

Московский государственный университет имени М. В. Ломоносова

На правах рукописи



Якубенко Антон Анатольевич

**Алгоритмы построения трехмерных моделей объектов
с регулярной структурой по фотографиям
при взаимодействии с пользователем
для виртуальных сред**

Специальность 05.13.11 – математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Автореферат
диссертации на соискание учёной степени
кандидата физико-математических наук

Москва – 2013

Работа выполнена на кафедре автоматизации систем вычислительных комплексов факультета вычислительной математики и кибернетики Московского государственного университета имени М.В. Ломоносова.

Научный руководитель: Кандидат физико-математических наук, доцент
Баяковский Юрий Матвеевич

Официальные оппоненты: Доктор физико-математических наук, профессор
Института прикладной математики им. М.В. Келдыша
Российской академии наук Соколов Сергей Михайлович
Кандидат технических наук, доцент Национального
Исследовательского Ядерного Университета (МИФИ)
Сафонов Илья Владимирович

Ведущая организация: Государственный научно–исследовательский институт
авиационных систем (ГосНИИАС)

Защита состоится 29 марта 2013 г. в 11 часов на заседании диссертационного совета Д 501.001.44 при Московском государственном университете имени М.В. Ломоносова по адресу: 119991, ГСП-1, Москва, Ленинские горы, МГУ, 2-й учебный корпус, факультет ВМК, аудитория 685. Желаящие присутствовать на заседании диссертационного совета должны сообщить об этом за 2 дня до указанной даты по тел. (495) 939-30-10 (для оформления заявки на пропуск).

С диссертацией можно ознакомиться в Фундаментальной библиотеке МГУ имени М.В. Ломоносова. С текстом автореферата можно ознакомиться на официальном сайте ВМК МГУ имени М.В. Ломоносова <http://www.cmc.msu.ru> в разделе «Наука» - «Работа диссертационных советов» - «Д 501.001.44».

Автореферат разослан 22 февраля 2013 г.

Учёный секретарь
диссертационного совета
профессор



Н.П. Трифонов

Общая характеристика работы

Объект исследования и актуальность работы

Виртуальная реальность – это компьютерная модель мира, передаваемая человеку через его ощущения, в первую очередь, через зрение. Для практических целей особенно значимо создание систем виртуальной реальности, моделирующих реальный мир. Основным содержанием таких систем являются трехмерные модели объектов. Для создания качественных моделей существует ряд способов, но они требуют значительных затрат времени на ручную работу и, соответственно, больших материальных вложений.

Чтобы сократить это время можно учитывать специфику класса объектов. Одним из таких классов объектов являются объекты с регулярной структурой (рис. 1). Такие объекты могут быть представлены одной или несколькими смежными плоскостями, на которых с некоторой регулярностью расположены повторяющиеся элементы.



Рис 1. Объекты реального мира с регулярной структурой.

Для практики среди подобных объектов наиболее значимы здания. Их трехмерные модели используются для геоинформационных систем, например, для картографических веб-сервисов и мобильных навигационных систем. Такими сервисами пользуются сотни миллионов человек. В мире миллионы зданий и нужны эффективные инструменты для создания их трехмерных моделей. Для построения реалистичных моделей обычно используются фотографии. Так как полностью автоматические алгоритмы часто не дают желаемого результата, то можно использовать взаимодействие с пользователем.

Цель работы

Целью работы является разработка алгоритмов и программной системы, позволяющих по фотографиям при взаимодействии с пользователем построить трехмерные модели объектов с регулярной структурой за меньшее время по сравнению с существующими аналогами.

Для достижения поставленной цели в работе предложен следующий алгоритм:

1. Из фотографий создаются прямоугольные текстуры плоскостей объекта, упрощающие дальнейший анализ, и строится трехмерная модель объекта.
2. По текстуре объекта определяется его регулярная структура, в том числе используемая для клонирования трехмерных моделей регулярных элементов.
3. На текстурах объекта определяются посторонние объекты, и текстура под ними восстанавливается с учетом регулярной структуры объекта.

Исходя из предложенной схемы алгоритма, возникают следующие задачи:

1. Предложить модель регулярной структуры объекта и алгоритм ее вычисления по его изображению.
2. Разработать алгоритм восстановления текстуры объекта с учетом его структуры.
3. Разработать алгоритм ректификации текстур и построения трехмерной модели части объекта, видимой на одной фотографии.

Научная новизна работы

Разработаны новые алгоритмы и программная система, позволяющие построить трехмерные модели объектов с регулярной структурой по фотографиям при меньшем объеме взаимодействия с пользователем и, соответственно, быстрее, чем в аналогах.

Предложена новая модель регулярной структуры объекта и разработан новый автоматический алгоритм ее определения по изображению. Алгоритм позволяет точнее определять регулярную структуру большего числа объектов, чем существующие решения.

Разработан новый алгоритм восстановления частей текстуры объекта, невидимых или загороженных объектами переднего плана, с учетом регулярной структуры объектов. Алгоритм требует от пользователя задания единственного простого параметра, после чего работает полностью автоматически. По сравнению с аналогами использование регулярной структуры объекта минимизирует взаимодействие с пользователем.

Разработан новый автоматизированный алгоритм получения текстур плоскостей объекта и построения трехмерной модели объекта по фотографии. Предложенный алгоритм позволяет извлечь из фотографии текстуры всех видимых на ней плоскостей объекта и построить их трехмерную модель при меньшем взаимодействии с пользователем по сравнению с существующими ручными подходами и в более контролируемой манере, чем в случае автоматических существующих алгоритмов.

Практическая значимость и реализация

Разработаны и доведены до практической реализации алгоритмы построения трехмерных моделей объектов с регулярной структурой. Программные реализации описываемых в диссертации алгоритмов удовлетворяют всем требованиям и ограничениям, сформулированным при постановке задач.

На основе предложенных алгоритмов разработана система построения трехмерных моделей объектов с регулярной структурой по фотографиям, требующая существенно меньшего объема взаимодействия с пользователем, чем существующие аналоги. Система разрабатывалась в лаборатории компьютерной графики и мультимедиа (ЛКГиМ) кафедры автоматизации систем и вычислительных комплексов (АСВК) факультета вычислительной математики и кибернетики (ВМК) Московского государственного университета имени М. В. Ломоносова в рамках проектов с компаниями Samsung Advanced Institute of Technology и ООО «Медиа Софт Интегро». Система используется для создания трехмерных моделей зданий города Москвы и других городов мира.

Апробация работы

Основные результаты работы докладывались и обсуждались на:

- 20-ой международной конференции по компьютерной графике и машинному зрению GraphiCon 2010, Россия, Санкт-Петербург, 2010;
- 19-ой международной конференции по компьютерной графике и машинному зрению GraphiCon 2009, Россия, Москва, 2009;
- 10-ой международной конференции по компьютерному зрению European Conference on Computer Vision (ECCV) 2008, France, Marseille, 2008;
- 7-ой международной конференции в области математической кибернетики и теоретической информатики Интеллектуализация обработки информации (ИОИ) 2008, Украина, Алушта, 2008;
- 18-ой международной конференции по компьютерной графике и машинному зрению GraphiCon 2008, Россия, Москва, 2008;
- 15-ой международной научной конференции студентов, аспирантов и молодых учёных «Ломоносов-2008», Россия, Москва, 2008;
- 14-ой международной научной конференции студентов, аспирантов и молодых учёных «Ломоносов-2007», Россия, Москва, 2007;

- семинаре по компьютерной графике и машинному зрению Ю. М. Баяковского (факультет ВМК МГУ);
- семинаре аспирантов кафедры АСВК факультета ВМК МГУ под руководством Л. Н. Королева.

Публикации

По результатам работы автором опубликовано 11 научных работ, из них 3 статьи в рецензируемых журналах [1-3], 2 из которых включены в список ВАК [1, 2], 5 статей [4-8] и 3 тезисные публикации [9-11] в сборниках трудов международных конференций. Автор работы является соавтором 4 патентов в России, США и Южной Корее.

Структура и объем работы

Диссертация состоит из введения, пяти глав, заключения и списка литературы. Содержание работы изложено на 143 страницах. Список литературы включает 106 наименований. В работе содержится 70 рисунков.

Содержание работы

В первой главе приводится обзор существующих подходов к получению трехмерных моделей объектов, описывается схема предложенного алгоритма построения трехмерных моделей объектов с регулярной структурой по фотографиям при взаимодействии с пользователем, задается методология сравнения с аналогами.

В первом разделе рассматриваются современные подходы к созданию трехмерных объектов реального мира с регулярной структурой. Средства трехмерного моделирования общего назначения в комбинации с редакторами для фотографий позволяют вручную смоделировать любой объект с фотореалистичным качеством и высокой геометрической точностью. Но это требует огромных временных и, соответственно, финансовых затрат.

Фотограмметрические системы позволяют рисовать объекты непосредственно на фотографиях. Но для этого на фотографиях нужно вручную выделить множество соответствующих точек, по которым вычисляются параметры камер. Большое число часто снятых последовательных фотографий можно автоматически сопоставлять и по ним восстанавливать разреженное или плотное облако трехмерных точек. Такой подход «структуры из движения» часто путает точки на объектах с регулярной структурой.

Более надежным способом получения плотных и точных облаков трехмерных точек является использование лазерных сканеров, в том числе, дорогих систем мобильного картографирования. Они позволяют автоматизировать процесс сбора данных. Но эффективный анализ огромных массивов данных остается открытым вопросом.

Во втором разделе описывается схема предложенного алгоритма. Входными данными являются разрозненные фотографии объекта. Получение таких данных проще, дешевле и гибче. На выходе требуется получить трехмерную полигональную модель, представляющую собой геометрию и текстуру плоскостей объекта и его регулярных элементов. Для зданий получаемые модели соответствуют второму и третьему уровням детальности по международному стандарту CityGML для трехмерных карт.

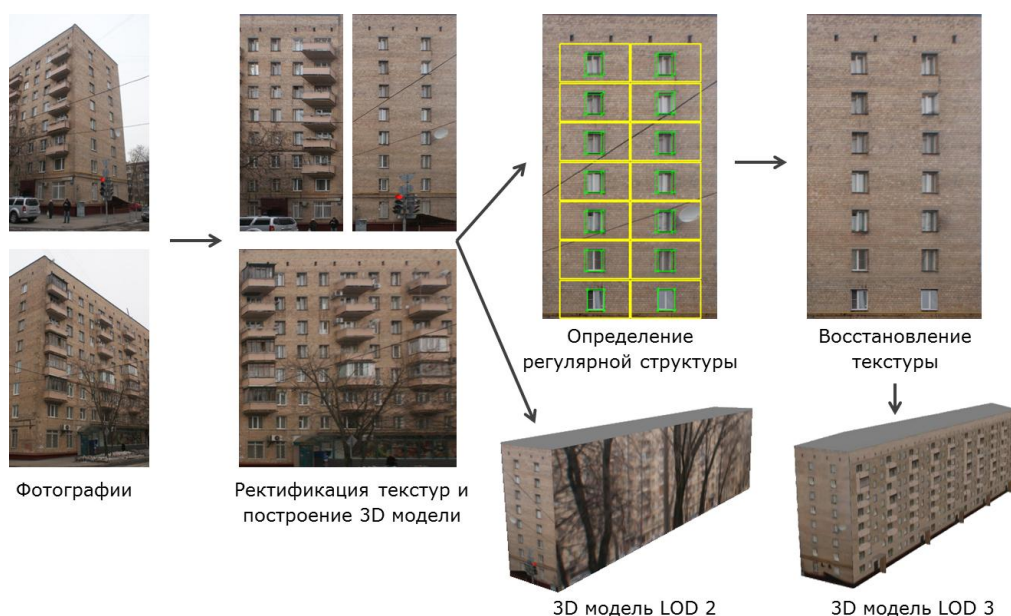


Рис. 2. Схема предложенного алгоритма.

Схема предложенного алгоритма (рис. 2):

1. Сначала из фотографий создаются прямоугольные текстуры плоскостей объекта и строится трехмерная модель второго уровня детальности. Компенсация перспективных искажений упрощает дальнейший анализ.
2. Затем определяется регулярная структура объекта по его текстуре. С ее помощью можно, в том числе, клонировать повторяющиеся трехмерные элементы и сжать текстуру модели.
3. На текстурах объекта могут присутствовать посторонние объекты. Например, в случае зданий, фасад могут загромождать деревья и провода. Необходимо определить такие посторонние объекты и восстановить текстуру под ними. При этом учитывается регулярная структура объекта.

В третьем разделе задается методология сравнения с аналогами. Проводится сравнение как итоговой системы, так и алгоритмов для решения подзадач. В области трехмерного моделирования и интерпретации объектов с регулярной структурой не существует единой базы изображений для тестирования. Для сравнения были выбраны изображения из доступных баз и собственные фотографии.

Вторая глава посвящена задаче получения ректифицированных текстур плоскостей объекта и построения его простой трехмерной модели по фотографиям. Для плоских объектов ректифицированное изображение компенсирует перспективные искажения. В главе приводится обзор существующих подходов решения данной задачи, и описываются предложенные новые алгоритмы.

В первом разделе формулируется постановка задачи. Требуется построить трехмерную модель, состоящую из вертикальных плоскостей, и получить из исходных фотографий единую ректифицированную текстуру.

Во втором разделе проводится обзор существующих алгоритмов получения ректифицированных текстур из фотографий. Самым популярным способом получения таких текстур является выделение на изображении четырехугольника, соответствующего прямоугольнику в пространстве, например, стены.

Альтернативным способом является задание прямоугольной системы координат через точки схода. Точки схода – это точки на изображении, где пересекаются проекции параллельных в пространстве линий. Например, в Google SketchUp пользователь должен задать на фотографии две пары отрезков, определяющих горизонтальные точки схода для перпендикулярных плоскостей. В заданной системе координат можно осуществлять трехмерное моделирование с использованием модели перспективной геометрии.

Недостатком всех существующих методов является необходимость долгого взаимодействия с пользователем для создания точной ректифицированной текстуры.

В третьем разделе описывается предложенный алгоритм решения задачи (рис. 3).

1. Исправление углов наклона и крена камеры (рис. 3б). При этом вертикальные в пространстве линии становятся вертикальными на изображении.
2. Определение уровня горизонта и точек схода (рис. 3в). Горизонт на фотографиях с исправленным углом наклона камеры является горизонтальной линией. Пользователь может подправить автоматически определенное положение горизонта вручную.

3. Задание линии контура объекта и его вертикальных границ (рис. 3г). Пользователь указывает на фотографии проекцию линии сечения объекта какой-либо горизонтальной плоскостью на изображение. При рисовании такой ломаной ее отрезки автоматически привязываются к направлениям на вычисленные точки схода. Точками задаются верхняя и нижняя границы объекта. Такие данные позволяют получить трехмерную модель видимых плоскостей объекта и ректифицированные текстуры для каждой плоскости.

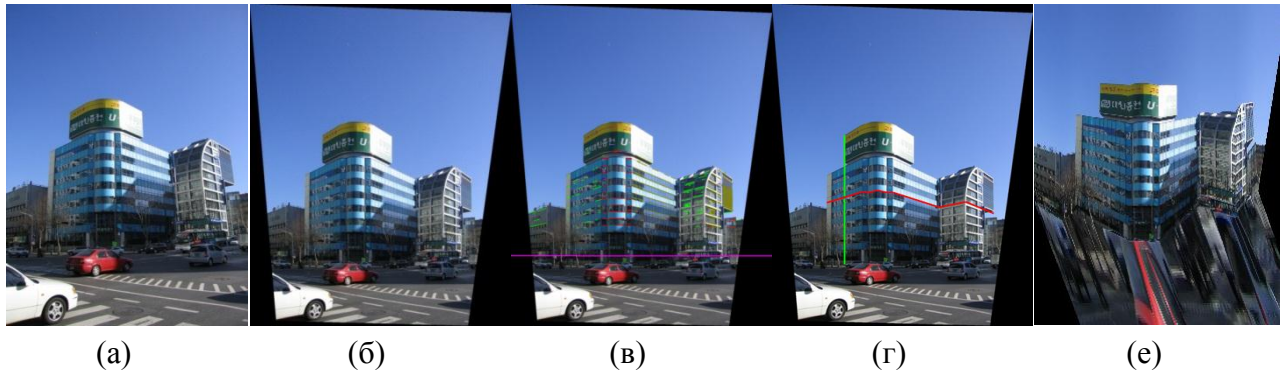


Рис. 3. Схема алгоритма получения ректифицированных текстур объекта и построения его трехмерной модели по фотографии.

Предлагается вычислять виртуальный вид с нулевыми углами наклона и крена, что позволяет упростить дальнейший анализ. Пусть I – исходное изображение, I' – искомый виртуальный вид. Ориентация камеры для изображений I и I' отличается только углами наклона aX и крена aZ . Искомое преобразование изображения I к I' можно описать поворотной гомографией H , которая может быть параметризована этими двумя углами:

$$H = \begin{bmatrix} \cos(aZ) & \sin(aZ) & 0 \\ -\cos(aX) * \sin(aZ) & \cos(aX) * \cos(aZ) & \sin(aX) \\ \sin(aX) * \sin(aZ) & -\sin(aX) * \cos(aZ) & \cos(aX) \end{bmatrix}$$

На изображениях часто присутствуют прямые отрезки. Находятся такие отрезки l_i , $i = 1..N$, которые соответствуют вертикальной точке схода. Пусть после применения к отрезкам l_i преобразования H они переходят в отрезки k_i . Составляется целевая функция G отклонения отрезков k_i от вертикалей. Для этого можно использовать, например, сумму квадратов разницы в X координатах концов k_i^1 и k_i^2 отрезков k_i :

$$G_i = (k_i^1.x - k_i^2.x)^2, \quad G = \sum_{i=1}^N G_i$$

Находятся такие углы наклона aX и крена aZ , что рассчитанное по ним преобразование H дает минимум функции G для заданных отрезков l_i .

$$k_i = H(l_i) = F(l_i, H) = F(l_i, aX, aZ), G = G(k_i)$$

$$(aX, aZ) = \operatorname{argmin} G$$

Минимизация заданной функции осуществляется с помощью алгоритма градиентного спуска. В качестве первого приближения задаются нулевые углы aX и aZ .

Для поиска линий, принадлежащих вертикальной точке схода используется устойчивый алгоритм оценки параметров модели на основе случайных выборок (RANSAC). В предположении, что камера не была слишком сильно повернута, также отбрасываются отрезки, отклоняющиеся от вертикали на угол более 30 градусов. В случае плохой работы алгоритма пользователь может вручную указать несколько линий, которые должны быть вертикальными.

На изображении после коррекции углов наклона и крена все точки схода имеют одинаковую y -координату. Для оценки уровня горизонта используется y -координата точки схода, к которой относится наибольшее число линий. Для оценки остальных точек схода линии группируются и для каждой группы вычисляется точка схода. Группы параллельных линий формируют пики распределения x -координат пересечений линий с горизонтом. Для кластеризации применяется алгоритм сдвига среднего.

В четвертом разделе описывается тестирование и сравнение предложенного алгоритма с аналогами. Проведенные эксперименты с алгоритмом исправления угла наклона и крена камеры показали, что на 96 фотографий из 100 отклонение линий, которые должны быть вертикальными, от вертикалей не превышает 0.2% от диагонали изображения. Для тестирования алгоритмов определения горизонта и точек схода были отобраны 20 изображений. На 75% изображений отклонение вычисленного автоматическим алгоритмом горизонта от горизонта, вычисленного по вручную заданным точкам схода, не превышало 5% от высоты изображения. Точки схода служат для привязки при выделении линии контура объекта. Отобранные изображения содержат 29 плоскостей объектов. Привязка к точкам схода была использована в 67% из них.

Для сравнения алгоритма получения ректифицированных текстур объекта и построения его трехмерной модели по фотографиям были выбраны следующие подходы: выделение стен по одной с помощью задания четырехугольника в Adobe Photoshop и задание двух точек схода и геометрии в Google SketchUp. Проведенное тестирование показало (рис. 4), что предложенный алгоритм позволяет в среднем в 1.85 раза быстрее получать ректифицированные текстуры, чем альтернативные методы. При этом разница становится все более заметной при росте числа плоскостей объекта на одной фотографии.

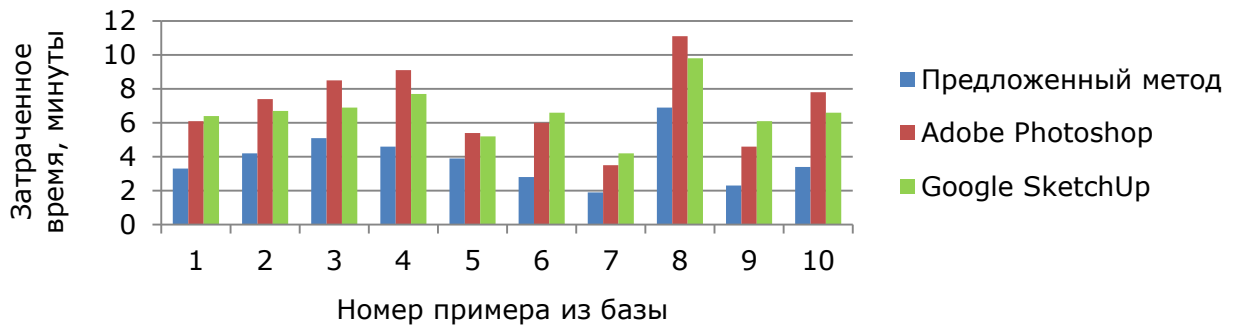


Рис. 4. Результаты сравнения предложенного алгоритма получения ректифицированных текстур объекта и построения его трехмерной модели по фотографиям с аналогами.

Третья глава посвящена задаче извлечения информации о регулярной структуре объекта из его ректифицированной текстуры. В ней приводится обзор существующих постановок задачи и алгоритмов ее решения. Описывается предложенная новая модель регулярной структуры объекта и алгоритм ее вычисления по изображению объекта.

В первом разделе формулируется постановка задачи. Требуется определить регулярную структуру объекта по его ректифицированной текстуре для последующего определения областей переднего плана и сжатия текстуры трехмерной модели.

Во втором разделе проводится обзор существующих подходов к решению поставленной задачи. Анализ пиков автокорреляции изображения подходит только для случаев одной строгой регулярности, которая занимает большую часть изображения без значительных окклюзий. Для более сложных случаев можно использовать разреженные фрагменты изображения. Они могут выбираться случайным образом или как окрестности особенностей. Похожие между собой фрагменты группируются с помощью оценки некоторой модели регулярности. Качество работы таких алгоритмов резко падает при анализе объектов с более сложными регулярными структурами.

В задачах определения структуры объектов, исследователи часто ограничиваются классом объектов «здания», как наиболее актуальным для практики. Тогда задача может формулироваться как поиск окон. Для решения используется детектор подобный Viola-Jones или эвристические подходы, учитывающие похожесть углов окон и наличие градиента по периметру окон. Окна сильно отличаются друг от друга, поэтому текущие результаты алгоритмов поиска окон не дают финального решения.

Лучшие алгоритмы для интерпретации фасадов пытаются описать входное изображение высокоуровневой моделью, например, грамматическим выводом. Выбор правил грамматики (например, для вертикальных или горизонтальных разделений на пролеты и этажи) и их параметров (например, координат разделений) может выполняться

с помощью случайных Марковских цепей. Текущие алгоритмы с трудом справляются с окклюзиями и сильно зависят от правил грамматического вывода и выбора метрик.

В третьем разделе описывается предложенный алгоритм. Существующие модели для описания регулярных структур либо слишком просты и не описывают многие объекты, либо слишком сложны для надежного определения по изображению. Поэтому была предложена новая модель регулярной структуры объекта – набор непересекающихся решеток похожих между собой прямоугольных фрагментов (рис. 5). Оси решетки параллельны осям изображения. Каждая решетка описывается 6 параметрами:

(cx, cy) – X и Y координаты верхнего левого угла решетки,

(sx, sy) – размер ячейки решетки,

$(rows, columns)$ – число рядов и столбцов в решетке.



Рис. 5. Предложенная модель регулярной структуры объектов.

Для вывода такой модели по ректифицированному изображению объекта был предложен следующий алгоритм (рис. 6). Главный шаг алгоритма – последовательный выбор лучших решеток, не пересекающихся с ранее найденными. Вводится функционал качества решетки и осуществляется жадный поиск лучшей решетки. Для снижения времени работы предлагается алгоритм сокращения пространства поиска за счет поиска и анализа повторяющихся прямоугольников. Лучшие решетки расширяются на области с загораживающимися объектами за счет снижения порогов на сходство ячеек решеток в функционале качества решетки. Из-за дискретизации пространства поиска и ограничений на качество выбираемых решеток, некоторые решетки могут быть пропущены. Поэтому дополнительно ищутся решетки с ячейками, похожими на ячейки уже найденных решеток. Такие решетки также расширяются. Когда больше нельзя найти новых решеток, все найденные решетки расширяются на области с сильно загораживающими их

посторонними объектами за счет повторного снижения порогов на сходство ячеек в функционале качества решеток. Если какие-то решетки нашлись неправильно, то пользователь может подправить размеры и положение решеток, а также создать новые.



Рис. 6. Схема работы алгоритма определения регулярной структуры объекта.

На изображении ищутся повторяющиеся примитивные элементы, например, прямоугольники. Чтобы не находить лишние прямоугольники в области шума и мелкой текстуры объекта, вычисляются края на пирамиде изображений, то есть краем считается тот пиксель, который является краем как на исходном изображении, так и на его уменьшенных копиях. Для каждого пикселя определяется, может ли он быть углом прямоугольника. Например, пиксель с краями сверху и слева может стать верхним левым углом прямоугольника. Анализируются пары углов, которые могли бы быть стороной прямоугольника и отбираются самые частые расстояния между ними. Производится перебор всех прямоугольников с найденными углами и самых частых размеров. Сравниваются пары фрагментов изображения, соответствующих прямоугольникам одинакового размера, с помощью нормализованной кросс-корреляции. Сохраняется расстояние между прямоугольниками, если значение метрики достаточно велико. Самые частые расстояния (sx, sy) становятся кандидатами размеров ячеек решеток. Центры прямоугольников становятся кандидатами центров ячеек решеток и таким образом определяют (cx, cy) . Такой алгоритм позволяет существенно снизить число комбинаций параметров решеток для осуществления полного перебора.

Решетки выглядят естественно, если в центре их ячеек находится некоторый элемент, например, прямоугольник. Такое центрирование решеток предотвращает потерю отдельных рядов и столбцов ячеек вблизи краев изображения. Выбор лучшей решетки формулируется как задача оптимизации функционала качества решетки G :

$$G = (M * N)^{-t_1} \sum_{i=1}^M \sum_{j=1}^N Q_{i,j},$$

где M – число рядов решетки, N – число столбцов решетки, $t_1 = -0.8$ – регуляризующий параметр, штрафующий маленькие решетки, а $Q_{i,j}$ – значение меры качества ячейки $c_{i,j}$:

$Q_{i,j} = Similar(c_{i,j}, c_{i-1,j}) + Similar(c_{i,j}, c_{i,j-1}) + HasRect(c_{i,j}) + Symmetry(c_{i,j})$, где:

- $Similar(c_{i,j}, c_{k,l})$ – значение меры сходства между ячейками $c_{i,j}$ и $c_{k,l}$. Оно равно значению нормализованной кросс-корреляции соответствующих фрагментов изображения, если оно больше порога t_2 , и $-t_2$ в противном случае.
- $HasRect(c_{i,j})$ равно t_2 , если внутри ячейки $c_{i,j}$ есть прямоугольник и $-t_2$ в противном случае. Это первое слагаемое, которое служит для центрирования решеток.
- $Symmetry(c_{i,j})$ – значение меры горизонтальной симметрии ячейки $c_{i,j}$. Оно равно значению нормализованной кросс-корреляции левой и правой частей ячейки, если оно больше порога t_2 , и $-t_2$ в противном случае. Эта составляющая весовой функции нужна, так как, если центрировать решетки, используя только информацию о прямоугольниках, то наравне с правильными решетками, центрированными по основным повторяющимся элементам, будут найдены ошибочные, центрированные по другим прямоугольным элементам.

В четвертом разделе проводится сравнение предложенного алгоритма с двумя современными аналогами на 25 изображениях из собранной базы. Алгоритм [1 – Park, ECCV'2008] был выбран как представитель алгоритмов для восстановления регулярностей общего вида. Он основан на сравнении окрестностей особых точек и итеративном построении гибких решеток. Алгоритм [2 – Wenzel, PRIA'2008] использует для описания структуры более сложную модель – линейную комбинацию сдвигов вдоль осей изображения с соответствующими коэффициентами, а также регуляризацию согласно принципу минимальной длины описания. Подбираются такие параметры модели, чтобы похожие пары SIFT особенности переходили друг в друга.

Выбранные алгоритмы не пытаются центрировать решетки, поэтому они могут терять некоторые строки и столбцы решеток. Сначала оценивается только результат нахождения размеров ячеек решеток (рис. 7). Считается, что размер ячеек решеток определен верно, если в финальном результате есть решетка с верным размером ячеек.

Найденная решетка считается определенной верно, если все ее ячейки визуально похожи, более не подлежат дроблению, и в решетке верное число строк и столбцов. Для оценки (рис. 8) вычисляется отношение числа верно найденных ячеек к должному общему числу ячеек. Сравнение с алгоритмом [2] не проводится, так как для позиционирования решетки он требует указания ее начала вручную.

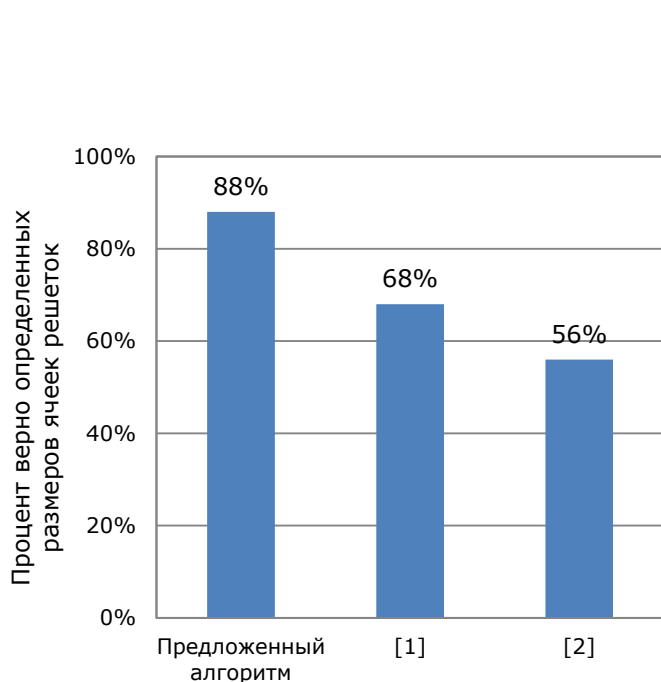


Рис. 7. Оценка результатов нахождения размеров ячеек решеток.

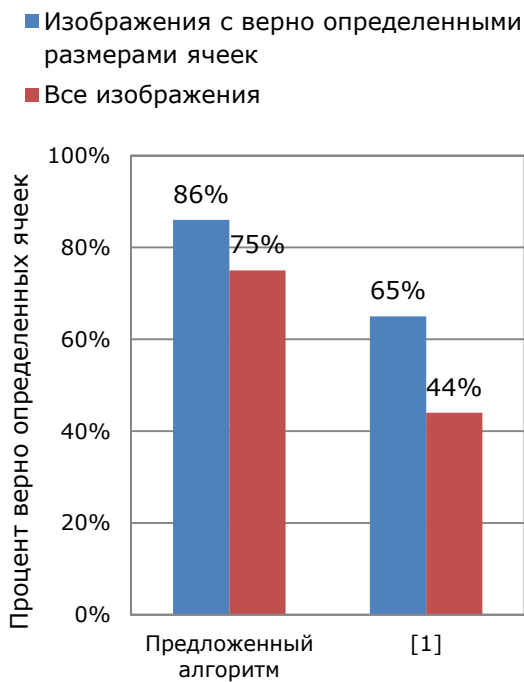


Рис. 8. Оценка доли верно найденных похожих элементов.

Четвертая глава посвящена задаче определения областей посторонних объектов на текстуре объекта и восстановления этих областей. Приводится обзор существующих алгоритмов сегментации и восстановления изображений. Описывается предложенный новый алгоритм восстановления текстуры объектов с учетом их регулярной структуры.

В первом разделе формулируется постановка задачи. На изображениях объект может быть загорожен другими объектами, например, фасад здания могут загораживать деревья и провода. Требуется выделить области переднего плана и восстановить текстуру под ними так, чтобы результат казался визуально естественным. Требуется возможность коррекции результата автоматических алгоритмов на каждом из двух шагов.

Во втором разделе проводится обзор существующих алгоритмов сегментации изображений. Современные автоматические алгоритмы не способны решать произвольные задачи сегментации с гарантированным результатом. Поэтому всё большее внимание стало уделяться интерактивной сегментации изображений. Большинство

современных алгоритмов сегментации являются развитием алгоритма GraphCut. Алгоритм трактует всё изображение как граф, каждая вершина которого соответствует пикселю изображения. Вершины, соответствующие соседним пикселям, связываются ребрами. Также добавляются 2 терминальные вершины, называемые истоком и стоком, которые связываются с остальными вершинами. На ребрах графа определяется весовая функция.

Пользователь указывает несколько пикселей (семена), принадлежащих объекту и фону. Вершины графа, соответствующие семенам объекта и фона, связываются соответственно с истоком и стоком ребрами с бесконечно большим весом. В графе находится минимальный разрез, который делит граф на две части. Пиксели, попавшие в один подграф с истоком, считаются объектом, остальные – фоном. Бесконечный вес ребер между семенами обеспечивает выполнение заданных пользователем ограничений. Чем больше отличаются цвета соседних пикселей, тем вес ребра между ними меньше, а значит более вероятно, что разрез графа пройдет между ними.

Для современных алгоритмов интерактивной сегментации изображений текстуры объектов с регулярной структурой, загороженные объектами переднего плана, являются сложной задачей. Существующие алгоритмы в основном рассчитаны на выделение одного объекта с гладкой границей на контрастном фоне. В данном случае есть несколько объектов с рваными границами и порой не сильно отличающихся по цвету от фона.

В третьем разделе проводится обзор алгоритмов восстановления и синтеза изображений. Их можно грубо разделить на два больших класса. Алгоритмы первого класса основаны на составлении и решении уравнений в частных производных. Они хорошо подходят только для восстановления узких и небольших областей, таких как царапины на фотографиях. В противном случае результат получается очень размытым.

Методы второго класса основаны на копировании информации с остальной части изображения на неизвестную область попиксельно или небольшими фрагментами. Некоторые алгоритмы жадно заполняют неизвестную область, часто с использованием функции приоритета. Другие же формулируют специальную функцию энергии, и находят такое копирование фрагментов, которое минимизирует эту функцию.

Современные алгоритмы позволяют быстро и качественно восстановить изображение естественной природы в небольших областях без сильной структуры. Текстуры объектов с регулярной структурой обладают явно выраженной структурой, требующей сохранения. Алгоритмы, учитывающие особенности объектов интереса,

ограничиваются простыми случаями загромождения и, например, не справляются со случаями, когда объекты переднего плана загромождают около половины ячеек.

В четвертом разделе описывается предложенный алгоритм. Алгоритм получает на вход ректифицированную текстуру объекта и решетки его регулярной структуры. Алгоритм работает независимо с каждым типом решеток. Схема алгоритма (рис. 9):

1. Нахождение «чистых» (не загроможденных объектами переднего плана) ячеек. Пользователь может указать такие ячейки и вручную.
2. Сегментация объектов переднего плана в остальных ячейках. Пользователь может варьировать критерий отнесения области к объектам переднего плана, регулируя единственный параметр.
3. Восстановление текстуры. Области переднего плана замещаются соответствующими областями из чистой ячейки соответствующего типа.

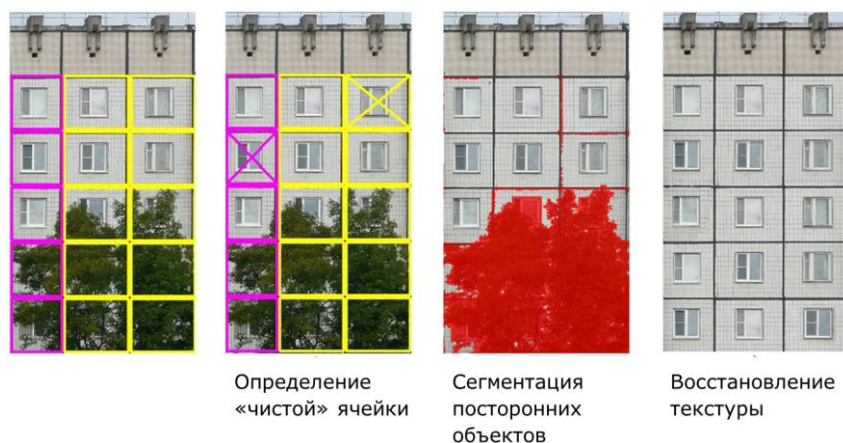


Рис. 9. Схема алгоритма восстановления текстуры объекта с учетом его структуры.

Алгоритм нахождения чистой ячейки основан на предположении, что чистые ячейки похожи между собой сильнее, чем ячейки, загроможденные объектами переднего плана. Так как существующие способы сравнения изображений не позволяют надежно определять похожие ячейки, была предложена новая метрика. Чтобы сделать метрику инвариантной к изменениям в освещении сравниваются не сами изображения, а градиенты на них. Для компенсации возможных небольших несоответствий в положении ячеек решеток предлагается размывать найденные на изображении края с помощью преобразования расстояний (distance transform). Для каждого пикселя изображения записывается расстояние до ближайшего к нему края. Сравнение полученных изображений ведется попиксельно как сумма квадратов разностей. Чистой ячейкой объявляется та, сумма расстояний от которой до других по метрике минимальна.

Задачу сегментации можно сформулировать как необходимость присвоения каждому пикселю каждой ячейки метки объекта или метки объектов переднего плана. В данной работе для сегментации используется модель Марковского случайного поля (MRF) – графическая модель, состоящая из набора случайных величин и связей между ними. Всем пикселям изображения ставятся в соответствии случайные величины, принимающие значения цветов этих пикселей, и неизвестные случайные величины, соответствующие метке пикселя. Проставляются связи, показывающие непосредственно зависящие друг от друга пиксели. Необходимо максимизировать общую апостериорную вероятность модели за счет выбора значений меток. Задачу максимизации вероятности можно представить как задачу минимизации энергии на графе. Вершины такого графа соответствуют пикселям изображения. Каждой вершине сопоставляется метка. Ребра графа строятся между пикселями, связанными друг с другом в модели. Функция энергии состоит из двух видов слагаемых: бинарных, которые формулируются для каждой связи между пикселями, и унарных, которые формулируются для каждого пикселя:

$$E = \sum_p E_u(p, l) + \sum_{p,q} E_b(p, q, l_p, l_q)$$

где p, q – различные пиксели, а l – метка, которая может принимать бинарное значение «объект» или «объект переднего плана». Используемая функция энергии удовлетворяет условиям регулярности и может быть минимизирована алгоритмом разреза графа. Граф строится для каждой имеющейся ячейки.

Значения бинарных слагаемых для соседних точек отражают тот факт, что близкие по значению пиксели скорее всего относятся к одному классу. В случае различных меток пикселей энергия зависит от цвета соседних пикселей и обратно пропорциональна их схожести. В случае одинаковых меток энергия равна нулю.

$$E_b(p, q, l_p, l_q) = \begin{cases} \exp(-\|I(p) - I(q)\|), & l_p \neq l_q \\ 0, & l_p = l_q \end{cases}$$

Унарные слагаемые отражают вероятности пикселей быть частью объекта или частью объектов переднего плана. Известна только одна чистая ячейка, соответственно можно считать, что чем ближе пиксель по цвету к соответствующему пикселю чистой ячейки, тем больше у него шансов быть частью объекта. При метке «объект переднего плана» значение унарного слагаемого является просто некоторой константой.

$$E_u(p, l) = \begin{cases} \exp(-\lambda * \|I(p) - I_{best}(p)\|), & l = facade \\ \exp(-\lambda * C), & l = foreground \end{cases}$$

где I_{best} – абсолютно чистая ячейка, а C – экспериментально подобранная константа.

Лямбда остается единственным параметром системы, который показывает важность бинарных слагаемых по сравнению с унарными. Изменяя только его, можно добиться приемлемого результата. Единого значения лямбда для всех случаев не существует. Это обусловлено различиями в освещенности ячеек и характере текстуры.

Для восстановления текстуры в найденных областях в них попиксельно копируется чистая ячейка. Для компенсации различий в освещенности ячеек и неточности нахождения решеток, применяется метод вставки части одного изображения в другое, основанный на построении и решении уравнения Пуассона с краевыми условиями Дирихле. Метод учитывает перепады цвета во вставляемой части и краевые условия векторного поля градиентов из целевого изображения.

В пятом разделе проводится тестирование и сравнение разработанных алгоритмов. Для тестирования алгоритма поиска чистой ячейки всем ячейкам была назначена одна из трех меток: чистая, слабо загороженная, сильно загороженная. Чистые ячейки выбирались в 90% случаев, а в оставшихся 10% выбирались слабо загороженные ячейки.

В качестве системы для сравнения алгоритмов сегментации и восстановления был выбран Adobe Photoshop CS5. В нем реализовано множество эффективных инструментов выделения и восстановления изображений, в частности, один из наиболее мощных на сегодняшний день алгоритмов восстановления изображения PatchMatch. Проведенное тестирование показало (рис. 10) прирост производительности в 1.56 раза.

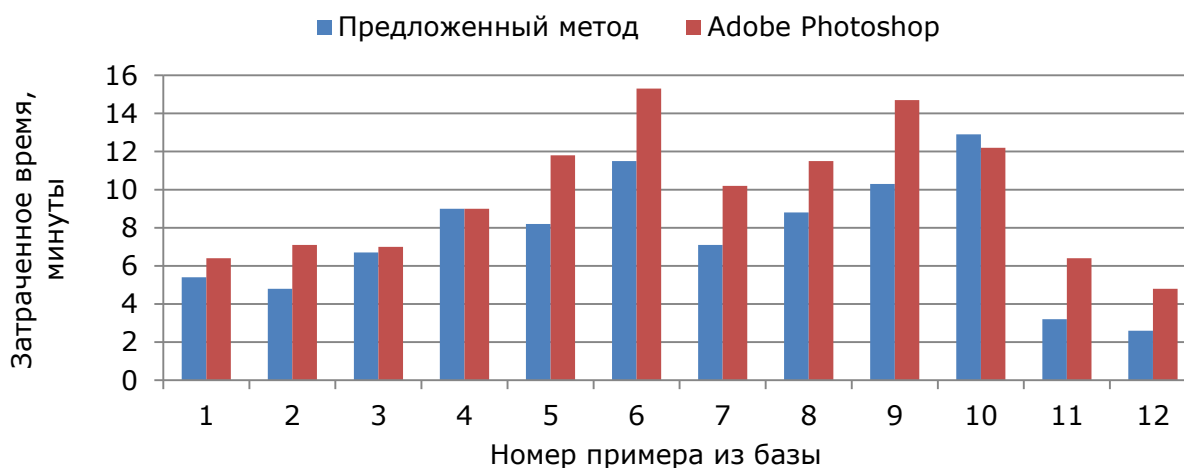


Рис. 10. Результаты сравнения предложенного алгоритма восстановления текстуры объекта с учетом его структуры с системой Adobe Photoshop.

В пятой главе описываются детали реализации всей программной системы и приводятся результаты сравнения с аналогами.

В первом разделе дается описание организации разработанной модульной программной системы. Каждый модуль оформлен в виде динамически подключаемой библиотеки. Основные структуры данных заданы в общем модуле, подключаемом статически. Обмен между модулями происходит через оперативную память и файлы.

Во втором разделе описываются все модули системы. Главный модуль отвечает за сериализацию проекта в формате XML и обеспечивает взаимодействие между модулями. Проект создается для создания трехмерной модели одного объекта.

Модуль подготовки текстур позволяет пользователю загрузить фотографии и получить из них ректифицированные текстуры. Точность ректификации можно проверить с помощью сетки из горизонтальных и вертикальных линий.

Модуль сшивки текстур позволяет сшить все текстуры в единое полотно, точно позиционируя текстуры между собой, корректируя разницу в освещенности текстур и делая незаметный шов при сшивке.

Модуль задания регулярной структуры объекта и трехмерного моделирования его элементов позволяет построить набор непересекающихся решеток, определяющих структуру объекта, и смоделировать трехмерные элементы путем выбора плоскости для рисования, задания плоского объекта, применения к нему операции создания смещенной по нормали копии, клонирования с учетом структуры объекта.

Модуль редактирования текстур позволяет отсегментировать области объектов переднего плана и восстановить изображение в данных областях с учетом ранее определенной регулярной структуры объекта. В модуле реализован эффективный инструмент клонирования регионов для ручного восстановления текстуры, который автоматически уточняет выбранное пользователем смещение в пределах нескольких пикселей. Для восстановления сильно загороженных областей разработан инструмент синтеза текстуры, который автоматически находит лучшее смещение выделенного пользователем прямоугольника и клонирует его.

Модуль экспорта позволяет выбрать баланс между детальностью геометрии и текстуры модели и размером файла модели, и сохранить полученную трехмерную модель в один из популярных форматов (Collada, Microsoft DirectX). Для каждого типа решетки пользователь может выбрать одну или несколько уникальных ячеек, которыми будут заменены все остальные элементы. В результирующей текстуре будет только один экземпляр текстур уникальных элементов, а текстуры всех остальных будут дублироваться с помощью задания соответствующих текстурных координат.

Модуль визуализации позволяет просмотреть текущую трехмерную модель из любого модуля. Используется библиотека Microsoft DirectX.

В третьем разделе проводится сравнение полного цикла моделирования по фотографиям в предложенной системы и наиболее эффективной на сегодняшний день комбинации системы для трехмерного моделирования Google SketchUp 8 и редактора фотографий Adobe Photoshop CS5. Замерялось время моделирования (рис. 11) для достижения одинаково высокого визуального качества на 10 отобранных наборах данных.

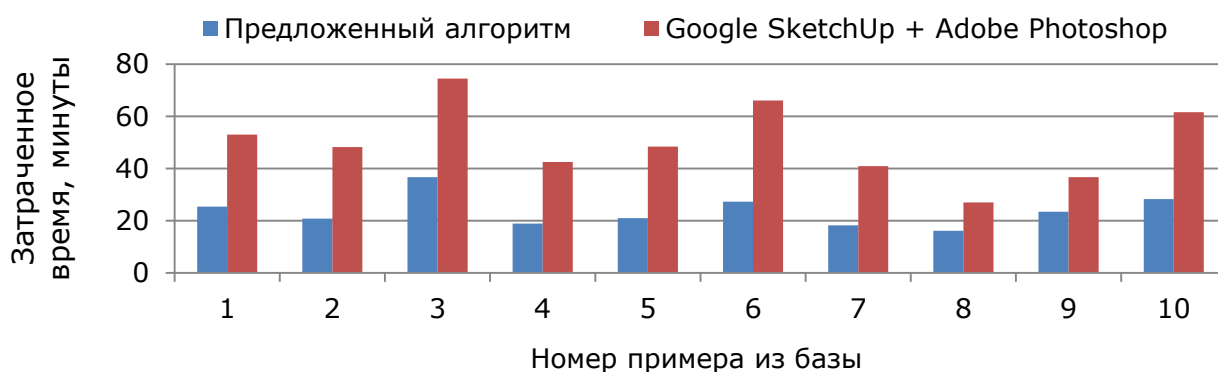


Рис. 11. Результаты сравнения предложенной программной системы.

Основные результаты работы

В результате работы разработаны новые алгоритмы и программная система, позволяющие по фотографиям при взаимодействии с пользователем построить трехмерные модели объектов с регулярной структурой, в частности, зданий за меньшее время по сравнению с существующими аналогами. В том числе:

1. Предложена новая модель регулярной структуры объекта и разработан новый автоматический алгоритм ее вычисления по изображению объекта. Алгоритм позволяет верно определять регулярную структуру большего числа объектов, чем современные алгоритмы.
2. Разработан новый алгоритм восстановления текстуры объекта, учитывающий его структуру. Выделение областей посторонних объектов регулируется единственным параметром. За счет автоматического восстановления текстуры требуется минимальное взаимодействие с пользователем по сравнению с аналогами.
3. Разработан новый алгоритм ректификации текстур и построения трехмерной модели части объекта, видимой на одной фотографии. Автоматизированный алгоритм требует меньшего объема взаимодействий с пользователем по сравнению с существующими ручными и автоматизированными средствами.

Список публикаций

1. *А. Якубенко, В. Кононов, И. Мизин, В. Конушин, А. Конушин.* Восстановление структуры и текстуры фасадов городских зданий. // «Программирование», 2011, №5, стр. 61-75.
2. *А. Якубенко, И. Мизин, А. Конушин.* Поиск регулярных решеток на текстуре фасадов зданий. // Программные продукты и системы, 2010, № 4, стр. 208 - 213.
3. *А. Якубенко.* Построение трехмерных моделей зданий по фотографиям для систем виртуальной реальности. // Программные системы и инструменты, 2012.
4. *A. Yakubenko, I. Mizin, A. Konushin.* Automatic Extraction of Regular Grids from Rectified Facade Image. // в трудах конференции Графикон 2010, стр. 100-106.
5. *В. Кононов, В. Конушин, А. Якубенко, А. Конушин.* Очищение текстур фасадов зданий с использованием их структуры // в трудах конференции Графикон 2009, стр. 283-286.
6. *O. Barinova, V. Konushin, A. Yakubenko, K. Lee, H. Lim, A. Konushin.* Fast Automatic Single-View 3-d Reconstruction of Urban Scenes. // In Proc. of European Conference on Computer Vision, 2008 (II), pp. 100-113.
7. *О. Баринаова, В. Конушин, А. Якубенко, А. Конушин.* Быстрый метод семантической сегментации изображений для автоматической трехмерной реконструкции городских сцен по одной фотографии. // в трудах конференции Интеллектуализация обработки информации 2008, стр. 22-24.
8. *О. Баринаова, В. Конушин, А. Соболев, А. Кузьмишкина, А. Якубенко, А. Конушин.* Быстрая автоматическая трехмерная реконструкция городских сцен по одному изображению. // в трудах конференции Графикон 2008, стр. 234-242.
9. *И. Мизин, А. Якубенко.* Автоматизированный поиск повторяющихся элементов на ректифицированных изображениях фасадов городских зданий // тезисы конференции «Ломоносов» 2009, стр. 54.
10. *А. Богданов, А. Якубенко.* Трехмерная реконструкция городских зданий по одному изображению на основе метода автоматической ректификации // тезисы конференции «Ломоносов» 2008, стр. 55-56.
11. *А. Якубенко, А. Конушин.* Трехмерная реконструкция городских зданий по изображениям // тезисы конференции «Ломоносов» 2007, стр. 49.