

Выводы:

1. Создание современных информационных систем, реализующих единое информационное пространство, должно опираться не только на существующие физически реализованные системы.

2. Исследования автора показывают, что имеется исторически сложившаяся система представления информации, защищенная от ее потери человечеством в случае глобальных катаклизмов.

3. Важным направлением исследований является применение развитых математических те-

рий для создания защищенных от воздействий средств хранения и представления информации.

4. Как показывают исследования, возможно создание глобальных информационных систем, в которых информация представлена в виде комплексных чисел с целью оптимизации требуемых ресурсов в условиях постоянно растущих объемов хранимой, обрабатываемой и передаваемой информации. Перспективным является применение в этих целях квантовых компьютерных технологий.

СПИСОК ЛИТЕРАТУРЫ

1. Воробьев Е. Г. Расчет эффективности информационных атак внешнего нарушителя на объекты информатизации с распределенной инфраструктурой // Изв. СПбГЭТУ «ЛЭТИ». 2015. № 1. С. 23–28.

2. Нильсен М., Чанг И. Квантовые вычисления и квантовая информация. М.: Мир, 2006.

E. G. Vorobiev

Saint-Petersburg state electrotechnical university «LETI»

COMPLEX NUMBERS AND OPTIMIZATION OF MEANS OF STORAGE OF INFORMATION IN GLOBAL INFORMATION SYSTEMS

In article the analysis of possibility of use of the theory of complex numbers for optimization of means of storage of information in global information systems is carried out. The uniform theory of submission of information and the characteristic of information fields in a form, simple and available to the reader, are for the first time considered.

Complex numbers, common information space, information capacity of a point of space, characteristic of information fields

УДК 004.896:004.42

Дао Зуй Нам, С. А. Ивановский

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Оптимизация алгоритма локализации мобильного робота с использованием триангуляции карты

Рассматривается приближенный алгоритм локализации мобильного робота, основанный на использовании триангуляции простого многоугольника, представляющего карту. Приводятся численные результаты экспериментальных исследований оптимизированного алгоритма и их интерпретация.

Вычислительная геометрия, робототехника, локализация робота, триангуляция полигона, пересечение многоугольников, приближенный алгоритм, оптимизация алгоритма

Для решения прикладной задачи локализации мобильного робота (ЗЛМР), которая относится к области робототехники, используются методы и

алгоритмы вычислительной геометрии. Содержательно ЗЛМР для случая плоскости формулируется следующим образом. Мобильный робот мо-

жет перемещаться во внешней среде, которую можно представить как свободное пространство на плоскости, ограниченное стеной (преградой). Будем считать, что эту внешнюю среду можно описать простым плоским многоугольником, внутренность которого соответствует свободному пространству, а граница многоугольника (без самопересечений) соответствует преграде. Пусть робот снабжен картой внешней среды в виде плоского простого многоугольника P с n вершинами без отверстий. В исходном состоянии мобильный робот помещен в заранее неизвестное место среды, т. е. в некоторую точку p в пределах многоугольника P . Робот снабжен компасом и сенсорным устройством, с помощью которого он осуществляет круговой обзор и определяет расстояние до преграды. Перед роботом стоит задача определить свое истинное местоположение во внешней среде, т. е. локализовать себя на карте. Для этого робот, во-первых, может осмотреть свою окрестность и соотнести видимую область (так называемый многоугольник видимости) $V = V(p)$ с картой. Если на карте имеется лишь один фрагмент, совпадающий с многоугольником видимости, то задача решена. Если таких фрагментов несколько, то необходимо определить, какой из них соответствует начальному местоположению робота. Для этого на основании анализа многоугольников P и V робот должен сгенерировать множество H всех гипотез о своем местоположении $p_i \in P$ таким образом, что область видимости в точке p_i конгруэнтна V . Далее робот, перемещаясь и обозревая окрестность, может устранить все неправильные гипотезы о своем местоположении и таким образом определить свое истинное начальное местоположение. При этом требуется, чтобы суммарная длина проделанных перемещений робота была минимальной.

Оказывается, оптимизационная задача локализации мобильного робота является NP-трудной задачей [1]. Поэтому в основном рассматриваются приближенные (полиномиальные по вычислительной сложности) алгоритмы локализации мобильного робота [2]–[4]. При этом, как правило, внимание акцентируется на характеристиках, описывающих отклонения от стоимости оптимального решения (суммарной длины перемещений робота). Характеристики вычислительной сложности таких алгоритмов оцениваются асимптотически, а данные о реальном времени работы алгоритмов, как правило, не приводятся, в том числе, видимо, из-за высокой вычислительной сложности алгоритмов, например $O(n^5 \log n)$ [2] или $\Omega(n^{12})$ [4]. В [5] и [6]

предложены приближенные алгоритмы решения ЗИМР, основанные на предварительной триангуляции простого многоугольника, представляющего карту. Далее будут рассмотрены модификации этих алгоритмов, более эффективные по времени вычислений.

Известные алгоритмы локализации мобильного робота [2]–[4] включают две фазы: генерацию гипотез [7] и проверку гипотез. В фазе генерации гипотез вычисляется множество гипотетических местоположений робота $p_1, p_2, \dots, p_k \in P$, которые соответствуют наблюдениям роботом окружающей среды в его начальном положении. В фазе проверки гипотез исключаются неправильные гипотезы.

В фазе генерации гипотез строится множество гипотез $H = \{h_1, h_2, \dots, h_k\} (\forall i \in 1..k \mid h_i : p = p_i)$, определяемых гипотетическими местоположениями в P , в которых робот мог бы быть расположен первоначально. Далее выбирается произвольное гипотетическое p_i местоположение из H , чтобы служить точкой привязки. Затем для каждого гипотетического местоположения $p_j, 1 \leq j \leq k (j \neq i)$ определяется вектор переноса $t_j = p_i - p_j$, который соответствует переводу местоположения p_j в $p_i (p_i = p_j + t_j)$. Вычисляются копии P_1, P_2, \dots, P_k многоугольника карты P , соответствующие множеству гипотез H , таким образом, что P_j преобразуется в $P = P_i$ смещением на t_j (можно считать, что при этом экземпляр P_i преобразуется в себя смещением на $t_i = 0$).

Вычислять пересечение многоугольников необходимо в алгоритмах [3] и [4], а также в двух рассматриваемых далее алгоритмах [5] и [6]. Отметим, что 1) при вычислении кратчайшего пути внутри простого многоугольника из одной точки в другую алгоритм [8], использующий предварительную триангуляцию многоугольника с последующим использованием «воронки» (funnel), более эффективен, чем другие подходы, например на основе построения графа видимости [9]; 2) на основе использования триангуляции многоугольника можно более эффективно вычислять многоугольники видимости и их скелеты.

Дадим описание предлагаемой модификации алгоритма локализации мобильного робота с использованием триангуляции карты, представленного ранее в [6]. Предлагаемое описание отличается уточнением и более подробным представле-

нием некоторых шагов этих алгоритмов (а также оценок их сложности, а в дальнейшем – их более эффективной программной реализацией), в том числе за счет более систематического использования триангуляции карты.

Как уже отмечалось, триангуляцию многоугольника карты целесообразно использовать в алгоритмах локализации как вспомогательное действие при вычислении кратчайшего пути из одной точки внутри простого многоугольника в другую. Однако используя триангуляцию исходного многоугольника как предобработку, во-первых, можно эффективно реализовать и некоторые другие действия, например построение многоугольника видимости, а во-вторых, можно использовать разбиение карты на множество треугольников для выбора перемещений робота на этапе отсека гипотез. Например, обследование пути робота может происходить по центрам треугольников или по средним точкам ребер триангуляции. Можно рассматривать триангуляцию как альтернативу разбиению многоугольника на ячейки видимости, использованному в [2], [7]. Количество ячеек триангуляции (т. е. треугольники) – $O(n)$, что позволяет надеяться на ускорение работы алгоритма локализации при их использовании.

Алгоритм локализации робота с использованием триангуляции карты [6] можно описать с учетом данных ранее определений. Пусть на входе алгоритма заданы многоугольник карты P и робот, помещенный в неизвестное начальное местоположение в P . Алгоритм состоит из следующих действий:

1. Вычислить в относительных координатах многоугольник видимости $V(p)$ и его скелет $V^*(p)$ по данным сенсоров робота в текущем начальном местоположении робота p , соответствие которому на карте неизвестно. Пусть m – число вершин многоугольника $V^*(p)$.

2. Выполнить триангуляцию многоугольника карты.

3. Сгенерировать множество H из k гипотез на карте P , которые соответствуют полученному многоугольнику видимости V . Для этого скелет многоугольника видимости V^* , заданный в относительных координатах, сопоставляется с последовательностью вершин многоугольника с учетом маркировки ребер скелета. В местах совпадений на основе использования триангуляции карты вычисляется скелет многоугольника видимости относительно предполагаемого положения робо-

та, и эти 2 скелета сравниваются. В случае совпадения не только скелетов, но и многоугольников видимости фиксируется новая гипотеза.

4. [Операции в пп. 4–9 осуществляются далее для всех активных (не отклоненных пока) гипотез h_j ($j = 1, 2, \dots, k'$), начиная с $k' = k$.] Выбрать произвольную гипотезу h_i из множества активных гипотез H и соответствующую точку гипотетического местоположения как исходную для построения пересечений. Переносами вершин получить триангулированные «смещенные» многоугольники, соответствующие прочим гипотезам (фактически триангуляция, заданная номерами вершин, не изменяется, а «смещаются» лишь координаты вершин).

5. Для активных гипотез вычислить связный компонент $F = \text{Inter } S(P_1, P_2, \dots, P_{k'})$, содержащий стартовую точку. Вычислить триангуляцию F .

6. Найти точку r в множестве точек на серединах ребер триангуляции и центров треугольников в пределах многоугольника F как такую точку, которая является ближайшей к текущему положению робота среди точек, возможно, устраняющих лишние гипотезы. Для этого запускается поиск в ширину на графе триангуляции. В каждой обследуемой вершине определяется скелет многоугольника видимости $V^*(q)$ из соответствующей ей точки q . Этот скелет сопоставляется со скелетами видимости соответствующих точек q_j в экземплярах карты активных гипотез, и определяется возможность отклонения гипотез при планируемом перемещении робота в эту точку. Для всех таких точек вычисляется кратчайший путь от текущего положения робота и выбирается точка r с минимальной длиной пути. При этом для вычисления кратчайшего пути можно уже на этапе поиска в ширину получить набор треугольников от текущего положения робота к точке r и затем сразу построить воронку для вычисления кратчайшего пути. При данной модификации важно, что дерево поиска в ширину обходится не полностью, так как обрываются поддеревья, корнями которых являются найденные точки возможного устранения гипотез. Это ограничение поиска в ширину, возможно, приведет к удлинению суммарного перемещения робота, но сокращает время работы алгоритма.

7. Переместить робот в точку r .

8. Устранить гипотезы, сравнивая данные о текущем многоугольнике видимости, выдаваемые роботом в точке r , с данными о видимости, вычисленными во всех эквивалентных точках, соот-

ветствующих активным гипотезам. При этом, если подтверждается одна гипотеза, то устраняется одна или более других, т. е. в итоге устраняется хотя бы одна из гипотез.

9. Пусть E – множество гипотез, устраненных на предыдущем шаге. Заменить k' на $k' - |E|$. Повторять шаги 4–9, пока в множестве активных гипотез H не останется только одна гипотеза ($k' = 1$), которая и будет соответствовать истинному начальному местоположению робота.

Вычислительная сложность этого алгоритма по этапам и суммарно приведена в табл. 1. Отметим, что при выполненной предварительной триангуляции кратчайший путь от текущего положения робота до анализируемой точки вычисляется в худшем случае за время $O(n)$ [8], однако эту операцию требуется применять многократно на шаге 6 каждой итерации алгоритма, что в значительной степени определяет суммарную сложность алгоритма локализации.

Триангуляция простого многоугольника теоретически имеет сложность $O(n)$ [10] и может быть реализована, например, известным эффективным и практичным алгоритмом [11] за время $O(n \log^* n)$ (здесь $\log^* n$ – итерированный логарифм), что на практике в ЗЛМР дает фактически $O(n)$. Выходом этого алгоритма является множество треугольников, заданных номерами своих вершин. Для дальнейшего использования в алгоритме локализации множество треугольников преобразуется за время $O(n)$ в специальную структуру данных [12]. Это представление, по сути, является одним из адаптированных к триан-

гуляции вариантов реберного списка, используемого для представления планарного подразделения плоскости [9]. Каждый треугольник в этой структуре представлен своими тремя вершинами и тремя указателями на соседние треугольники, смежные с ним через его стороны.

Структура триангуляции получается в 2 этапа. Сначала по заданному множеству треугольников, образующему триангуляцию, для каждой вершины триангуляции формируется список треугольников, в которые входит эта вершина. Затем формируется собственно структура триангуляции, при этом информация о прилегающих треугольниках получается для каждой пары вершин данного треугольника посредством анализа списков, полученных на первом этапе. Структура триангуляции дополняется массивами входов для вершин и ребер триангуляции.

Использование структуры триангуляции позволяет эффективно реализовать некоторые базовые операции алгоритма локализации. Например, систематически используемая операция построения многоугольника видимости и его скелета реализована на основе поиска в ширину на графе, двойственном триангуляции. При этом удается, как правило, избежать просмотра всех вершин многоугольника карты, анализируя треугольники, смежные с текущим.

Для анализа эффективности предложенных алгоритмов проводилось экспериментальное исследование, основанное на их программной реализации (Visual C++ 2010), а также реализациях алгоритмов [2]–[4] и сравнении таких их характе-

Таблица 1

Шаги	Действия	Сложность укрупненных действий
1, 2	Триангуляция многоугольника карты	$O(n \log^* n)$
3, 4	Генерация гипотез	$O(mn) + O(kn)$
5	Построение пересечений относительно выбранной гипотезы и триангуляция F (k' – число активных гипотез)	$(k' - 1)O(n) + O(n \log^* n)$
6, 7	Обследование $4k'(n - 2)$ точек на ребрах и центрах треугольников для устранения гипотез. Вычисление кратчайших путей до точек, устраняющих гипотезы, для определения ближайшей из них	$4k'(n - 2)O(n) = k'O(n^2)$
8	Сравнение данных о многоугольниках видимости для активных гипотез и текущего положения робота	$k'O(n)$
Полная сложность: $O(mn) + O(kn) + O(n \log^* n) + \sum_{k'=1}^{k-1} [k'O(n^2) + k'O(n)] = O(n^4)$		

ристик, как длина суммарного пути перемещения робота и время работы алгоритма локализации. Ранее было установлено [13], что алгоритм [3], использующий рандомизацию при проверке гипотез, более эффективен, чем алгоритм [2], использующий декомпозицию карты на ячейки видимости и имеющий асимптотическую сложность $O(n^5 \log n)$, а также более эффективен, чем алгоритм [4], основанный на решении полугрупповой задачи Штейнера и имеющий вычислительную сложность $\Omega(n^{12})$. В связи с этим далее приводится сравнение именно с алгоритмом [3]. На алгоритмы, участвовавшие в эксперименте, будем ссылаться, обозначив их для краткости римскими цифрами: I – алгоритм локализации мобильного робота (АЛМР) с использованием триангуляции карты; II – АЛМР с использованием рандомизации при проверке гипотез [3] (число точек, случайным образом размещаемых в исследуемой области, $X = 100$); III – тот же АЛМР [3], но число точек, случайным образом размещаемых в исследуемой области, $X = 500$; IV – АЛМР с использованием окон в многоугольнике карты [5].

При проведении эксперимента использовалась генерация карт различных модельных типов. Генерация осуществлялась по нескольким задаваемым шаблонам. Сочетание параметров, задающих размеры шаблонов, определяет общий размер карты n , и при такой генерации его числовое значение можно регулировать, как правило, лишь

приблизительно. На рис. 1 приведен пример генерации карты размера $n = 672$ и с числом гипотез $k = 144$ при любом из обозначенных на рисунке начальных положений p_i ($i = 1 \dots 144$). Отметим, что такая структура карты обеспечивает относительно большое отношение k/n , а именно $k/n = 0.21$.

Предварительный анализ показал, что полученные характеристики алгоритмов (длина пути d , пройденного роботом до окончания локализации, и время t работы алгоритма локализации) существенно зависят от начального расположения робота (от номера гипотезы). По этой причине целесообразно усреднять по гипотезам не сами характеристики, полученные для разных алгоритмов, а их отношения, которые и будут характеризовать сравнительную эффективность алгоритмов. Выберем характеристики алгоритма с триангуляцией (номер алгоритма I) в качестве «эталона» для сравнения. Другими словами, для алгоритмов (с номером a) будут вычислены (здесь i – номер гипотезы)

$$s_i^{(a)} = \frac{x_i^{(a)}}{x_i^{(I)}}, \text{ где } x = d \text{ или } x = t.$$

Усреднением по k гипотезам получим $\bar{s}^{(a)} = \frac{1}{k} \sum_{i=1}^k s_i^{(a)}$. При этом $\bar{s}_i^{(I)} = 1$ и $\bar{s}^{(I)} = 1$.

В табл. 2 приведены значения отношений $\bar{s}^{(a)}$ для конфигурации, изображенной на рис. 1.

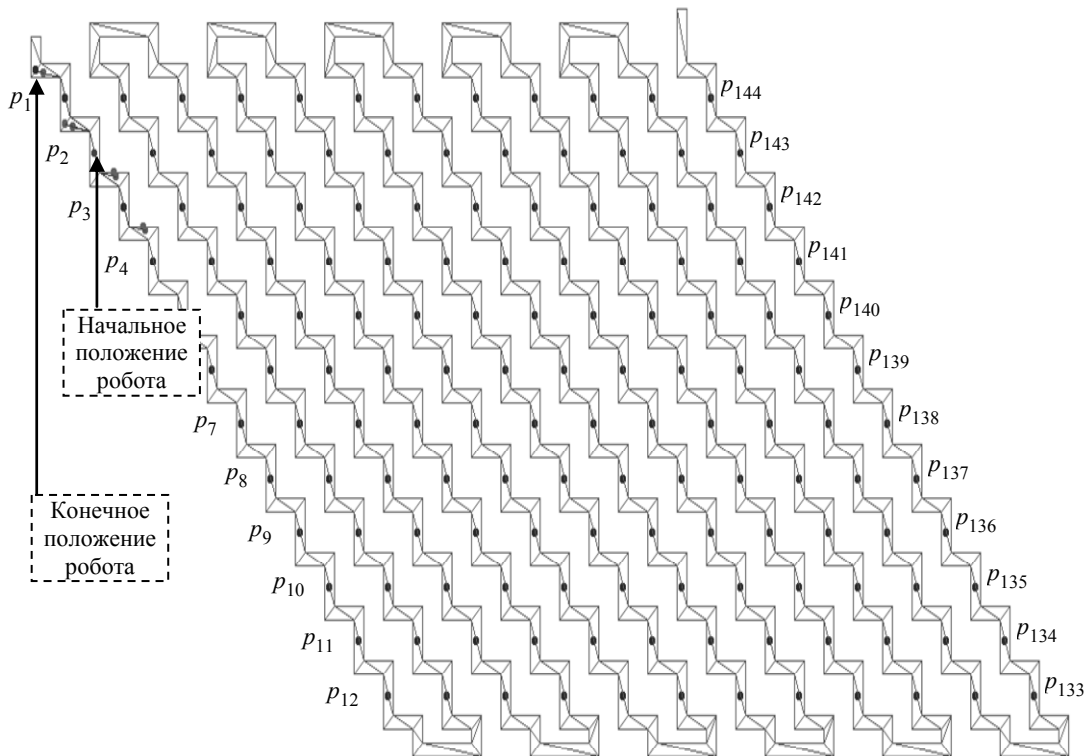


Рис. 1

Таблица 2

Характеристика	Номер алгоритма			
	I	II	III	IV
Длина пути	1	1.57	1.01	1.63
Время локализации	1	17.1	26.7	26.9

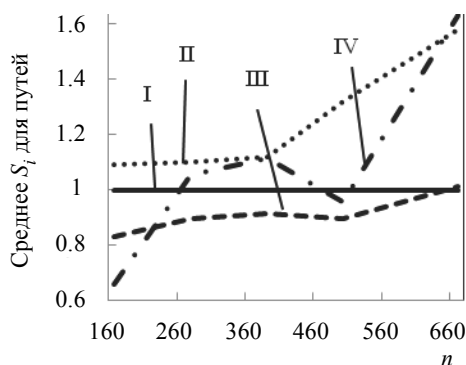


Рис. 2

Алгоритм I на основе триангуляции показывает здесь значительно лучшее время, но при этом дает несколько большую длину пути, чем алгоритм с рандомизацией в варианте III. Аналогичные соотношения получены и на других конфигурациях карты. На рис. 2 и 3 приведены графики зависимостей средних значений указанных отношений от размера n многоугольника карты. Различные типы линий графиков на рис. 2–3 относятся к алгоритмам с соответствующими номерами.

Данные на этих рисунках показывают, что в приведенном диапазоне значений размера карты n алгоритм I показывает существенно меньшее

время работы, а вторым по времени работы оказывается алгоритм II.

Проведенные эксперименты с другими модельными конфигурациями карты показали аналогичные результаты и позволяют сделать вывод,

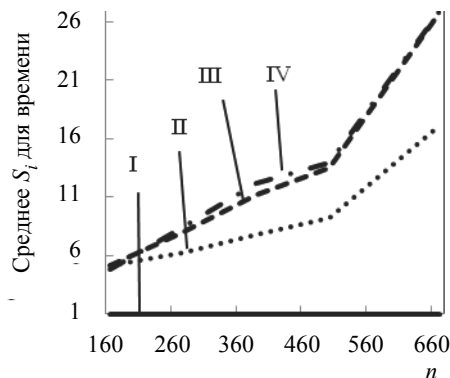


Рис. 3

что рассмотренные модификации приближенных алгоритмов ЛМР обеспечивают сравнимую точность, но время работы двух модифицированных алгоритмов ЛМР с использованием триангуляции карты и окон в многоугольнике карты меньше, чем время работы других. При этом значительно лучшие по времени результаты показывает алгоритм I, систематически использующий триангуляцию карты. Этот результат отличается от ранее полученного по исходным (немодифицированным) версиям алгоритмов [5] и [6], использующих триангуляцию, где лучшие результаты показывал алгоритм II.

СПИСОК ЛИТЕРАТУРЫ

1. Dudek G., Jenkin M. Computational Principles of Mobile Robotics. Cambridge University Press, 2010.
2. Dudek G., Romanik K., Whitesides S. Localizing a robot with minimum travel // SIAM J. Comput. 1998. Vol. 27. P. 583-604.
3. Rao M., Dudek G., Whitesides S. Randomized algorithms for minimum distance localization // Intern. J. Robotics Research. 2007. Vol. 26. P. 917-934.
4. A near-tight approximation algorithm for the robot localization problem / S. Koenig, J. S. B. Mitchell, A. Mudgal, C. Tovey // SIAM J. Comput. 2009. Vol. 39. P. 461-490.
5. Дао Зуй Нам, Ивановский С. А. Приближенный алгоритм локализации мобильного робота с использованием окон в многоугольнике карты // Изв. СПбГЭТУ «ЛЭТИ». 2014. № 3. С. 38-43.
6. Дао Зуй Нам, Ивановский С. А. Приближенные алгоритмы локализации мобильного робота // Науч. вестн. НГТУ. 2014. № 2. С. 109-121.
7. Guibas L. J., Motwani R., Raghavan P. The robot localization problem // SIAM J. Comput. 1997. Vol. 26. P. 1120-1138.
8. Hershberger J., Snoeyink J. Computing minimum length paths of a given homotopy class // Computational Geometry. Theory and Applications. 1994. № 4. P. 63-98.
9. Computational Geometry: Algorithms and Applications / M. De Berg, O. Cheong, M. Van Kreveld, M. Overmars. Berlin; Heidelberg: Springer-Verlag, 2008.
10. Chazelle B. Triangulating a simple polygon in linear time // Discrete & Computational Geometry. 1991. № 6(1). P. 485-524.
11. Seidel R. A Simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons // Computational Geometry. Theory and Applications. 1991. № 1(1). P. 51-54.
12. Скворцов А. В. Триангуляция Делоне и ее применение. Томск: Изд-во Том. ун-та, 2002.
13. Дао Зуй Нам, Ивановский С. А. Экспериментальный анализ алгоритмов локализации мобильного робота // Изв. СПбГЭТУ «ЛЭТИ». 2014. № 1. С. 19-24.

Dao Duy Nam, S. A. Ivanovskiy
Saint-Petersburg state electrotechnical university «LETI»

OPTIMIZATION ALGORITHM LOCALIZATION MOBILE ROBOT WITH USING TRIANGULATION MAP

We consider approximation algorithm for the robot localization problem. The algorithm is based on triangulation of a simple polygon representing a map. On the basis of their program implementation conducted experimental studies of this algorithm. The numerical results and their interpretation are given.

Computational geometry, robotics, robot localization, polygon triangulation, overlay polygon, approximation algorithm, algorithm complexity

УДК 004.82, 004.89

А. В. Малов
Motorola Solutions

С. В. Родионов
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Реализация упрощенного алгоритма Байеса в среде функционального программирования COMMON LISP

Рассматриваются основные задачи и особенности интеллектуального анализа данных. Иллюстрируются преимущества применения среды и языка функционального программирования COMMON LISP для решения типовых задач интеллектуального анализа данных на примере реализации упрощенного алгоритма Байеса.

Интеллектуальный анализ данных, функциональное программирование, упрощенный алгоритм Байеса

В настоящее время наблюдается интенсивное развитие информационных технологий. В частности, активно развиваются технологии искусственного интеллекта, в том числе интеллектуальный анализ данных (Data mining). Он является одним из современных подходов к анализу больших массивов данных и предназначен для выявления скрытых нетривиальных знаний, которые позволят сформулировать практически полезные гипотезы.

Для того чтобы сформулированные гипотезы не носили случайного характера и были характерными для заданной предметной области, они должны быть логически объяснимы и представлены в виде, доступном для дальнейшего анализа

и использования. Такие представления называются моделями. Модель, представляющая собой формальное описание полученных из данных знаний, является результатом работы методов и алгоритмов Data mining и отражает вид представления полученных знаний.

Широко распространен подход к интеллектуальному анализу данных посредством первоначального построения или выбора модели с последующим ее применением для выполнения функций Data mining, таких, как классификация, аппроксимация, кластеризация, ассоциирование и построение последовательностей, определение атрибутов, прогнозирование временных рядов, определение аномалий [1]. При этом имеется