

4. Р 50-601-46-2004 Рекомендации. Методика менеджмента процессов в системе качества. М.: Изд-во стандартов, 2004.

5. Стандарты и рекомендации для гарантии качества высшего образования в европейском пространстве / Аккредитация в образовании. Йошкар-Ола, 2008. 58 с.

6. Татаринов Ю. С., Власенко С. В., Макин В. А. Интеграция компонентов корпоративных информа-

ционных сред вузов на базе применения современных систем электронного документооборота и делопроизводства с элементами BPM // Тр. XX Всерос. науч.-методической конф. Телематика. СПб., 2013. 147 с.

7. Татаринов Ю. С., Власенко С. В., Макин В. А. Актуальные задачи автоматизации систем качества вузов, построенных на концепциях типовой модели // Тр. XVIII Всерос. науч.-методической конф. Телематика. СПб., 2011. 186 с.

Yu. S. Tatarinov, S. V. Vlasenko, V. A. Makin, A. V. Nasedkin
Saint-Petersburg State Electrotechnical University «LETI»

PROSPECTS OF CREATION A REPLICABLE UNIVERSITY QUALITY MANAGEMENT SYSTEM BASED ON THE ADVANCED DOCUMENT AUTOMATION SYSTEM WITH BPM ELEMENTS

Reviewing approaches for create a replicable university quality management system. Given a comparison classes of systems suitable for this task. Approved the case of using document automation system with BPM elements as a base of university quality management system.

Quality management system, Document automation, information systems

004.896:004.42

Дао Зуй Нам, С. А. Ивановский

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

Приближенный алгоритм локализации мобильного робота с использованием окон в многоугольнике карты

Рассматривается приближенный алгоритм локализации мобильного робота, снабженного картой в виде простого многоугольника. Гипотезам локализации соответствуют экземпляры карты с отметкой предполагаемого положения робота. Алгоритм основан на использовании оверлея и пересечения экземпляров карты. В пересечении выделяются так называемые окна, «заглядывая» в которые робот отсекает ложные гипотезы. Приведены численные результаты экспериментальных исследований этого алгоритма и их интерпретация.

Вычислительная геометрия, робототехника, локализация робота, оверлей многоугольников, сложность алгоритма, приближенный алгоритм

Задача локализации мобильного робота [1] состоит в определении координат робота в системе отсчета, связанной с внешней средой. Робот снабжен картой внешней среды в виде плоского простого многоугольника P с n вершинами без отверстий. Мобильный робот помещен в заранее неизвестное место p в пределах P (рис. 1). Для решения задачи локализации робота, во-первых, должен, обзревая свою окрестность и соотнося полученный многоугольник видимости $V = V(p)$ с картой, определить, является ли его начальное

местоположение единственным. Затем на основании анализа многоугольников P и V робот должен сгенерировать множество H всех гипотез о своем местоположении $p_i \in P$ таким образом, чтобы область видимости в точке p_i была конгруэнтна V . Далее робот должен определить свое истинное начальное местоположение, перемещаясь и обзревая окрестность, чтобы устранить все неправильные гипотезы о своем местоположении. При этом суммарная длина перемещений робота должна быть минимальной.

Известные алгоритмы локализации мобильного робота [2]–[5] включают две фазы: генерацию гипотез и проверку гипотез. В фазе генерации гипотез вычисляется множество гипотетических местоположений $p_1, p_2, \dots, p_k \in P$, которые соответствуют наблюдениям робота в его начальном положении. В фазе проверки гипотез исключаются неправильные гипотезы. Оптимизационная задача локализации мобильного робота является NP-трудной задачей [2]. Предлагается приближенный (полиномиальный) алгоритм локализации мобильного робота с использованием окон в многоугольнике карты.

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 $t_i = 0$). Точка гипотетического местоположения p_j в каждом многоугольнике P_j переходит при таком переносе в точку p_i многоугольника $P = P_i$.

Определим теперь *оверлей (наложение) многоугольников* (многоугольники рассматриваются здесь как подразбиение плоскости на внутреннюю и внешнюю грани, а стороны многоугольников рассматриваются как ребра такого подразбиения плоскости) [6], [7].

Определение. Оверлей многоугольников P_1, P_2, \dots, P_k относительно выделенного экземпляра многоугольника карты $P = P_i$ является структурой, полученной объединением множеств ребер всех смещенных многоугольников P_j , $1 \leq j \leq k$. Это множество ребер порождает разбиение плоскости на грани.

В более общем случае можно определить оверлей (наложение) $\text{Overlay}(S_1, S_2)$ двух подразбиений S_1 и S_2 как подразбиение $\text{Overlay}(S_1, S_2)$, такое, что существует грань F в $\text{Overlay}(S_1, S_2)$, если

и только если имеются грани F_1 в S_1 и F_2 в S_2 , такие, что F является максимальным связным подмножеством в $F_1 \cap F_2$ [6]. На рис. 1 приведен пример оверлея многоугольников $\text{Overlay}(P_1, P_2)$.

Пересечение многоугольников P_1, P_2, \dots, P_k относительно выделенного экземпляра многоугольника карты $P = P_i$ определим как *пересечение смещенных многоугольников* P_j , $1 \leq j \leq k$. Это пересечение является гранью (или набором граней) оверлея многоугольников. На рис. 1 грань оверлея, которая является пересечением $\text{Intersection}(P_1, P_2)$ многоугольников P_1 и P_2 , заштрихована.

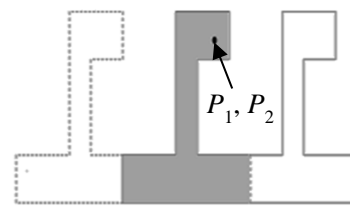


Рис. 1

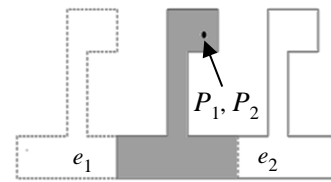


Рис. 2

Определение. Внутреннее ребро оверлея многоугольников является таким ребром их пересечения (одним из нескольких), которое отделяет область пересечения от других внутренних граней оверлея, в противоположность тем ребрам, которые принадлежат пересечению, но отделяют область пересечения от внешней грани оверлея на двумерной плоскости (внутренние ребра e_1 и e_2 на рис. 2).

Рассмотрим гипотезу h_j ($h_j \neq h_i$) и обозначим как F_{ij} такую грань в пересечении $\text{Intersection}(P_i, P_j)$, которая содержит начальную позицию γ_0 (см., например, F_{12} на рис. 3). Грань F_{ij} имеет самое большее $2n$ ребер [4].

Каждое из $O(n)$ ребер $e \subset F_{ij}$ может быть ребром одного из трех типов: (i) e лежит на границе P_i , но не P_j ; (ii) e лежит на границе P_j , но не P_i ; (iii) e лежит на границе и P_i , и P_j . Робот может различить h_i и h_j , если и только если робот видит ребро e типа (i) или (ii).

На рис. 3, *a* показан многоугольник P с двумя гипотезами h_1 и h_2 ; на рис. 3, *б* – наложение P_1 и P_2 с выделенной гранью F_{12} , содержащей γ_0 .

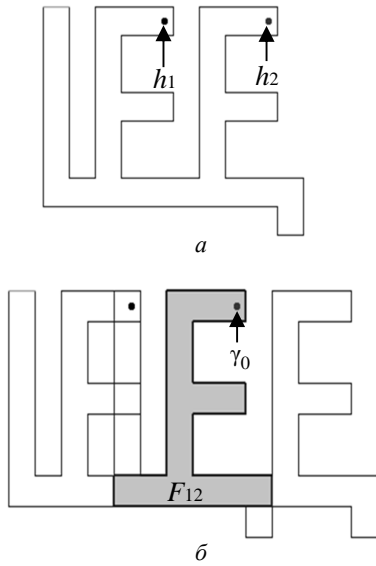


Рис. 3

Если из точки γ_0 видно какое-либо ребро типа (i) или типа (ii), то робот может различить h_i и h_j , не перемещаясь из γ_0 . Предположим, что все ребра F_{ij} , которые видимы из γ_0 , имеют тип (iii). Пусть ребро e грани F_{ij} имеет тип (i) или тип (ii). Множество $VP(e)$ точек F_{ij} , которые видимы с некоторой точки ребра e , является простым многоугольником (многоугольником видимости e) в пределах F_{ij} , который предположительно не включает точку γ_0 . В грани F_{ij} существует хорда $w(e)$, которая лежит на границе $VP(e)$ и отделяет e от γ_0 . Отрезок $w(e)$ можно назвать *окном* [4] (рис. 4).

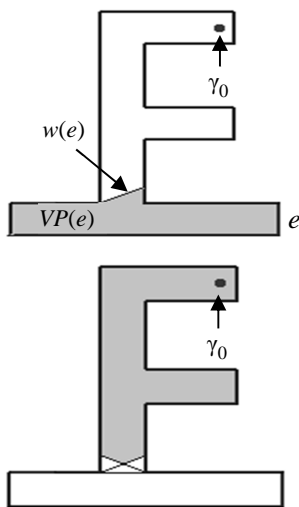


Рис. 4

На рис. 4 слева изображен многоугольник видимости $VP(e)$ и показаны ребро e типа (i) и соответствующая хорда (окно) $w(e)$. На рис. 4 справа показаны все окна $w(e)$ для ребер грани F_{ij} , имеющих тип (i) или (ii).

С учетом данных определений можно описать предлагаемый алгоритм локализации робота с использованием окон карты (АЛРОК). Пусть на входе алгоритма заданы многоугольник карты P и робот, помещенный в неизвестное начальное местоположение в P . Алгоритм АЛРОК состоит из следующих действий:

1. Вычислить многоугольник видимости V по данным сенсоров робота в текущем неизвестном начальном местоположении робота.
2. Сгенерировать множество гипотез H на карте P , которые соответствуют полученному многоугольнику видимости V .
3. Выбрать произвольную гипотезу h_i из H и соответствующую точку гипотетического местоположения как исходную для построения оверлеев.

Операции в пп. 4–7 осуществляются для всех активных (не отклоненных пока) гипотез h_j ($j = 1, 2, \dots, k'$).

4. Построить $Overlay(P_i, P_j)$ и $Intersection(P_i, P_j)$ экземпляров карты P_i и P_j .
5. Вычислить для $Intersection(P_i, P_j)$ связный компонент F_{ij} , содержащий стартовую точку.
6. Построить все окна компонента пересечения F_{ij} .
7. Вычислить средние точки всех окон в F_{ij} и найти среди них точку r_j , ближайшую к текущему положению робота, т. е. найти «ближайшее» окно в F_{ij} .

8. Вычислить $Intersection(P_1, P_2, \dots, P_{k'})$ для активных гипотез и связный компонент F , содержащий стартовую точку. Среди тех точек r_j , которые попадают в F , найти ближайшую к текущему положению робота точку r .

9. Переместить робота в точку r .
10. Устранить неподтвержденные гипотезы, сравнивая данные о текущем многоугольнике видимости, выдаваемые роботом в точке r , с данными о видимости, вычисленными во всех эквивалентных точках, соответствующих всем активным гипотезам.

Таблица 1

Шаг	Действия	Сложность алгоритма
1, 2	Генерация гипотез	$O(mn^2)$
4–7, 8	Построение оверлейных пересечений относительно выбранной гипотезы	$O(k'n \log n)$, где k' – число активных гипотез
6, 8	Построение «окон» связанного компонента оверлейного пересечения	$O(2k'n^2)$
7, 8	Обследование $2k'n$ точек на серединах окон. Вычисление кратчайших путей для определения ближайших окон	$k'O(n \log^* n) + 2k'nO(n) = k'O(n^2)$
10	Сравнение данных о многоугольниках видимости для активных гипотез и текущего положения робота	$2k'nO(n)$
Полная сложность		$O(mn^2) + \sum_{k'=1}^{k-1} [O(k'n \log n) + O(2k'n^2) + k'O(n^2) + 2k'nO(n)] = O(n^4)$

Пусть E – множество устраненных гипотез. Повторять шаги 3–10, пока в множестве активных гипотез $H-E$ не останется только одна гипотеза, которая и будет соответствовать истинному начальному местоположению робота.

Сложность этого алгоритма (по шагам и суммарно) приведена в табл. 1.

Экспериментальное исследование алгоритма основано на его программной реализации (Visual C++ 2010), а также реализациях алгоритмов [3] и [5] и сравнении таких их характеристик, как длина суммарного пути перемещения робота и время работы алгоритма локализации. При проведении эксперимента использовалась генерация карт различных модельных типов. Пример такой генерации и работы алгоритма локализации с исполь-

зованием окон в многоугольнике карты приведен на рис. 5. Здесь размер карты $n = 746$, а число гипотез $k = 7$.

В табл. 2 приведены численные результаты эксперимента для этих значений параметров $n = 746$ и $k = 7$. Здесь для трех приближенных алгоритмов локализации мобильного робота (алгоритм 1 – АЛРОК, алгоритм 2 [3] и алгоритм 3 – локализация робота с использованием триангуляции карты [5]) указаны № – номер гипотезы (в соответствии с нумерацией на рис. 5), d – длина пути, пройденного роботом при локализации, и t – время работы алгоритма. В рандомизированном алгоритме 2 [3] были использованы 2 варианта для количества случайных точек $X = 100$ и $X = 500$.

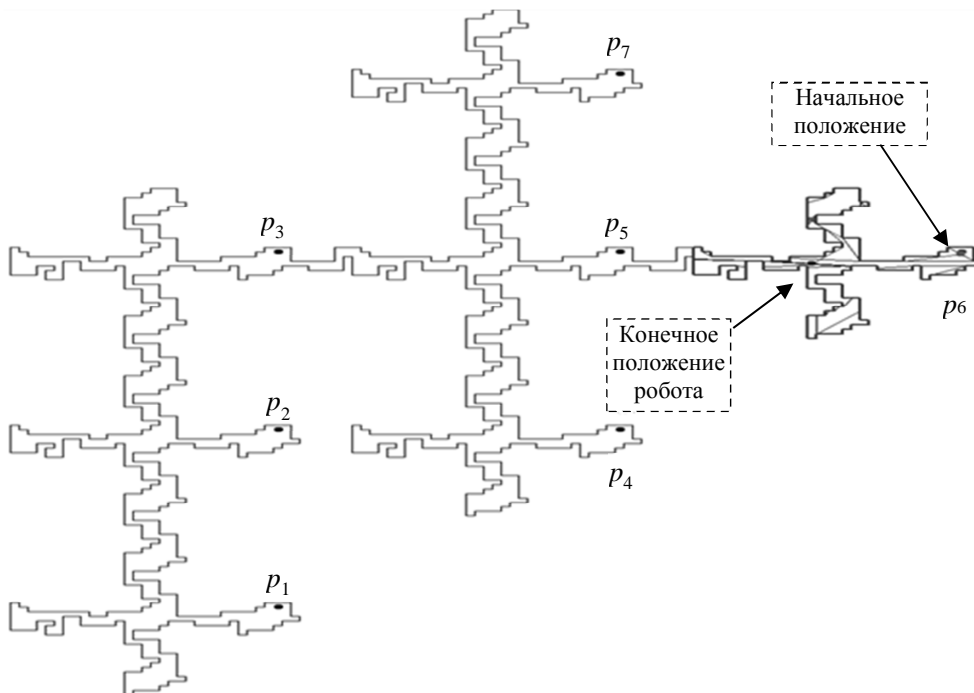


Рис. 5

Таблица 2

№	n = 746							
	d, о. е.				t, с			
	Алгоритм 1	Алгоритм 2		Алгоритм 3	Алгоритм 1	Алгоритм 2		Алгоритм 3
	X = 100	X = 500		X = 100	X = 500			
1	500,5	529,2	504,9	464,9	273,6	788,5	2890,0	1800,2
2	220,2	266,0	225,1	303,9	230,0	683,8	2462,1	1653,1
3	69,3	113,5	76,4	90,2	178,2	499,2	1694,3	1145,9
4	499,9	520,1	457,5	461,6	278,1	808,6	2409,2	1843,7
5	70,3	117,0	96,8	91,2	178,0	459,5	1665,0	1083,0
6	175,5	235,9	172,4	92,7	186,5	519,6	2364,6	973,8
7	175,5	245,5	240,4	302,1	190,3	494,3	2709,4	1560,9

Таблица 3

№	n = 746					
	d, о. е.			t, с		
	$s_i^{(2,1)} = \frac{d_i^{(2)}}{d_i^{(1)}}$	$s_i^{(2',1)} = \frac{d_i^{(2')}}{d_i^{(1)}}$	$s_i^{(3,1)} = \frac{d_i^{(3)}}{d_i^{(1)}}$	$s_i^{(2,1)} = \frac{t_i^{(2)}}{t_i^{(1)}}$	$s_i^{(2',1)} = \frac{t_i^{(2')}}{t_i^{(1)}}$	$s_i^{(3,1)} = \frac{t_i^{(3)}}{t_i^{(1)}}$
1	1,06	1,01	0,93	2,88	10,56	6,58
2	1,21	1,02	1,38	2,97	10,70	7,19
3	1,64	1,10	1,30	2,80	9,51	6,43
4	1,04	0,92	0,92	2,91	8,66	6,63
5	1,66	1,38	1,30	2,58	9,35	6,08
6	1,34	0,98	0,53	2,79	12,68	5,22
7	1,40	1,37	1,72	2,60	14,24	8,20
$\bar{s}^{(l,1)} = \frac{1}{7} \sum_{j=1}^7 s_j^{(l,1)}$	1,34	1,11	1,15	2,79	10,82	6,62

Для сравнения результатов эксперимента определим средние из зафиксированных значений d и t . Например, $\bar{d} = \frac{1}{7} \sum_{i=1}^7 d_i$, $\bar{t} = \frac{1}{7} \sum_{i=1}^7 t_i$,

$s_i^{(2,1)} = \frac{d_i^{(2)}}{d_i^{(1)}}$, $s_i^{(2',1)} = \frac{d_i^{(2')}}{d_i^{(1)}}$, $s_i^{(3,1)} = \frac{d_i^{(3)}}{d_i^{(1)}}$,

$s_i^{(2,1)} = \frac{t_i^{(2)}}{t_i^{(1)}}$, $s_i^{(2',1)} = \frac{t_i^{(2')}}{t_i^{(1)}}$ и $s_i^{(3,1)} = \frac{t_i^{(3)}}{t_i^{(1)}}$, где i – номер гипотезы, а в верхних индексах в скобках указаны номера алгоритмов. В табл. 3 приведены

отношения s_i для различных пар алгоритмов и их средние значения.

Различные типы линий графиков на рис. 6 и 7 соответствуют отношениям: 1 – отношению $\bar{s}^{(2,1)}$; 2 – отношению $\bar{s}^{(2',1)}$; 3 – отношению $\bar{s}^{(3,1)}$. На рис. 6 приведен график среднего отношения длин путей как функции от размера карты n , на рис. 7 – график среднего отношения времени работы алгоритмов как функции от размера карты n . Сравнительный анализ данных, соответствующих рис. 6, 7, показывает следующее.

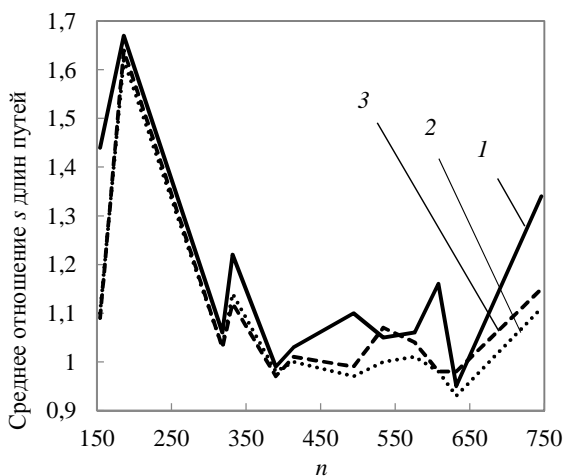


Рис. 6

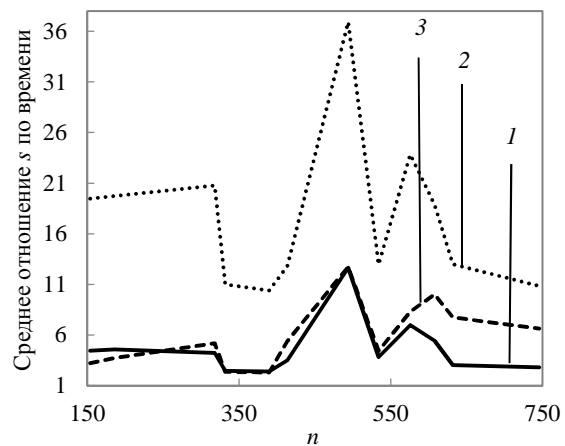


Рис. 7

Значения среднего пути для алгоритмов 1, 2' при $X = 500$ и 3 очень близки, а для алгоритма 2 при $X = 100$ несколько больше. Например, для случая $n = 746$, соответствующего табл. 1, имеем:

$$\bar{d}_1 = 244,46; \bar{d}_2/\bar{d}_1 = 1,34;$$

$$\bar{d}_{2'}/\bar{d}_1 = 1,11; \bar{d}_3/\bar{d}_1 = 1,15.$$

Здесь нижние индексы соответствуют номерам алгоритмов, а для алгоритма 2 использован индекс 2 при $X = 100$ и индекс 2' при $X = 500$.

Проведенные эксперименты с другими модельными конфигурациями карты показали аналогичные результаты и позволяют сделать вывод, что все рассмотренные алгоритмы обеспечивают сравнимую точность, но время работы алгоритма 1 меньше, чем у других алгоритмов, и он является более

предпочтительным по совокупности двух характеристик: времени работы и точности решения.

Самое большое время в среднем показывает алгоритм 2', а наименьшее время – алгоритм 1. При $n = 746$, например,

$$\bar{t}_1 = 216,39 \text{ с}; \text{ а } \bar{t}_2/\bar{t}_1 = 2,79;$$

$$\bar{t}_{2'}/\bar{t}_1 = 10,82; \bar{t}_3/\bar{t}_1 = 6,62.$$

Однако время работы и асимптотическая сложность $O(n^4)$ предложенного алгоритма все-таки велики. Их уменьшения можно ожидать за счет оптимизации наиболее трудоемких шагов алгоритма, а также за счет их реализации на параллельных структурах, например с использованием графических ускорителей и гетерогенных вычислительных структур [8], [9].

СПИСОК ЛИТЕРАТУРЫ

1. Guibas L. J., Motwani R., Raghavan P. The robot localization problem// SIAM J. Comput. 1997. Vol. 26. P. 1120–1138.
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// Internat. 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. Экспериментальный анализ алгоритмов локализации мобильного робота// Изв. СПбГЭТУ «ЛЭТИ». 2014. № 1. С. 45–51.
6. Computational Geometry: Algorithms and Applications / M. De Berg, O. Cheong, M. Van Kreveld, M. Overmars. Berlin, Heidelberg: Springer-Verlag, 2008. 386 p.
7. Скворцов А. В. Построение объединения, пересечения и разности произвольных многоугольников в среднем за линейное время с помощью триангуляции // Вычислительные методы и программирование. 2002. Т. 3. С. 116–123.
8. Qi M., Cao T.-T., Tan T.-S. Computing 2D Constrained Delaunay Triangulation Using Graphics Hardware // Technical Report # TRB3/11. 2011. March / School of Computing; National University of Singapore. Singapore, 2011. 9 p.
9. Сандерс Дж., Кэндрот Э. Технология CUDA в примерах: введение в программирование графических процессоров. М.: ДМК Пресс, 2011. 232 с.

Dao Duy Nam, S. A. Ivanovskiy

Saint-Petersburg State Electrotechnical University «LETI»

APPROXIMATION ALGORITHM LOCALIZATION MOBILE ROBOT WITH USING WINDOWS IN POLYGON MAP

Approximation algorithm for the robot localization problem with the map in the form of a simple polygon is considered. Hypotheses localization correspond to copies maps from assumed mark of robot's position. The algorithm is based on using overlay and intersection copies maps. At the intersection so-called windows are segregated, "looking" in which the robot eliminates false hypotheses. On the basis of their program implementation conducted experimental studies of these algorithms. The numerical results and their interpretation are given.

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