

Бочаров Н.А.¹, Парамонов Н.Б.², Сапачев И.Д.¹

¹ АО «МЦСТ», Москва, Россия

² ПАО «ИНЭУМ им И.С. Брука», Москва, Россия

РЕАЛИЗАЦИЯ АЛГОРИТМОВ ГРУППОВОГО УПРАВЛЕНИЯ НА ЯЗЫКЕ JAVA В СРЕДЕ ОС «ЭЛЬБРУС»*

АННОТАЦИЯ

Цель работы — разработка программного комплекса для отладки и демонстрации задач управления движением групп роботов, а также наблюдения за группой роботов с использованием нескольких камер с использованием Java в среде «Эльбрус». Показана возможность использования вычислительных средств ряда «Эльбрус» в качестве вычислителей для задач управления и наблюдения за группой роботов.

КЛЮЧЕВЫЕ СЛОВА

Моделирование движения; поиск пути на графе; моделирование алгоритмов группового управления.

Nikita Bocharov¹, Nikolay Paramonov², Ilya Sapachev¹

¹ «MCST», Moscow, Russia

² «INEUM im. I.S. Bruka», Moscow, Russia

IMPLEMENTATION OF ALGORITHMS OF GROUP CONTROL ON JAVA LANGUAGE IN OS «ELBRUS» ENVIRONMENT

ABSTRACT

The purpose of this work is developing the software for debugging and demonstration of tasks of motion control of robots group and monitoring a robots group using multiple cameras using Java language in the OS «Elbrus» environment. The article displays possibility of using computer systems of «Elbrus» series as computer for robots control and monitoring tasks.

KEYWORDS

Motion modelling; search path on the graph; modelling algorithms of group control.

Введение

Современной тенденцией развития наземной робототехники является постепенный переход от дистанционно управляемых к полуавтономным, а в перспективе - к автономным робототехническим комплексам (РТК). При этом важным направлением является применение не единичных образцов, а группировок роботов, способных более эффективно выполнять определенные задачи. В связи с этим актуальным является создание интеллектуальных систем управления группировками автономных наземных робототехнических комплексов. Существенным, но не решенным вопросом создания систем управления РТК является оснащение вычислительной техникой, разработанной на базе отечественных микропроцессоров и программного обеспечения отечественной разработки.

Поскольку робототехника является одним из перспективных направлений применения вычислительных комплексов (ВК) и программного обеспечения семейства «Эльбрус» [1], то целью данной работы стала оценка возможностей существующих СВТ на базе микропроцессоров «Эльбрус» для систем управления РТК на колесных и гусеничных шасси, а так же создание модели алгоритмов для управления и наблюдения за группой роботов.

Результатом данной работы должен являться программный комплекс, позволяющий моделировать поведение группы автономных роботов на местности и моделировать работу группы средств наблюдения за роботами. Данный программный комплекс должен позволять определять

* Труды I Международной научной конференции «Конвергентные когнитивно-информационные технологии» (Convergent'2016), Москва, 25-26 ноября, 2016

временные и нагрузочные характеристики, которые позволят проводить работы по оптимизации программного обеспечения в ОС «Эльбрус».

Постановка задачи

В предыдущей работе [2] авторами было проведено моделирование алгоритмов поиска пути отдельным роботом и алгоритмов технического зрения. Движение робота было сведено к поиску пути на графе, соответственно был проведен анализ существующих алгоритмов поиска пути на графе. В качестве используемого алгоритма был выбран алгоритм A^* с некоторыми изменениями в эвристической функции для учета дополнительных параметров. Для повышения реалистичности модели в нее был включен учет таких параметров робота, как: скорость, ускорение, радиус поворота, радиус обнаружения препятствий. В данной работе планировалось расширить данную модель для моделирования уже не одного, а группы из нескольких роботов.

Формально, в задаче обрабатываются три множества: множество роботов, множество целей и множество внешних факторов. В разрабатываемой программе должно быть моделирование всех этих множеств.

Задача поиска пути

Задача поиска пути была сведена к задаче поиска пути на графе из узла-старта до узла-финиша [3, 4]. Пара $(V(G), E(G))$ называется графом, если $V(G)$ — непустое конечное множество элементов, называемых узлами, а $E(G)$ — конечное множество неупорядоченных пар различных элементов из $V(G)$, называемых рёбрами [5]. При этом поиск необходимо осуществлять, учитывая проходимость различных опорных поверхностей и радиус поворота робота.

Для задачи поиска пути роботом также необходимо учитывать длину пути, что можно обозначить через вес рёбер. При этом, если робот может переместиться из точки A в точку B , то не обязательно, что он может и переместиться из точки B в точку A . Например, если точка A — это точка на открытой местности, а точка B — точка у стены с азимутом, направленным прямо в стену. Таким образом, для поставленной задачи подходит использование взвешенного ориентированного графа.

Узел графа по определению представляет собой элемент графа, обозначающий объект любой природы. В данной задаче узел графа обозначает область на местности, в которую робот имеет возможность встать. Так как необходимо учитывать радиус поворота робота, то необходимо и учитывать азимут робота при построении пути. Таким образом, узел графа задается не только координатами x, y , но и азимутом.

Ребром графа обозначается траектория между узлами, которые она соединяет, по которой может перемещаться робот. При этом траектория строится с учетом радиуса поворота робота. Так как при поиске пути надо учитывать проходимость различных опорных поверхностей, то вес ребра обозначим как средний коэффициент проходимости на всём протяжении траектории данного ребра.

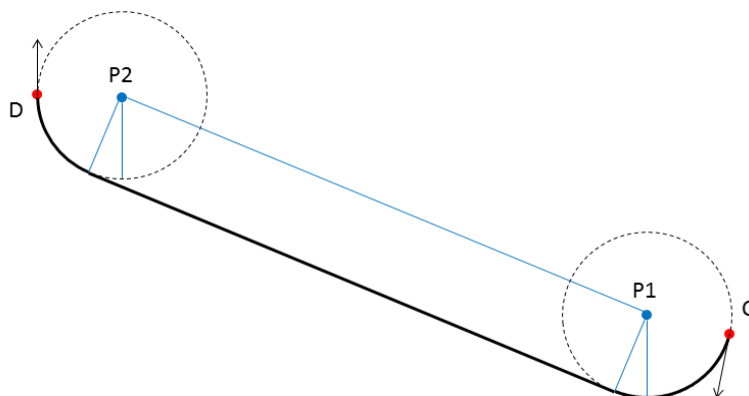


Рисунок 1. Обход окружностей в одном направлении

Путь робота проходит от узла к узлу. Но эти узлы не могут быть соединены ребрами, являющимися прямыми траекториями, так как робот не имеет возможность мгновенно развернуться в нужном направлении. Поэтому между узлами необходимо находить траекторию с

плавными поворотами, соответствующими радиусу поворота робота. При этом необходимо проверять эти траектории на наличие препятствий на пути.

Учет радиуса поворота происходит при построении траектории от узла к узлу. Соответственно, каждый узел, помимо координат, характеризовался еще и азимутом. Итоговая траектория между двумя узлами состоит из трех сегментов - двух дуг и отрезка прямой. При этом могут возникнуть два случая, различающиеся подсчетом основных точек: когда обе окружности обходятся в одном направлении и наоборот. Выбирается та, которая имеет наименьшую длину. Варианты обхода окружностей изображены на рисунках 1 и 2.

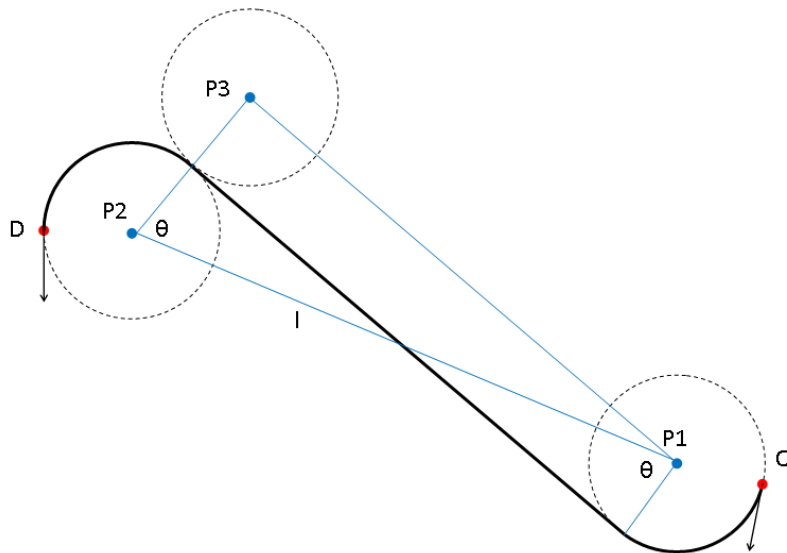


Рисунок 2. Обход окружностей в противоположных направлениях

Задача расчета проходимости

Задача поиска наилучшего пути могла бы сводиться к задаче поиска кратчайшего пути. Но в реальности данное решение не всегда приемлемо. Путь может проходить по различным опорным поверхностям. Например, есть более длинный путь по асфальтированной дороге и более короткий по песку. Заранее известно, что робот движется по песку намного медленнее, чем по асфальту и поэтому лучший путь был бы более длинным.

Робот оценивает проходимость двумя методами: посредством анализа загружаемой в него карты и посредством своих датчиков.

Первый метод формирует карту проходимости на основе загружаемой в робота карты местности. Карта местности отображает виды опорных поверхностей, а карта проходимости отображает коэффициенты проходимости этих поверхностей.

Второй метод анализирует реальную проходимость и меняет карту проходимости, если реальный коэффициент отличается от коэффициента на карте.

Расчет карты проходимости

При расчете карты проходимости карта местности в формате OpenStreetMap разбивается на квадраты, соответствующие размерам робота. В каждом таком квадрате считается средний коэффициент проходимости по каждому пикселю. Если хотя бы один пиксель в этой области окажется непроходимым, тогда вся область считается непроходимой. Иначе, робот помечает эту область рассчитанным средним коэффициентом проходимости.

Но с растровыми картами могут возникнуть некоторые проблемы. Во-первых, между разными областями на карте в результате сглаживания, некоторые пиксели не соответствуют ни одному цвету из легенды. Во-вторых, в результате того же сглаживания, некоторые пиксели могут обозначать непроходимую область, хотя на самом деле это не так. И в-третьих, некоторые пиксели могут немного отличаться от цветов в легенде (обычно не больше чем на 2 в одном из каналов). Эти проблемы решаются путем сглаживания, использование скользящего окна и введения небольшой погрешности. Пример формирования карты местности и сформированной карты проходимости приведен на рисунке 3.

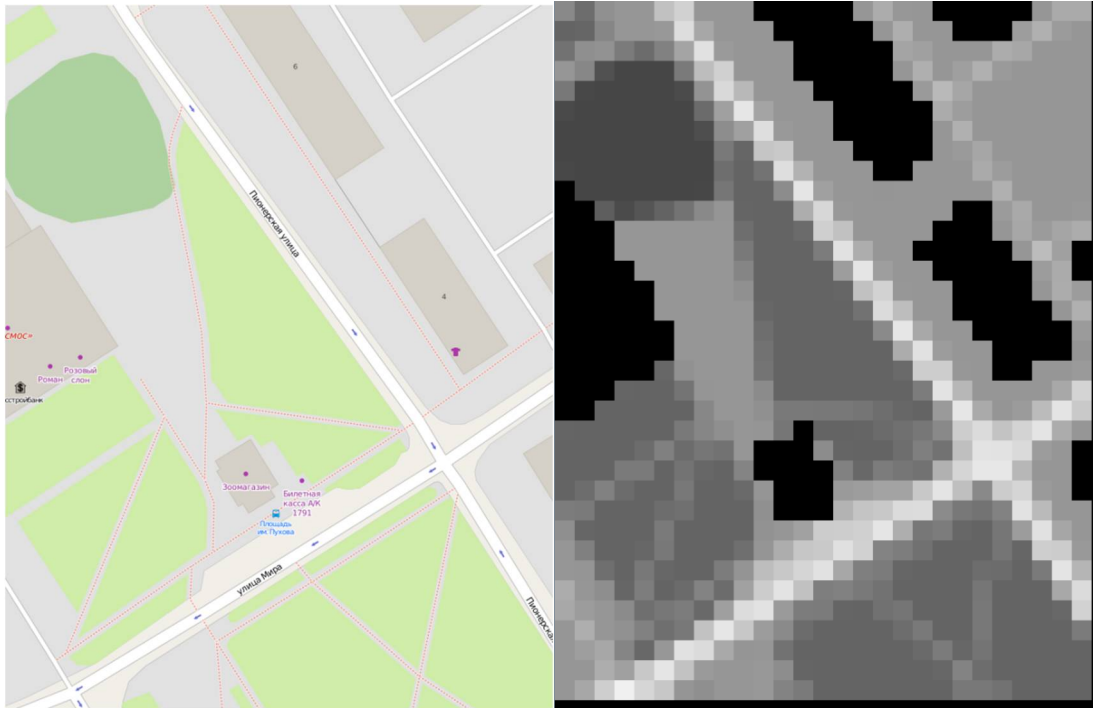


Рисунок 3. Пример карты местности и карты проходимости

Алгоритм поиска пути

Алгоритмы поиска пути ищут путь на графе из стартового узла в узел-финиш. При этом, в зависимости от алгоритма, путь может быть кратчайшим. Кроме того, некоторые алгоритмы позволяют учитывать вес узлов.

Алгоритм A* считается одним из лучших алгоритмов поиска пути [6]. Он объединяет в себя достоинства двух алгоритмов: учет длины пути из алгоритма Дейкстры и учет эвристической функции из алгоритма «лучший первый».

Алгоритм A* [7] использует формулу эвристики, которой в общем случае имеет вид:

$$f(n) = g(n) + h(n),$$

где $f(n)$ — значение оценки для узла n , $g(n)$ — стоимость пути из узла-старта в узел n , $h(n)$ — эвристическое приближение стоимости пути из узла n в узел-финиш.

Функция $h(n)$ должна быть допустимой эвристической оценкой, то есть не должна переоценивать расстояние до узла-финиша. Одним из способов задания такой функции является длина прямой, соединяющий узел n и узел-финиш.

Алгоритм работает аналогично алгоритму Дейкстры, где вместо длины пути учитывается функция $f(n)$, а когда узлу $n1$ устанавливается родитель $n2$, пересчитывается функция $g(n)$ следующим образом:

$$g(n1) = g(n2) + d(n1, n2),$$

где $d(n1, n2)$ — расстояние между узлами $n1$ и $n2$.

Так как подразумевается использование различных опорных поверхностей с разными коэффициентами проходимости и учет радиуса поворота, то функции $g(n)$ и $h(n)$ были изменены.

Функция $g(n)$ должна учитывать не только длину ребра, но и его вес. При этом желательнее внести штраф к поворотам. При указании узлу $n1$ родителя $n2$ происходит пересчет функции $g(n)$ следующим образом:

$$g(n) = g(n2) + d(n1, n2) * w(n1, n2) + r(n1, n2) * w(n1, n2),$$

где $d(n1, n2)$ — длина пути от узла $n1$ до узла $n2$, $w(n1, n2)$ — вес ребра, соединяющего узлы $n1$ и $n2$, $r(n1, n2)$ — суммарный угол поворота в радианах ребра, соединяющего узлы $n1$ и $n2$.

Оценка $h(n)$ рассчитывается как длина пути от узла n до узла финиша с учетом направлений узла n и узла финиша, умноженная на средний вес. Средний вес в данном случае рассчитывается как среднее значение веса в точках узла n и узла финиша.

Поиск пути группой роботов

Для разрабатываемой модели было принято использовать схему независимого управления с использованием общих ресурсов. Каждый робот движется независимо от других в свою целевую точку, считая остальных роботов препятствиями, которые надо объезжать. Общение между

роботами происходит через управляющую машину, которая уведомляется обо всех обнаруженных несоответствиях карты местности и карты реальности. Далее, управляющая машина уведомляет всех роботов о необходимости изменении карты проходимости, вследствие чего все роботы имеют одинаковую и актуальную карту проходимости, позволяющую более точно планировать маршрут. На рисунке 4 изображено окно программы с группой из трех роботов.

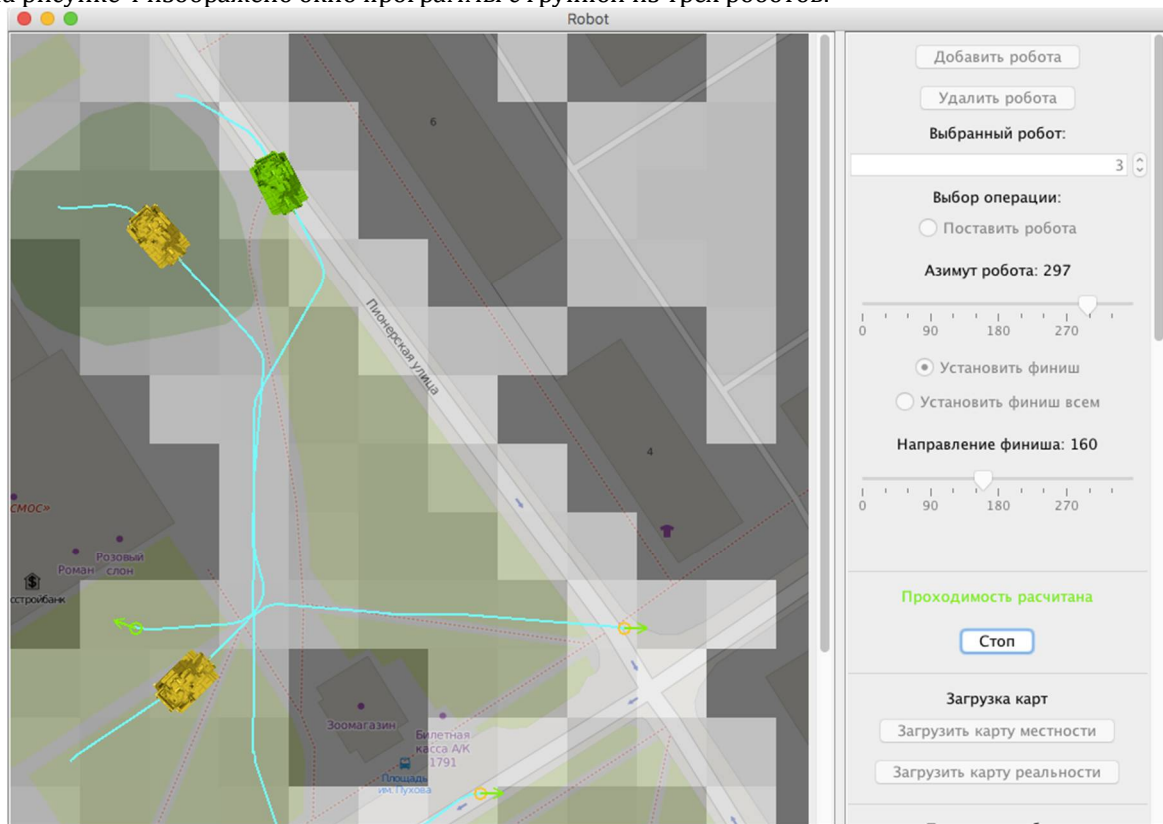


Рисунок 4. Окно программы с группой из трех роботов

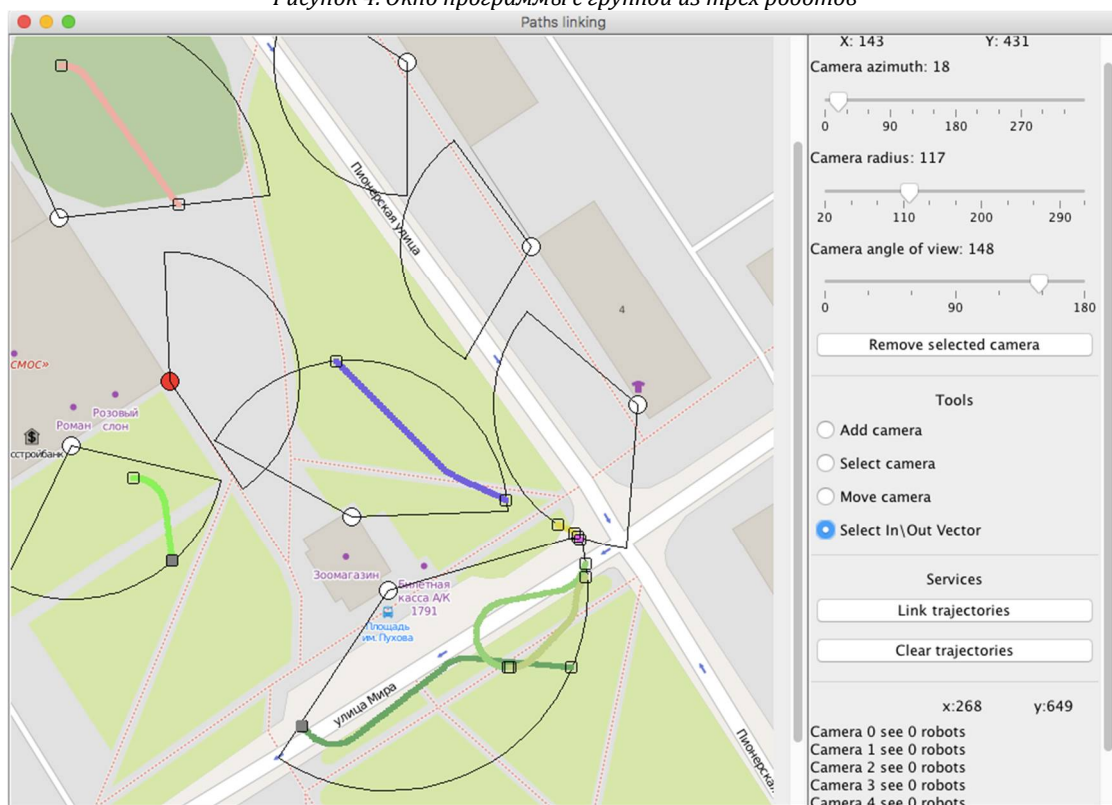


Рисунок 5. Окно программы наблюдения за группой роботов

Наблюдение за группой роботов

Наблюдение за роботами осуществляется группой камер с такими различными параметрами, как радиус обзора и угол обзора. В данной модели предполагается, что отдельная камера способна отследить в своей области видимости отдельных роботов и вести их от момента въезда в область видимости до момента выезда из области видимости. Если робот выезжает из области видимости на любое время большее минимального кванта, воспринимаемого программой, то считается, что камера его «потеряла» и следующий въезд будет восприниматься как въезд другим роботом. Далее алгоритм поиска соответствий ищет точки въезда и выезда и областей видимости, которые могли бы принадлежать одному роботу. При поиске соответствий помимо местоположения учитываются такие характеристики роботов, как: скорость, ускорение и азимут. Если области видимости камер перекрываются, и робот проехал через это перекрытие хотя бы в одной точке, то соответствие между траекториями с камер определяется отдельно и однозначно. На рисунке 5 изображено окно программы с добавленными камерами, обнаруженными траекториями и некоторыми соответствиями.

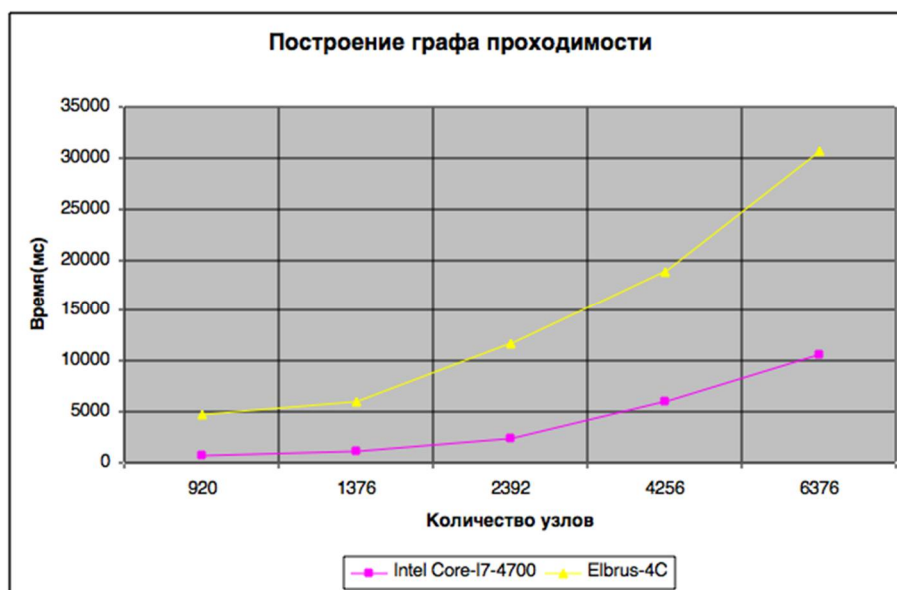


Рисунок 6. Временные характеристики алгоритма построения графа проходимости

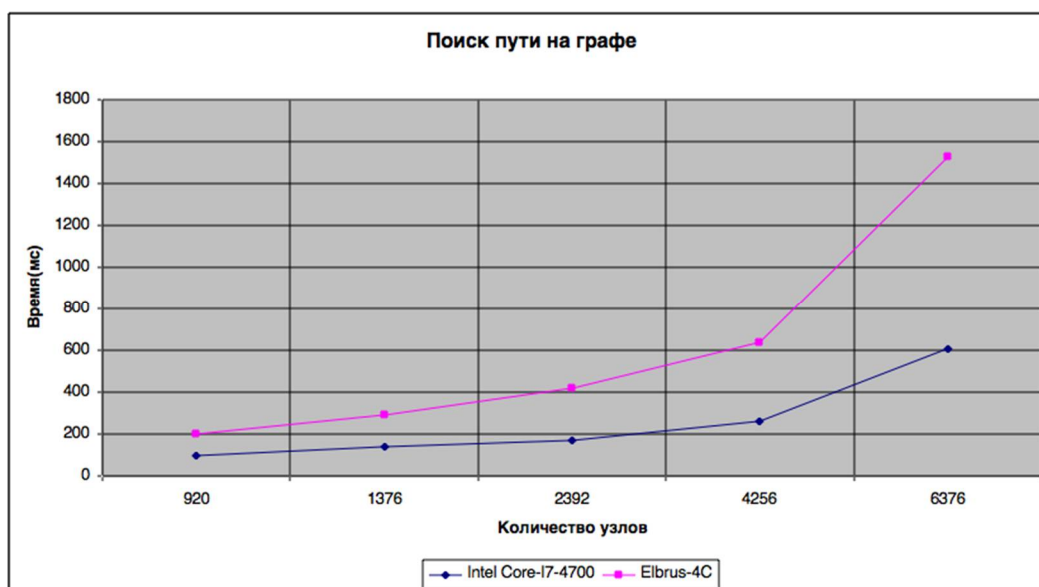


Рисунок 7. Временные характеристики алгоритма поиска пути

Тестирование

Разработанный программный стенд позволяет проводить эксперименты для оценки скорости работы алгоритмов поиска пути, как для одиночного робота, так и в составе группы. Было

проведено сравнительное тестирование на платформах Intel и Эльбрус. Для задачи поиска пути ключевые этапы – это построение графа проходимости и поиск пути на графе. Результаты представлены на рисунках 6 и 7. Тестирование на архитектуре Intel проводилось на процессоре Intel Core-i7 4700. Тестирование на архитектуре Эльбрус проводилось на ВК «Эльбрус-401 РС». Данный вычислительный комплекс представляет собой четырехпроцессорную систему с неравномерным доступом к памяти (NUMA). Каждый процессор имеет по 4 ядра с тактовой частотой 800 МГц.

Заключение

В статье показано, что вычислительные средства ряда “Эльбрус” могут удовлетворять требованиям, предъявляемым РТК в области планирования маршрута и наблюдения за группой роботов. Показаны временные характеристики для соответствующих алгоритмов.

Использование отечественных вычислительных средств и сертифицированного ОПО “Эльбрус” позволяет говорить о перспективах решения задач импортозамещения в области робототехники.

Литература

1. Парамонов Н.Б., Ржевский Д.А., Перекатов В.И. Доверенная программно-аппаратная среда «Эльбрус» бортовых вычислительных средств робототехнических комплексов // Вопросы радиоэлектроники, сер. ЭВТ, 2015, вып. 1.
2. Н.А. Бочаров, И.Д. Сапачев, Н.Б. Парамонов. Макеты робототехнических комплексов на языке Джава в среде ОС «Эльбрус»: Материалы 58 научной конференции МФТИ, 23 - 28 ноября 2015.
3. В.Е., Таланов В.А. Графы. Модели вычислений. Структуры данных. — Нижний Новгород: Издательство Нижегородского гос. университета, 2005. — 307 с.
4. Берж К. Теория графов и ее применения / Под ред. И. А. Вайнштейна. — Москва: Издательство иностранной литературы, 1962. — 320 с.
5. Уилсон Р. Введение в теорию графов. Пер с англ. М.: Мир, 1977. 208с.
6. Бруан Stout (оригинальная статья) Maxim Kamensky (перевод). Алгоритмы поиска пути [Электронный ресурс] // Программирование магических игр [Сайт] URL: <http://pmg.org.ru/ai/stout.htm> (дата обращения: 27.09.2016).
7. Алгоритм поиска A* [Электронный ресурс] // Википедия [Сайт] URL: https://ru.wikipedia.org/wiki/Алгоритм_поиска_A* (дата обращения 27.09.2016).

References

1. Paramonov N.B., Rzhevskiy D.A., Perekatov V.I. Doverennaya programmno-apparatnaya sreda «Elbrus» bortovykh vychislitel'nykh sredstv robototekhnicheskikh kompleksov // Voprosy radioelektroniki, ser. EVT, 2015, vyp. 1.
2. N.A. Bocharov, I.D. Sapachev, N.B. Paramonov. Makety robototekhnicheskikh kompleksov na yazyke Dzhava v srede OS «Elbrus»: Materialy 58 nauchnoy konferentsii MFIT, 23 - 28 noyabrya 2015.
3. V.E., Talanov V.A. Grafy. Modeli vychisleniy. Struktury dannykh. — Nizhniy Novgorod: Izdatelstvo Nizhegorodskogo gos. universiteta, 2005. — 307 s.
4. Berzh K. Teoriya grafov i ee primeneniya / Pod red. I. A. Vaynshteyna. — Moskva: Izdatel'stvo inostrannoy literatury, 1962. — 320 s.
5. Uilson R. Vvedenie v teoriyu grafov. Per s angl. M.: Mir, 1977. 208s.
6. Bryan Stout (original'naya stat'ya) Maxim Kamensky (perevod). Algoritmy poiska puti [Elektronnyy resurs] // Programmirovaniye magicheskikh igr [Sayt] URL: <http://pmg.org.ru/ai/stout.htm> (data obrashcheniya: 27.09.2016).
7. Algoritm poiska A* [Elektronnyy resurs] // Vikipediya [Sayt] URL: https://ru.wikipedia.org/wiki/Алгоритм_поиска_A* (data obrashcheniya 27.09.2016).

Поступила: 10.10.2016

Об авторах:

Бочаров Никита Алексеевич, студент магистратуры кафедры информатики и вычислительной техники Московского физико-технического института, инженер-программист в АО «МЦСТ», bocharov.na@phystech.edu;

Парамонов Николай Борисович, доктор технических наук, профессор, главный научный сотрудник АО «МЦСТ», paramonov_n_b@mail.ru;

Сапачев Илья Денисович, студент магистратуры кафедры компьютерных систем и технологий Московского инженерно-физического института, инженер-программист в АО «МЦСТ», sid.hugo7@gmail.com.