

С. Ю. Болотова, канд. физ.-мат. наук, доц., e-mail: bolotova.svetlana@gmail.com,
А. В. Зонов, преподаватель, e-mail: andryzonov@gmail.com,
А. П. Тютин, преподаватель, e-mail: yutin.ap@gmail.com,
Воронежский государственный университет

Навигация внутри помещений в мобильных приложениях

Навигация внутри помещений является достаточно сложной задачей, особенно в условиях больших многоэтажных зданий с множеством стен. Кроме этого, требуются высокая точность и скорость определения местоположения. В работе описаны основные идеи реализации мобильного приложения, позволяющего строить оптимальные маршруты внутри помещений с помощью Bluetooth-маячков и удовлетворяющего отмеченным выше требованиям.

Ключевые слова: навигация, bluetooth-маячки, beacons, мобильные приложения, местоположение, трилатерация, триангуляция, тесселяция, построение маршрутов

Введение

На настоящее время одной из актуальных тем в мобильных разработках является навигация внутри помещений. Это обстоятельство связано с тем, что здания становятся все более объемными и нередко имеют довольно сложную структуру, что затрудняет ориентацию в пространстве. Кроме того, решения, применяемые в навигации внутри помещений, помогают и в ориентировании вне зданий, на улице — там, где в условиях плотной застройки использование систем спутниковой навигации затруднено (нет спутников в прямой видимости, присутствует только отраженный/ослабленный/зашумленный сигнал GPS/Глонасс и т. д.).

Навигация внутри помещений имеет следующий ряд особенностей.

1. Местоположение внутри помещений необходимо определять быстро и с высокой точностью, чего не позволяет навигация GPS/ГЛОНАСС.

2. Присутствует такой фактор, как этажность здания.

3. В помещении, как правило, нет дорог, по которым можно было бы прокладывать маршруты.

Таким образом, узловыми моментами для решения задачи навигации внутри помещений являются вопросы определения местоположения с высокой точностью и построения маршрутов на открытой местности. Целью настоящей работы стали подбор и реализация алгоритмов программных систем, позволяющих найти ответы на поставленные задачи.

В ходе изучения рассматриваемой предметной области был сделан вывод, что поставленная задача на настоящий момент является актуальной, а работающих решений, удовлетворяющих необходимым требованиям, не существует.

Определению местоположения с помощью технологии Bluetooth (Bluetooth Indoor Positioning) посвя-

щена работа [1] Anja Bekkelien из Женевского университета. Однако ее задача заключалась в разработке подмодуля для глобального модуля позиционирования (GPM), в основе которого работает совокупность сенсоров. В настоящей работе ставится условие реализации самодостаточной навигационной системы, использующей в своей основе технологию iBeacons.

Технология iBeacons, представленная Apple в iOS 7, расширила возможности библиотеки для работы со службами геолокации. Вместо геолокации по широте и долготе iBeacon использует низкоэнергетический сигнал Bluetooth, который обнаруживается телефоном. Соответствующий формат данных строго типизирован и детально описан на официальном сайте Apple для разработчиков.

Построение маршрутов на открытой местности изучала Anita Graser из Австрийского технологического института. В ее работе [2] рассмотрены вопросы внедрения работы с открытыми пространствами в OpenStreetMap — веб-картографическом проекте по созданию географической карты мира. Однако описанные в указанной работе алгоритмы не учитывают препятствий на открытых пространствах, а реализация не учитывает специфики мобильных платформ.

Технология Bluetooth Beacons

Для определения местоположения мобильного устройства нередко используются Bluetooth-маячки (beacons).

Bluetooth-маячки — это обычные Bluetooth 4.0 LE (Low Energy) устройства, т. е. их роль может с успехом выполнять любое устройство, оснащенное чипом BLE (Bluetooth Low Energy), например, смартфон iPhone. Для этой цели необходимо установить маячки по периметру помещения, а координаты их расположения сохранить в базе данных. Маячки с заданной

периодичностью будут осуществлять широковещательную рассылку, содержащую идентифицирующую их информацию. Пользовательское приложение, циклично получая эти сигналы, сможет с помощью базы данных определить координаты маячков. Далее на основе анализа интенсивности посылаемого сигнала RSSI (*Received Signal Strength Indicator*) может быть вычислено местоположение пользователя [3].

Алгоритм определения местоположения

Получаемый параметр RSSI позволяет определить удаленность маячка от приемника сигнала — смартфона. Это осуществляется делением текущего значения RSSI на эталонное значение интенсивности, измеренное на расстоянии 1 м. Очевидно, что значение RSSI обратно пропорционально удаленности смартфона от маячка. Однако вследствие физического эффекта интерференции сигналов значение RSSI между двумя стационарными объектами непостоянно (рис. 1, 2).

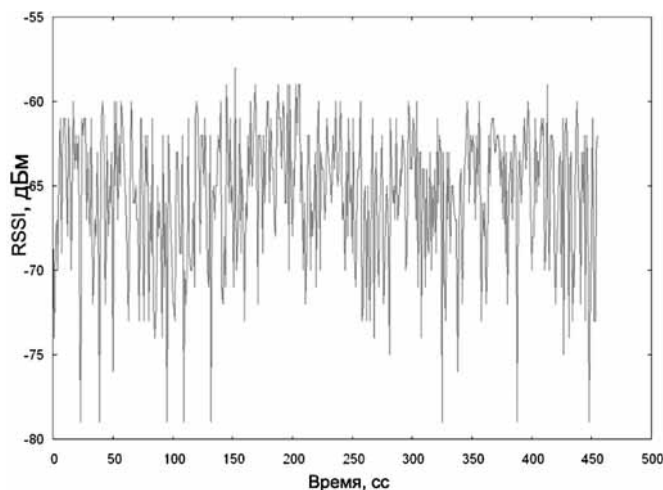


Рис. 1. Значения RSSI между маячком и устройством на фиксированном расстоянии 3 м

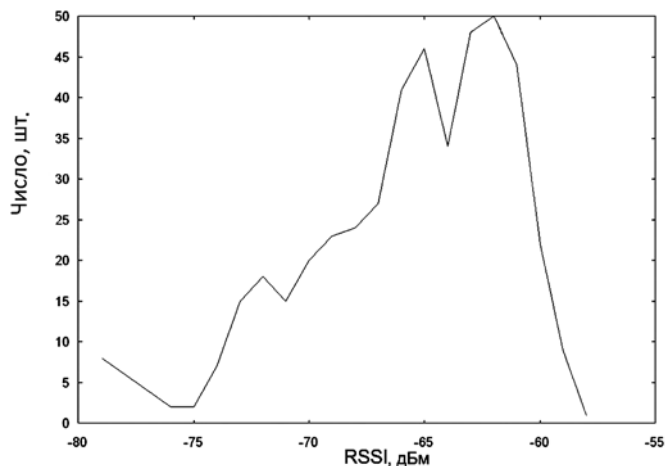


Рис. 2. Гистограмма значений для графика, показанного на рис. 1

Как видно представленных графиках, при использовании единственного маячка можно получать разную удаленность. Графически ситуацию можно представить окружностью с центром в точке с координатами расположения маячка и радиусом, равным вычисленному расстоянию до него. Радиус этой окружности даже при неподвижном состоянии приемника будет меняться с течением времени. Используя информацию от двух маячков, можно получить две окружности с изменяющимися радиусами и, возможно, одной или двумя точками пересечения. Если маячки дают точную информацию об удаленности, то используя два маячка, точное местоположение можно рассчитать лишь находясь на одной линии с ними. При отклонении от этой линии получатся две точки пересечения и две возможные координаты. Однако, поскольку значение RSSI колеблется, точек пересечения может и не быть. Отсюда можно сделать вывод, что для однозначного определения местоположения необходимо как минимум три маячка.

Алгоритм нахождения координат по трем или более точкам и удалению от них называется алгоритмом трилатерации [4]. Его графическое представление дано на рис. 3.

Задача трехмерной трилатерации решается путем нахождения координат пересечения трех сфер, которые определяют решением системы уравнений. Вначале составим систему уравнений для трех сфер. Поскольку любые три точки, не лежащие на одной прямой, определяют в пространстве плоскость, поместим начало координат этой плоскости в центр

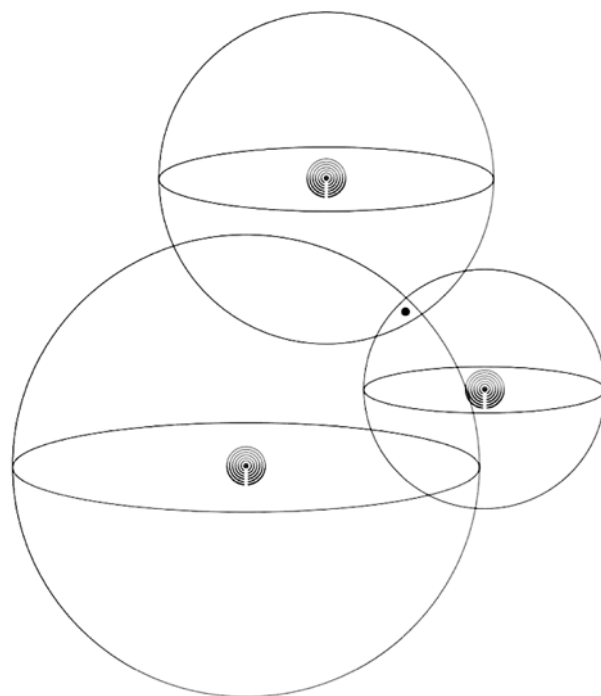


Рис. 3. Алгоритм трилатерации — графическое представление

одной из сфер, а центр второй сферы поместим на ось координат Ox :

$$\begin{aligned} r_1^2 &= x^2 + y^2 + z^2, \\ r_2^2 &= (x - d)^2 + y^2 + z^2, \\ r_3^2 &= (x - i)^2 + (y - j)^2 + z^2, \end{aligned}$$

где r_1, r_2, r_3 — радиусы сфер; d — расстояние между центрами первой и второй сфер по оси Ox ; i — расстояние между центрами первой и третьей сфер по оси Ox ; j — расстояние между центрами первой и третьей сфер по оси Oy .

В точке (x, y, z) будет располагаться приемник сигналов:

$$\begin{aligned} x &= \frac{r_1^2 - r_2^2 + d^2}{2d}, \\ y &= \frac{r_1^2 - r_3^2 - x^2 + (x - i)^2 + j^2}{2j} = \\ &= \frac{r_1^2 - r_3^2 + i^2 + j^2}{2j} - \frac{i}{j}x, \\ z &= \pm \sqrt{r_1^2 - x^2 - y^2}. \end{aligned}$$

Таким образом, получены уравнения для вычисления координат (x, y, z) приемника. Очевидно, координата высоты z может принимать два значения. Это обстоятельство обусловлено тем, что передатчики и приемник могут находиться не в одной плоскости. Для нашей задачи можно рассматривать только нижнее значение, так как маячки располагаются выше пользовательских устройств.

Благодаря тому требованию, что в рассматриваемой системе все маячки должны лежать в одной плоскости, для решения поставленной задачи положением на вертикальной оси можно пренебречь. Устройство пользователя — приемник находится в другой плоскости. Однако в силу плотного покрытия области маячками и фильтрации удаленных маячков можно считать, что приемник находится в той же плоскости.

В нашем случае координаты вычисляются в помещениях со стенами, столбами и электронными устройствами. Такие помещения характеризуются наличием "шума". Представим области действия маячков в виде окружностей, где центр окружности — точка расположения маячка, а радиус определяется силой сигнала. Пересечением подобных окружностей может являться область, пустая или нет. Поэтому целесообразно модифицировать рассматриваемый алгоритм, воспользовавшись методом итеративной трилатерации [5]. Суть этого алгоритма состоит в итеративной минимизации ошибки:

$$|f_i| = \left| \frac{d_i - \sqrt{(x_i - x_e)^2 + (y_i - y_e)^2}}{d_i} \right|,$$

где $|f_i|$ — вещественное значение, характеризующее ошибку, а (x_e, y_e) — тривиальная начальная оценка. Необходимо итеративно высчитывать значение $|f_i|$ до тех пор, пока ошибка не уменьшится до заданного значения.

Однако эмпирические наблюдения показывают, что в результате все еще остается "шум", который проявляется как изменение координат пользователей при отсутствии физического движения. Фильтрация такого "шума" может быть выполнена с помощью вмонтированных в устройствах сенсоров. Для решения этой части задачи подходят методы на основе рекурсивной оценки Байеса.

С указанной целью в рамках настоящей работы были рассмотрены возможности применения фильтра Калмана [6] и многочастотного фильтра [7]. Фильтр Калмана предназначен для рекурсивного дооценивания вектора состояния априорно известной динамической системы. Для расчета текущего состояния системы необходимо знать текущее измерение, а также предыдущее состояние самого фильтра. Алгоритм работает в два этапа. На этапе прогнозирования фильтр Калмана экстраполирует значения переменных состояния, а также их неопределенности. На втором этапе по данным измерения (полученного с некоторой погрешностью) результат экстраполяции уточняется. Благодаря пошаговой природе алгоритма он может в реальном времени отслеживать состояние объекта [8].

Многочастотный фильтр основан на последовательном методе Монте-Карло. Это рекурсивный алгоритм численного решения задач оценивания (фильтрации, сглаживания), особенно актуальный для нелинейных и негауссовских случаев. В отличие от фильтра Калмана, многочастотные фильтры не зависят от методов линеаризации или аппроксимации.

Экспериментальным путем было установлено, что из двух рассмотренных вариантов фильтр Калмана обеспечивает лучшее сглаживание.

Построение маршрутов на открытой местности

Перейдем к вопросу построения маршрутов. Поскольку здание обычно представляется не набором коридоров/лабиринтов, а некоторым числом помещений, то, как было сказано ранее, изначально отсутствует заданное множество дорог. Процесс решения рассматриваемой задачи можно разбить на следующие два этапа.

1. Представив помещение в виде полигона, провести его триангуляцию.

2. Приняв вершины получившихся треугольников за вершины графа, а их стороны — за его ребра, определить ближайшие к стартовой и конечной точкам вершины и найти кратчайший путь от одной до другой.

По определению, простой полигон — упорядоченная последовательность точек V_0, \dots, V_n . Соседние вер-

шины соединены ребром $\langle V_i, V_{i+1} \rangle$, где $0 \leq i \leq n - 2$.

Ребро $\langle V_{n-1}, V_0 \rangle$ соединяет первую и последнюю вершины. Каждая вершина принадлежит ровно двум ребрам. Единственное место пересечения любых двух ребер — вершина полигона.

Если полигон простой, то во время прохода по его ребрам внутренняя область всегда находится с одной и той же стороны. Полигон будем считать направленным против часовой стрелки, если при проходе вдоль ребер внутренняя область находится слева.

Триангуляция — это разбиение геометрического объекта на симплексы, т. е. n -мерные обобщения треугольника. В нашем случае объект находится на плоскости, соответственно, полигон разбивается на треугольники.

Для триангуляции существует множество алгоритмов. Рассмотрим один из них (полностью описан в работе [9]), называемый *ear clipping* (отсекание "ушей", "ушная" триангуляция). Сложность данного алгоритма — $O(n^2)$.

"Ухом" полигона называется треугольник, образованный тремя последовательными вершинами $V_{i_0}, V_{i_1}, V_{i_2}$, где V_{i_1} — выпуклая вершина (ее внутренний угол меньше 180°), отрезок $\langle V_{i_0}, V_{i_2} \rangle$ полностью лежит внутри полигона, при этом ни одна вершина полигона, кроме $V_{i_0}, V_{i_1}, V_{i_2}$, не принадлежит треугольнику. Отрезок $\langle V_{i_0}, V_{i_2} \rangle$ называется диагональю полигона. Вершина V_{i_1} называется "кончиком уха". Треугольник (полигон, для которого $n = 32$) состоит из одного "уха", однако "кончиком уха" можно считать любую из трех вершин. Полигон с четырьмя и более сторонами всегда имеет как минимум два непересекающихся "уха".

Предлагается рекурсивный подход к решению задачи "ушной" триангуляции. Если возможно найти "уха" в полигоне с $n \geq 4$ вершинами и "отрезать" его, то останется полигон из $n - 1$ вершин и можно повторить процесс. Такой алгоритм характеризуется сложностью $O(n^3)$.

Однако существует модификация этого алгоритма, приводящая к сложности $O(n^2)$ [9]. Первый шаг заключается в том, чтобы хранить полигон в двусвязном кольцевом списке, что дает возможность быстрого удаления кончиков ушей. Создание такого списка реализуется алгоритмом со сложностью $O(n)$. Вторым шагом является проход по всем вершинам и поиск "ушей". Для каждой выпуклой вершины V_i возьмем треугольник $\langle V_{i-1}, V_i, V_{i+1} \rangle$ и проверим, существует ли среди остальных вершин полигона такая, которая лежит в этом треугольнике. Если ни одна из них не лежит внутри треугольника, треугольник будет являться "ухом".

Еще одним шагом оптимизации рассматриваемого алгоритма может быть использование следующего факта. Очевидно, что если в проверяемый треугольник попадает какая-либо другая вершина многоугольника, то существует невыпуклая вершина, находящаяся внутри этого треугольника. Таким

образом, проверять на нахождение внутри треугольника нужно только те вершины, углы которых являются невыпуклыми (внутренний по отношению к полигону угол больше 180°).

Структура данных для полигона состоит из четырех двусвязных списков:

- вершины полигона хранятся в циклическом списке;
- выпуклые вершины хранятся в линейном списке;
- невыпуклые вершины хранятся в линейном списке;
- "кончики ушей" хранятся в циклическом списке.

Создав первые три списка, можно начать удалять "уши" по одному. Если V_i — удаляемая из полигона вершина, описание углов вершин V_{i-1} и V_{i+1} может измениться следующим образом [9]:

- если один из этих углов выпуклый, то он останется выпуклым;
- если один из углов V_j принадлежал другому "уху", то после удаления "уха" V_i "уха" V_j может исчезнуть;
- если один из углов невыпуклый, он может стать выпуклым, а также может создать новое "уха".

Таким образом, после удаления "уха" V_i , если соседняя вершина выпуклая, необходимо проверить, не образует ли она новое "уха", обойдя невыпуклые вершины и проверив принадлежность этих вершин треугольнику "уха". Всего есть $O(n)$ "ушей". Каждое обновление соседних вершин влечет проверку, занимающую $O(n)$ операций. Таким образом, весь процесс "отрезания ушей" представляет собой алгоритм сложности $O(n^2)$.

"Ушной" алгоритм также может быть применен к полигонам с дырками. Дыркой называется полигон, полностью лежащий внутри триангулируемого полигона. В таком случае перед применением алгоритма триангуляции необходимо соединить внешний полигон и все его дырки в одну последовательность точек.

Представим себе полигон с одной дыркой. Он состоит из внешнего и внутреннего полигонов. Порядок вершин во внешнем и внутреннем полигонах должен быть противоположным. Задача заключается в том, чтобы найти во внешнем и внутреннем полигонах вершины, между которыми можно построить отрезок, не пересекающийся ни с одним из других отрезков полигонов. В работе [9] описан один из способов нахождения таких вершин, суть которого представляется следующим алгоритмом.

Шаг 1. Найти точку M внутреннего полигона с максимальным значением по оси x .

Шаг 2. Найти самую ближнюю к точке M точку пересечения I луча $M + t(1, 0)$ со сторонами внешнего полигона.

Шаг 3. Если I — вершина внешнего полигона, искомая точка найдена.

Шаг 4. Иначе I — точка отрезка $\langle V_i, V_{i+1} \rangle$. Выбрать точку P — конец отрезка $\langle V_i, V_{i+1} \rangle$ с большим значением по оси x .

Шаг 5. Проверить невыпуклые вершины внешнего полигона (кроме P) на содержание внутри треугольника $\langle M, I, P \rangle$. Если ни одна из них не лежит внутри треугольника, то точка P является искомой.

Шаг 6. Иначе хотя бы одна невыпуклая вершина лежит внутри треугольника. Среди этих вершин выбрать такую вершину R , что угол IMR минимален. Если таких точек несколько, то все они лежат на одном луче, выходящем из M , и необходимо выбрать ближайшую к M точку. Точка R является искомой.

Вершины M и P необходимо дублировать и соединить полигоны в один двумя отрезками. Полученный полигон можно триангулировать как обычный полигон без дырок.

Полигон может иметь несколько дырок. В таком случае нужно применить описанный выше алгоритм к каждой из этих дырок последовательно, выбирая из списка оставшихся такую, которая содержит вершину с максимальным значением x .

Было замечено, что при малом числе углов полигона после триангуляции может получиться достаточно разреженный граф, что не позволит построить оптимальный маршрут через его вершины. Поэтому было решено провести уплотнение полигональной сетки. Это можно сделать с помощью *тесселяции*. Тесселяция — это процесс разбиения полигональной сетки на более мелкие многоугольники. В результате тесселяции граф получается более плотным, и кратчайшие пути становятся гораздо более естественными для движения пешехода.

Итак, для выполнения триангуляции данные, описывающие карту помещения, должны представляться набором полигонов, где координаты углов внешних стен помещения перечислены по часовой стрелке, а координаты углов внутренних стен и других преград (дырок в полигоне) — против часовой стрелки, или наоборот.

Поскольку помещение может представлять невыпуклый многоугольник, а внутри него могут оказаться дополнительные преграды, необходим алгоритм триангуляции, учитывающий эти два обстоятельства.

На выходе алгоритма триангуляции и, если нужно, тесселяции, должен получиться набор вершин треугольников. Далее этот набор необходимо преобразовать в неориентированный взвешенный граф: вершины треугольников становятся вершинами графа, а стороны — его ребрами. При этом длина стороны становится весом соответствующего ребра.

Для построения маршрута достаточно использовать один из алгоритмов поиска кратчайшего пути, например, алгоритм Дейкстры.

Применяя описанные выше операции, получим маршрут, который представляется на карте ломаной линией. В качестве пути дальнейшего улучшения результатов можно рассмотреть сглаживание полу-

ченного маршрута, например, с помощью кубических сплайнов.

Заключение

Результатом представленного исследования стал работоспособный прототип приложения, который удовлетворяет требованиям к производительности, точности определения местоположения, а также умеет строить оптимальные маршруты внутри помещений. На основе проведенных экспериментов были получены следующие результаты.

- Описанные алгоритмы обладают достаточной производительностью, чтобы проводить все расчеты в реальном времени с учетом ограничения мобильных устройств, на которых прототип тестировался.

- Погрешность вычислений координат составляет порядка 15 м. Для ее уменьшения необходимо учитывать, в первую очередь, внутренние датчики.

- Погрешность увеличивается при движении и поворотах. Поэтому в текущей реализации невозможно избежать небольших "подергиваний" точки, обозначающей местоположение пользователя на карте при движениях такого типа.

- Удастся построить достаточно правдоподобный, оптимальный маршрут в помещении, не имеющем определенных дорог.

Список литературы

1. **Bekkelien A., Deriaz M., Marchand-Maillet S.** Bluetooth indoor positioning. Master's thesis, University of Geneva. 2012. 49 p.
2. **Graser A.** Integrating Open Spaces Into OpenStreetMap Routing Graphs for Realistic Crossing Behavior in Pedestrian Navigation // *GI_Forum — Journal for Geographic Information Science*. 2016. N. 1. P. 217–230.
3. **Wu B.-F., Jen C.-L., Chang K. C.** Neural fuzzy based indoor localization by Kalman filtering with propagation channel modeling // In *IEEE International Conference on Systems, Man and Cybernetics*. 2007. P. 812–817.
4. **Lau E.-E. L., Chung W.-Y.** Enhanced RSSI-based real-time user location tracking system for indoor and outdoor environments // In *ICCIT 07: Proceedings of the 2007 International Conference on Convergence Information Technology*. Washington, DC, USA. 2007. P. 1213–1218.
5. **Thrun S., Burgard W., Fox D.** Probabilistic Robotics. Early Draft-MIT Press. 2005. 492 p.
6. **Chen Z.** Bayesian filtering: From Kalman filters to particle filters, and beyond. Statistics. McMaster University, Hamilton, 2003. 69 p.
7. **Rekleitis I. M.** A particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02. Centre for Intelligent Machines. University St., Montreal, Quebec, Canada H3A 2A. 2004. 36 p.
8. **Carpenter J., Clifford P., Fernhead P.** An improved particle filter for non-linear problems // *EE Proceedings-Radar, Sonar and Navigation*. 1999. P. 2–7.
9. **Eberly D.** Triangulation by ear clipping. Geometric Tools, LLC. 2015. 13 p.

Indoor Navigation in Mobile Applications

S. Yu. Bolotova, bolotova.svetlana@gmail.com, **A. V. Zonov**, andryzonov@gmail.com,
A. P. Tutin, tyutin.ap@gmail.com, The Voronezh State University, Voronezh,
394018, Russian Federation

Corresponding author:

Bolotova Svetlana Yu., Associate Professor, The Voronezh State University, 394018, Voronezh, Russian Federation
E-mail: bolotova.svetlana@gmail.com

Received on October 2, 2017
Accepted on October 16, 2017

The present article describes basic ideas of implementation of mobile application dedicated to indoor navigation. Indoor navigation is a complex issue, especially when it comes to large buildings with lots of rooms and rich infrastructure. The goal of the research was to achieve high accuracy and performance of mobile device positioning and indoor routes calculation.

Bluetooth beacons were suggested to determine mobile device location. Analyzing the received signal strength indicator (RSSI) and beacons position on the map provided an ability to calculate the distance from each beacon to the mobile device.

Trilateration algorithms were composed with Kalman filter which allowed for fast and precise device positioning.

To calculate pedestrian-friendly indoor routes the authors addressed game development methods: room map was suggested to be in a form of a polygon with holes, which was decomposed via triangulation and tessellation algorithms. Among all triangulation algorithms that work with "holed" polygons the modification of Earcut triangulation algorithm was considered to be one of the most suitable and efficient. As a result, getting a polygonal mesh of high density allowed to use this mesh as a graph and apply any path searching algorithm to calculate the determined route.

The result of the whole research was a working prototype of a mobile application which satisfied performance and accuracy requirements. Preferred algorithms have good potential and can be used as a basis for future improvements. Inaccuracy of measurements should always be considered and its handling is a self-sufficient problem.

Keywords: navigation, Bluetooth beacons, beacons, mobile applications, location, trilateration, triangulation, tessellation, route building

For citation:

Bolotova S. Yu., Zonov A. V., Tutin A. P. Indoor Navigation in Mobile Applications, *Programmnaya Ingeneria*, 2018, vol. 9, no. 1, pp. 29–34.

DOI: 10.17587/prin.9.29-34

References

1. **Bekkelien A., Deriaz M., Marchand-Maillet S.** *Bluetooth indoor positioning*, Master's thesis, University of Geneva, 2012, 49 p.
2. **Graser A.** Integrating Open Spaces Into OpenStreetMap Routing Graphs for Realistic Crossing Behavior in Pedestrian Navigation, *GI Forum — Journal for Geographic Information Science*, 2016, no. 1, pp. 217–230.
3. **Wu B.-F., Jen C.-L., Chang K. C.** Neural fuzzy based indoor localization by Kalman filtering with propagation channel modeling, *In IEEE International Conference on Systems, Man and Cybernetics*, 2007, pp. 812–817.
4. **Lau E.-E. L., Chung W.-Y.** Enhanced RSSI-based real-time user location tracking system for indoor and outdoor environments, *In ICCIT 07: Proceedings of the 2007 International Conference on Convergence Information Technology*, Washington, DC, USA, 2007, pp. 1213–1218.
5. **Thrun S., Burgard W., Fox D.** *Probabilistic Robotics*, Early Draft-MIT Press, 2005, 492 p.
6. **Chen Z.** *Bayesian filtering: From Kalman filters to particle filters, and beyond*, Statistics, McMaster University, Hamilton, 2003, 69 p.
7. **Rekleitis I. M.** A particle filter tutorial for mobile robot localization, Technical Report TR-CIM-04-02, Centre for Intelligent Machines, University St., Montreal, Quebec, Canada H3A 2A, 2004, 36 p.
8. **Carpenter J., Clifford P., Fernhead P.** An improved particle filter for non-linear problems, *EE Proceedings-Radar, Sonar and Navigation*, 1999, pp. 2–7.
9. **Eberly D.** *Triangulation by ear clipping*, Geometric Tools, LLC, 2015, 13 p.