



ПРОИЗВОДИТЕЛЬНОСТЬ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ С ПРОЦЕССОРОМ «ЭЛБРУС-8С» НА ЗАДАЧАХ РОБОТОТЕХНИЧЕСКОГО КОМПЛЕКСА

PERFORMANCE OF COMPUTER SYSTEMS WITH ELBRUS-8S PROCESSOR FOR ROBOTIC SYSTEMS TASKS

УДК 004.94

БОЧАРОВ НИКИТА АЛЕКСЕЕВИЧ¹

Инженер-программист 1 категории

bocharov.na@phystech.edu

BOCHAROV NIKITA A.¹

Master, Engineer-programmer of the 1st category

bocharov.na@phystech.edu

ПАРАМОНОВ НИКОЛАЙ БОРИСОВИЧ¹

Д. т. н., профессор, главный научный сотрудник

paramonov_n_b@mail.ru

PARAMONOV NIKOLAY B.²

Sc.D, professor, Chief Researcher

paramonov_n_b@mail.ru

ТИМОФЕЕВ ГЕННАДИЙ СЕРГЕЕВИЧ²

К. т. н., старший научный сотрудник

tgs@teleformis.ru

TIMOFEEV GENNADY S.²

Candidate of Technical Sciences, Senior Researcher

tgs@teleformis.ru

ПАНОВА ОЛЬГА ЮРЬЕВНА¹

Инженер-программист

panova.o.y@gmail.com

PANOVA OLGA YU.¹

Software Engineer

panova.o.y@gmail.com

¹ АО «МЦСТ»

117105, Москва, ул. Нагатинская, 1, стр. 23

² ПАО Институт электронных управляющих машин

им. И. С. Брука

119334, Москва, ул. Вавилова, 24

¹ MCST JSC

bld. 23, 1 Nagatinskaya St., Moscow, 117105, Russia

² Brook INEUM PJSC

24 Vavilov St., Moscow, 119334, Russia

Цель работы — исследовать применимость вычислительных комплексов на базе микропроцессора Эльбрус-8С для решения задач робототехнических комплексов. Авторами были разработаны программные средства для моделирования задач управления движением робота и обработки стереоизображений и получены временные характеристики для этих моделей.

Ключевые слова: высокопроизводительные вычислительные системы; моделирование движения; моделирование стереопары; алгоритмы трехмерной реконструкции.

The purpose of this work is to research the applicability of computer systems based on Elbrus-8S microprocessor for solving robotic systems tasks. The authors have developed software to simulate the tasks of robot motion control and for processing stereo images and have received temporal characteristics for these models.

Keywords: high-performance computing systems; simulation of motion; simulation of stereo-pair; algorithms of three-dimensional reconstruction.

ВВЕДЕНИЕ

Современной тенденцией развития наземной робототехники является постепенный переход от дистанционно управляемых к полуавтономным, а в перспективе — к автономным робототехническим комплексам (РТК). В связи с этим, актуальным является создание интеллектуальных систем управления для таких РТК. Существенным, но не решенным вопросом создания систем управления РТК является оснащение вычислительной техникой, разработанной на базе отечественных микропроцессоров и программного обеспечения отечественной разработки [1].

Поскольку робототехника является одним из перспективных направлений применения вычислительных комплексов (ВК) и программного обеспечения семейства «Эльбрус» [2], то целью данной работы было исследовать и показать применимость микропроцессора «Эльбрус-8С» [3] для решения задач когнитивного управления группой роботов.

Результатом данной работы является программный комплекс, позволяющий моделировать поведение робота на местности. Также были разработаны вспомогательные программы для моделирования систем технического зрения роботов и средств резервирования. Данный программный комплекс позволил определить временные и нагрузочные характеристики, которые позволили сделать вывод о применимости микропроцессора «Эльбрус-8С» для решения задач когнитивного управления группой роботов, а также помог проводить работы по оптимизации программного обеспечения в ОС «Эльбрус».

ПОСТАНОВКА ЗАДАЧИ

В работе [4] авторами было проведено моделирование алгоритмов поиска пути и работы со системой стереозрения отдельным роботом на прошлом поколении процессоров «Эльбрус». Для тестирования был использован вычислительный комплекс, состоящий



из четырех процессоров «Эльбрус-4С», объединенных в кластер и использующих общую NUMA-память.

Движение робота в модели 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 работе [5] было проведено улучшение алгоритмов для моделирования уже не одного, а группы из нескольких роботов. В данной работе были проведены оптимизации алгоритмов и проведено моделирование на новом поколении микропроцессоров «Эльбрус-8С».

Формально, в рассматриваемой задаче поиска пути группой роботов обрабатываются три множества: множество роботов, множество целей и множество внешних факторов. В разработанной программе было реализовано моделирование всех этих множеств.

ЗАДАЧА ПОИСКА ПУТИ

Задача поиска пути была сведена к задаче поиска пути на графе из узла-старта до узла-финиша [6, 7]. 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 ребрами. 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 ребер. 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 учитывать и азимут робота при построении пути. Таким образом, узел графа задается не только координатами x, y , но и азимутом.

Ребра графа обозначаются 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 всем протяжении траектории данного ребра.

Путь робота проходит от узла к узлу. Но эти узлы не могут быть соединены ребрами, являющимися прямыми траекториями, так как робот не имеет возможности 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,

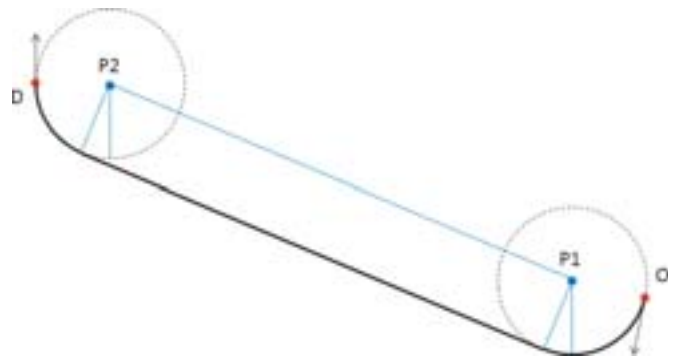


Рис. 1. Обход окружностей в одном направлении

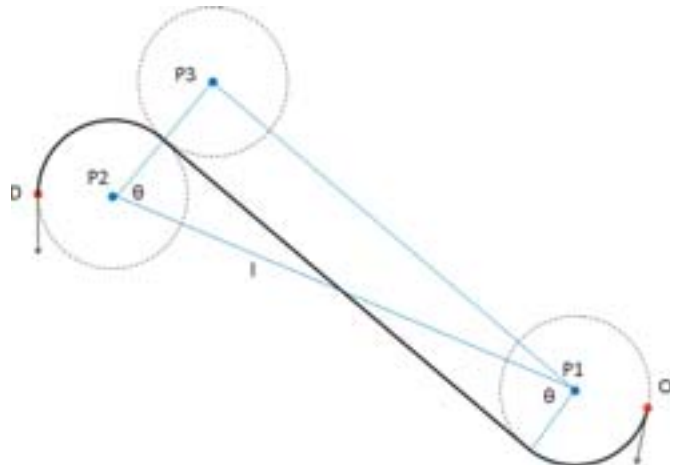


Рис. 2. Обход окружностей в противоположных направлениях

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. При построении пути есть два случая, различающиеся расчетом траектории: когда обе окружности поворота обходятся в одном направлении (например, обе по часовой стрелке) и когда две окружности поворота обходятся в противоположных направлениях (одна окружность по часовой стрелке, другая — против часовой стрелки). В обоих случаях задача построения траектории сводится к поиску угла первой окружности, на котором начинается движение по прямому участку траектории, угла второй окружности, на котором заканчивается движение по прямому участку, и длины этого прямого участка. Варианты обхода окружностей изображены на рис. 1 и 2.

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 проходимости и меняет карту проходимости, если реальный коэффициент отличен 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).

Первая проблема решается запоминанием значения веса последнего определенного пикселя. То есть, если цвет текущего пикселя не содержится в легенде, то это переходный цвет от одной области к другой, и с большой вероятностью эта область имеет вес предыдущего проанализированного пикселя.

Вторая проблема решается проверкой всех восьми пикселей вокруг обнаруженного непроходимого. По этим пикселям считается средний коэффициент проходимости, но если обнаружен хотя бы еще один непроходимый пиксель, то этот коэффициент становится равным 0. Затем рассчитанный по восьми пикселям коэффициент проходимости присваивается текущему пикселю.

Третья проблема решается сверкой с легендой, хранящейся в работе, с некоторой погрешностью.

Пример формирования карты местности и сформированной карты проходимости приведен на рис. 3.

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.



Рис. 3. Пример карты местности и карты проходимости

Алгоритм A* считается одним из лучших алгоритмов поиска пути [7]. Он объединяет в себя достоинства двух алгоритмов: учет длины пути из алгоритма Дейкстры и учет эвристической функции из алгоритма «лучший первый».

Алгоритм A* [8] использует формулу эвристики, которая 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, соединяющей 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 желательно 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.

АЛГОРИТМ ДВИЖЕНИЯ РОБОТА ОТ УЗЛА К УЗЛУ

На вход данному алгоритму подается связь, по которой робот должен проехать от текущего узла к следующему узлу. При выполнении движения происходит цикл до тех пор, пока связь, которую робот должен проехать, не будет пройдена. Цикл также может прерываться замеченными изменениями карты.

В каждой итерации цикла робот проверяет свои сенсоры. Если при этой проверке было замечено сильное изменение какого-либо квадрата, то движение прерывается. Иначе, робот изменяет свою скорость, следуя одному из следующих вариантов:

- если сенсорами была замечена плохо проходимая область, то происходит торможение до минимальной скорости;
- если робот достиг момента торможения до финиша, то происходит торможение до нуля;
- если роботу была дана команда остановки, то происходит торможение до нуля;
- если ничего из вышеперечисленного не выполнено, то происходит ускорение до обычной скорости.

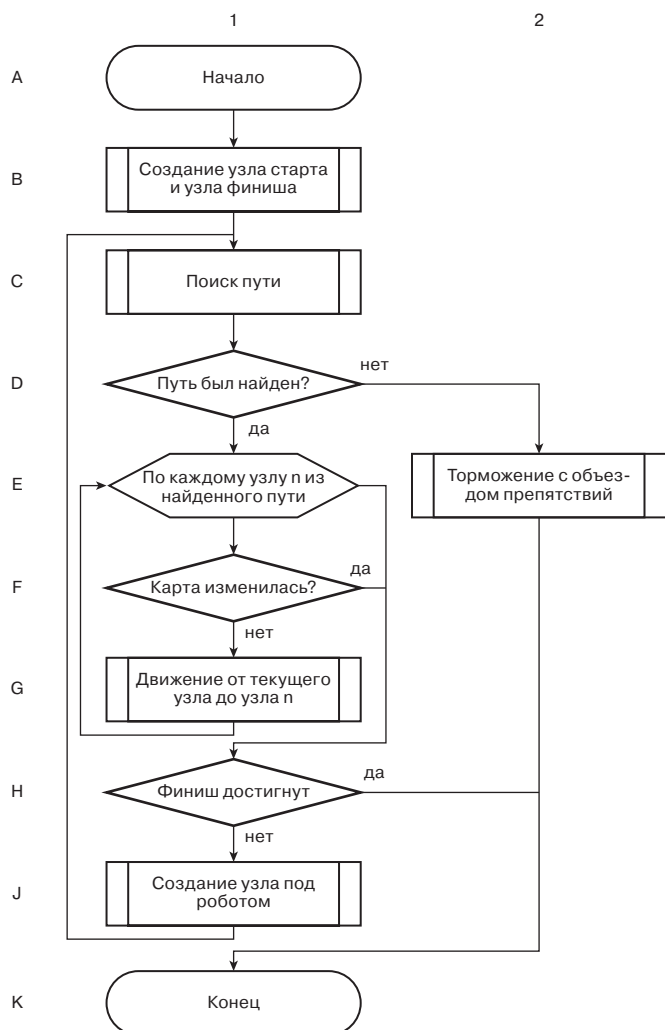


Рис. 4. Схема алгоритма движения робота по всему пути

После того как робот проедет некоторое расстояние, он проверяет условие начала торможения. Если скорость не стала равна нулю вследствие остановки, то робот продолжает движение.

АЛГОРИТМ ДВИЖЕНИЯ РОБОТА ПО ВСЕМУ ПУТИ

Перед началом движения робот создает узел старта и узел финиша. После этого происходит поиск пути с помощью алгоритма A*. Если путь не был найден, то происходит плавная остановка с объездом препятствий. В противном случае происходит движение до каждого узла из найденного пути по алгоритму движения робота от узла к узлу. При этом, если было замечено изменение карты, то движение прерывается.

При выходе из цикла проверяется, достиг ли робот финиша. Если нет, то считается, что из цикла прохода по всему пути робот вышел из-за изменения карты. Тогда под роботом создается узел и пересчитывается путь. Иначе движение алгоритм заканчивает свою работу.

После движения робота, на карте проходимости часто происходят изменения, но изменения весов в связях графа происходят только, когда алгоритм поиска пути их проанализирует. Таким образом, рекомендуется после окончания движения робота пересчитывать веса всех связей в графе, иначе алгоритм поиска пути не будет находить лучший путь.

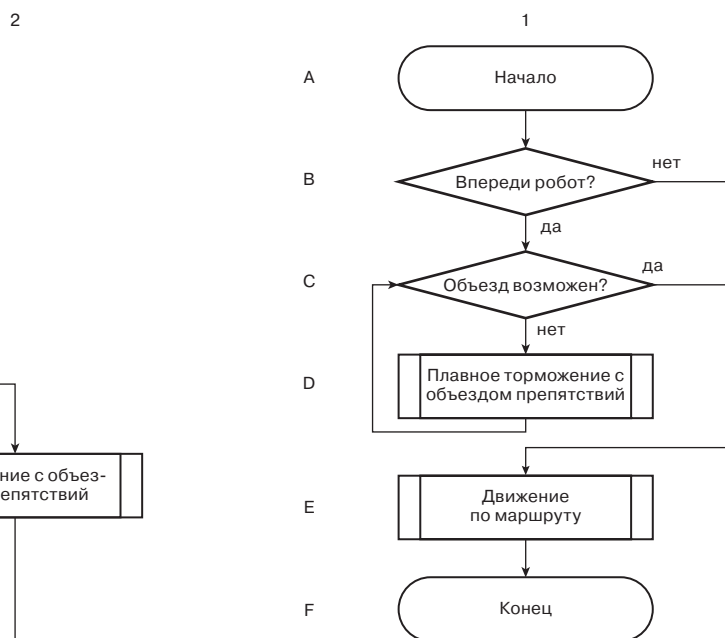


Рис. 5. Блок-схема алгоритма поведения робота при обнаружении другого робота

Блок-схема алгоритма движения робота по всему пути приведена на рис. 4.

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 внешнюю управляющую машину, которая уведомляется обо всех обнаруженных несоответствиях карты местности и карты реальности. Далее, внешняя управляющая машина уведомляет всех роботов о необходимости изменения карты проходимости, вследствие чего все роботы имеют одинаковую и актуальную карту проходимости, позволяющую более точно планировать маршрут. Блок-схема алгоритма поведения робота при обнаружении другого робота перед собой приведена на рис. 5. На рис. 6 изображено окно программы с группой из трех роботов.



Рис. 6. Окно программы с группой из трех роботов



СИСТЕМА СТЕРЕОЗРЕНИЯ

Система стереоскопического зрения решает задачи обнаружения препятствий роботом на расстояниях свыше 30 метров. Применение данной системы возможно и на меньших расстояниях в случае неожиданного выхода из строя датчиков, использовавшихся на небольших расстояниях, например, сканирующих лазерных дальномеров. Эта система является одной из важнейших систем для навигации в автономном РТК. Для ее моделирования был разработан программный стенд с использованием открытой библиотеки OpenCV. Программный стенд реализует алгоритмы калибровки стереокамеры и алгоритма стереорекострукции.

В качестве способа калибровки был выбран путь смешанной калибровки. В основе данного подхода лежит использование простого плоского калибровочного шаблона в виде «шахматной доски». Шаблон снимается каждой камерой в некотором количестве положений, после чего по снимкам оцениваются параметры камеры. Данный подход не требует применения высокоточного шаблона и вспомогательного оборудования, при этом обеспечивает достаточную (до 1 мм) точность и пригоден для применения в полевых условиях.

В модели была реализована возможность установить параметры стереопары (расстояние между камерами, поворот) и провести калибровку с использованием виртуальной шахматной доски. В программную модель заложены несколько тестов, которые можно использовать для быстрой проверки работоспособности алгоритмов и оценки времени выполнения. Также в модель можно загружать сторонние изображения, в таком случае калибровка не требуется, на этапе стереорекострукции программа сама подберет подходящие параметры. На рис. 7 представлено окно программы с полученной картой смещений для одного из встроенных тестов.

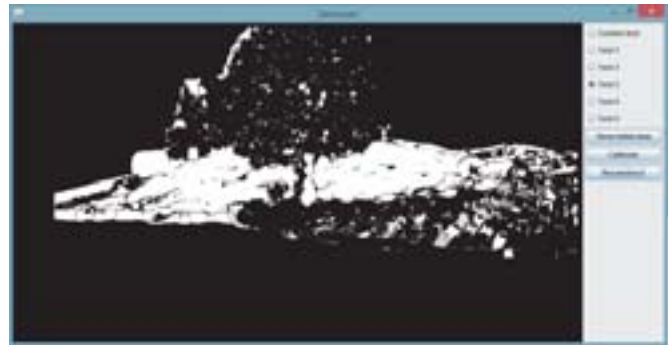


Рис. 7. Результат трехмерной реконструкции для встроенного теста

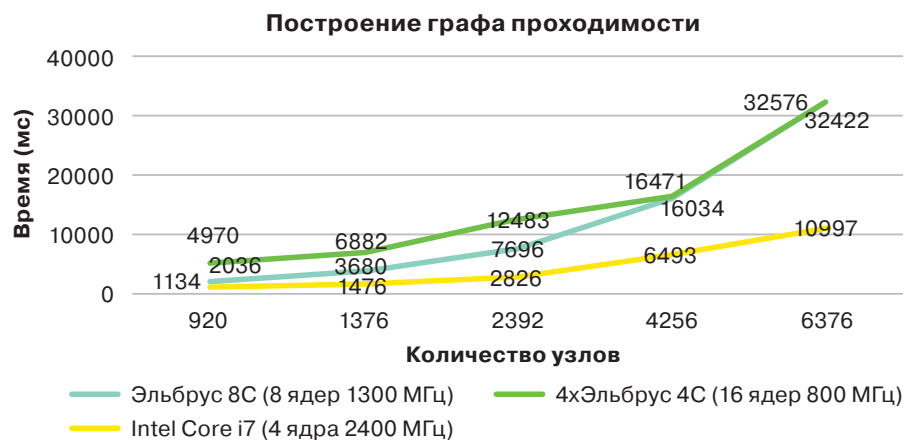


Рис. 8. Временные характеристики алгоритма построения графа проходимости

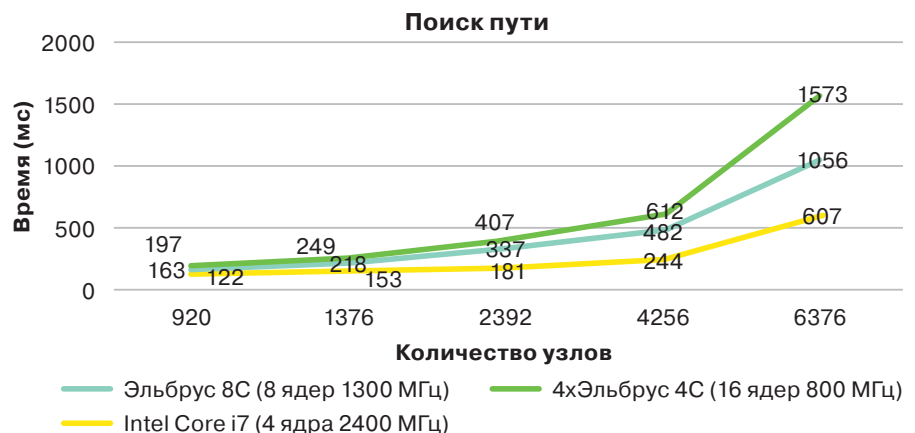


Рис. 9. Временные характеристики алгоритма поиска пути

ТЕСТИРОВАНИЕ

Разработанный программный стенд позволяет проводить эксперименты для оценки скорости работы алгоритмов поиска пути и стереозрения. Было проведено сравнительное моделирование на микропроцессорах Intel и Эльбрус прошлого и текущего поколений. Для задачи поиска пути ключевые этапы — это построение графа проходимости и поиск пути на графе. Результаты моделирования алгоритмов построения графа проходимости и поиска пути представлены на рис. 8 и 9 соответственно. Результаты моделирования алгоритма стереорекострукции представлены на рис. 10. Тестирование на архитектуре Intel проводилось на процессоре Intel Core-i7 4700. Тестирование на архитектуре Эльбрус проводилось на ВК «Эльбрус-401 РС» и ВК «Эльбрус-801». «Эльбрус-401 РС»

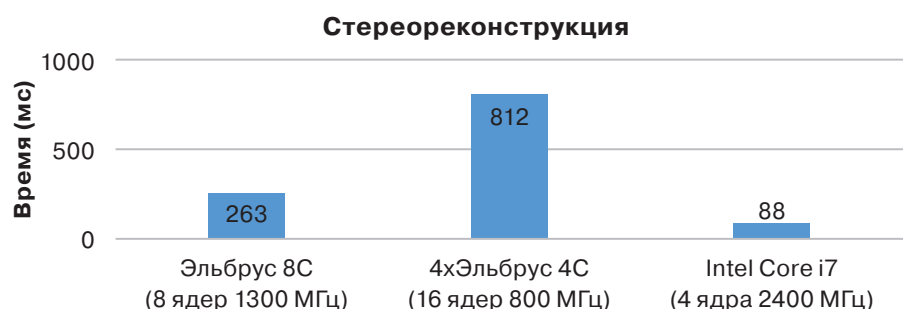


Рис. 10. Затраты времени на стереорекострукцию по стереопаре



представляет собой четырехпроцессорную систему с неравномерным доступом к памяти (NUMA). Каждый процессор имеет по 4 ядра с тактовой частотой 800 МГц. «Эльбрус-801» представляет собой вычислительный комплекс, содержащий один процессор Эльбрус-8С. Процессор имеет 8 ядер с тактовой частотой 1300 МГц.

ЗАКЛЮЧЕНИЕ

Проведенное моделирование показало возможность применения общего программного обеспечения и средств вычислительной техники на основе микропроцессора «Эльбрус-8С» для решения задач движения робота и обработки системы технического зрения.

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

Авторы считают, что в данной работе новыми являются следующие положения и результаты: разработаны тестовые программы для моделирования задач движения робота и системы стереозрения, получены временные характеристики для процессора «Эльбрус-8С».

Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект № 17-29-03297)

ЛИТЕРАТУРА

1. Бычков И. Н., Глухов В. И., Трушкин К. А. Доверенная программно-аппаратная платформа «Эльбрус». Отечественное решение для АСУ ТП КВО // Журнал «Информатизация и Системы Управления в Промышленности», № 1(49), 2014. — С. 66–71.
2. 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.
3. Альфонсо Д. М., Деменко Р. В., Кожин А. С., Кожин Е. С., Колычев Р. Е., Костенко В. О., Поляков Н. Ю., Смирнова Е. В., Смирнов Д. А., Смолянов П. А., Тихорский В. В. Микроархитектура восьмиядерного универсального микропроцессора «Эльбрус-8С» // Вопросы радиоэлектроники. 2016. Т. 4. № 3. — С. 6–13.
4. Бочаров Н. А., Сапачев И. Д., Парамонов Н. Б. Макеты задач робототехнических комплексов на языке Java в среде ОС «Эльбрус» // Наноиндустрия. Спецвыпуск 2017(74). Международный форум «Микроэлектроника-2016», 2-я научная конференция «Интегральные и микроэлектронные модули». Сборник докладов — С. 122–127.
5. Бочаров Н. А., Парамонов Н. Б., Сапачев И. Д. Реализация алгоритмов группового управления на языке в среде ОС «Эльбрус» // Современные информационные технологии и ИТ-образование, 2016, Том 12, № 1. — С. 108–115.
6. Алексеев В. Е., Таланов В. А. Графы. Модели вычислений. Структуры данных. — Нижний Новгород: Издательство Нижегородского гос. университета, 2005. — 307 с.
7. Берж К. Теория графов и ее применения / Под ред. И. А. Вайнштейна. — Москва: Издательство иностранной литературы, 1962. — 320 с.
8. Bryan 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: 25.08.2017).

КНИГИ ИЗДАТЕЛЬСТВА "ТЕХНОСФЕРА"



Цена 636 руб.

ОСНОВЫ СХЕМОТЕХНИКИ МИКРОЭЛЕКТРОННЫХ УСТРОЙСТВ

А.И. Белоус, В.А. Емельянов,
А.С. Турцевич

М: ТЕХНОСФЕРА, 2012. — 472 с.
ISBN 978-5-94836-307-3

В книге представлен анализ особенностей работы, методы проектирования и основы практического применения цифровых микросхем в составе современных микроэлектронных устройств, предложен большой набор эффективных схемотехнических решений базовых элементов для реализации требований, предъявляемых к микроэлектронным устройствам, приведено детальное описание принципов работы и правил применения современных базовых элементов в составе микроэлектронных устройств.

Издание ориентировано на широкий круг инженерно-технических работников, ученых, студентов и аспирантов, специализирующихся в области разработки, организации производства и эксплуатации радиоэлектронной, бытовой, промышленной и специальной техники, информационно-коммуникационных, телекоммуникационных и навигационных применений, использующих современные микроэлектронные устройства.

КАК ЗАКАЗАТЬ НАШИ КНИГИ?

☎ 125319, Москва, а/я 91; ☎ +7 (495) 234-0110; ☎ +7 (495) 956-3346; ✉ knigi@technosphera.ru, sales@technosphera.ru