

УДК 004.043+021

ОРГАНИЗАЦИЯ ХРАНЕНИЯ ДАННЫХ В РАМКАХ ПРОБЛЕМЫ АНАЛИЗА КАЧЕСТВА ДОРОЖНОГО ПОКРЫТИЯ

М. А. Артемов, С. В. Бабкин

Воронежский государственный университет

Поступила в редакцию 05.11.2014 г.

Аннотация. В рамках проблемы анализа качества дорожного покрытия рассматриваются различные варианты хранения данных, поступающих от анализатора. При этом принимается во внимание тот факт, что анализатор создан на основе сенсора Kinect. Производится оценка количества сохраняемых данных на основе имеющихся статистических данных о протяженности дорог, а также об их качестве. На основе полученной оценки вычисляется время выполнения запросов на удаление, изменение и добавление информации. Хранение данных должно быть организовано таким образом, чтобы можно было обойтись без использования кэширования поступающих данных.

Ключевые слова: дорожное покрытие, информационная система, база данных, анализатор поверхности.

Annotation. Within problem of analysis of pavement different ways of storing data are reviewed, data is coming from analyzer. The fact, which is taken in account, is that analyzer is based on the Kinect. Approximation of quantity of stored data is performed using statistical data about overall length and quality of roads. Based on this evaluation it is possible to calculate runtime of add, change and removal requests. Storing data must be organized in such a way, that it could be possible to avoid using caching of incoming data.

Keywords: pavement, information systems, surface analysis, storing data.

ВВЕДЕНИЕ

Проблема мониторинга качества дорожного покрытия является одной из актуальных задач, так как общая длина дорог России достаточно велика (~1 млн. км) и увеличивается в среднем на 3 % в год.

Поддержание приемлемого качества дорог позволяет обеспечивать стабильность окружающей экологической обстановки и не увеличивать число «пробок» на важных магистралях крупных городов. С учетом ограничений на человеческие и материальные ресурсы, следует правильно расставить приоритеты, чтобы установить оптимальную последовательность проведения ремонта до-

рог. К сожалению, существующие способы анализа качества дорожного покрытия и используемые для этого устройства не отвечают предъявляемым требованиям к системе анализа качества таких покрытий. Так, ни одна из существующих технологий не позволяет должным образом оценить состояние дорог в пределах района, города или области. Во многом это связано со слишком большой стоимостью существующих комплексов анализа покрытия дороги. Поэтому, проанализировав существующие технологии, было решено создать анализатор качества дорожного покрытия на основе сенсора Kinect [1]. Kinect, помимо стандартной видеокамеры, позволяющей снимать видео в формате вплоть до FULL HD, имеет так называемую камеру глубины, которая состоит из инфракрасного приемника и

передатчика. Передатчик излучает уникальную инфракрасную матрицу, а приемник получает искаженное изображение и на основе искажений определяет расстояние до каждой конкретной точки – это надо перенести ниже в следующий раздел

Целью работы является обоснование оптимального способа организации большого объема хранимых данных, определяющих качество дорожной сети, на основе анализа требований к дорожному покрытию.

СТРУКТУРА ДАННЫХ В ХРАНИЛИЩЕ

На основе анализа существующих технологий оценки состояния дорожного покрытия было решено создать анализатор качества дороги на основе сенсора Kinect [1]. Этот сенсор, помимо стандартной видеокамеры, позволяющей получать видео достаточно высокого качества вплоть до FULL HD, имеет и так называемую камеру глубины, которая состоит из инфракрасного передатчика и приемника. Передатчик излучает уникальную инфракрасную сетку, а приемник получает искаженное изображение сетки и на основе таких искажений определяет расстояние до каждой конкретной точки.

Благодаря наличию приемника и передатчика глубины у сенсора Kinect, существует возможность получать данные о рельефе дороги в режиме реального времени. При этом данные поступают в анализатор в виде так называемых «кадров глубины». Они похожи на обычные цветные кадры, получаемые от видеокамеры. Единственная разница заключается в том, что каждый пиксель кадра глубины задан не цветом, а расстоянием от сенсора до объекта. Подробные способы анализа изображений рассмотрены в [2–6].

Таким образом, кадр глубины представляет собой трехмерное множество точек, а инструментарий разработчика, предоставляемый вместе с сенсором, позволяет переводить такие координаты в обычные трехмерные. В этом случае система координат будет привязана к сенсору, ось Y направлена вверх, а ось Z – совпадать с направлением Kinect (см. рис. 1).

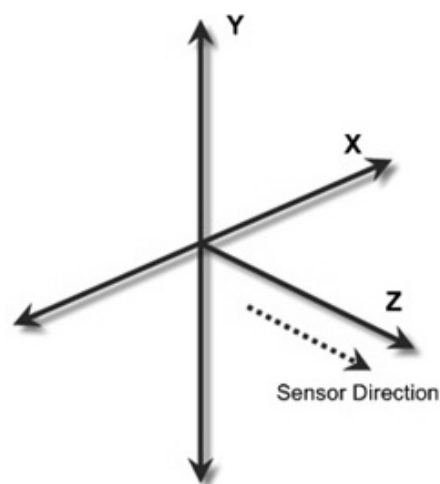


Рис. 1. Трехмерная система координат сенсора Kinect

Проведя детальный анализ различных подходов [7–8], было принято решение анализировать участок дороги, попавший в область видимости сенсора, целиком, а не искать на нем отдельные дефекты.

Каждый участок дороги будет характеризоваться следующими параметрами:

- 1) дата и время проведения анализа;
- 2) глобальные координаты широты и долготы;
- 3) агрегированная оценка (в диапазоне от 0 до 100), показывающая степень неровности дороги;
- 4) цветная фотография выбранного участка дороги.

Глобальные координаты нужны для обеспечения возможности отметить дефекты исследуемой дороги на карте, дата и время проведения анализа требуются, чтобы можно было удалить результаты устаревшего анализа.

Наиболее подходящим способом хранения информации в рамках проведения исследования качества дорожного покрытия будет использование реляционной базы данных. Во-первых, данные, которые нужно сохранить, имеют одинаковую структуру. Во-вторых, должен быть обеспечен доступ к данным в параллельном режиме. Тем не менее, необходимо провести детальный анализ количества записей в БД для того, чтобы определить, какие структуры данных использовать для наиболее быстрого выполнения поступающих запросов.

ОЦЕНКА КОЛИЧЕСТВА ДАННЫХ В БАЗЕ ДАННЫХ

Прежде, чем определять такие требования к производительности, как скорость ответа на запросы чтения из таблицы, скорость выполнения запросов на внесение, изменение и удаление данных, необходимо оценить количество записей в БД, которое определяется числом дефектов покрытия исследуемых дорог. Это число, в свою очередь, будет зависеть от следующих факторов:

- качество дороги – число дефектов на единицу длины;
- количество дорожных полос в каждую сторону;
- общая протяженность дорог.

Будем считать, что в одном километре дороги среднего качества содержится 15 дефектов (число взято с точностью до порядка), данные о которых необходимо сохранить. В таком случае для оценки количества записей во всей БД необходимо определить общую протяженность дорог. Для г. Воронежа она составляет ~ 2000 км. Так как среднее количество полос дорог города равняется двум, то примерное число записей в БД для этого города будет составлять $2000 \times 2 \times 15 = 60000$.

Так как протяженность дорог в целом по России составляет ≈ 982 тыс. км, то, следовательно, в пределах страны придется обрабатывать примерно $982000 \times 3 \times 15 \approx 4 \times 10^7$ записей.

Полученные значения указывают на то, что даже простые манипуляции с базой данных, такие как поиск определенной записи или её удаление, могут занимать время до нескольких секунд. Это связано, во-первых, с тем, что обработка данных об участках дороги будет производиться в режиме реального времени, то есть 24 раза в секунду, и, во-вторых, запросы к БД могут приходиться одновременно от нескольких анализаторов с Kinect, так как программа планируется к использованию в пределах всей страны.

Существует два способа решения проблемы слишком большой длительности обработки запросов к базе данных: либо необходимо реализовать кэширование поступающей ин-

формации на уровне приложения с последующей передачей данных в БД, либо следует ввести дополнительные структуры данных (например, бинарное дерево поиска), позволяющие выполнять запросы намного быстрее.

В рамках решения проблемы анализа качества дорожного покрытия необходимо выполнять следующие действия над данными в БД:

- 1) добавлять новую запись в таблицу;
- 2) выполнять запрос на получение набора записей, находящихся в некотором прямоугольном районе, например, для отображения неровностей на карте;
- 3) удалять «устаревшие» записи из таблицы.

Удаление устаревших записей производится в том случае, когда анализатор движется по дороге, которая ранее уже была подвергнута анализу, и устаревшие данные нужно будет заменить новыми. Такая функция нужна для наиболее удобного использования анализатора качества, а именно, оператор не должен сам решать, какие записи уже устарели, и их надо удалить. Это позволит избежать ошибок человеческого фактора.

Тем не менее, чтобы правильно реализовать функцию удаления устаревших данных, придется отказаться от кэширования поступающих данных в БД и искать решение путем использования таких структур данных, как индексы.

АНАЛИЗ ВРЕМЕННОЙ СЛОЖНОСТИ ЗАПРОСОВ К БД

Для оценки времени выполнения каждого типа запроса к БД необходимо найти временную сложность выполнения каждого типа запроса: выборки, добавления и удаления. Без использования дополнительных структур данных перечисленные функции добавления, удаления и выборки будут выполняться за $O(1)$, $O(N)$ и $O(N)$, соответственно, где N – количество записей в БД.

Необходимо отметить, что функция запроса записей, относящихся к заданному прямоугольному району, будет выполняться реже, чем, например, запрос на удаление, по-

этому время выполнения запроса в несколько секунд еще можно назвать приемлемым. Но функции удаления и добавления записей должны работать в режиме реального времени без задержек. Это значит, что без использования индексов удаление записей будет происходить так долго, что это не позволит выполнять такие запросы несколько раз в секунду.

Один из способов ускорения выполнения запросов к БД является использование структуры данных «индекс», реализованной на основе так называемого «красно-черного дерева». Применение таких структур позволяет отсортировать все записи БД по некоторому ключу и после этого находить нужную запись по ключу за логарифмическое время. Это значит, что если необходимо найти запись в таблице размером в 4×10^7 записей, то на поиск необходимо затратить не более 30 шагов. В сравнении с обычным перебором по всей таблице это увеличивает скорость выполнения запроса на несколько порядков.

Более того, красно-черное дерево устроено так, что при изменении или добавлении одного элемента, перестройка его структуры происходит также за логарифмическое время. Таким образом, добавление нового элемента в таблицу будет осуществляться не более чем за 30 шагов.

Отметим, что прежде чем использовать индекс, необходимо решить, по какому ключу его нужно построить. Самые высокие требования к времени выполнения предъявляются к запросу на удаление устаревших записей. К сожалению, невозможно удалять записи по их идентификатору. В связи с этим было принято решение удалять записи, описывающие дефекты дороги, находящиеся в пределах некоторой зоны (3×3 м). Технически такой запрос будет выполняться для всех записей, находящихся в некоторых диапазонах широты и долготы.

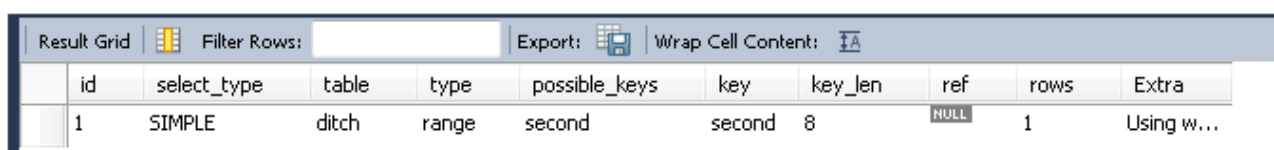
Чтобы понять, каким образом и как быстро будет выполнен некоторый запрос, надо проанализировать результат команды EXPLAIN EXTENDED DELETE FROM.... Команда EXPLAIN EXTENDED показывает информацию о том, каким образом будет выполнен запрос СУБД.

Если в таблице присутствует индекс, например, по широте, то будет выведен следующий результат (см. рис. 2).

Наиболее значимыми здесь являются колонки TYPE и KEY_LEN. Первая из них указывает на то, каким будет способ обхода по таблице. Если там стоит значение ALL, то это означает, что будет просмотрена вся таблица, и запрос будет выполняться медленнее всего. Значение RANGE в случае использования индекса по широте означает, что будет осуществлен бинарный поиск первой записи за логарифмическое время, после чего записи будут просматриваться до того момента, пока не попадется запись, которая не находится в диапазоне допустимой широты. Время выполнения такого типа запроса, в основном, зависит от того, сколько записей попало в диапазон допустимых широт.

Колонка KEY_LEN показывает, сколько байтов ключа будет использовано при определении диапазона. В текущей ситуации значение 8 байтов равно размеру одной переменной вещественного типа, которая как раз и задает широту. Эта колонка очень важна для определения скорости выполнения запроса, если используется ключ, построенный по нескольким полям. Если значение KEY_LEN невелико (например, 4 из 12 байтов), то это означает, что диапазон допустимых записей будет выбран достаточно большим, и запрос будет осуществляться долго, вплоть до одной секунды.

Результат выполнения команды EXPLAIN EXTENDED при использовании индекса по долготе будет аналогичным. Более того, если



id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	ditch	range	second	second	8	NULL	1	Using w...

Рис. 2. Разбор запроса к БД с индексом по широте

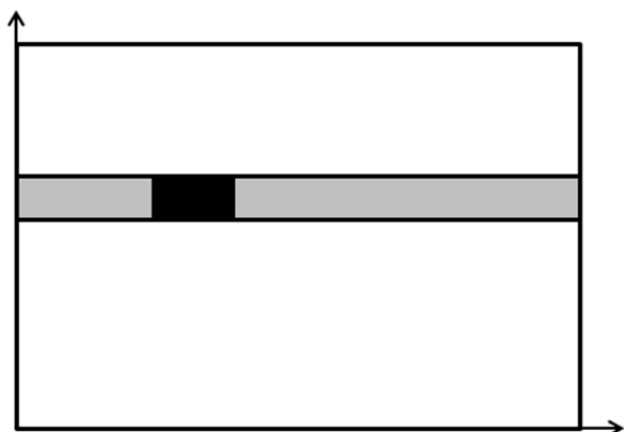


Рис. 3. Зоны поиска с индексом по широте

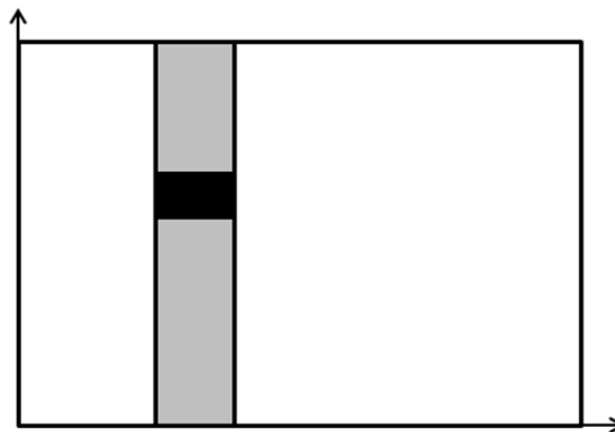


Рис. 4. Зоны поиска с индексом по долготе

ввести ключ как по широте, так и по долготе, значение KEY_LEN по-прежнему будет равно 8 вместо ожидаемых 16. Это происходит из-за того, что невозможно построить структуру данных, которая будет разделять записи по двум диапазонам сразу.

В действительности ход запроса может быть представлен таким образом, как указано на рис. 3.

На рис. 3 представлена координатная плоскость с горизонтальной осью долготы и вертикальной – широты. Белый прямоугольник показывает все возможные координаты, которые могут иметь записи, черный прямоугольник – область, которую необходимо найти. Серый прямоугольник показывает, какие записи система обработает, чтобы получить искомые данные. Основываясь на рисунке, можно сделать вывод, что запрос в данном случае будет недостаточно эффективным, так как серая область существенно превышает черную. Чтобы это показать, можно привести такой пример. Пусть надо удалить записи, входящие в зону размером 4×4 м. С учетом размеров нашей страны получается, что необходимо обработать записи, находящиеся в пределах области размером $4 \times 8 \times 10^6$ м, то есть в $2 \cdot 10^6$ раза большей по сравнению с реальной зоной поиска.

При использовании либо индекса по долготе, либо составного индекса ситуация остается аналогичной предыдущей (рис. 4).

Чтобы получить нужную скорость выполнения запросов (менее 0.1 секунды), необходимо создать индекс по составному ключу. Для этого в таблицу добавляется новое целочисленное поле, указывающее номер зоны, в которой расположена запись. Например, всю координатную плоскость можно разбить на вертикальные полосы равной ширины и пронумеровать их. Тогда номер зоны можно будет вычислить, используя значение долготы. В таком случае необходимо построить индекс по номеру зоны и по широте. Однако следует учесть, что ключ целиком будет использоваться только в том случае, если задан только один номер зоны и диапазон широт (рис. 5).

KEY_LEN показывает, что при выполнении запроса используется целочисленное поле (4 байта) и вещественное (8 байт).

При таком способе составления запросов к базе данных зона поиска искомой области значительно сокращается (рис. 6).

В случае использования составного индекса и построения запросов с указанием одного номера зоны, область поиска будет лишь незначительно больше искомой области. В худшем случае ширина области поиска будет больше искомой области на две ширины зоны.

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	ditch	range	PRIMARY	PRIM...	12	NULL	1	Using w...

Рис. 5. Разбор запроса с составным индексом

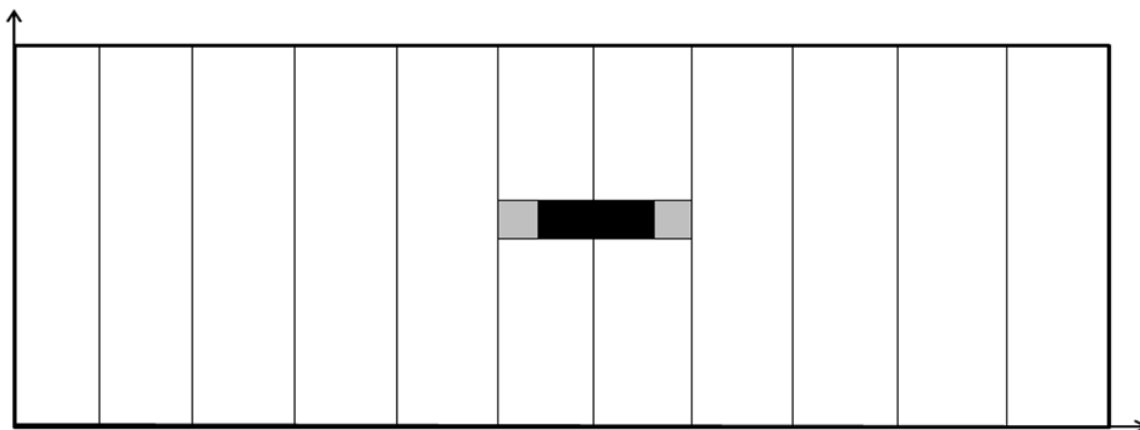


Рис. 6. Зоны поиска с составным индексом

Таким образом, задав ширину зоны в 50 м, область поиска для искомой зоны 4×4 м будет не более 4×100 м, а чаще всего 4×50 м. На практике это значит, что необходимо обработать не более 60 точек. Таким образом, запрос на удаление устаревших записей будет осуществлен за 30 (бинарный поиск) + $60 = 90$ шагов, что обеспечит значительный запас по производительности.

Стоит отметить, что ширина зоны не может быть слишком малой. С одной стороны, чем меньше ширина зоны, тем меньше будет разница между областью поиска и искомой областью. Но в этом случае уменьшение ширины зоны лишь незначительно ускорит выполнение запроса на удаление записей из таблицы. С другой стороны, ограничивающим фактором при построении запросов является обязательное задание только одной зоны в запросе. Это означает, например, что при выполнении запроса на получение записей из большого района надо будет выполнить намного больше отдельных запросов к БД, чем при использовании более широких. Например, количество запросов при ширине зоны 3 м. будет в 10 раз больше, чем при ширине зоны в 30 м.

ВЫВОДЫ

В настоящей работе определены требования к системе контроля качества дорожного покрытия, сделан вывод на основании анализа поступающей из анализатора информации, что наиболее рациональным способом организации хранимых данных является ре-

ляционная БД. Были исследованы различные варианты индексирования в таблице БД, наиболее оптимальным признано применение составного индекса. Использование номера зоны как дополнительной характеристики (поля в таблице БД), а также составного индекса по номеру зоны и по долготе позволяет выполнять все запросы к базе данных в рамках решения проблемы анализа качества дорожного покрытия за приемлемое время (менее 0.1 секунды). При построении запроса к БД с указанием одного номера зоны возможно удаление устаревших данных «на лету». Эта возможность позволяет отказаться от кэширования поступающих данных, и, таким образом, избежать конфликтных ситуаций при добавлении новой информации в БД.

Описанный выше способ организации хранения данных позволяет проводить распараллеливание вычислений. Например, разделение координатной плоскости на зоны дает возможность выделить необходимое количество серверов, ответственных за выбранные зоны. Это позволяет основной программе, выполняющей запрос в режиме параллельной работы, делать запросы к соответствующему серверу, и, дождавшись ответов и собрав их воедино, получить окончательный результат в виде набора дефектов, входящих в искомую прямоугольную область.

СПИСОК ЛИТЕРАТУРЫ

1. Артемов М. А. Создание перспективного анализатора качества дорожного покрытия / М. А. Артемов, С. В. Бабкин, М. В. Кир-

гинцев и др. // Современное общество, образование и наука : сб. научных трудов по материалам Международной научно-практической конференции. – Тамбов, 2014. – Ч. 7. – С. 13–15.

2. Крыловецкий А. А. Автоматическое совмещение поверхностей в системах компьютерного зрения / А. А. Крыловецкий, И. С. Черников, С. Д. Кургалин // Математическое моделирование. – 2013. – Т. 25, № 3. – С. 33–46.

3. Атанов А. В. Параллельный алгоритм реконструкции двумерных объектов на основе радиальных базисных функций / А. В. Атанов, А. А. Крыловецкий, С. Д. Кургалин // Известия ЮФУ. Технические науки. – 2012. – № 6. – С. 195–198.

4. Пространственная реконструкция в системах компьютерного зрения на основе веб-камер / А. В. Атанов, С. Д. Кургалин, А. А. Крыловецкий и др. // Вестник Воронеж. гос. ун-та. Сер. Системный анализ и информационные технологии. – 2011. – № 2. – С. 149–153.

5. Атанов А. В. Параллельный алгоритм реконструкции объектов по неупорядоченному набору точек на основе радиальных базисных функций / А. В. Атанов, А. А. Крыловецкий, С. Д. Кургалин // Вестник Воронежского государственного технического университета. – 2012. – Т. 8, № 10-2. – С. 13–15.

6. Протасов С. И. Подход к решению задачи ректификации стереоизображений по сцене без калибровки камер / С. И. Протасов, А. А. Крыловецкий, С. Д. Кургалин // Информационные технологии. – 2013. – № 2. – С. 61–65.

Бабкин С. В. – магистр 1-го курса, Воронежский государственный университет, факультет ПММ, кафедра Программного обеспечения и администрирования информационных систем. Тел.: (473) 270-20-15
E-mail: bsv5555@yandex.ru

Артемов М. А. – д.ф.-м.н, Воронежский государственный университет, ф-т ПММ, заведующий кафедрой Программного обеспечения и администрирования информационных систем. Тел.: (473) 220-83-37
E-mail: Artemov_m_a@mail.ru

7. Бабкин С. В. Критерии оценки качества дорожного покрытия / С. В. Бабкин, М. А. Артемов // Современное общество, образование и наука : сб. научных трудов по материалам Международной научно-практической конференции. – Тамбов, 2014. – Ч. 7. – С. 18–21.

8. Артемов М. А. Способы анализа дорожного покрытия / М. А. Артемов, С. В. Бабкин, М. В. Киргинцев // Современное общество, образование и наука: сб. научных трудов по материалам международной научно-практической конференции. – Тамбов, 2014. – Ч. 7. – С. 11–13.

9. Авдеев Ю. В. Разработка алгоритма определения координат в задаче дистанционного управления движением машинно-тракторных агрегатов / Ю. В. Авдеев, А. Д. Кононов, А. А. Кононов // Механизация и электрификация сельского хозяйства : межведомственный тематический сборник. – Вып. 46. – Минск, 2012. – С. 24–31.

10. Артемов М. А. Особенности использования фазовых измерений систем типа ГЛОНАСС для повышения точности в навигационных задачах / М. А. Артемов, Д. Е. Кочкин // Вестник Воронежского государственного университета. Сер. Системный анализ и информационные технологии. – 2010. – № 2. – С. 19–26.

11. Авдеев Ю. В. Разработка алгоритма определения координат в задаче дистанционного управления движением машинно-тракторных агрегатов / Ю. В. Авдеев, А. Д. Кононов, А. А. Кононов // Механизация и электрификация сельского хозяйства : межведомственный тематический сборник. – Вып. 46. – Минск, 2012. – С. 24–31.

Babkin S. V. – master 1st course, Voronezh State University, AMM faculty, Chair of Programming Software and Administration of Information Systems. Tel. : (473) 270-20-15
E-mail: bsv5555@yandex.ru

Artemov M. A. – Doctor of Phys. And Mathem. Science, Professor, Voronezh State University, AMM Faculty, Head of the Chair of Programming Software and Administration of Information Systems. Tel.: (473) 220-83-37
E-mail: Artemov_m_a@mail.ru