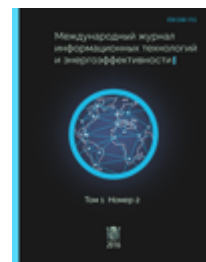




Международный журнал информационных технологий и энергоэффективности

Сайт журнала: <http://www.openaccessscience.ru/index.php/ijcse/>



УДК 681.3.019

СПОСОБ И АЛГОРИТМЫ ПОИСКА МАРШРУТОВ ПРОТЯЖЕННЫХ ОБЪЕКТОВ НА ЦИФРОВОЙ КАРТЕ МЕСТНОСТИ

¹Лашков А.А., ²Зернов М.М.

Филиал ФГБОУ ВО «Национальный исследовательский университет «МЭИ» в г. Смоленске, Россия, (214013, г. Смоленск, Энергетический проезд, 1), e-mail: ¹lahkandr010@mail.ru, ²zmmioml@yandex.ru

Данная статья посвящена рассмотрению возможных путей решения задачи поиска маршрута протяженного объекта с использованием навигационного графа. Предложены этапы и особенности реализации способа поиска маршрутов протяженных объектов на цифровой карте местности. Выделены основные этапы поиска маршрутов на цифровой карте местности. Дано описание формирования представления цифровой карты местности с учетом коэффициента проходимости объектов этой карты. Дано описание и формирование навигационного графа по скелету свободной области цифровой карты местности. Описано получение скелета свободной области из диаграммы Вороного, построенного на основе сегментного представления препятствия. В роли препятствий выступают объекты цифровой карты местности, которые соответствуют выбранному коэффициенту проходимости. Сформулирован модифицированный способ поиска кратчайшего пути на графах A-star.

Ключевые слова: навигационный граф, КД-дерево, скелет, диаграмма Вороного.

METHOD AND ALGORITHMS FOR FINDING ROUTES OF EXTENDED OBJECTS ON A DIGITAL TERRAIN MAP

¹Lashkov A.A., ²Zernov M.M.

^{1,2}Smolensk Branch of the National Research University "Moscow Power Engineering Institute", Smolensk, Russia (214013, Smolensk, Energeticheskyy proezd, 1), e-mail: ¹lahkandr010@mail.ru, ²zmmioml@yandex.ru

This article is devoted to the consideration of possible ways to solve the problem of finding the route of an extended object using a navigation graph. The stages and features of the implementation of the method of searching for routes of extended objects on a digital terrain map are proposed. The main stages of route search on the digital map of the area are highlighted. The description of the formation of the representation of a digital terrain map is given, taking into account the coefficient of patency of the objects of this map. The description and formation of the navigation graph on the skeleton of the free area of the digital terrain map is given. It is described how to obtain a skeleton of a free area from a Voronoi diagram constructed on the basis of a segmental representation of an obstacle. The objects of the digital terrain map that correspond to the selected cross-country coefficient act as obstacles. A modified method for finding the shortest path on A-star graphs is formulated.

Keywords: navigation graph, KD-tree, skeleton, Voronoi diagram.

Введение

Данная работа является продолжением статьи «Применение навигационного графа для решения задачи поиска маршрута габаритного объекта по цифровой карте местности» [1]. По результатам анализа способов и алгоритмов построения маршрута на цифровой карте

местности(ЦКМ), рассмотренных в статьях [3-6], можно выделить основные этапы поиска маршрутов на ЦКМ, описанные ниже.

1. Формирование представления карты.

Включает в себя этапы обработки карты такие как:

- выбор объектов на основе характеристик (площади, проходимости, типа и т.д.);
- корректировка объектов (например, обработка самопересечение площадных объектов);
- упрощение карты (аппроксимация границ объектов, объединение объектов и т.д.).

2. Формирование графа для поиска пути.

Чаще всего для поиска пути используется граф, например, граф контрольных точек(waypoints)[7], граф видимости[3] и т.д.

3. Нахождение пути на графе.

Существует множество алгоритмов поиска пути на графе, наиболее известны алгоритмы Дейкстры и A-star [8].

4. Оптимизация маршрута.

Данный этап заключается в уточнение маршрута, например в уменьшении протяженности, уменьшении или увеличении количества узлов в зависимости от точности и т.д.

Способ и алгоритмы поиска маршрутов протяженных объектов на цифровой карте местности

На основе вышеперечисленных пунктов предлагается способ поиска маршрутов протяженных объектов на цифровой карте местности, состоящий из следующих этапов.

1. Формирование карты проходимости на основе ЦКМ, выделение свободных областей и препятствий. Формирование карты препятствий.

2. Формирование скелета свободной области на основе диаграммы Вороного и выборка вершин данной диаграммы, не лежащих на препятствиях.

3. Формирование навигационного графа, позволяющей охарактеризовать свободную область. Выбор вершин для графа из диаграммы Вороного, на основе расстояния до ближайшего препятствия, и их соединение.

4. Поиск кратчайшего пути с учетом ширины объекта на основе алгоритма A-star во взвешенном графе, адаптированного к использованию оценки расстояния до препятствия от вершины навигационного графа.

5. Улучшение построенного пути за счет удаления лишних вершин и итерационного уточнения пути.

Формирование карты проходимости на основе ЦКМ, выделение свободных областей и препятствий. Формирование карты препятствий

Начальный этап заключается в формировании карты проходимости на основе ЦКМ, выделении свободных областей и препятствий. Необходимо преобразовать ЦКМ в форму удобную для построения навигационного графа, а для этого нужно выделить область, занимаемую препятствиями (бездорожьем) и свободную область. В результате решения,

данной задачи формируется список объектов карты, представляющие собой многоугольники со своими коэффициентами проходимости, изменяемые от 0(свободен) до 1(непроходим).

Для этого различным типам объектов ЦКМ (слоям) должны быть сопоставлены значения коэффициентов проходимости, т.е. заранее составлены справочники с коэффициентами проходимости для каждого объекта. Опираясь на данные справочника, составляются списки объектов-препятствий (например, выбрать все объекты как препятствия, значения коэффициента проходимости которых больше 0.5).

После того как выбраны объекты-препятствия необходимо создать карту препятствий (набор непересекающихся многоугольников). Так как цифровая карта местности может содержать самопересечение объектов и пересечение объектов между собой, то необходимо заменять каждый многоугольник с самопересечением на внешний контур или разделять на несколько многоугольников, и объединять пересекающиеся объекты. Данные действия способствуют корректному построению скелета свободной области.

Таким образом, можно предложить следующий алгоритм формирования карты препятствий на основе ЦКМ.

Итак, есть типы перемещаемых объектов (человек, колёсный транспорт, гусеничный транспорт, и т.д.). Каждому типу перемещаемого объекта для каждого типа объектов карты сопоставлена своя проходимость. У каждого типа перемещаемого объекта свой порог проходимости.

1. Установка верхней границы коэффициента проходимости, в зависимости от типа перемещаемого объекта.
2. Т.к. ЦКМ имеет разные типы объектов(леса, водоемы, кварталы и т. д.) сгруппированные по слоям, то формируются наборы объектов для каждого слоя.
3. В каждой группе удаляются объекты, коэффициент которых больше установленного порога для указанного типа объекта, т.е. «проходимые» объекты.
4. Проход по каждой группе и удаление самопересечения(рисунок 1.а и 1.б)
5. Объединение в каждой группе объектов, которые пересекаются (рисунок 1.в).
6. Объединение объектов между группами, также сопровождается устранением пересечений.

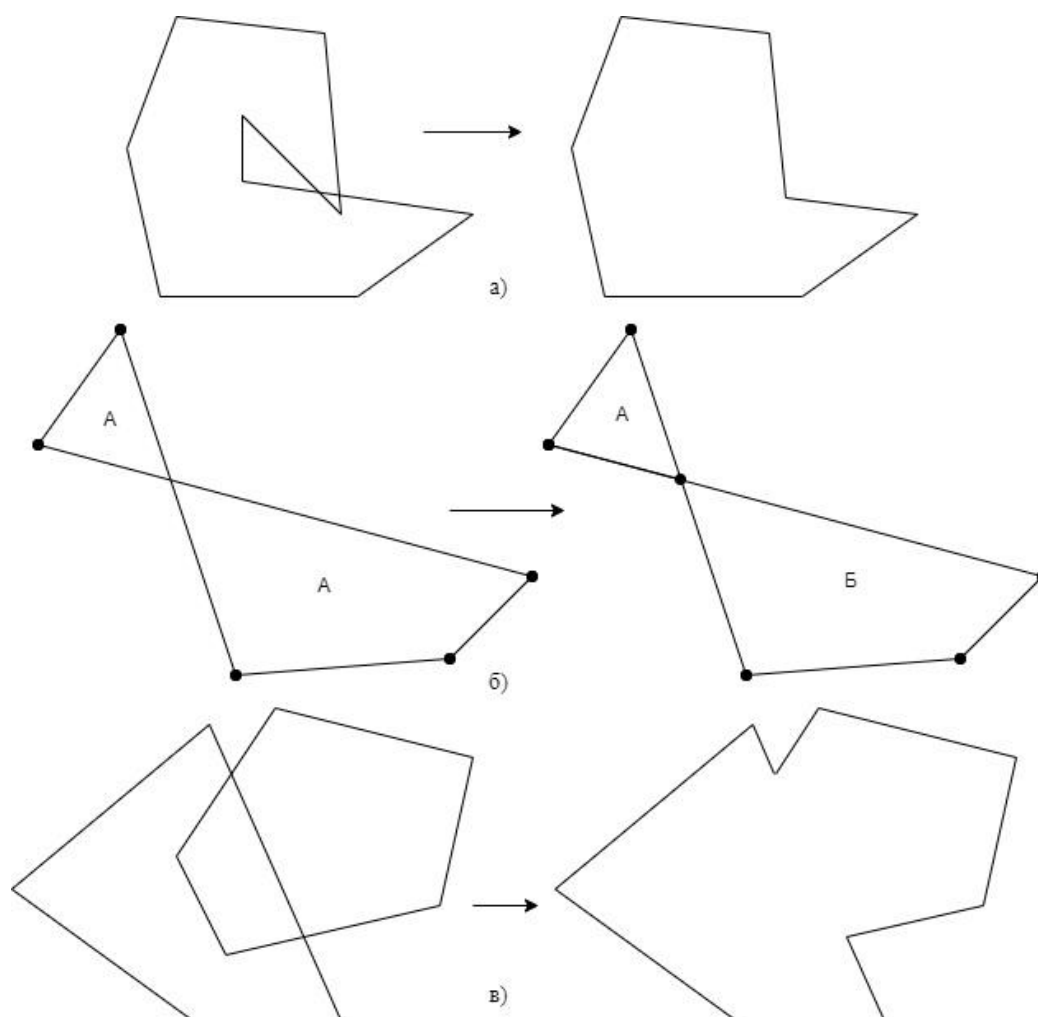


Рисунок 1 – Операции устранения самопересечения и операция объединения пересекающихся объектов (*а* – замена многоугольника с самопересечением на его контур, *б* – деление многоугольника с самопересечением на несколько многоугольников, *в* – объединение пересекающихся объектов).

Операции устранения самопересечения и операция объединения пересекающихся объектов выполняются в соответствии с алгоритмом, рассмотренным в [2]. Таким образом получаем набор непроходимых непересекающихся многоугольников, который представляет карту препятствий.

Формирование скелета свободной области

Строим скелет карты препятствий на основе Диаграммы Вороного. Используем алгоритм построения для сегментов[3], т.е. необходимо получить из набора препятствий(карты препятствий) набор сегментов, для чего достаточно обойти каждый многоугольник объекта-препятствия и сформировать сегменты. Ребра диаграммы Вороного используются в качестве элементов скелета (рисунок 2).

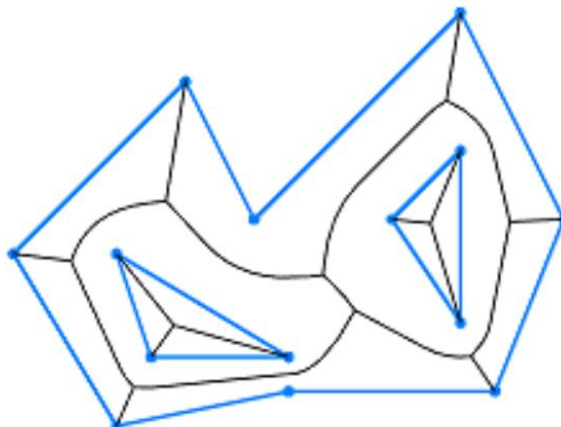


Рисунок 2 – Пример скелета (черные линии)

В результате построения ребра диаграммы Вороного будут лежать, как внутри препятствий, так и на внешней совокупности свободных областей, значит следующим шагом будет определение ребер диаграмм Вороного, которые лежат вне препятствий, т.е. на свободной области, такие ребра и будут являться скелетом карты препятствий. Т.к. все полученные ребра диаграммы Вороного формируют набор компонент связности, то достаточно будет начать проходить все узлы таких графов и при этом определять, лежат ли они на препятствии или нет. Допустим v – какой-то узел, если v лежит на препятствии, то обходим граф и помечаем все узлы данного графа, как узлы, которые лежат на препятствии, тоже самое делаем и если v лежит на свободной области.

Для каждого из оставшихся узлов находим минимальное расстояние до ближайшего препятствия. Пусть - узел v_i с наибольшим минимальным расстоянием l_m , убираем узел v_i из рассмотрения и все узлы, которые покрывает окружность с центром в узле v_i и радиусом l_m , после чего переходим к следующему нерассмотренному узлу. Повторяем до тех пор, пока узлы не закончатся. Далее формируем новые связи(ребра) между выбранными узлами на основе старых связей, т.е. допустим была последовательность связанных узлов $v_i, v_{i+1}, v_{i+2}, \dots, v_{i+n}$, допустим v_i и v_{i+n} имеют наибольшие минимальные расстояния до ближайшего препятствия и покрывают узлы $v_{i+1}, v_{i+2}, \dots, v_{i+n-1}$, то формируем связь между v_i и v_{i+n} .

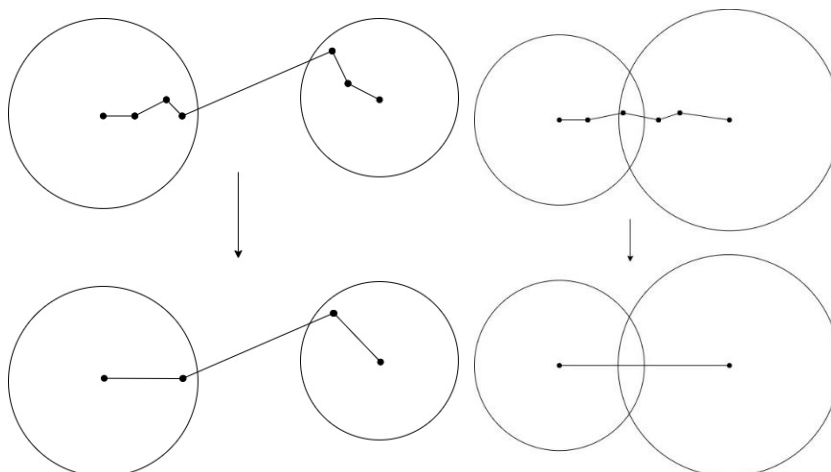


Рисунок 3 – Устранение покрытых вершин окружностями вершин

Как видно из рисунка 3, если окружности пересекаются, то вершины, являющиеся центрами этих окружностей, соединяются одним ребром, иначе маршрутом из 3-х. Таким образом уменьшается количество узлов и образуется навигационный граф, который используется для построения итогового маршрута.

Поиск кратчайшего пути с учетом ширины объекта на основе алгоритма A-star

По сформированному навигационному графу ищем путь, используя алгоритм поиска кратчайшего пути на графе A-star с учетом ширины объекта. Используется стандартный алгоритм A-star, но с дополнительной проверкой на то пройдет ли данный объект по ребру или нет. Формируется описывающий прямоугольник вокруг ребра, т.е. ребро как бы «раздувается» (рисунок 4) во все стороны на половину длины ширины объекта, а после проверяется пересекается ли построенный многоугольник с ближайшими препятствиями, если да, то он подходит для дальнейшего рассмотрения в алгоритме A-star, иначе убираем данное ребро из рассмотрения. Для быстрого нахождения ближайшего препятствия можно построить КД-дерево[9], позволяющее ускорить поиск ближайшего препятствия к точке. Таким образом формируется путь для объекта с определённой шириной, по которому он может перемещаться, не задевая препятствий.

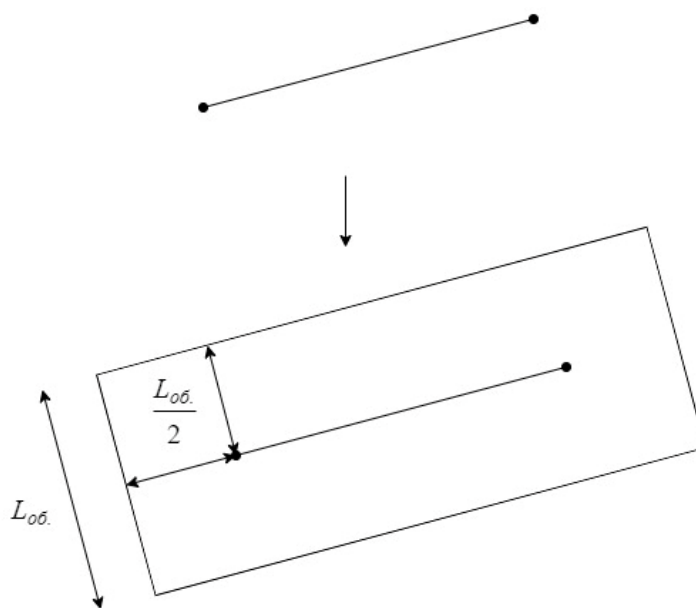


Рисунок 4 – «Раздутие» ребра

Модифицированный алгоритм поиска маршрута имеет шаги, описанные ниже.

Обозначим веса вершин:

- G - стоимость передвижения из начальной вершины к данной вершине, следуя найденному пути к данной вершине;
- F – стоимость равная сумме G и функции $h(x)$ для данной вершины.

Шаги алгоритма A^* :

1) Формирование «открытого» списка (в начале работы алгоритма содержит начальную вершину), в котором находятся вершины, которые необходимо просмотреть, и «закрытого» списка, который содержит просмотренные вершины.

2) Выбрать в качестве текущей вершины «С» вершину с наименьшим значением F и поместить в закрытый список.

3) Нахождение всех соседних вершин «N», который не входят в закрытый список, и соответствующих инцидентных ребер «E».

4) Каждое ребро «раздувается» (рисунок 4) и проверяется пересечение с ближайшими препятствиями, если пересечения есть, то убираем из рассмотрения соответствующую соседнюю вершину «N», т.е. помещаем ее в закрытый список.

4) Расчет для каждого оставшегося в рассмотрении соседа «N» временного G_TEMP равное сумме веса G текущей вершины «С» и веса ребра «С»-«N». Если «N» находится в открытом списке и его текущее G больше временного G_TEMP , или «N» не находится в открытом списке, то соответственно обновляем в открытом списке или добавляем в открытый список вершину «N», где $G = G_TEMP$ и рассчитываем F , еще указываем для «N» вершину, из которой пришли («С»).

5) Необходимо повторить описанные выше шаги с шага 2 до тех пор, пока не будет посещен пункт назначения.

После нахождения кратчайшего пути применяем алгоритмы улучшения пути за счет удаления лишних вершин и итерационного уточнения траектории пути, рассмотренные в [4].

Для каждой вершины v_i проверяется возможность создания ребра между вершиной v_i и v_{i+2} . Формируем ребро и определяем, лежит ли оно на расстоянии в половину ширины перемещаемого объекта от ближайших препятствий, если нет, то удаляется вершина v_{i+1} , соединяются ребром вершины v_i и v_{i+2} , и следующей для рассмотрения берется вершина v_{i+2} , если да, сохраняется вершина v_{i+1} и рассматривается как следующая вершина для обработки. Данные действия продолжаются пока количество вершин пути уменьшается.

Дальнейшим шагом будет сокращение длины пути путем как можно близкого огибания препятствия («натягивание пути»). Допустим берем вершину v на кратчайшем пути, берем инцидентные для выбранной вершины ребра e_1 и e_2 . Вдоль ребер e_1 и e_2 от вершины v откладываем заранее найденный шаг h , при этом получаем две вершины $v_{1(1h)}$ и $v_{2(1h)}$ на соответствующих ребрах. Далее проверяется возможность создания ребра между вершинами $v_{1(1h)}$ и $v_{2(1h)}$, как это было описано при удалении избыточных вершин, если вершины можно соединить, то продолжаем откладывать вдоль ребер e_1 и e_2 вершины $v_{1(2h)}$ и $v_{2(2h)}$ с шагом $2h$ и также пытаемся их соединить. Процесс продолжается до тех пор, пока мы можем соединять вновь отложенные вершины, или пока не будет достигнута конечная точка одного из ребер. После чего мы заменяем вершину v последними вершинами $v_{1(ih)}$ и $v_{2(ih)}$, которые прошли проверку на соединение между собой, и переходим к следующей вершине, если таких вершин нет, то сохраняем вершину v и переходим к следующей вершине. Данные действия продолжаются до тех пор, пока вершины заменяются, как только замены прекращаются, уменьшается шаг h , после чего продолжается выполнение алгоритма с новым шагом, пока шаг не достигнет определенного допустимого значения (устанавливается пользователем или рассчитывается, изначальный шаг можно взять как среднее арифметическое минимальных расстояний вершин пути до препятствий деленный на коэффициент). После чего повторяем алгоритм удаления избыточных вершин.

В итоге получаем следующий набор шагов, изображенный на рисунке 5.

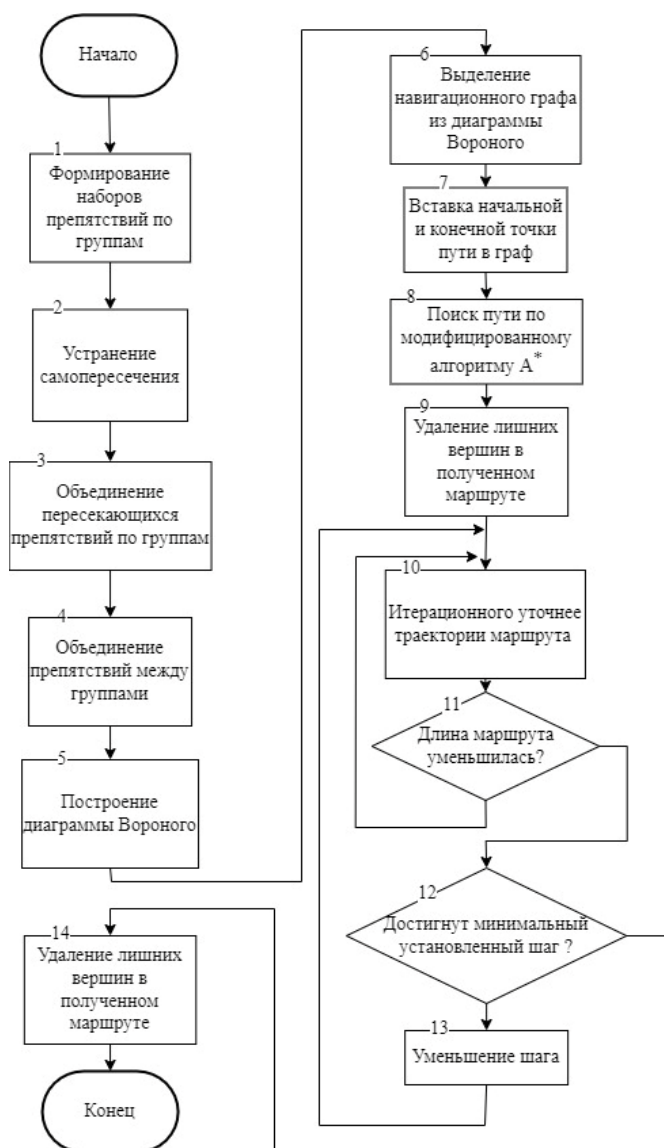


Рисунок 5 – Способ поиска маршрутов протяженных объектов на цифровой карте местности

Заключение

Предложен способ поиска маршрутов протяженных объектов на цифровой карте местности, характеризующиеся следующими особенностями:

- поиск маршрута осуществляется на цифровой карте местности с учетом коэффициента проходимости объектов;
- учет ширины объекта при формировании маршрута;
- применением навигационного графа, построенного на основе скелета проходимых участков;
- применением модифицированного алгоритма поиска пути на графах A-star;
- алгоритмами уточнения траектории построенного маршрута.

Список литературы

1. Лашков А.А., Зернов М.М. Применение навигационного графа для решения задачи поиска маршрута габаритного объекта по цифровой карте местности // Международный журнал информационных технологий и энергоэффективности, [S.1.], v. 7, n. 1(23), С. 3-16, апр. 2022. ISSN 2500-1752
2. THE BOOST.POLYGON LIBRARY [Электронный ресурс] URL: https://www.boost.org/doc/libs/1_66_0/libs/polygon/doc/index.htm (дата обращения 15.05.22)
3. Mark B. et al. Computational geometry algorithms and applications. – Springer, 2008.
4. Bhattacharya P., Gavrilova M. L. Roadmap-based path planning-using the voronoi diagram for a clearance-based shortest path //IEEE Robotics & Automation Magazine. – 2008. – Т. 15. – №. 2. – С. 58-66.
5. Pettré J., Grillon H., Thalmann D. Crowds of moving objects: Navigation planning and simulation //ACM SIGGRAPH 2008 classes. – 2008. – С. 1-7.
6. Pettré J. et al. Real-time navigating crowds: scalable simulation and rendering //Computer Animation and Virtual Worlds. – 2006. – Т. 17. – №. 3-4. – С. 445-455.
7. Toma A. I. et al. Waypoint Planning Networks //arXiv preprint arXiv:2105.00312. – 2021.
8. Rachmawati D., Gustin L. Analysis of Dijkstra’s Algorithm and A* Algorithm in Shortest Path Problem //Journal of Physics: Conference Series. – IOP Publishing, 2020. – Т. 1566. – №. 1. – С. 012061.
9. Анатолия КД-Деревьев [Электронный ресурс] URL: <https://habr.com/ru/post/312882/> (дата обращения 5.05.22)

References

1. Lashkov A.A., Zernov M.M. Application of the navigation graph to solve the problem of finding the route of a dimensional object on a digital terrain map // International Journal of Information Technology and Energy Efficiency, [S.L.], v. 7, No. 1(23), pp 3-16, Apr. 2022. ISSN 2500-1752
 2. THE BOOST.POLYGON LIBRARY [E-resource] URL: https://www.boost.org/doc/libs/1_66_0/libs/polygon/doc/index.htm (Address date 15.05.22)
 3. Mark B. et al. Computational geometry algorithms and applications. – Springer, 2008.
 4. Bhattacharya P., Gavrilova M. L. Roadmap-based path planning-using the voronoi diagram for a clearance-based shortest path //IEEE Robotics & Automation Magazine. – 2008. – Т. 15. – №. 2. – pp. 58-66.
 5. Pettré J., Grillon H., Thalmann D. Crowds of moving objects: Navigation planning and simulation //ACM SIGGRAPH 2008 classes. – 2008. – pp 1-7.
 6. Pettré J. et al. Real-time navigating crowds: scalable simulation and rendering //Computer Animation and Virtual Worlds. – 2006. – Т. 17. – №. 3-4. – pp. 445-455.
 7. Toma A. I. et al. Waypoint Planning Networks //arXiv preprint arXiv:2105.00312. – 2021.
 8. Rachmawati D., Gustin L. Analysis of Dijkstra’s Algorithm and A* Algorithm in Shortest Path Problem //Journal of Physics: Conference Series. – IOP Publishing, 2020. – Т. 1566. – №. 1. – pp. 012061.
 9. Anatomy of KD-Trees [E-resource] URL: <https://habr.com/ru/post/312882/> / (Address date 5.05.22)
-