К

УДК 004.75:004.021

А. А. Койбаш, Т. В. Завадская, С. В. Кривошеев
Государственное образовательное учреждение высшего профессионального образования
«Донецкий национальный технический университет», г. Донецк
83001, г. Донецк, ул. Артёма, 58

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ТЕХНОЛОГИИ NVIDIA CUDA В РАСПРЕДЕЛЕННЫХ КОМПЬЮТЕРНЫХ СИМУЛЯТОРАХ ПОДВИЖНЫХ ОБЪЕКТОВ

A. A. Koibash, T. V. Zavadskaya, S. V. Kryvosheev
State Educational Institution of Higher Education «Donetsk national technical University», Donetsk city
83001, Donetsk, Artema str., 58

RESEARCH OF THE EFFECTIVENESS OF NVIDIA CUDA TECHNOLOGIES IN DISTRIBUTED COMPUTER SIMULATORS OF MOVING OBJECTS

О. А. Койбаш, Т. В. Завадська, С. В. Кривошеєв
Державна освітня установа вищої професійної освіти
«Донецький національний технічний університет», м. Донецьк
83001, м. Донецьк, вул. Артема, 58

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ТЕХНОЛОГІЇ NVIDIA CUDA У РОЗПОДІЛЕНИХ КОМП'ЮТЕРНИХ СИМУЛЯТОРАХ РУХОМИХ ОБ'ЄКТІВ

В работе рассмотрены пути применения алгоритма A* в компьютерных симуляторах для построения траектории движения подвижного объекта. Для повышения быстродействия построения траектории движения предложено использование сортирующего дерева и хеш-таблиц. Проведено исследование работы алгоритма с использованием этих структур. Результаты тестирования показали значительное увеличение скорости вычислений при использовании в распределенных компьютерных симуляторах.
Ключевые слова: генерация оптимальной траектории, алгоритм A*, распределенный симулятор, сортирующее дерево.

The ways of applying the algorithm A * in computer simulators for prediction the trajectory of a moving object are considered in this paper. To improve the performance of the traectory generation, the use of a sorting tree and hash tables was proposed. The study of the algorithm with the use of these structures. The test results showed a significant increase in the speed of the calculations in distributed computer simulators.
Keywords: optimal trajectory generation, A * distributed simulator, sorting tree.

У роботі розглянуті шляхи застосування алгоритму A * в комп'ютерних симуляторах для побудови траєкторії руху рухомого об'єкта. Для підвищення швидкодії побудови траєкторії руху запропоновано використання сортувального дерева і хеш-таблиць. Проведено дослідження роботи алгоритму з використанням цих структур. Результати тестування показали значне збільшення швидкості обчислень при використанні в розподілених комп'ютерних симуляторах.
Ключові слова: генерація оптимальної траєкторії, алгоритм A*, розподілений симулятор, сортувальне дерево.

Введение

Сегодня инженерная техника используется во многих отраслях: от строительства и сельского хозяйства до промышленности и космонавтики. Она позволяет значительным образом обезопасить человеческий труд и повысить его эффективность. Однако для управления такой техникой необходим соответствующий квалифицированный персонал.

Целью исследования является выявление путей применения технологии NVIDIA CUDA при реализации алгоритма A* в распределенных компьютерных симуляторах для снижения времени построения оптимальной траектории движения подвижного объекта по заданным критериям. **Основная задача** заключается в оценке прироста производительности при генерации траектории с помощью распределенных вычислительных систем на базе GPU при использовании сортирующего дерева и хеш-таблиц для алгоритма A*.

Применение компьютерных симуляторов для подготовки операторов подвижных объектов

Одним из направлений тренировки кадров является использование компьютерных симуляторов, которые позволят моделировать штатные и внештатные ситуации. Одна из особенностей этого подхода заключается в возможности реализации экстремальных ситуаций, которые либо тяжело воспроизвести в реальных условиях, либо могут быть опасны для жизни и здоровья операторов техники. Подобные системы имеют возможность детального разбора учебного процесса в любой момент времени, а также визуальное представление ситуации: от графиков, описывающих ситуацию в целом, до графического отображения инженерной техники и окружающих объектов. Кроме того, компьютерный симулятор может быть реализован как распределенная система, что позволяет одновременное обучение нескольких операторов.

Одной из составляющей симулятора выступает модуль прогнозирования траектории, который предназначен для определения кратчайшего маршрута до цели. Следовательно, появляется возможность сравнить путь, составленный оператором, от оптимального, который реализует алгоритм программного обеспечения по заданному критерию. Также это позволяет минимизировать влияние человеческого фактора на траекторию движения подвижного объекта и снизит затраты ресурсов (топлива и/или времени), необходимых на преодоление пути.

Использование технологии CUDA для прогнозирования траектории движения подвижного объекта

В работах [1–3] определено, что наилучшим методом поиска пути в случае с тяжелой инженерной техникой является алгоритм A*. Благодаря использованию эвристики он сможет быстро найти маршрут до целевого пункта. Для ускорения поиска пути были исследованы различные многопоточные реализации. Первый подход заключался в многопоточном просчете параметров соседних вершин в ходе работы алгоритма. При этом вычислялся один маршрут. Альтернативным вариантом выступал многопоточный просчет сразу нескольких маршрутов – каждый поток полностью вычислял свой собственный маршрут. Наилучший результат показал поиск нескольких путей. Сравнительная диаграмма показана на рис. 1. В левых столбцах показано время вычисления сразу нескольких путей, в правых – время вычисления одного маршрута в несколько потоков.

Кроме параллельной реализации необходимо также учесть места падения производительности при реализации самого алгоритма A*:

1) Циклический поиск минимума – поиск вершины по полю F.

2) Произвольный доступ к элементу списка по координатам.

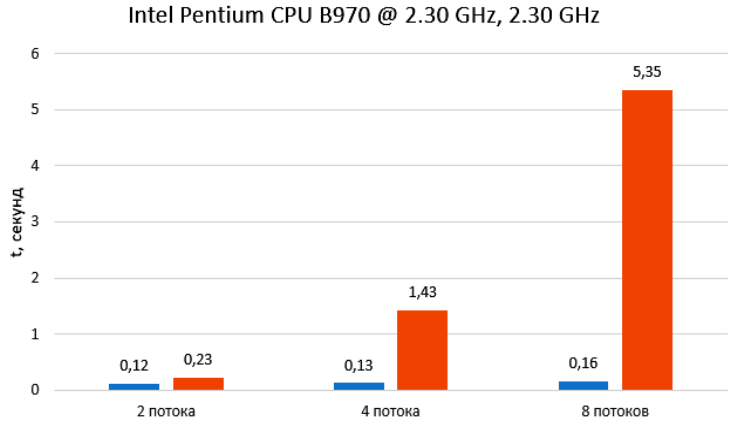

Рисунок 1 – Сравнительная диаграмма разных многопоточных реализаций [4]

На малых картах падение производительности незначительное, однако на больших картах (например, 2048×2048) произойдет серьёзное снижение времени поиска пути.

Для решения первой проблемы использовано сортирующее дерево, называемое также двоичной кучей. Для такого дерева выполняется следующее условие: приоритет каждой вершины больше приоритетов её потомков [5]. Оценка сложности операций добавления и восстановления свойств дерева имеет сложность O(log(N)), поскольку оно имеет логарифмическую высоту. Изъятие минимального элемента всегда выполняется в одну операцию – забрать самый верхний элемент.

Для решения второй проблемы использована хеш-таблица. В среднем все три её операции (вставка, поиск и удаление) выполняются за время O(1) [5]. Элементы таблицы – пары ссылок, первая из которых указывает на элемент двоичной кучи. Вторая ссылка указывает на следующий элемент таблицы, так как коллизии (совпадения хеш-значений при разных входных данных) разрешаются методом цепочек. Поскольку элементы открытого списка в ходе алгоритма A* должны находиться по совпадению координат (x; y), для нормального распределения и минимализации коллизий выбрана хеш-функция вида $k*x \oplus y$, где k – степень двойки. При этом хеш-значение будет также и индексом массива. Переменная k выбиралась исходя из следующих условий: при больших k вероятность коллизий сокращается, однако вырастает потребление памяти и времени для выделения этой памяти; при меньших k потребление ресурсов снижается, однако увеличивается вероятность возникновения коллизий. При k=64 количество совпадений в худшем случае составляет 31 элемент на одну ячейку таблицы, однако результаты теста показали, что уровень коллизий составляет не выше двух [6].

Был проведён анализ методов оптимизации алгоритма A* с помощью специальных структур. Диаграмма времени вычисления пути представлена на рис. 2.

Таким образом, при необходимости прямого доступа вычисляется хеш-значение, по нему напрямую находится элемент в хеш-таблице, который содержит ссылку на двоичную кучу. Несмотря на необходимость поддержки взаимодействия сортирующего дерева и хеш-таблицы, а также постоянного обновления ссылок в обеих

структурах при изменении в одной из них, такой подход показывает значительное повышение производительности.

Исследование показало, что параллельный подход с использованием оптимизирующих структур может серьёзно повысить время поиска пути. Использование результатов многопоточной реализации показало, что просчет сразу нескольких маршрутов является более быстрым. Следовательно, можно определить, куда тяжелая инженерная техника может попасть через интервал времени и какие точки необходимо просчитывать.

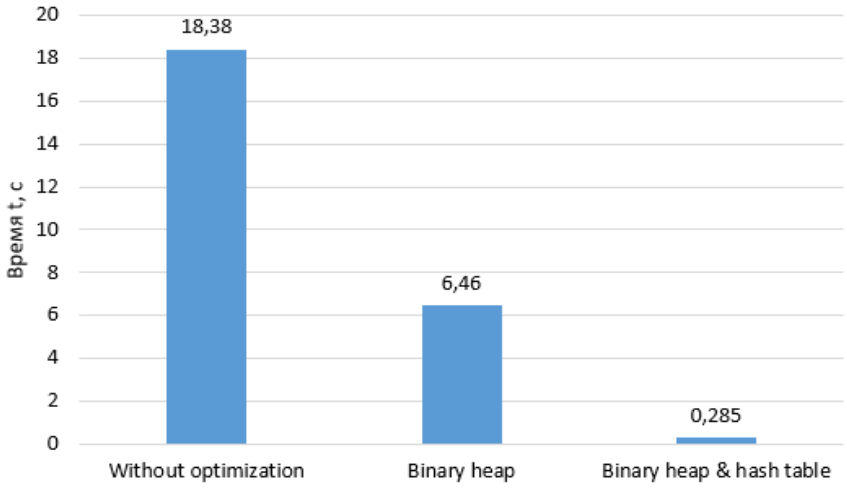

Рисунок 2 – Диаграмма времени поиска пути

Объект может перемещаться назад и вперёд, а также поворачивать на 45° и 90° [6]. Когда объект только начинает перемещение, нужно просчитывать точки позади него. В случае перемещения объект не сможет резко двигаться назад, поэтому точки сзади просчитывать не нужно и имеет смысл просчитать больше точек впереди объекта. Пример карты точек, куда может переместиться подвижный объект, показан на рис. 3.

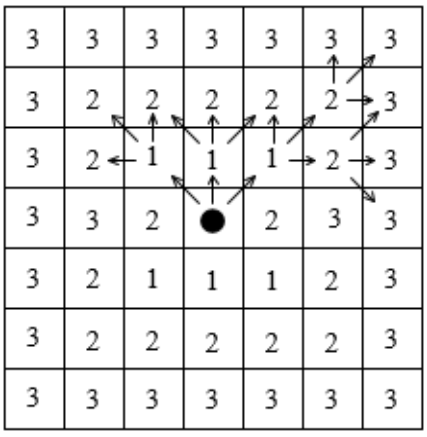

Рисунок 3– Пример карты точек, в которые объект может переместиться через определённый интервал времени [7]

Для таких параллельных вычислений наиболее оптимальным решением является использование графических процессоров, максимально направленных на параллелизм.

Для прогнозирования траектории движения целесообразно использовать архитектуру параллельных вычислений NVIDIA CUDA [1].

Принцип вычисления графических процессоров – SIMD (один поток команд, множественный поток данных). Следовательно, разработанный алгоритм может принимать точку старта, точку финиша, а также карту местности, что будет вполне достаточно для просчета пути.

Работа программы вычисления пути происходит в несколько этапов:

– выделение памяти и инициализация структур;
– поиск пути с очисткой памяти;
– освобождение памяти;
– завершение работы.

Необходимо осуществить выравнивание структур в памяти. Для этого в памяти выделяется место поочередно под каждый тип для N потоков. В первую очередь место выделяется под карты 2048×2048 ячеек и по 1 байту на каждую ячейку. Их размер составляет N*4 МБ. Следующие участки выделяются уже динамически в куче в следующем порядке:

Хеш-таблицы размером N * 1 МБ.

Двоичные кучи со стартовым размером в 100 000 элементов. Поскольку размер узла кучи составляет 24 байта, начальный размер кучи – N * 2.4 МБ.

Векторы пути со стартовым размером в 20 000 элементов или N * 0.16 МБ.

Схема расположения структур в памяти изображена на рис. 4.

	[0]	[1]	[2]	[3]	...	[N-2]	[N-1]	[N]
Maps	4 МБ	4 МБ	4 МБ	4 МБ		4 МБ	4 МБ	4 МБ
	[0]	[1]	[2]	[3]	...	[N-2]	[N-1]	[N]
HashTables	1 МБ	1 МБ	1 МБ	1 МБ		1 МБ	1 МБ	1 МБ
	[0]	[1]	[2]	[3]	...	[N-2]	[N-1]	[N]
Heaps	2.4 МБ	2.4 МБ	2.4 МБ	2.4 МБ		2.4 МБ	2.4 МБ	2.4 МБ
	[0]	[1]	[2]	[3]	...	[N-2]	[N-1]	[N]
Paths	0.16 МБ	0.16 МБ	0.16 МБ	0.16 МБ		0.16 МБ	0.16 МБ	0.16 МБ

Рисунок 4 – Схема расположения структур в памяти

Суммарный минимальный объем памяти для каждого потока составляет 7.56 МБ. Структуры Heap и Path – динамические массивы, основанные на векторном классе, поэтому в случае нехватки памяти могут занять дополнительное место.

Алгоритм протестирован на графических процессорах – на двух видеокартах разных компьютеров. IntelPentiumCPUG2020 @ 2.90GHz 2.90GHz и видеокартой Nvidia-GeForceGT 630. Другой компьютер является ноутбуком, обладает процессором Intel-PentiumCPUB970 @ 2.30 GHz, 2.30 GHz, интегрированной видеокартой IntelSandyBridge и дискретной NvidiaGeForceGT 610m. Сравнительная характеристика показана в табл. 1.

Для тестирования времени поиска пути проведены замеры на малых, средних и больших картах. Для двух видеокарт выбраны маршруты длиной 32, 64, 128, 256, 512, 1024, 1536 и 2048 точек, сами маршруты вычисляются в 32 потока. Траектория прокладывается от верхней левой точки (0; 0) до правой центральной (1023; 2047).

За каждый маршрут отвечает отдельный блок CUDA. На графическом процессоре могут вычисляться одновременно различные маршруты, однако для тестирования один и тот же маршрут был продублирован для 32 потоков.

Таблица 1 – Сравнительная характеристика видеокарт

Наименование	Nvidia GeForceGT 610m	Nvidia GeForceGT 630
Тех-процесс, нм	40	40
Транзисторов, млн	585	585
Частота работы ядра, MHz	900	810
Частота работы памяти, Mhz	900	900
Объём памяти, Гб	2048	1024
Разрядность шины памяти, бит	64	128

Диаграмма зависимости времени выполнения от длины маршрута для видеокарты NvidiaGeForceGT 630 показана на рис. 5.

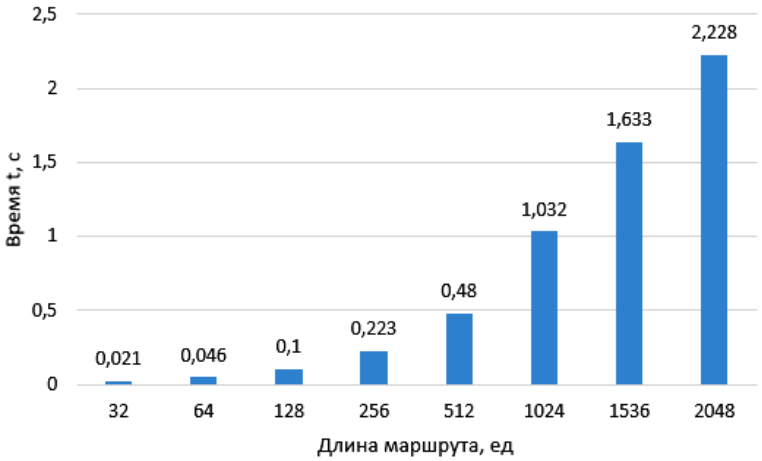

Рисунок 5 – Диаграмма времени поиска пути для GeForce 630

Из рис. 5 видно, что время растет нелинейно: при увеличении длины маршрута в 2 раза время поиска растет приблизительно в 2.1 раза.

Диаграмма зависимости времени выполнения от длины маршрута для видеокарты NvidiaGeForceGT 610m показана на рисунке 6.

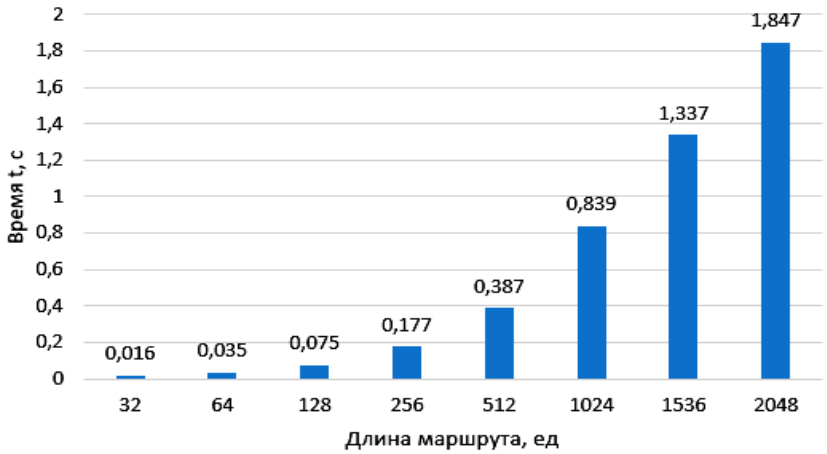

Рисунок 6 – Диаграмма времени поиска пути для GeForce 610m

Выводы

Зависимость также нелинейная, но время выполнения отличается от реализации с применением Nvidia 630.

Результаты исследований показывают, что многопоточный расчет сразу нескольких маршрутов с использованием сортирующих деревьев и хеш-таблиц показывает существенное повышение времени вычисления пути. Таким образом, использование графических процессоров для ускорения прогнозирования траектории движения инженерной техники возможно и определять кратчайший путь до цели допустимо динамически – непосредственно во время передвижения объекта.

Список литературы

1. Койбаш А. А. Прогнозирование траектории движения подвижного объекта распределенного симулятора тяжелой инженерной техники [Текст] / А. А. Койбаш, С. В. Кривошеев // Информатика, управляющие системы, математическое и компьютерное моделирование (ИУСМКМ – 2016): материалы VII междунар. науч.-техн. конф., Донецк, 2016. / редкол. А. Ю. Харитонов и др. – Донецк : ДонНТУ, 2016. – С. 343–346.
2. Койбаш А. А. Подсистема прогнозирования траектории движения подвижного объекта распределенного симулятора тяжелой инженерной техники [Текст] / А. А. Койбаш, С. В. Кривошеев // Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС – 2016): сборник научных трудов I научно-практической конференции (студенческая секция), Донецк, 2016. – Донецк : ДонНТУ, 2016. – С. 262–265.
3. Кривошеев С. В. Пути снижения времени прогнозирования траектории движения подвижного объекта распределенного симулятора тяжелой инженерной техники [Текст] / С. В. Кривошеев, А. А. Койбаш // Современные тенденции развития и перспективы внедрения инновационных технологий в машиностроении, образовании и экономике : материалы IV междунар. науч.-практ. конф., Азов 2017 / редкол. С. В. Жуков и др. – Азов : Технологический институт (филиал) ДГТУ, 2017. – С. 51–54.
4. Койбаш А. А. Модификация алгоритма A* для прогнозирования траектории движения подвижного объекта в распределенных вычислительных системах [Текст] / А. А. Койбаш, Т. В. Завадская, С. В. Кривошеев // Международный рецензируемый научно-теоретический журнал «Проблемы искусственного интеллекта». – 2018. – № 3 (10). – С. 65–73.
5. Томас Х. Кормен. Алгоритмы: построение и анализ [Текст] / Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн; 3-е изд.; Пер. с англ. Красиков И.В. – М. : «Вильямс», 2013. – 1328 с.
6. Койбаш А. А. Прогнозирование траектории движения подвижного объекта распределенного симулятора тяжелой инженерной техники [Текст] / А. А. Койбаш, Т. В. Завадская, С. В. Кривошеев // Информатика, управляющие системы, математическое и компьютерное моделирование (ИУСМКМ – 2018) : материалы IX междунар. науч.-техн. конф., Донецк, 2018. / редкол. Р. В. Мальчева и др. – Донецк : ДонНТУ, 2018. – С. 187–191.
7. Анилович В. Я. Конструирование и расчет сельскохозяйственных тракторов [Текст] : справочное пособие / В. Я. Анилович, Ю. Т. Водолажченко. – 2-е изд.; перераб. и доп. – М.: «Машиностроение», 1976. – 456 с.

References

1. Koibash A.A., Kryvosheev S.V. Prognozirovanie traektorii dvizheniya podvizhnogo ob"ekta raspredelennogo simulyatora tyazheloj inzhenernoj tekhniki. [Prediction of the motion trajectory of a moving object of a distributed simulator of heavy engineering equipment]. *Informatika, upravlyayushchie sistemy, matematicheskoe i komp'yuternoe modelirovanie* (IUSMKM – 2016): materialy VII mezhdunar. nauch.-tekhn. konf., Doneck, 2016 / redkol. A. YU. Haritonov i dr. [Computer science, control systems, mathematical and computer modeling] Doneck, DonNTU, 2016. pp. 343–346.

2. Koibash A.A., Kryvosheev S.V. Podsistema prognozirovaniya traektorii dvizheniya podvizhnogo ob"ekta raspredelennogo simulyatora tyazheloj inzhenernoj tekhniki. [Subsystem for predicting the motion trajectory of a moving object of a distributed heavy engineering simulator]. *Programmnaya inzheneriya: metody i tekhnologii razrabotki informacionno-vychislitel'nyh sistem* (PIIVS – 2016): sbornik nauchnyh trudov I nauchno-prakticheskoj konferencii (studencheskaya sekciya), Doneck, 2016. [Software Engineering: Methods and Technologies for Developing Information-Computing systems], Doneck, DonNTU, 2016. pp. 262–265.
3. Kryvosheev S.V., Koibash A.A. Puti snizheniya vremeni prognozirovaniya traektorii dvizheniya podvizhnogo ob"ekta raspredelennogo simulyatora tyazheloj inzhenernoj tekhniki [Ways to reduce the time to predict the motion trajectory of a moving object of a distributed heavy engineering simulator. *Sovremennye tendencii razvitiya i perspektivy vnedreniya innovacionnyh tekhnologij v mashinostroenii, obrazovanii i ehkonomike: materialy IV mezhdunar. nauch.-prakt. konf.*, Azov 2017. / redkol. S. V. ZHukov i dr. [Modern development trends and perspectives of implementation innovative technologies in mechanical engineering, education and economy] Azov: Tekhnologicheskij institut (filial) DGTU [Technological Institute DGTU], 2017. pp. 51–54.
4. Koibash A.A., Zavadskaja T.V., Kryvosheev S.V. Modifikatsiya algoritma A * dlya prognozirovaniya trayektoriy dvizheniya podvizhnogo ob"yekta v raspredelennykh vychislitel'nykh sistemakh. [Modification algorithm A* for prediction of mobile object motion trajectory at distributed systems] *Mezhdunarodnyy retsenziruyemyy nauchno-teoreticheskiy zhurnal «Problemy iskusstvennogo intellekta»* [Problems of Artificial Intelligenc], 2018, no. 3 (10), pp. 65–73.
5. Tomas H. Kormen. Algoritmy: postroenie i analiz [Algorithms: construction and analysis]/ Tomas H. Kormen, CHarl'z I. Lejzerson, Ronal'd L. Rivest, Klifford SHtajn; 3-e izd.; Per. s angl. Krasikov I.V. M., Vil'yams, 2013, 1328 p.
6. Koibash A.A., Zavadskaya T.V., Kryvosheev S.V. Prognozirovanie traektorii dvizheniya podvizhnogo ob'ekta raspredelennogo simulyatora tyazheloj inzhenernoj tekhniki. [Prediction of the motion trajectory of a moving object of a distributed simulator of heavy engineering equipment.]. *Informatika, upravlyayushchie sistemy, matematicheskoe i komp'yuternoe modelirovanie* (IUSMKM – 2018): materialy IX mezhdunar. nauch.-tekhn. konf., [Computer science, control systems, mathematical and computer modeling] Doneck, 2018. / redkol. R. V. Mal'cheva i dr. Doneck: DonNTU, 2018. pp. 187–191.
7. Anilovich V.Y., Vodolazhchenko Y.T. *Konstruirovaniye i raschet sel'skokhozyaystvennykh traktorov* / spravochnoye posobiye [Design and calculation of agricultural tractors / reference manual], M., Mashinostroyeniye, 1976. 456 p.

RESUME

A. A. Koibash, T. V. Zavadskaya, S.V. Kryvosheev

Research of the Effectiveness of Nvidia Cuda Technologies in Distributed Computer Simulators Of Moving Objects

Background: The ways of applying the algorithm A * in computer simulators for prediction the trajectory of a moving object are considered in this paper. To improve the performance of the traectory generation, the use of a sorting tree and hash tables was proposed. The study of the algorithm with the use of these structures. The test results showed a significant increase in the speed of the calculations in distributed computer simulators.

Materials and methods: development environment Microsoft Visual Studio and programming languages C ++ and C #.

Results: A modified algorithm for trajectory generation is proposed.

Conclusion: A multi-threaded route with optimizing structures can give a performance boost. For prediction of different trajectories is advisable to use the NVIDIA CUDA architecture, which will allow to calculate dozens of routes.

РЕЗЮМЕ

А. А. Койбаш, Т. В. Завадская, С.В. Кривошеев
Исследование эффективности технологии NVIDIA CUDA
в распределенных компьютерных симуляторах подвижных объектов

История вопроса. В работе рассмотрены пути применения алгоритма A* в компьютерных симуляторах для построения траектории движения подвижного объекта. Для повышения быстродействия построения траектории движения предложено использование сортирующего дерева и хеш-таблиц. Проведено исследование работы алгоритма с использованием этих структур. Результаты тестирования показали значительное увеличение скорости вычислений при использовании в распределенных компьютерных симуляторах.

Материалы и методы. В работе используется среда разработки Microsoft Visual Studio. А также языки программирования C++ и C#.

Результаты. Предложен алгоритм для генерации траектории движения.

Заключение. Многопоточный маршрут с оптимизирующими структурами может дать прирост производительности. Для прогнозирования различных траекторий движения целесообразно использовать архитектуру NVidiaCUDA, что позволит вычислять одновременно множество возможных путей перемещения одного или нескольких объектов.

Статья поступила в редакцию 20.09.2018.