогащането на BIM моделите с информация за достъпност, заетост на помещенията и качество въздуха дава възможност за планиране на маршрути според нуждите на потребностите. Интеграцията с технологии за допълнена реалност дава възможност за наслояване на виртуална навигационна информация върху физическото пространство на сградата.

Системите на база RFID и NFC технологии се надграждат с блокчейн за осигуряване сигурност и автентичност на навигационните данни. Свързването с мобилни приложения за плащания дава възможност за създаване на услуги, при които навигацията се комбинира с транзакции – при преминаване през транспортни системи или паркинги. Криптографските протоколи дават защита на личната информация и позволяват обслужване според нуждите на потребителя.

BLE алгоритмите за локализация се усъвършенстват чрез прилагане на машинно обучение за моделиране на радио средата. Това дава по-точно определяне на местоположението в сложни и динамични интериорни условия. Чрез федеративно обучение устройствата могат да обменят знания за радиочестотните характеристики на заобикалящата среда без споделяне на лични данни – запазвайки поверителността на потребителите. Интеграцията с меш мрежи дава възможност за създаване на децентрализирана и самовъзстановяваща се

навигационна инфраструктура, която се адаптира към промени в физическото пространство, като ремонти или промени в интериора.

Допълнената (AR) и виртуална реалност (VR) могат да бъдат използвани при съвременните навигационните системи. Смесената реалност (MR), която обединява елементи от AR и VR, осигурява база за създаване на интерактивни навигационни преживявания, в които виртуални обекти се вписват в реалната среда. Холографските дисплеи могат да разширят възможностите, като предоставят триизмерни навигационни указания, които се проектират в зрителното поле на потребителя, създавайки усещане за навигация „в пространството“.

Пространственото изчисляване (Spatial Computing) може да се използва за трансформиране на заобикалящата среда в интерактивен интерфейс чрез който потребителите взаимодействат с навигационните системи. Това включва използването на жестове, например задаване на посоки или посочване на обекти. Проследяването на движението на очите добавя ново измерение за управление и дава възможност на потребителите да контролират системата с поглед. Технологията може да се използва за хора с ограничена подвижност и осигурява достъпен начин за навигация.

Невроинтерфейсите представляват граница на взаимодействието човек-машина в навигационните системи. Мозъчно-компютърните интерфейси могат да интерпретират намерения за навигация от мозъчните сигнали и позволяват управление на системата чрез мисли. Този подход може да се използва за хора с тежки увреждания, които не могат да използват традиционните интерфейси.

Мултимодалните интерфейси се развиват и включват разпознаване на емоции и обработка на чувствата. Тези системи могат да разбират емоционалното състояние на потребителя, като анализират гласа, физиологичните сигнали или изражението на лицето. В зависимост от това навигационната система може да променя начина, по който дава инструкции. Ако засече, че потребителят е стресиран или объркан, тя може да предложи подробни обяснения или лесен маршрут.

Социалните аспекти на навигацията се развиват към системи с колективен интелект, които използват знанията и опита на общността за подобряване на навигационните услуги. Социалните алгоритми могат да препоръчват маршрути на база предпочитанията на хора със сходни профили и интереси.

Интелектът на рояка (Swarm Intelligence) осигурява възможност на групи потребители да координират движението си за намаляване на задръстванията и повишаване ефективността на придвижването. Този подход се прилага при организми като мравки, пчели, риби и птици, които координират поведението си чрез локални правила и обмен на информация.

9.3 Научни и технологични предизвикателства

Навигационните системи с изкуствен интелект в бъдеще ще зависят от технологичния прогрес и от решаването на основните проблеми в тази област. Тези проблеми продължават да се отразяват негативно на точността, надеждността и приложимостта на съвременните решения.

Един от основните проблеми при локализация е работата в сложна градска среда. Многопътното разпространение на сигналите причинява интерференция. Така се променят параметрите на GPS сигналите, което затруднява тяхното предвиждане. Необходимо е създаване на адаптивни алгоритми. Тези алгоритми трябва да разделят директните и отразените сигнали в реално време. Това изисква провеждането на изследвания в две области: цифрова обработка на сигнали и статистическа теория за оценка.

Интериорните навигационни системи се подобряват с квантови сензори като акселерометри, магнитометри и барометри. При подобни системи ще има проблеми, свързани с декохерентността и стабилизацията в реални условия. Декохерентност означава загуба квантови свойства (суперпозиция и заплитане) при контакт с външната среда. Използването на квантови сензори осигурява възможност за по-висока точност при локализация при сравнение с класическите методи.

Проблемът при локализацията в сгради е отразяването и затихването на радиовълните. Проектантите на навигационни системи трябва да разработят математически модели, които описват взаимодействието на радиосигналите с материалите и хората в сградите. Тези модели трябва да отчитат както геометрията на помещенията, така и електромагнитните свойства на материалите. Чрез машинно обучение могат да се създадат адаптивни модели. Проблемът е, че тяхното обучение ще изискват големи количества данни. Тези данни трябва се събират при различни интериорни среди и времеви интервали.

.

Проблемът с мащабируемостта на навигационните системи се увеличава с увеличаването броя на свързаните устройства. Алгоритми, които работят за хиляди потребители, може да не са подходящи за милиони устройства. За поддържане висока производителност при голям брой потребители се изисква разработване на разпределени алгоритми. Това изисква изследвания в областта на разпределените системи и паралелните изчисления.

Енергийната ефективност остава предизвикателство за носими и вградени устройства. За поддържане добра точност при минимално енергопотребление са необходими методи за оптимизиране работата на сензорите. Техники като приблизителни изчисления и вероятностни изчисления дават възможност за намаляване консумацията на енергия, като се допуска загуба в точността.

Технологиите за проследяване на хора правят трудно гарантирането на поверителността и сигурността личните данни. За създаване на навигационни услуги без разкриване точното местоположение на потребителите се изисква разработване на алгоритми с висока поверителност. Това изисква изследвания в областта на криптографията. Вместо алгоритмите да работят с личните данни на клиентите е възможно те да работят с шифрираните данни. Това е възможно ако се използва хомоморфно шифриране. Практическата приложимост на хомоморфното шифриране в навигационните системи все още е техническо предизвикателство.

Възможностите за интердисциплинарни изследвания в областта навигацията са широки. Свързването с неврологичните изследвания помага да се разбере как човешкият мозък обработва пространствената информация, което може да послужи за създаване интуитивни навигационни интерфейси. Изследванията върху хипокампуса и енториналната кора разкриват как се формира пространствената памет и как хората се ориентират, а тези механизми се прилагат като модел за изграждане изкуствени навигационни системи.

Точната локализация на хора в големи сгради изисква интердисциплинарен подход. Решението на този проблем предполага комбиниране на технологии с познания от социологията и антропологията. Например, ориентирането в пространството варира при различните културни групи.

Зависимостта на хората от технологиите е проблем при навигация. Терминът „GPS-деменция“ се свързва с влошени умения за ориентиране поради продължителна

употреба на навигационни системи. Необходими са изследвания в областта на когнитивната психология. Целта е намиране на технологии за навигация, които не засягат естествените способности на хората.

При разработването на навигационни системи е необходимо да се отчитат регулаторните рамки и стандарти. Съвместимостта между навигационните системи на различни производители е важна и изисква използването на отворени стандарти. Регулаторните рамки трябва да решат два проблема: как да стимулират иновациите и как да защитят обществен интерес (равен достъп до технологии, защита личните данни).

Използването на машинно обучение при системите за навигация предполага някои етични предизвикателства. Ако системите са обучени със социално небалансирани данни те могат да имат специфични пристрастия. Например, алгоритмите за планиране на маршрути могат да избягват някои квартали.

Социалните ефекти, породени от навигационните технологии изискват изследвания и планиране. Промените в градската среда поради внедряването на навигационни системи обуславят последствия за социалните взаимоотношения и културното наследство. Навигационните системи се проектират за подкрепа човешкото развитие и благосъстояние.

Бъдещето на системите за навигация и локализация предполага използването на интердисциплинарен подход при тяхното проектиране. Имплементирането на новите технологии трябва да е съобразено със защита на личните данни и отчитане на социалните влияния на тези системи. Методите и алгоритмите в тази монография са добра отправна точка за създаване на интелигентни и точни навигационни системи в бъдещите умни градове. За да се реализира пълния потенциал на новите технологии ще са необходими продължителни усилия на специалисти от различни специалности.

Aumepamypa

- Aharchi, M., & Kbir, M. H. A. (2021). Localization and navigation system for blind persons using stereo vision and a GIS. In *WITS 2020: Proceedings of the 6th International Conference on Wireless Technologies, Embedded, and Intelligent Systems* (pp. 365-376). Singapore: Springer Singapore.
- Alfaresi, B., Barlian, T., Ardianto, F., & Hurairah, M. (2020). Path loss propagation evaluation u modelling based ecc-model in lowlu area on 1800 mhz frequency. *Journal of Robotics u Control (JRC)*, 1(5), 167-172.
- Aljazzar, H. u Leue, S., (2011). K*: A heuristic search algorithm for finding the k shortest paths. *Artif. Intell.*, 175(18), 2129-2154.
- Barbeau, S., Labrador, M. A., Perez, A., Winters, P., Georggi, N., Aguilar, D., & Perez, R. (2008, September). Dynamic management of real-time location data on GPS-enabled mobile phones. In *2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services u Technologies* (pp. 343-348). IEEE.
- Bijlani, N., Maldonado, O. M., Nilforooshan, R., Barnaghi, P., Kouchaki, S., & CR&T Group. (2024). Utilizing graph neural networks for adverse health detection and personalized decision making in sensor-based remote monitoring for dementia care. *Computers in biology and medicine*, 183, 109287.
- Brena, R. F., García-Vázquez, J. P., Galván-Tejada, C. E., Muñoz-Rodriguez, D., Vargas-Rosales, C., & Fangmeyer, J. (2017). Evolution of indoor positioning technologies: A survey. *Journal of Sensors*, 2017.
- Brown, R. G., & Hwang, P. Y. C. (2012). *Introduction to Ruom Signals u Applied Kalman Filtering with Matlab Exercises* (4th ed.). John Wiley & Sons.
- Cannizzaro, D., Zafiri, M., Jahier Pagliari, D., Patt , E., Macii, E., Poncino, M., & Acquaviva, A. (2019). A comparison analysis of BLE-based algorithms for localization in industrial environments. *Electronics*, 9(1), 44.
- Cantón Paterna, V., Calveras Auge, A., Paradells Aspas, J., & Perez Bullones, M. A. (2017). A bluetooth low energy indoor positioning system with channel diversity, weighted trilateration u kalman filtering. *Sensors*, 17(12), 2927.
- Chaudhary, A., Gupta, H. P., Shukla, K. K., & Dutta, T. (2020). Sensor signals-based early dementia detection system using travel pattern classification. *IEEE Sensors Journal*, 20(23), 14474-14481.
- Chen, L., Ahriz, I., & Le Ruyet, D. (2020). AoA-aware probabilistic indoor location fingerprinting using channel state information. *IEEE internet of things journal*, 7(11), 10868-10883.

- Chen, X., Liu, D., Luo, J., Chen, T., Zhang, G., Rong, X., & Li, Y. (2022). Realization of indoor and outdoor localization and navigation for quadruped robots. *Procedia Computer Science*, 209, 84-92.
- Chen, Y., Jiang, K., Zheng, Y., Li, C., & Yu, N. (2009). Trajectory simplification method for location-based social networking services. In *Proceedings of the 2009 international workshop on location based social networks* (pp. 33-40).
- Cheung, C. K., & Shi, W. (2006). Positional error modeling for line simplification based on automatic shape similarity analysis in GIS. *Computers & geosciences*, 32(4), 462-475.
- Crowdconnected. (2024). Indoor positioning technology review 2024. Retrieved from <https://www.crowdconnected.com/blog/indoor-positioning-tech-update-2025/>
- Dah-Jing Jwo, Chung-Min Huang (2007). An adaptive fuzzy strong tracking Kalman filter for GPS/INS navigation. In *IECON 2007-33rd Annual Conference of the IEEE Industrial Electronics Society* (pp. 2266-2271). IEEE.
- Dey, A. K. (2001). Understanding π using context. *Personal and Ubiquitous Computing*, 5(1), 4-7.
- Diakité, A. A., & Zlatanova, S. (2018). Spatial subdivision of complex indoor environments for 3D indoor navigation. *International Journal of Geographical Information Science*, 32(2), 213-235.
- Dijkstra, E.W., (1959). A note on two problems in connexion with graphs. *Numer. Math.*, 1, 269–271.
- Dinh-Van, N., Nashashibi, F., Thanh-Huong, N., & Castelli, E. (2017, March). Indoor Intelligent Vehicle localization using WiFi received signal strength indicator. In *2017 IEEE MTT-S international conference on microwaves for intelligent mobility (ICMIM)* (pp. 33-36). IEEE.
- Doer, C., & Trommer, G.F. (2020). An EKF Based Approach to Radar Inertial Odometry. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 152-159.
- Dokeroglu, T., Sevinc, E., Kucukyilmaz, T., & Cosar, A. (2019). A survey on new generation metaheuristic algorithms. *Computers & Industrial Engineering*, 137, 106040.
- Douglas, D. H., & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2), 112-122.
- Dritsas, E., & Trigka, M. (2025). Federated Learning for IoT: A Survey of Techniques, Challenges, and Applications. *Journal of Sensor and Actuator Networks*, 14(1), 9.
- Dubois, B., Hampel, H., Feldman, H. H., Scheltens, P., Aisen, P., Irieu, S., ... & Washington, D. C. (2016). Preclinical Alzheimer's disease: definition, natural history, and diagnostic criteria. *Alzheimer's & Dementia*, 12(3), 292-323.

- Eastman, C., Teicholz, P., Sacks, R., & Liston, K. (2018). *BIM hubook: A guide to building information modeling for owners, managers, designers, engineers u contractors*. John Wiley & Sons.
- Elsanhoury, M., Koljonen, J., Välisuo, P., Elmusrati, M., & Kuusniemi, H. (2021). Survey on recent advances in integrated GNSS towards seamless navigation using multi-sensor fusion technology. *Proceedings of the 34th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2021*, 2754-2765. <https://doi.org/10.33012/2021.17961>
- Enge, P. K. (1994). The global positioning system: Signals, measurements, и performance. *International Journal of Wireless Information Networks*, 1(2), 83-105.
- Eppstein, D., (1999). Finding the k Shortest Paths. *SIAM J. Comput.*, 28, 652–673.
- Fallah, N., Apostolopoulos, I., Bekris, K., & Folmer, E. (2013). Indoor human navigation systems: A survey. *Interacting with Computers*, 25(1), 21-33.
- Faragher, R., & Harle, R. (2015). Location fingerprinting with bluetooth low energy beacons. *IEEE journal on Selected Areas in Communications*, 33(11), 2418-2428.
- Farsi, M., Elhosseini, M. A., Badawy, M., Ali, H. A., & Eldin, H. Z. (2019). Deployment techniques in wireless sensor networks, coverage и connectivity: A survey. *IEEE access*, 7, 28940-28954.
- Fernies, H., Costa, P., Filipe, V., Paredes, H., & Barroso, J. (2019). A review of assistive spatial orientation и navigation technologies for the visually impaired. *Universal Access in the Information Society*, 18(2), 155-168. <https://doi.org/10.1007/s10209-017-0570-8>
- Fernio, N., McMeekin, D. A., & Murray, I. (2023). Route planning methods in indoor navigation tools for vision impaired persons: a systematic review. *Disability и Rehabilitation: Assistive Technology*, 18(6), 763-782.
- Ford, E., Milne, R., & Curlewis, K. (2023). Ethical issues when using digital biomarkers and artificial intelligence for the early detection of dementia. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(3), e1492.
- Gao, Y., Yuan, J., Jiang, J., Sun, Q., & Zhang, X. (2023). VIDO: A Robust и Consistent Monocular Visual-Inertial-Depth Odometry. *IEEE Transactions on Intelligent Transportation Systems*, 24(3), 2976-2992.
- Ghasemieh, A., & Kashef, R. (2024). Advanced Monocular Outdoor Pose Estimation in Autonomous Systems: Leveraging Optical Flow, Depth Estimation, и Semantic Segmentation with Dynamic Object Removal. *Sensors*, 24(24). <https://doi.org/10.3390/s24248040>
- Giudice, N. A., Guenther, B. A., Kaplan, T. M., Anderson, S. M., Knuesel, R. J., & Cioffi, J. F. (2020). Use of an indoor navigation system by sighted and blind travelers: Performance similarities across visual status and age. *ACM Transactions on Accessible Computing (TACCESS)*, 13(3), 1-27.

- Goldberg, A.V. и Radzik, T., (1993). A heuristic improvement of the Bellman-Ford algorithm. *Appl. Math. Lett.*, 6.
- Grieves, M. (2014). Digital Twin: Manufacturing Excellence through Virtual Factory Replication. *Digital Manufacturing*, 1(1), 1-7.
- Groves, P. D. (2013). *Principles of GNSS, inertial, u multisensor integrated navigation systems* (2nd ed.). Artech House.
- Grzechca, D. E., Pelczar, P., & Chruszczyk, L. (2016). Analysis of object location accuracy for iBeacon technology based on the RSSI path loss model и fingerprint map. *International Journal of Electronics u Telecommunications*.
- Guo, G., Chen, R., Yan, K., Li, P., Yuan, L., & Chen, L. (2023). Multichannel и multi-RSS based BLE range estimation for indoor tracking of commercial smartphones. *IEEE Sensors Journal*, 23(24), 30728-30738.
- Hamedani M.; Caneva S.; Mancardi G.L.; Ali P.A.; Fiaschi P.; Massa F.; Schenone A.; Pardini M. (2024). Toward Quantitative Neurology: Sensors to Assess Motor Deficits in Dementia. *Journal of Alzheimer's Disease*, 101(4), 1083-1106. <https://doi.org/10.3233/JAD-240559>
- Hart, P.E., Nilsson, N.J., и Raphael, B., (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.*, 4(2), 100-107.
- Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, u prediction* (Vol. 2, pp. 1-758). New York: springer.
- He, W., Ho, P. H., & Tapolcai, J. (2017). Beacon deployment for unambiguous positioning. *IEEE Internet of Things Journal*, 4(5), 1370-1379.
- Hensel, S., B. Marinov, M., & Obert, M. (2022). 3d lidar based slam system evaluation with low-cost real-time kinematics gps solution. *Computation*, 10(9), 154.
- Hershberger, J. E., & Snoeyink, J. (1992). Speeding up the Douglas-Peucker line-simplification algorithm. in Proc. of the 5th Int. Symp. on Spatial Data Handling, vol. 1, 1992, pp. 134-143.
- Hershberger, J. E., & Snoeyink, J. (1992). Speeding up the Douglas-Peucker line simplification algorithm, In: Proceedings of the 5th Int. Symposium on Spatial Data Handling, vol. 1, pp. 134-143.
- Hershberger, J., Maxel, M., и Sur, S., (2007). Finding the k Shortest Simple Paths: A New Algorithm и its Implementation. *ACM Trans. Algorithms*, 3(4), 45–56.
- Hofmann-Wellenhof, B., Lichtenegger, H., & Wasle, E. (2007). *GNSS–Global Navigation Satellite Systems: GPS, GLONASS, Galileo, u more*. Springer-Verlag.
- Hopfield, H. S. (1969). Two-quartic tropospheric refractivity profile for correcting satellite data. *Journal of Geophysical Research*, 74(18), 4487-4499. <https://doi.org/10.1029/JC074i018p04487>

- Hou, X., & Bergmann, J. (2022). HeadSLAM: Pedestrian SLAM with head-mounted sensors. *Sensors*, 22(4), 1593.
- Hu, C., Chen, W., Chen, Y., & Liu, D. (2009). Adaptive Kalman filtering for vehicle navigation. *Positioning*, 1(4).
- Hu, M., Pang, Y., & Gao, L. (2023). Advances in silicon-based integrated LiDAR. *Sensors*, 23(13), 5920.
- Ienca, M., Schneble, C., Kressig, R. W., & Elger, B. (2018). Digital health u the ethics of Alzheimer's disease research: The emerging role of digital biomarkers. *Journal of Alzheimer's Disease*, 67(3), 835-846.
- Irwansyah, A. S., Heryadi, B., Dewi, D. K., Saputra, R. P., & Abidin, Z. (2024). ROS-based multi-sensor integrated localization system for cost-effective and accurate indoor navigation system. *International Journal of Intelligent Robotics and Applications*, 1-19.
- Jack Jr, C. R., Bennett, D. A., Blennow, K., Carrillo, M. C., Dunn, B., Haeberlein, S. B., ... & Silverberg, N. (2018). NIA-AA research framework: toward a biological definition of Alzheimer's disease. *Alzheimer's & dementia*, 14(4), 535-562.
- Jamshed, M., Shahzad, A., Riaz, F., & Kim, K. (2024). Exploring inertial sensor-based balance biomarkers for early detection of mild cognitive impairment. *Scientific Reports*, 14(1), 9829.
- Jeon, K. E., She, J., Soonsawad, P., & Ng, P. C. (2018). Ble beacons for internet of things applications: Survey, challenges, u opportunities. *IEEE Internet of Things Journal*, 5(2), 811-828.
- Ji, H., Xie, L., Wang, C., Yin, Y., & Lu, S. (2015). CrowdSensing: A crowd-sourcing based indoor navigation using RFID-based delay tolerant network. *Journal of Network and Computer Applications*, 52, 79-89.
- Jonell P.; Moëll B.; Håkansson K.; Henter G.E.; Kuchurenko T.; Mikheeva O.; Hagman G.; Holleman J.; Kivipelto M.; Kjellström H.; Gustafson J.; Beskow J. (2021). Multimodal Capture of Patient Behaviour for Improved Detection of Early Dementia: Clinical Feasibility u Preliminary Results. *Frontiers in Computer Science*, 3(), -. <https://doi.org/10.3389/fcomp.2021.642633>
- Kang, H. K., & Li, K. J. (2017). A standard indoor spatial data model—OGC IndoorGML and implementation approaches. *ISPRS International Journal of Geo-Information*, 6(4), 116.
- Kaplan, E. D., & Hegarty, C. J. (Eds.). (2017). *Understuing GPS/GNSS: Principles u applications* (3rd ed.). Artech House.
- Kar, K., Datta, S., Pal, M., & Ghatak, R. (2016, January). Motley Keenan model of in-building coverage analysis of IEEE 802.11 n WLAN signal in electronics u communication engineering department of National Institute of Technology Durgapur. In *2016 international conference on microelectronics, computing u communications (MicroCom)* (pp. 1-6). IEEE.

- Katoh, N., и коллектив, (1982). An efficient algorithm for K shortest simple paths. *Networks*, 12(4), 411–427.
- Kaye, J., Mattek, N., Dodge, H. H., Campbell, I., Hayes, T., Austin, D., ... & Pavel, M. (2014). Unobtrusive measurement of daily computer use to detect mild cognitive impairment. *Alzheimer's & Dementia*, 10(1), 10-17.
- Kelly, A. (1994). A 3d space formulation of a navigation kalman filter for autonomous vehicles. *The Robotics Institute, Carnegie Mellon University*.
- Khan, D., Cheng, Z., Uchiyama, H., Ali, S., Asshad, M., & Kiyokawa, K. (2022). Recent advances in vision-based indoor navigation: A systematic literature review. *Computers & Graphics*, 104, 24-45.
- Kim, I. J. (2025). Recent advancements in indoor electronic travel aids for the blind or visually impaired: a comprehensive review of technologies и implementations. *Universal Access in the Information Society*, 24(1), 173-193.
- Kim, J., Cheon, S., & Lim, J. (2022). IoT-based unobtrusive physical activity monitoring system for predicting dementia. *Ieee Access*, 10, 26078-26089.
- Klobuchar, J. A. (2007). Ionospheric time-delay algorithm for single-frequency GPS users. *IEEE Transactions on aerospace and electronic systems*, (3), 325-331.
- Kulkarni, A., & Beulet, P. (2023). Sensor-Based Adaptive Estimation in a Hybrid Environment Employing State Estimator Filters. *Intelligent Automation & Soft Computing*, 37(1).
- Kumar, N., Ferbin, F. J., Doss, A. S. A., & Thao, N. G. M. (2025, March). Hybrid Perception in Autonomous Mobile Robots: Integrating 2D and 3D SLAM with ROS2 for Enhanced Navigation and Mapping. In *2025 1st International Conference on Consumer Technology (ICCT-Pacific)* (pp. 1-4). IEEE.
- Kun, D. P., Varga, E. B., и Toth, Z. (2017). Ontology based navigation model of the ILONA system. In: *IEEE 15th International Symposium on Applied Machine Intelligence и Informatics (SAMi)*, pp. 479-484.
- Kuriakose, B., Shrestha, R., & Simes, F. E. (2017). Tools и technologies for blind и visually impaired navigation support: A review. *IETE Technical Review*, 37(1), 3-18.
- Киула, H., Chidurala, V., Cao, Y., & Li, X. (2024). Localization of a BLE Device Based on Single-Device RSSI и DOA Measurements. *Network*, 4(2), 196-216.
- Lan, H., Yu, C., Zhuang, Y., Li, Y., & El-Sheimy, N. (2015). A novel kalman filter with state constraint approach for the integration of multiple pedestrian navigation systems. *Micromachines*, 6(7), 926-952.

- Lee, K., Lee, J. и Kwan, M. P., (2017). Location-based service using ontology-based semantic queries: A study with a focus on indoor activities in a university context. *Computers, Environment и Urban Systems*, 62, 41-52.
- Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3), 50-60.
- Li, Z., Liu, C., Gao, J., & Li, D. (2024). Current status и future trends of meter-level indoor positioning technology: A review. *Remote Sensing*, 16(2), 398.
- Lin, Z., и коллектив, (2017). Hybrid Spatial Data Model for Indoor Space: Combined Topology и Grid. *ISPRS International Journal of Geo-Information*, 6(11), 343.
- Liu, C., Zhang, Y., Ma, L., Huang, Y., Liu, K., & Wang, G. (2025). Unmanned Aerial Vehicle–Unmanned Ground Vehicle Centric Visual Semantic Simultaneous Localization and Mapping Framework with Remote Interaction for Dynamic Scenarios. *Drones*, 9(6), 424.
- Liu, J. (2024). Rfid (radio frequency identification) localization and application. *Applied Sciences*, 14(13), 5932.
- Liu, J., Yang, Z., Zlatanova, S., Li, S., & Yu, B. (2025). Indoor Localization Methods for Smartphones with Multi-Source Sensors Fusion: Tasks, Challenges, Strategies, и Perspectives. *Sensors*, 25(6), 1806.
- Liu, T., Li, B., Chen, G. E., Yang, L., Qiao, J., & Chen, W. (2023). Tightly coupled integration of GNSS/UWB/VIO for reliable and seamless positioning. *IEEE Transactions on Intelligent Transportation Systems*, 25(2), 2116-2128.
- Lukashchuk, A., Yildirim, H. K., Bancora, A., Lihachev, G., Liu, Y., Qiu, Z., ... & Kippenberg, T. J. (2024). Photonic-electronic integrated circuit-based coherent LiDAR engine. *Nature Communications*, 15(1), 3134.
- Łukasik, S., Szott, S., & Leszczuk, M. (2024). Multimodal image-based indoor localization with machine learning—A systematic review. *Sensors*, 24(18), 6051.
- Luo, X., O'Brien, W. J., & Julien, C. L. (2011). Comparative evaluation of Received Signal-Strength Index (RSSI) based indoor localization techniques for construction jobsites. *Advanced Engineering Informatics*, 25(2), 355-363.
- Mai, C., Chen, H., Zeng, L., Li, Z., Liu, G., Qiao, Z., Qu, Y., Li, L., & Li, L. (2024). A Smart Cane Based on 2D LiDAR и RGB-D Camera Sensor-Realizing Navigation и Obstacle Recognition. *Sensors*, 24(3). <https://doi.org/10.3390/s24030870>
- McKhann, G. M., Knopman, D. S., Chertkow, H., Hyman, B. T., Jack Jr, C. R., Kawas, C. H., ... & Phelps, C. H. (2011). The diagnosis of dementia due to Alzheimer's disease: recommendations

from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease. *Alzheimer's & dementia*, 7(3), 263-269.

Mekelleche, F., & Haffa, H. (2017). Classification и Comparison of Range-Based Localization Techniques in Wireless Sensor Networks. *J. Commun.*, 12(4), 221-227.

Menanno, M., Savino, M. M., & Accorsi, R. (2023). Digitalization of Fresh Chestnut Fruit Supply Chain through RFID: Evidence, Benefits and Managerial Implications. *Applied Sciences*, 13(8), 5086.

Mendoza-Silva, G. M., Torres-Sospedra, J., & Huerta, J. (2019). A meta-review of indoor positioning systems. *Sensors*, 19(20), 4507.

Merkel, S., Kessler, A., и Schmeck, H., (2014). Dynamic Multi-Objective Evacuation Path Planning in Mobile Ad Hoc Networks, Technical report.

Mohamed, A. H., & Schwarz, K. P. (1999). Adaptive Kalman filtering for INS/GPS. *Journal of geodesy*, 73(4), 193-203.

Müller K.; Fröhlich S.; Germano A.M.C.; Kondragunta J.; Agoitia Hurtado M.F.D.C.; Rudisch J.; Schmidt D.; Hirtz G.; Stollmann P.; Voelcker-Rehage C. (2020). Sensor-based systems for early detection of dementia (SENDa): A study protocol for a prospective cohort sequential study. *BMC Neurology*, 20(1), -. <https://doi.org/10.1186/s12883-020-01666-8>

Naseer, T., & Burgard, W. (2017, September). Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1525-1530). IEEE.

Nasreddine, Z. S., Phillips, N. A., Bédirian, V., Charbonneau, S., Whitehead, V., Collin, I., ... & Chertkow, H. (2005). The Montreal Cognitive Assessment, MoCA: a brief screening tool for mild cognitive impairment. *Journal of the American Geriatrics Society*, 53(4), 695-699.

Nawaz, H., Tahir, A., Ahmed, N., Fayyaz, U. U., Mahmood, T., Jaleel, A., ... & Abbasi, Q. (2021). Ultra-low-power, high-accuracy 434 MHz indoor positioning system for smart homes leveraging machine learning models. *Entropy*, 23(11), 1401.

Nguyen, D. V., Recalde, M. E. V., & Nashashibi, F. (2016, November). Low speed vehicle localization using wifi fingerprinting. In *2016 14th international conference on control, automation, robotics and vision (ICARCV)* (pp. 1-5). IEEE.

Obeidat, H., Shuaieb, W., Obeidat, O., & Abd-Alhameed, R. (2021). A review of indoor localization techniques и wireless technologies. *Wireless Personal Communications*, 119(1), 289-327.

- Ogle, J., Guensler, R., Bachman, W., Koutsak, M., & Wolf, J. (2002). Accuracy of global positioning system for determining driver performance parameters. *Transportation Research Record*, 1818(1), 12-24.
- Ozdenizci, B., Coskun, V., & Ok, K. (2015). NFC internal: An indoor navigation system. *Sensors*, 15(4), 7571-7595.
- Pahlavan, K., Krishnamurthy, P., & Geng, Y. (2015). Localization Challenges for the Emergence of the Smart World, Access. *IEEE*, 3(1), 1-11.
- Papadopoulos, K., Koustriava, E., & Kartasidou, L. (2012). Spatial coding of individuals with visual impairments. *The Journal of Special Education*, 46(3), 180-190.
- Park, J., Kim, J., & Kang, S. (2015). Ble-based accurate indoor location tracking for home и office. *Computer Science & Information Technology (CS & IT)*, 173-181.
- Pascacio, P., Casteleyn, S., Torres-Sospedra, J., Lohan, E. S., & Nurmi, J. (2021). Collaborative indoor positioning systems: A systematic review. *Sensors*, 21(3), 1002.
- Peng, J., Zhang, L., Fan, M., Zhao, N., Lei, L., He, Q., & Xia, J. (2023). An admission-control-based dynamic query tree protocol for fast moving RFID tag identification. *Applied Sciences*, 13(4), 2228.
- Pillette, L., Moreau, G., Normи, J. M., Perrier, M., Lécuyer, A., & Cogne, M. (2022). A systematic review of navigation assistance systems for people with dementia. *IEEE Transactions on Visualization u Computer Graphics*, 29(4), 2146-2165.
- Piva, B., & de Souza, C. C. (2017). Minimum stabbing rectangular partitions of rectilinear polygons. *Computers & Operations Research*, 80, 184-197.
- Pyt, P., Skrobacz, K., Jankowski-Mihułowicz, P., Węglarski, M., & Kamuda, K. (2023). Empowering Accessibility: BLE beacon-based IoT localization. *Electronics*, 12(19), 4012.
- Rawtaer, I., Mahendran, R., Kua, E. H., Tan, H. P., Tan, H. X., Lee, T. S., & Ng, T. P. (2020). Early detection of mild cognitive impairment with in-home sensors to monitor behavior patterns in community-dwelling senior citizens in Singapore: Cross-sectional feasibility study. *Journal of medical Internet research*, 22(5), e16854.
- Real, S., & Araujo, A. (2019). Navigation systems for the blind and visually impaired: Past work, challenges, and open problems. *Sensors*, 19(15), 3404.
- Research и Markets. (2025). Indoor positioning & indoor navigation market size & trends. Retrieved from <http://www.researchmarkets.com/report/ipin>
- Rezazadeh, J., Moradi, M., Ismail, A. S., & Dutkiewicz, E. (2014). Superior path planning mechanism for mobile beacon-assisted localization in wireless sensor networks. *IEEE Sensors Journal*, 14(9), 3052-3064.

- Rezazadeh, J., Moradi, M., Ismail, A. S., & Dutkiewicz, E. (2014). Superior path planning mechanism for mobile beacon-assisted localization in wireless sensor networks. *IEEE Sensors Journal*, 14(9), 3052-3064.
- Rezazadeh, J., Subramanian, R., Suresegaran, K., Kong, X., Moradi, M., & Khodamoradi, F. (2018). Novel iBeacon placement for indoor positioning in IoT. *IEEE Sensors Journal*, 18(24), 10240-10247.
- Roe, C. M., Babulal, G. M., Head, D., Stout, S. H., Vernon, E. K., Ghoshal, N., ... & Morris, J. C. (2021). Preclinical Alzheimer's disease и longitudinal driving decline. *Alzheimer's & Dementia: Translational Research & Clinical Interventions*, 3(1), 74-82.
- Sadowski, S., & Spachos, P. (2018). Rssi-based indoor localization with the internet of things. *IEEE access*, 6, 30149-30161.
- Sadowski, S., & Spachos, P. (2019). Optimization of BLE beacon density for RSSI-based indoor localization. In *2019 IEEE international conference on communications workshops (ICC Workshops)* (pp. 1-6). IEEE.
- Salducci, C., Bidel, Y., Cadoret, M., Darmon, S., Zahzam, N., Bonnin, A., ... & Bresson, A. (2024). Quantum sensing of acceleration and rotation by interfering magnetically launched atoms. *Science Advances*, 10(44), eadq4498.
- Sariman, M. S., Othman, M., Akir, R. M., Mahamad, A. K., & Rahman, M. A. (2024). A Review: Current Trend of Immersive Technologies for Indoor Navigation и the Algorithms. *Pertanika Journal of Science & Technology*, 32(2).
- Schilit, B., & Theimer, M. (1994). Disseminating active map information to mobile hosts. *IEEE Network*, 8(5), 22-32.
- Shaharuddin, S., Abdul Maulud, K. N., Syed Abdul Rahman, S. A. F., Che Ani, A. I., & Pradhan, B. (2023). The role of IoT sensor in smart building context for indoor fire hazard scenario: A systematic review of interdisciplinary articles. *Internet of Things (Netherlus)*.
- Shahat, E., Hyun, C. T., & Yeom, C. (2021). City digital twin potentials: A review and research agenda. *Sustainability*, 13(6), 3386.
- Shahzad, A., Dadlani, A., Lee, H., & Kim, K. (2022). Automated Prescreening of Mild Cognitive Impairment Using Shank-Mounted Inertial Sensors Based Gait Biomarkers. *IEEE Access* 10,(February 2022), 15835–15844. URL: <https://ieeexplore.ieee.org/abstract/document/9703339/>, Publisher: IEEE.
- Shi, W., & Cheung, C. (2006). Performance evaluation of line simplification algorithms for vector generalization. *The Cartographic Journal*, 43(1), 27-44.

- Spachos, P., & Plataniotis, K. (2020). BLE beacons in the smart city: Applications, challenges, и research opportunities. *IEEE Internet of Things Magazine*, 3(1), 14-18.
- Spilker Jr, J. J., Axelrad, P., Parkinson, B. W., & Enge, P. (Eds.). (1996). *Global positioning system: theory и applications, volume I*. American Institute of Aeronautics и Astronautics.
- Straits Research. (2025). Indoor positioning и navigation market size, share & growth chart by 2033. Retrieved from <https://straitsresearch.com/report/indoor-positioning-и-navigation-market>
- Straka, T., Vojtech, L., & Neruda, M. (2022). Simulation of radio signal propagation for UHF RFID technology in an indoor environment using Ray Tracing (Graphics) method. *Applied Sciences*, 12(21), 11065.
- Sun, X., Yeoh, W., и Koenig, S., (2010). Generalized Fringe-Retrieving A*: Faster moving target search on state lattices. In: *Proceedings - International Conference on Autonomous Agents и Multiagent Systems*, pp. 1081-1088.
- Szyc, K., Nikodem, M., & Zdunek, M. (2023). Bluetooth low energy indoor localization for large industrial areas и limited infrastructure. *Ad Hoc Networks*, 139, 103024.
- Tamura, B. K., Masaki, K. H., & Blanchette, P. (2020). Sleep и circadian rhythm disturbances in Alzheimer's disease. *Current Treatment Options in Neurology*, 9(4), 299-306.
- Tan, P., Tsinakwadi, T. H., Xu, Z., & Xu, H. (2022). Sing-Ant: RFID indoor positioning system using single antenna with multiple beams based on LANDMARC algorithm. *Applied Sciences*, 12(13), 6751.
- Tao, F., Zhang, H., Liu, A., & Nee, A. Y. (2018). Digital twin in industry: State-of-the-art. *IEEE Transactions on industrial informatics*, 15(4), 2405-2415.
- Tashakkori, H., Rajabifard, A., и Kalantari, M., (2015). A new 3D indoor/outdoor spatial model for indoor emergency response facilitation. *Building и Environment*, 89, 170-182.
- Templier, S., Cheiney, P., d'Armagnac de Castanet, Q., Gouraud, B., Porte, H., Napolitano, F., ... & Barrett, B. (2022). Tracking the vector acceleration with a hybrid quantum accelerometer triad. *Science Advances*, 8(45), eadd3854.
- Tian, Y., Huang, B., Jia, B., & Zhao, L. (2020). Optimizing AP и beacon placement in WiFi и BLE hybrid localization. *Journal of Network и Computer Applications*, 164, 102673.
- Tong, X., Su, Y., Li, Z., Si, C., Han, G., & Ning, J. (2023). A survey of indoor positioning systems based on a six-layer model. *Computer Networks*, 234, 109932.
- Virtue Market Research. (2024). Indoor positioning system market size, share & growth 2025-2030. Retrieved from <https://virtuemarketresearch.com/report/indoor-positioning-system-market>

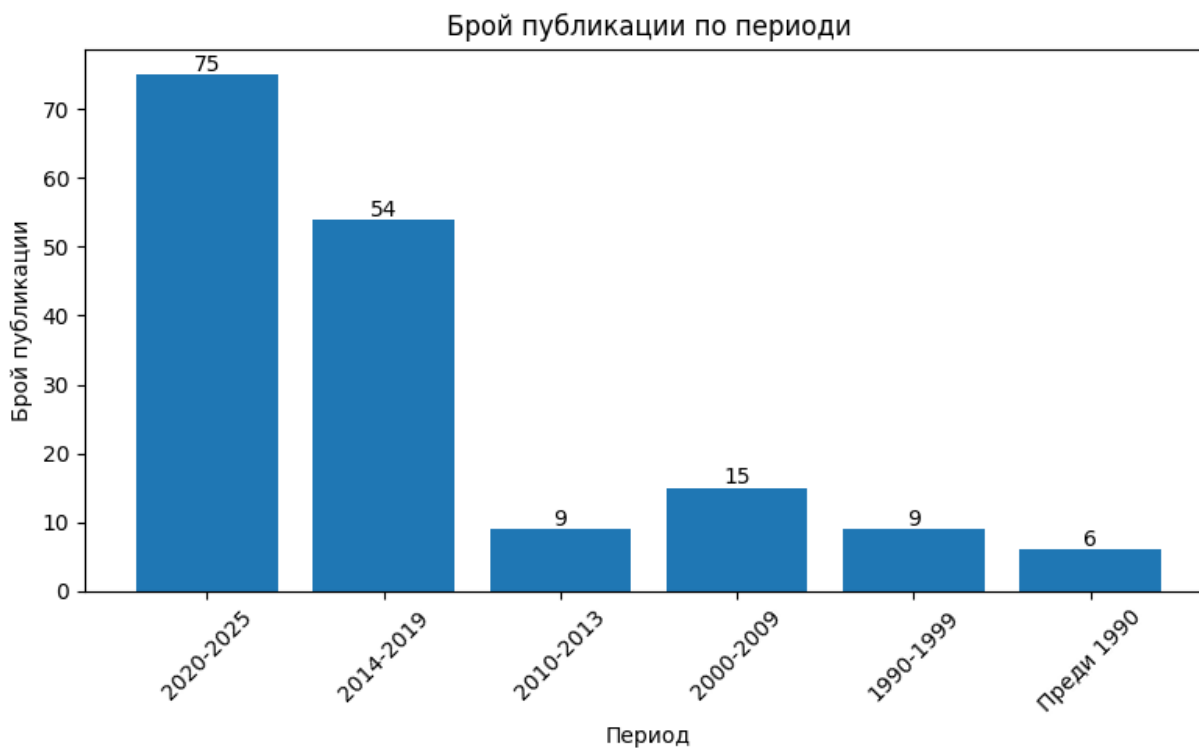
- Visvalingam, M., & Whyatt, J. D. (2017). Line generalization by repeated elimination of points. In *Lumarks in Mapping* (pp. 144-155). Routledge.
- Wagner, S., Fet, N., Handte, M., & Marrón, P. J. (2017, August). An approach for hybrid indoor/outdoor navigation. In *2017 International Conference on Intelligent Environments (IE)* (pp. 36-43). IEEE.
- Wei, K., Zheng, X., Wei, W., Song, M., Zhang, J., Gan, H., ... & Zhang, J. (2024, June). An algorithm of simultaneous localization and mapping for mobile robots based on graph optimization. In *International Conference on Remote Sensing, Mapping, and Image Processing (RSMIP 2024)* (Vol. 13167, pp. 157-168). SPIE.
- Wichmann, J. (2022). Indoor positioning systems in hospitals: A scoping review. *Digital Health*, 8, 20552076221081696.
- World Health Organization. (2019). *World report on vision*. World Health Organization.
- World Health Organization. (2023). *Dementia*. Retrieved from <https://www.who.int/news-room/fact-sheets/detail/dementia>
- Xiang, G., Dian, S., Zhao, N., & Wang, G. (2023). Semantic-Structure-Aware Multi-Level Information Fusion for Robust Global Orientation Optimization of Autonomous Mobile Robots. *Sensors*, 23(3). <https://doi.org/10.3390/s23031125>
- Xiaodong, G., Jiwei, H., Siyu, L., Jianhua, L., & Mingyi, D. (2018, March). Indoor localization method of intelligent mobile terminal based on BIM. In *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)* (pp. 1-9). IEEE.
- Xiong, Q., и коллектив, (2015). Multi-level indoor path planning method. *The International Archives of Photogrammetry, Remote Sensing и Spatial Information Sciences*, 40(4), 19.
- Xu, P., Kennedy, G. A., Zhao, F. Y., Zhang, W. J., & Van Schyndel, R. (2023). Wearable obstacle avoidance electronic travel aids for blind и visually impaired individuals: A systematic review. *IEEE Access*, 11, 66587-66613.
- Xu, W., Liu, L., Zlatanova, S., Penard, W., & Xiong, Q. (2018). A pedestrian tracking algorithm using grid-based indoor model. *Automation in Construction*, 92, 173-187..
- Yan, N., Chen, H., Lin, K., Li, Z., & Liu, Y. (2023). Fast and Effective Tag Searching for Multi-Group RFID Systems. *Applied Sciences*, 13(6), 3540.
- Yang, L., Chen, Y., Li, X. Y., Xiao, C., Li, M., & Liu, Y. (2014, September). Tagoram: real-time tracking of mobile RFID tags to high precision using COTS devices. In *MobiCom* (Vol. 10, pp. 2639108-2639111).

- Ye, W., He, N., Wang, J., & Yuan, X. (2023, October). A Navigation System for Guiding Blind People in Indoor and Outdoor. In *2023 International Conference on Computer Science and Automation Technology (CSAT)* (pp. 20-24). IEEE.
- Yen, J.Y., (1971). Finding the K Shortest Loopless Paths in a Network. *J. Manag. Sci.*, 17, 712–716.
- Yu, K., & Guo, Y. J. (2008). Statistical NLOS identification based on AOA, TOA, и signal strength. *IEEE Transactions on Vehicular Technology*, 58(1), 274-286.
- Zafari, F., & Papapanagiotou, I. (2015). Enhancing ibeacon based micro-location with particle filtering. In *2015 IEEE Global Communications Conference (GLOBECOM)* (pp. 1-7). IEEE.
- Zafari, F., Gkelias, A., & Leung, K. K. (2019). A survey of indoor localization systems и technologies. *IEEE Communications Surveys & Tutorials*, 21(3), 2568-2599.
- Zhang, X., Kwon, K., Henriksson, J., Luo, J., & Wu, M. C. (2022). A large-scale microelectromechanical-systems-based silicon photonics LiDAR. *Nature*, 603(7900), 253-258.
- Zhang, Y., Chen, H., Wang, J., & Liu, B. (2024). IoT-enabled smart cities: A hybrid systematic analysis of key research areas, challenges, и recommendations for future direction. *Discover Cities*, 1(1), 1-25.
- Zheng, Y., Li, Q., Chen, Y., Xie, X., & Ma, W. Y. (2008). Understanding mobility based on GPS data. In *Proceedings of the 10th international conference on Ubiquitous computing* (pp. 312-321).
- Zhou, B., Li, Q., Mao, Q., Tu, W., Zhang, X., & Chen, L. (2024). Smart cities и the IoT: An in-depth analysis of global research trends и future directions. *Discover Internet of Things*, 4(1), 1-20.
- Zlatanova, S., Liu, L., & Sithole, G. (2013, November). A conceptual framework of space subdivision for indoor navigation. In *Proceedings of the fifth ACM SIGSPATIAL international workshop on indoor spatial awareness* (pp. 37-41).
- Zou, H., Jiang, H., Luo, Y., Zhu, J., Lu, X., & Xie, L. (2016). Bluedetect: An ibeacon-enabled scheme for accurate and energy-efficient indoor-outdoor detection and seamless location-based service. *Sensors*, 16(2), 268.
- Zuo, W., & Shen, X. (2023). A contactless delivery solution for intelligent unmanned vehicles based on multi-source sensing signals. *Procedia Computer Science*, 228, 1000-1009.

Разпределение на цитираната литература по времеви периоди

Период	Брой публикации	Процент
2020-2025	75	44.64%
2014-2019	54	32.14%
2010-2013	9	5.36%
2000-2009	15	8.93%
1990-1999	9	5.36%
Преди 1990	6	3.57%
Всичко	168	100.00%

Графично представяне:



Публикации на автора

- RS-1. Ivanov, R. (2003). Automatic GPS-based vehicle tracking and localization information system. In *International Conference on Computer Systems and Technologies* (pp. 42-47).
- RS-2. Ivanov, R. (2008). Mobile GPS navigation application, adapted to visually impaired people. In *Proceeding of International Conference Automatics and Informatics* (Vol. 8, pp. 6-10).
- RS-3. Ivanov, R. (2009). Algorithm for GPS navigation, adapted for visually impaired people. In *Proceedings of the IASK International Conference E-Activity and Leading Technologies*
- RS-4. Ivanov, R. (2010). Indoor navigation system for visually impaired. In *Proceedings of the 11th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing on International Conference on Computer Systems and Technologies* (pp. 143-149).
- RS-5. Ivanov, R. (2010). On-line GPS Track Simplification Algorithm for Mobile Platforms. *Journal "Information Technologies and Control"*, (1).
- RS-6. Ivanov, R. (2011). A low-cost indoor navigation system for visually impaired and blind. *Communication and Cognition*, 44(3), 129.
- RS-7. Ivanov, R. (2011). Algorithm for blind navigation along a GPS track. In *Proceedings of the 12th International Conference on Computer Systems and Technologies* (pp. 372-379).
- RS-8. Ivanov, R. (2011). Matlab toolbox for raw GPS data processing and visualization, Списание "Автоматика и Информатика", No.1.
- RS-9. Ivanov, R. (2012). Real-time GPS track simplification algorithm for outdoor navigation of visually impaired. *Journal of Network and Computer Applications*, 35(5), 1559-1567
- RS-10. Ivanov, R. (2012). RSNAVI: an RFID-based context-aware indoor navigation system for the blind. In *Proceedings of the 13th international conference on computer systems and technologies* (pp. 313-320).
- RS-11. Ivanov, R. (2014). Blind-environment interaction through voice augmented objects. *Journal on Multimodal User Interfaces*, 8(4), 345-365.
- RS-12. Ivanov, R. (2016). An approach for microscopic path finding and obstacle avoidance for blind and visually impaired people. In *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016* (pp. 285-292).
- RS-13. Ivanov, R. (2017). An Algorithm for Micro-localization in Large Public Buildings. In *Proceedings of the 18th International Conference on Computer Systems and Technologies* (pp. 119-126).

- RS-14. Ivanov, R. (2017). An approach for developing indoor navigation systems for visually impaired people using Building Information Modeling. *Journal of Ambient Intelligence and Smart Environments*, 9(4), 449-467
- RS-15. Ivanov, R. (2018). An algorithm for on-the-fly shortest paths finding in multi-storey buildings using a hierarchical topology model. *International Journal of Geographical Information Science*, 32(12), 2362-2385
- RS-16. Ivanov, R. (2019). An approach to developing internet of things (IoT)-based services for smart museums. *Digital Presentation and Preservation of Cultural and Scientific Heritage*, 9, 111-122.
- RS-17. Ivanov, R. (2021). Accuracy analysis of BLE beacon-based localization in smart buildings. *Journal of Ambient Intelligence and Smart Environments*, 13(4), 325-344
- RS-18. Ivanov, R. (2021). Automatic Beacon Selection for Indoor Localization. In 2021 *International Conference Automatics and Informatics (ICAI)* (pp. 345-349). IEEE
- RS-19. Ivanov, R. (2022), Automatic Beacon Deployment for Indoor Localization, *Journal Automatica and Informatics*, No.1.
- RS-20. Ivanov, R. (2023). Exhibitexplorer: Enabling personalized content delivery in museums using contextual geofencing and artificial intelligence. *ISPRS International Journal of Geo-Information*, 12(10), 434.
- RS-21. Ivanov, R. (2024). Advanced Visitor Profiling for Personalized Museum Experiences Using Telemetry-Driven Smart Badges. *Electronics*, 13(20), 3977

Оценка на качеството на публикации на автора

Публикация индекс	Публикация тип	Индексиране	IF към годината на публикуване	IF за 2025	Цитирания
RS-1	доклад	-	0	0	6
RS-2	доклад	-	0	0	14
RS-3	доклад	-	0	0	7
RS-4	доклад	Scopus	0	0	125
RS-5	статия	-	0	0	2
RS-6	статия	-	0	0	5
RS-7	доклад	Scopus	0	0	8
RS-8	статия	-	0	0	0
RS-9	статия	Scopus, WoS	1,065	8,000	44
RS-10	доклад	Scopus	0	0	26
RS-11	статия	Scopus, WoS	1,031	2,100	20
RS-12	доклад	Scopus, WoS	0	0	6
RS-13	доклад	Scopus	0	0	3
RS-14	статия	Scopus, WoS	0,878	2,000	41
RS-15	статия	Scopus, WoS	2,370	5,100	10
RS-16	доклад	Scopus, WoS	0	0	3
RS-17	статия	Scopus, WoS	2,105	2,000	8
RS-18	доклад	Scopus	0	0	0
RS-19	статия	-	0	0	0
RS-20	статия	Scopus, WoS	3,400	2,800	43
RS-21	статия	Scopus, WoS	2,600	2,600	12
Всичко	10 доклада, 11 статии	14 Scopus, 9 WoS	13,449	24,600	383

За справка:

<http://scholar.google.com/citations?user=d7NmszgAAAAJ&hl=bg&oi=ao>

<http://orcid.org/0000-0002-8695-3085>

[ResearcherID: A-3265-2017](#)

[Scopus Author ID: 55781888700](#)

Приноси на автора

Настоящата монография систематизира и представя резултатите от дългогодишната научноизследователска дейност на автора в областта на локализацията и навигацията в интериорни и екстериорни среди. Приносите се отличават с изразена иновативност, практическа приложимост и интердисциплинарност, обединявайки алгоритмични, инженерни и приложни решения. Те могат да бъдат групирани в няколко основни направления:

1. Теоретично-методологични приноси

Теоретично-методологичните приноси могат да бъдат обобщени както следва:

- Систематизиране на знанията в областта на интелигентната локализация и навигация в интериорни и екстериорни среди.
- Критичен анализ на съществуващите решения, като се идентифицират техните предимства, недостатъци и предизвикателства.
- Класификация на навигационните системи и идентифициране на перспективи за бъдещо развитие, включително хибридни системи и контекстно-осъзнати подходи.

2. Алгоритмични приноси

Авторът предлага оригинални методи и алгоритми, които оптимизират процесите на локализация и навигация в различни среди:

- **Алгоритъм RSTS за опростяване на GPS траектории** – метод за обработка на GPS данни в реално време, позволяващ значително намаляване на изчислителните и енергийните ресурси на мобилни устройства, като същевременно се запазва необходимата точност. Емпиричните резултати показват предимства спрямо класическите методи (алгоритъм на Дъглас-Пюкер), което потвърждава неговата приложимост при мобилни и вградени системи.
- **Метод за намиране на K -най-кратки пътища в многоетажни сгради** – йерархичен модел на свързаност, позволяващ бързо изчисляване на

множество алтернативни маршрути, включително вертикални връзки (асансьори и стълбища). Подходът е особено ефективен за големи обществени сгради, където са необходими варианти за навигация и евакуация.

- **Методи за заобикаляне на препятствия и генериране на фини маршрути** – разработени са техники за буфериране и динамично избягване на прегради в реална среда. Тези решения са от ключово значение за системи, насочени към незрящи потребители или автономни агенти.
- **Алгоритми за автоматично разполагане и селекция на BLE маяци** – предложени са оптимизационни процедури, които намаляват броя на необходимите маяци при запазване на точност на локализацията. Включени са адаптивни методи за обработка на RSSI сигнали и алгоритми за микро локализация, водещи до устойчиви резултати дори в сложни интериорни среди.

3. Моделиране и интеграция на данни

В монографията се предлага оригинален подход за интеграция на пространствени и семантични модели:

- **BIM-базирано моделиране със семантичен слой** – разработени са структури за геометрично и функционално описание на сгради, които позволяват ефективно извличане на информация за навигация. Въведени са модели за запитвания, свързани с местоположението, които подпомагат контекстно-ориентирани приложения, включително за евакуация и управление на кризи.
- **XML скриптове и инструменти за моделиране на свързаността** – предлагат се формализирани методи за представяне на топологията на сгради, които могат да бъдат интегрирани с навигационни ядра и софтуерни системи.

4. Системни и приложни реализации

Всички предложени алгоритми и методи са с ориентация към практическа приложимост и разработване на прототипи:

- **RSNAVI** система за интериорна навигация на незрящи – реализирана на базата на RFID/NFC, системата комбинира семантични модели, маршрутизация и интуитивен интерфейс (аудио и тактилен). Тя е пример за социално значимо приложение на научните резултати.
- **Система за превенция и ранно откриване на деменция** – интегрира умни баджове, ESP32 шлюзове и бекенд платформа за мониторинг на поведенчески и локални показатели. Подходът предлага иновативен инструмент за медицинска и социална подкрепа.
- **ExhibitExplorer** – приложение за музеи и културни институции, което използва локализация и навигация за обогатено потребителско преживяване.
- **MATLAB инструментариум и приложения** – разработени са софтуерни инструменти за обработка на GPS и BLE данни, които улесняват изследователите и практиците при работа със сложни пространствени данни.

5. Емпирична верификация и научно влияние

Съществен принос е широката и систематична емпирична верификация:

- Проведени са обстойни експерименти в симулирани и реални среди, включително тестове за точност, устойчивост и ефективност.
- Визуализациите, статистическите анализи и сравненията с класически методи потвърждават предимствата на разработените решения.
- Научният принос е потвърден и чрез значителен брой самостоятелни публикации: 21 научни труда, от които 14 в Scopus и 7 в Web of Science. Част от тях са в списания с висок импакт фактор и със значителен брой цитирания, което свидетелства за международно признание.
- Най-силно въздействие имат публикации в престижни международни списания с импакт фактор и доклади с над 100 цитирания, което демонстрира реален принос към развитието на областта.

6. Обобщение

Монографията представя цялостен и последователен научен път, който обхваща:

1. **Теоретични приноси** – нови алгоритми и методи за локализация, оптимизация и маршрутизация.
2. **Моделиране и интеграция** – семантични и BIM модели за ефективно използване на пространствени данни.
3. **Приложни системи** – реални прототипи за незрящи, музеи и превенция на деменция.
4. **Емпирична валидност и научно признание** – множество експерименти, международни публикации и цитирания.