

Создание единой текстурной карты для объемной модели, полученной с помощью трехмерного сканера

Бобков Валерий¹, Кудряшов Алексей²
bobkov@dvo.ru | kudryashovA@dvo.ru

^{1,2} Институт автоматизации и процессов управления ДВО РАН, Владивосток, Россия;

В данной работе предложен метод генерации единой текстурной карты для компьютерной трехмерной модели, построенной по данным, полученным с разных точек наблюдения. В качестве источника для генерации текстуры служат фотоизображения самой модели. Большая часть существующих методов построения текстурной карты обладает очень высокой вычислительной сложностью и не всегда позволяет избежать заметных швов. Предложенный в данной работе подход предлагает решение этих проблем. Он основан на слиянии всех доступных текстур с учетом веса каждого отдельного пикселя в каждом треугольнике, учитывающий перекрытие и наблюдаемость этих треугольников. Для уменьшения времени работы предложены: алгоритм индексации треугольников в виде текстурной карты, который позволяет избежать полного перебора треугольников, алгоритм для быстрого расчета весов точек треугольника, а также используются многоядерные вычисления.

Ключевые слова: текстурирование, текстурная карта, воксельное пространство, трехмерная реконструкция, триангуляционная оболочка.

Creating a single texture map for the 3D model obtained by 3D scanner

Bobkov V.¹, Kudryashov A.²
bobkov@dvo.ru | kudryashovA@dvo.ru

^{1,2} Institute of Automation and Control Processes, Far Eastern Branch of RAS, Vladivostok, Russia

In this paper, we propose a method for generating a single texture map for 3D model reconstructed by 3D scanner. Simple image projection results in visible seams on the surface of the scans, and all the analogues of the current method considered are highly computationally complex. The proposed method of texturing offers a solution to these problems. The texturing algorithm is based on the blending of all available textures, taking into account the weight of each pixel in each triangle, as well the overlap and observability of these triangles. The algorithmic implementation of the method is optimized to improve computational performance of the method and the quality of the generated textured triangulation model.

Keywords: texturing, texture map, voxel space, 3d reconstruction, mesh.

1. Введение

Для создания реалистичных моделей объектов реального мира часто используются трехмерные сканеры, использующие стереокамеры. Они позволяют получить набор триангуляционных сеток (или карт высот), которые определяют форму объекта. Однако такие данные обладают большой избыточностью, поскольку сканирование ведется с большими перекрытиями. Поэтому существует задача объединения этих сеток. В данной работе используется воксельный подход [3]. Наряду с задачей объединения этих триангуляционных сеток, подразумевающей построение единой связной оболочки без дублирования участков поверхности, стоят задачи зашивки дыр и построение единой текстуры. Зашивка дыр происходит на этапе заполнения неявной функции в воксельном пространстве [4,6].

Существует довольно много подходов для текстурирования реконструированных объектов, однако для каждого метода реконструкции с его ограничениями необходимо подбирать свой метод создания текстуры. В [1, 2, 5] предложены несколько подходов, которые легли в основу предложенного метода.

Предлагаемый в настоящей статье метод построения единой целостной текстурной карты трехмерной модели основывается на более ранних работах авторов [7,8]. Работа выполнялась в контексте решения авторами более общей задачи по созданию трехмерных сцен на последовательностях изображений. Вклад авторов состоит в разработке оригинального алгоритма, в котором: а) производится построение гибридной весовой функции для слияния текстур, учитывающий следующие факторы: угол

наблюдения, тень/пересвет, близость к краю скана; б) обеспечивается оптимизация вычислений за счет эффективной структурной организации данных; в) достигается высокая скорость обработки данных за счет реализации параллельных вычислений на многоядерных процессорах.

2. Описание метода

Исходными данными являются трехмерная модель и набор ее фотоизображений с известной внешней и внутренней калибровкой камер. Построение самой модели выполняется модифицированным воксельным методом с зашивкой дыр, также разработанным авторами, но может выполняться и любым другим.

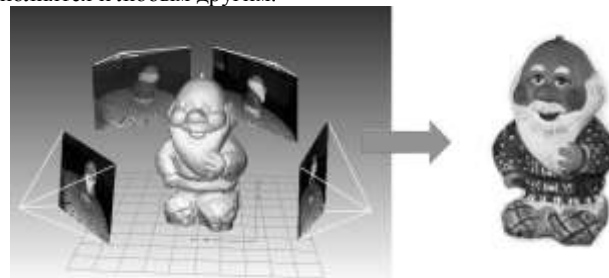


Рис. 1. Построение единой текстурной карты трехмерной модели по набору изображений, полученных с разных видов.

2.1 Предварительная фильтрация

Первая проверка – это проверка на наблюдаемость. Каждый треугольник построенной трехмерной модели проецируется на каждый из первоначальных видов. Для

каждого из них рассчитывается угол наклона от нормали треугольника к линии проходящей через центр каждой камеры. Очевидно, что если этот угол больше 90 градусов, то треугольник невидим и его можно отбросить. Для треугольников, у которых этот угол близок к 90 градусам, велика вероятность ошибки, но на случай, если нет другого более подходящего вида, эта проекция будет учтена. Вторая проверка – это проверка на видимость, то есть на то, что треугольник не перекрыт другими. В общем случае потребовался бы перебор всех треугольников модели, что заняло бы значительное время. Для оптимизации предлагается воспользоваться уже построенной индексной картой для первоначальной модели [7]. В случае использования модели, построенной другим методом объединения, такую индексную структуру надо дополнительно построить. Для всех углов треугольника объединённой модели получим значения индексов треугольников исходных сканов до объединения. Если расстояние между этими объектами меньше порогового значения, значит не существует объектов, которые бы загоразивали рассматриваемый треугольник. Особняком стоят треугольники, полученные в результате зашивки дыр – у них нет индексов на первоначальной индексной карте, поэтому их на видимость придется проверять либо с помощью полного перебора, либо, не использовать этот критерий, а использовать только критерий наблюдаемости. Логично, что для текстуры каждого треугольника нужно выбрать видимый участок, снятый с минимальным углом. Однако, на стыках текстур, полученных с разных видов, появляются заметные границы. Решению этой задачи посвящен следующий раздел статьи.

2.2 Попиксельное смешивание

Основной проблемой при нанесении текстуры на трехмерную модель с помощью исходных фотоизображений является различное освещение на разных видах. На рис. 2 видно, как один и тот же участок поверхности модели выглядит по-разному на разных видах.



Рис. 2. Влияние освещения на текстуру.

Решить эту задачу можно путем смешивания значений цвета для каждой точки текстуры с разных видов. Однако равномерное смешивание текстур приводит к не самым идеальным результатам. Дело в том, что виды не равнозначны, на некоторых из них могут присутствовать сильные тени или пересветы, на некоторых, нужный нам треугольник, снят под очень малым углом, или находится на границе скана, где обычно могут возникать ошибки сканирования. Поэтому было предложено использовать гибридный вес для учета всех предложенных критериев выбора текстуры по следующей формуле:

$$W = W_{осв}(W_{уг} + W_{гр}), \quad (1)$$

где $W_{осв}$ – вес по освещенности, функция имеет трапециевидальную форму, принимает значение 0 – для теней (в которых яркость пикселя равна 0), затем линейно возрастает до 1 и также линейно падает до 0 для пересветов (в которых яркость пикселя равна 255), $w_{уг}$ – вес по углу наблюдения, линейно возрастает от 0 до 1, в зависимости от угла между вектором нормали к треугольнику и линией наблюдения, проходящей через камеру и центр треугольника, $w_{гр}$ – вес по расстоянию, линейно возрастает от 0 до 1, в зависимости от расстояния от центра треугольника до края скана.

Первые два параметра не вызывают трудностей с расчетом, то расчёт расстояния от каждого треугольника до границы скана является крайне ресурсоемким, поэтому была предложена следующая оптимизация. Уже существует построенная структура индексов для всех треугольников, которая в двухмерном виде содержит индексы треугольников и нули там, где треугольников нет. Необходимо, для каждого ненулевого значения вычислить расстояние до ближайшего нуля. Поскольку мы используем найденные значения только в качестве веса, то нет необходимости в вычислении абсолютных значений в 3d, достаточно вычисления дистанции на плоскости изображения. Однако, прямой перебор всех ненулевых значений и поиск ближайшего нулевого в заданном радиусе будет занимать продолжительное время. Поэтому была предложена еще одна оптимизация. Каждому ненулевому значению индекса присваиваем максимальный вес, затем проверяем есть ли рядом нулевое значение, если есть, добавляем в список и присваиваем значение 1. Затем будем производить поиск значений равных максимальному весу только в соседних ячейках элементов из этого списка. Такую итерацию производим необходимое число раз. Данная оптимизация позволяет сократить время вычисления веса в 6-8 раз по сравнению с полным перебором.

На рис. 3б показана построенная карта весов по дистанции от края скана для одного вида модели «Змея» (рис.3а), а также гибридная карта весов с учетом всех весов: по освещению, углу и расстоянию до края скана (рис. 3в), рассчитанная по формуле (1).



Рис. 3. а) Изображение модели «Змея», б) карта весов по дистанции от края, в) гибридная карта с учетом всех весов.

Рис. 3. а) Изображение модели «Змея», б) карта весов по дистанции от края, в) гибридная карта с учетом всех весов.

Рассчитав вес для каждого пикселя будем строить финальную текстуру. Для каждого пикселя каждого треугольника вычисляем цвет по формуле (2):

$$C = \sum_{i=0}^n W_i c_i, \quad (2)$$

где n – число видов, c_i – исходный цвет с i -го вида, вес W_i должен быть нормирован так, чтобы сумма всех весов для всех видов была равна 1.

2.3 Создание компактной текстурной карты

Заполненные цветом треугольники проецируются на пустые изображения, аналогичные по параметрам исходным видам. Очевидно, что полученная таким образом текстура будет занимать площадь, равную площади всех исходных изображений, что крайне избыточно. Для построения компактной текстуры будем использовать следующий алгоритм. Спроецированные треугольники будут образовывать цветные участки, которые с помощью волнового алгоритма выделяются и заносятся каждый в отдельный массив (рис. 4).

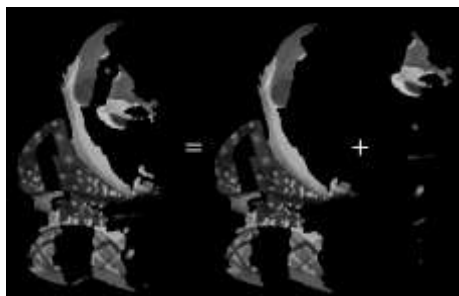


Рис. 4. Сегментация текстурных участков.

Туда же записываются индексы соответствующих треугольников. Теперь необходимо эти участки расположить максимально компактным образом. Представим эти участки в виде фигурок игры «Тетрис», только более сложной формы. Для скорости уменьшим их размер в несколько раз и представим в виде однобитовой карты (рис. 5).

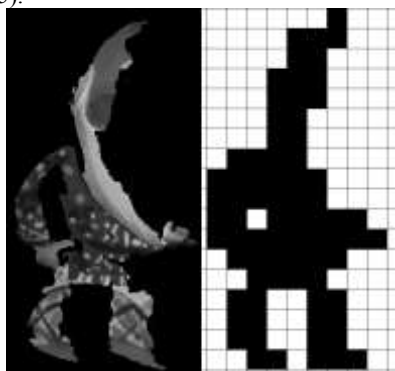


Рис. 5. Представление текстурного участка в виде однобитовой карты.

В первую очередь располагаем на финальной текстуре участки наибольшего размера, а в образовавшиеся промежутки укладываем все остальные. При перемещении необходимо пересчитать текстурные координаты.

3. Результаты

На следующих рисунках представлены несколько реконструированных моделей. Показаны несколько исходных изображений, сама модель и ее текстурная карта. В таблице 1 приведены параметры модели и время работы предложенного алгоритма. Там же, для сравнения, приведено время работы алгоритма UVAtlas [9], который также позволяет создать текстурную карту.

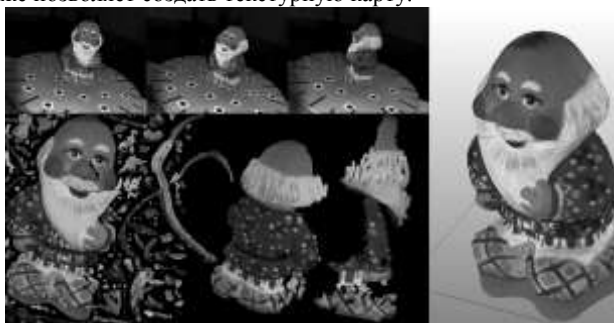


Рис. 6. Модель «Дед».

Отличие части текстурного участка от рис. 5 объясняется тем, что при создании компактной текстурной карты, часть треугольников можно перепроецировать на участки с большим количеством треугольников, для создания текстурных участков большей площади.



Рис. 7. Модель «Чайник».

Таблица 1. Сводная таблица результатов

Параметры \ Модель	Дед	Змея	Чайник
Треугольников в исходной модели, млн.	1,8	2,7	11,8
Количество видов	8	10	22
Время реконструкции, сек.	18	37	184
Время создания текстуры, сек.	6	12	99
Треугольников в финальной модели, млн.	0,3	0,6	4,8
Время работы UVAtlas, сек	84	213	>3800

В качестве недостатков метода стоит отметить то, что в областях, ненаблюдаемых ни на одном скане и в прилегающих к ним, могут создаваться артефакты. Также, помимо больших текстурных участков создается большое количество мелких, которыми закрашивается всего один или два треугольника. Дальнейшая работа будет направлена на решение указанных недостатков.

4. Заключение

В работе представлен метод создания единой текстурной карты для триангуляционной трехмерной компьютерной модели, по набору фотонизображения. Тестирование на реальных данных показало, что предложенные алгоритмические решения и структура данных с использованием октодеревя, индексации треугольников, вычисления гибридной весовой функции для каждой точки текстуры существенно повысили качество визуализации и производительность метода в сравнении с аналогами.

5. Благодарности

Работа выполнена при частичной финансовой поддержке РФФИ (грант № 15-07-00341, № 16-07-00350), Программы «Дальний Восток» (проект 15-I-4-011 о) и Программы Президиума РАН № I.33П «Фундаментальные проблемы математического моделирования».

6. Список литературы

- [1] Alshawabkeh Y., Naala N. Automatic multi-image phototexturing of complex 3D scenes. XVIII CIPA Int. Symposium, Torino. 2005. pp. 68-73.
- [2] Baumberg A. Blending Images for Texturing 3D Models. British Machine Vision Conference - BMVC, 2002. pp. 112-126.
- [3] Brian Curless, Marc Levoy. A Volumetric Method for Building Complex Models from Range Images. . In Proceedings of SIGGRAPH'96. 1996. P. 303-312

- [4] James Davis, Stephen R. Marschner, Matt Garr, Marc Levoy. Filling holes in complex surfaces using volumetric diffusion // First International Symposium on 3D Data Processing, Visualization, and Transmission, Padua, Italy. 2005. pp. 428–438.
- [5] R. Gal, Y. Wexler, E. Ofek, H. Hoppe, D. Cohen-Or. Seamless montage for texturing models Computer Graphics Forum 29 (2). 2009. pp. 479-486.
- [6] M. Kazhdan and H. Hoppe. Screened Poisson Surface Reconstruction. Transactions on Graphics, June 2013, Vol. 32, No. 3. pp. 29-42.
- [7] Бобков В.А., Кудряшов А.П. Воксельный метод построения триангуляционной поверхности по множеству видов // Информатика и системы управления, №2. 2012. С. 31-38.
- [8] Кудряшов А.П., Черкашин А.С. Построение единой триангуляционной поверхности по набору видов с зашивкой дыр // Информатика и системы управления. №1. 2015. С.36-40.
- [9] <https://uvatlas.codeplex.com/>

Об авторах

Бобков Валерий Александрович, д.т.н., заведующий лабораторией машинной графики института автоматки и процессов управления ДВО РАН.

Кудряшов Алексей Павлович, к.т.н., младший научный сотрудник лаборатории машинной графики института автоматки и процессов управления ДВО РАН.