

УДК 519.677

© 2012 г. **В.А. Бобков**, д-р техн. наук,  
**А.П. Кудряшов**, канд. техн. наук  
(Институт автоматизации и процессов управления ДВО РАН, Владивосток)

## **ВОКСЕЛЬНЫЙ МЕТОД ПОСТРОЕНИЯ ТРИАНГУЛЯЦИОННОЙ ПОВЕРХНОСТИ ПО МНОЖЕСТВУ ВИДОВ\***

Описан алгоритм построения триангуляционной 3D модели сцены по множественному набору пространственных данных, основанный на воксельном подходе. Предложена оптимизированная структура исходных данных, обеспечивающая повышение производительности алгоритма. Реализована вычислительная схема с организацией параллельных вычислений на графических процессорах средствами языка CUDA.

**Ключевые слова:** воксельное пространство, трехмерная реконструкция, триангуляционная оболочка, «марширующие кубики», CUDA.

### **Введение**

Один из подходов к созданию 3D-компьютерных моделей реальных объектов и сцен основывается на использовании дальностных данных, получаемых с помощью лазерных дальномерных приборов или методами компьютерного зрения по изображениям. В литературе на такие данные ссылаются как на дальностные изображения (range images), карты глубин (depth maps). В отличие от обычных изображений с каждым пикселем range image/depth map связывается расстояние до видимой точки объекта в пространстве сцены. Эти пространственные данные могут быть представлены в виде облака точек в трехмерном пространстве или в виде триангуляционной поверхности. Технологии получения дальностных данных и генерации цифровых пространственных моделей на практике применяются в робототехнике, геоинформатике, производстве, медицине и в других приложениях. Нетривиальной задачей при построении трехмерных моделей по последовательности дальностных изображений, представленных в виде отдельных триангуляционных сеток (оболочек), является задача объединения этих триангуляционных сеток. Ее решение подразумевает построение единой связной оболочки, отвечающей исходным данным, без дублирования участков поверхности, видимых с нескольких видов. Важным требованием к алгоритмическому решению задачи является высокая скорость обработки данных, поскольку для насыщенных

\* Работа выполнена при финансовой поддержке РФФИ (проект №11-07-00088-а) и Программы №15 Президиума РАН.

сцен со сложными объектами может потребоваться много видов с большим количеством описывающих треугольников (порядка  $10^5$ ,  $10^6$ ).

Естественный путь к решению указанной задачи состоит в сопряжении исходных сеток с устранением избыточного триангуляционного описания посредством совместного анализа данных в 3D-пространстве сцены и в 2D-пространствах изображений видов. Подобная схема решения была реализована в [1]. Однако она, как отмечают авторы, полностью не исключает дублирования. Более интересным представляется так называемый воксельный подход, основанный на использовании воксельной структуры пространства сцены. Эффективная реализация такого подхода была предложена в работе [2]. Основным преимуществом предложенного решения является сведение исходной задачи к известной задаче построения изоповерхности в скалярном поле. Значительным недостатком данного метода можно считать его высокую ресурсоемкость, из-за чего реконструкция даже небольшой сцены по нескольким десяткам изображений занимает много часов [4]. Однако в последнее время, с появлением технологий, использующих для вычислений графический процессор, – таких как CUDA (*Compute Unified Device Architecture*) от компании NVIDIA, стало возможным увеличить скорость расчета ресурсоемких алгоритмов в десятки и сотни раз. В работе [5] можно ознакомиться и с систематизированным обзором некоторых других методов реконструкции поверхности по плотному набору дальностных данных.

Представленный в настоящей статье метод построения единой сеточной модели основывается на подходе, предложенном в [2]. Работа выполнялась в контексте решения авторами более общей задачи по созданию трехмерных моделей на множественных последовательностях изображений с построением карт глубин [3]. Вклад авторов состоит в предлагаемой оптимизации вычислений за счет эффективной структурной организации данных и реализация параллельных вычислений на графических процессорах, что в целом обеспечивает более высокую скорость обработки данных.

### Описание метода

Задача формулируется следующим образом. Имеются  $n$  видов (ракурсов сцены), для каждого из которых на предварительном этапе была построена видимая поверхность в виде пространственной триангуляционной сетки. Триангуляционная сетка строилась по 2D-триангуляции на регулярной пиксельной решетке изображения с использованием карты глубин. Заметим, что карта глубин вычисляется для точек, видимых на данном виде. Цель – построить единую текстурированную триангуляционную оболочку, которая и будет трехмерной триангуляционной моделью сцены.

Метод работает в воксельном пространстве сцены. Используется непрерывная неявная функция  $D(V_i)$ , представленная значениями в узлах воксельной решетки. Функция конструируется как взвешенная сумма получаемых для  $n$  видов расстояний  $d_1(V_i), d_2(V_i), \dots, d_n(V_i)$  от точки  $V_i$  до ближайшей поверхности (рис. 1). Расстояние  $d_j$  берется на луче, направленном из центра проекций вида  $j$  в точку  $V_i$ , и является величиной со знаком (положительное для точек, находящихся пе-

ред поверхностью, и отрицательное – для точек за поверхностью).

В качестве весовой функции  $W(\mathbf{V}_i)$  берется скалярное произведение вектора нормали к поверхности и вектора направления на точку наблюдения. Такой выбор исходит из предположения, что при корреляционном подходе к определению расстояния до поверхности степень неопределенности для наклонных поверхностей выше, чем для поверхностей, наблюдаемых под прямым углом.

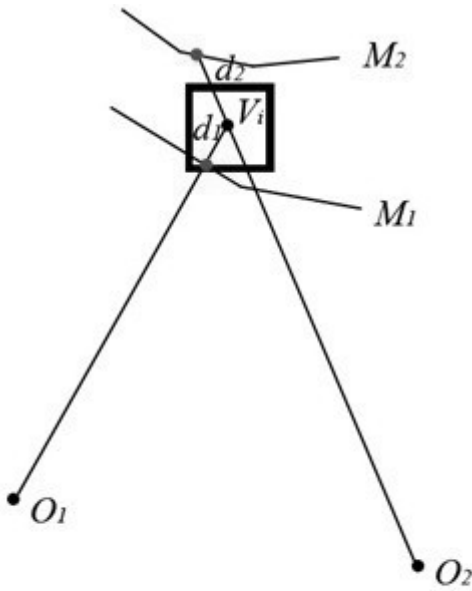


Рис. 1. Построение скалярной функции.

Для вычисления  $D$  и  $W$  используются следующие инкрементальные правила:

$$D_{j+1}(\mathbf{V}_i) = [W_j(\mathbf{V}_i)D_j(\mathbf{V}_i) + w_{j+1}(\mathbf{V}_i)d_{j+1}(\mathbf{V}_i)] / [W_j(\mathbf{V}_i) + w_{j+1}(\mathbf{V}_i)],$$

$$W_{j+1}(\mathbf{V}_i) = W_j(\mathbf{V}_i) + w_{j+1}(\mathbf{V}_i),$$

где  $i$  – номер вокселя;  $j$  – номер вида.

Весовая функция  $W(\mathbf{V}_i)$  – константа для всех вокселей до видимой поверхности (для данного вида), а для вокселей за поверхностью линейно убывает до нуля в пределах  $\epsilon$ -окрестности. Такой выбор области определения весовой функции направлен на предотвращение возникновения ложных поверхностей.

В результате последовательной обработки всех видов каждому вокселю присваивается значение интегральной функции расстояния  $D(\mathbf{V}_i)$  (величина со знаком) и значение интегральной весовой функции  $W(\mathbf{V}_i)$ . Тогда построение изоповерхности  $D(\mathbf{V}_i) = 0$  и является решением нашей задачи. Преимущество сведения исходной задачи к такой постановке – то, что построение изоповерхности скалярного поля, заданного на воксельной решетке, можно выполнить с помощью известного алгоритма марширующих кубиков [4].

### Алгоритмическая реализация

Предполагается, что для каждого из видов определены параметры камеры. Вначале создается воксельное пространство сцены. Для определения границ сцены просматриваются все узлы исходных сеток и находятся максимальные и минимальные значения по всем координатам. Размерность воксельного пространства задается пользователем. Она определяет детализацию результирующей трехмерной сцены.

Заметим, что наиболее ресурсоемкой частью представленного выше метода является определение расстояний от вокселей до ближайшей поверхности при формировании скалярного поля.

Вычислительная трудоемкость возникает из-за необходимости перебора всех треугольников, с вычислением соответствующих расстояний. Чтобы избежать избыточных вычислений, предлагается механизм с предварительной индексацией всех треугольников сетки каждого вида (рис. 2). Для этого используется само изображение этого вида.

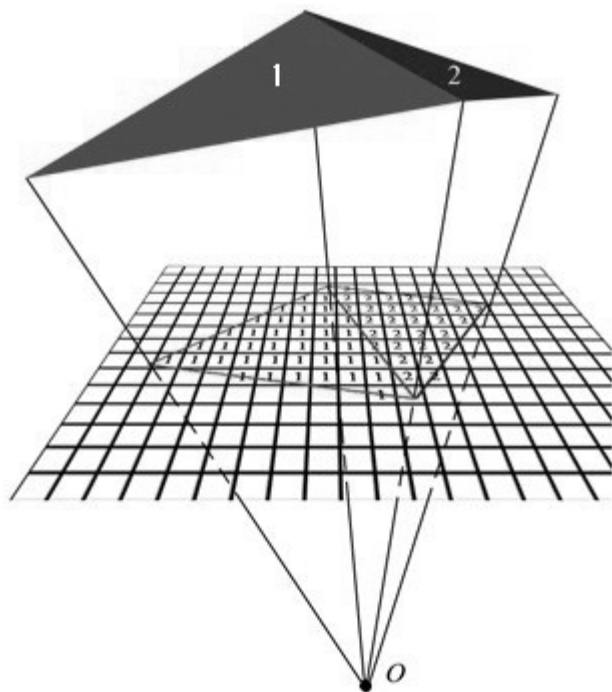


Рис. 2. Индексация пикселей изображения.

Построим проекционную матрицу каждого  $i$ -го вида по формуле:

$$H_i = \begin{pmatrix} c_{11}^i f & c_{12}^i f & c_{13}^i \\ c_{21}^i f & c_{22}^i f & c_{23}^i \\ c_{31}^i f & c_{32}^i f & c_{33}^i \\ c_{41}^i f & c_{42}^i f & c_{43}^i \end{pmatrix} \quad (1)$$

где  $c_{kl}^i$  – элементы матрицы камеры  $i$ -го вида;  $f$  – фокус.

Тогда можно спроецировать каждый треугольник исходной сетки  $i$ -го вида на  $i$ -е изображение по формуле:

$$P_{tex}^i = H_i \times P, \quad (2)$$

где  $P$  – координаты точки в трехмерном пространстве. В результате каждый пиксель изображения приобретает индекс, указывающий на принадлежность к определенному треугольнику.

Такая процедура возможна, поскольку исходная сетка для каждого вида является «видимой», т.е. не содержит невидимых на этом виде граней. Все непроиндексированные пиксели приравниваются к одному значению. Использование индексов при последующей работе алгоритма позволяет избежать полного перебора по всем треугольникам каждого вида и существенно уменьшает время работы алгоритма.

При реализации метода был разработан дополнительный алгоритм с использованием альфа-канала, который необходим для отделения реконструируемого объекта от фона. Для каждого пикселя проиндексированного изображения проверяется его значение в альфа-канале исходного изображения. Если это значение равно нулю (т.е. пиксель является фоном), то этому пикселю присваивается то же значение, что и непроиндексированным пикселям, что исключает построение поверхностей, относящихся к фону.

На последнем этапе, после формирования скалярного поля с помощью алгоритма «марширующих кубиков», выполняется построение изоповерхности  $D(V_i) = 0$ , которая и является решением задачи построения общей триангуляционной сетки по множеству видов.

Работу основной части метода можно описать следующим образом:

1) чтение исходных данных из файлов (изображения, параметры камер, триангуляционные сетки);

2) цикл по видам:

индексирование всех треугольников исходной сетки;

удаление индексов треугольников фона, используя значения альфа-канала (при его наличии);

3) цикл по вокселям:

цикл по видам –

воксел проецируется на индексированное изображение.

Если проекция не попадает в область, где индекс не соответствует ни одному треугольнику, то вычисляется расстояние от воксела до соответствующего треугольника сетки  $d(V_i)$ ;

вычисляется значение весовой функции  $w(V_i)$ .

результат  $d(V_i)*w(V_i)$  добавляется в  $D(V)$ ;

4) по полученной воксельной структуре строится триангуляционная сетка методом «марширующих кубиков».

5) цикл по треугольникам полученной сетки:

треугольник проецируется на все исходные изображения. Изображение с максимальной площадью проекции используется в качестве текстуры;

б) вывод полученной триангуляционной сетки в формате DirectX.

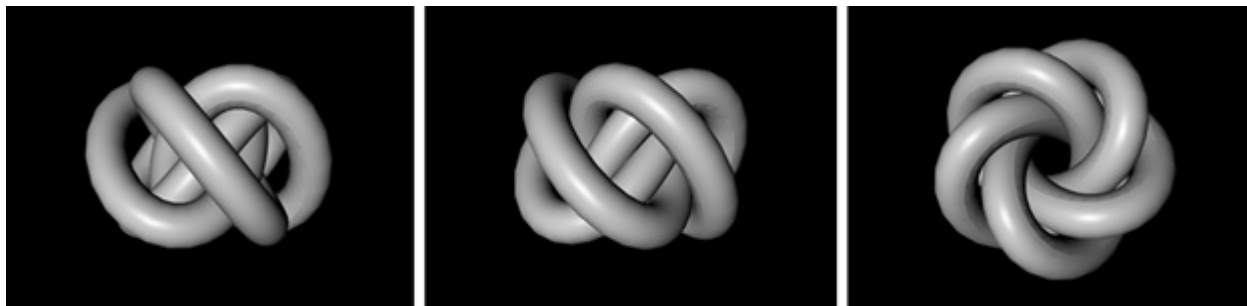
Поскольку метод работает с большими объемами данных, была реализована схема центральный процессор + графические процессоры, где наиболее ресурсоемкие вычисления выполняются на графической плате. Самая ресурсозатратная часть метода – это проход по всем вокселям сцены с вычислением значений скалярной функции  $D(V)$ . Поскольку вычисления этих значений для каждого вокселя являются независимыми, алгоритм хорошо распараллеливается с помощью технологии CUDA.

### Вычислительные эксперименты

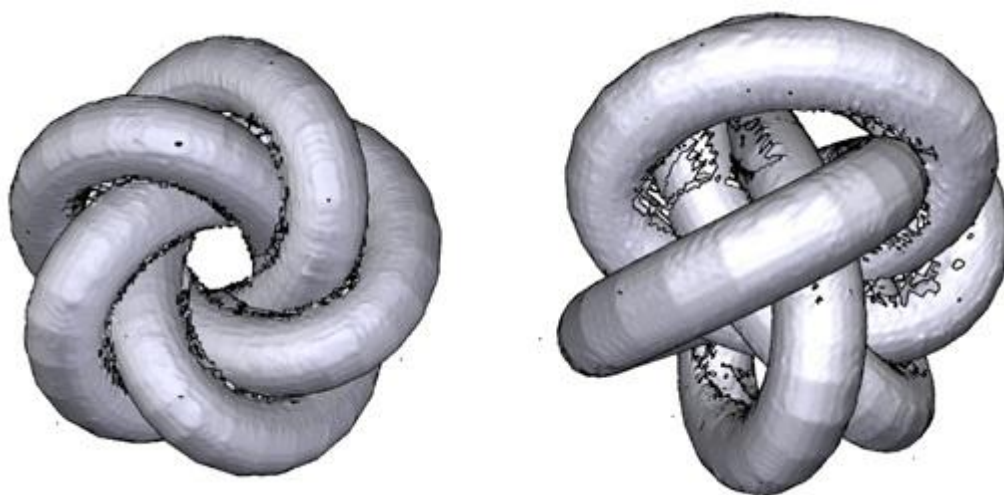
Для получения сравнительных оценок эффективности предложенного метода были проведены вычислительные эксперименты на модельных сценах 1 и 2 (рис. 3, 5) и реальной сцене 3 «Temple» (рис. 7) (<http://vision.middlebury.edu/mview/>) [5].

Параметры используемого вычислительного оборудования: процессор Intel Core Duo 2,6 ГГц, графическая плата NVidia GeForce 9600GT. Во-первых, сравнивались производительности двух алгоритмических реализаций на центральном процессоре – предложенной в настоящей работе и аналогичной описанной в [2]. Во-вторых, оценивалось ускорение вычислений предложенного алгоритма за счет реализации параллельных вычислений на графической плате. Результаты вычислительных экспериментов приведены в таблице и на рис. 4, 6, 8. Как видно из таблицы, время выполнения практически не зависит от сложности сцены (количество треугольников), что подтверждает эффективность реализованного механизма индексации. Предложенный механизм индексации при реализации алгоритма на центральном процессоре ускоряет вычисления в 12-20 раз. Использование графических процессоров дает дополнительное ускорение в 7-15 раз. Таким образом, суммарное ускорение оценивается в 200-300 раз.

В таблице указано время вычисления (сек.). Первое значение – время работы аналога алгоритма [2], второе – время работы предложенного алгоритма с индексацией треугольников, третье – время работы алгоритма с использованием CUDA.



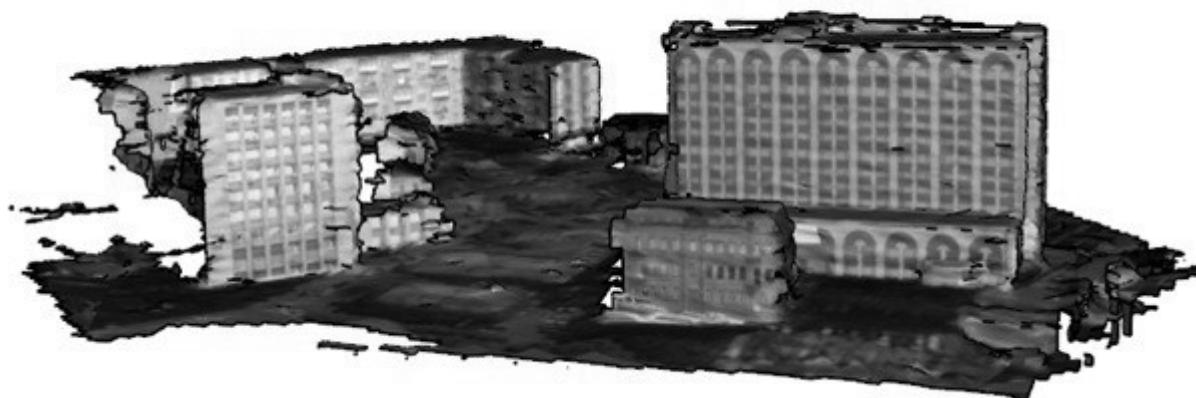
*Рис. 3.* Исходные виды (3 из 6) сцены 1.



*Рис. 4.* Реконструкция сцены 1 по шести видам.



*Рис. 5.* Исходные виды сцены 2 (3 из 11).



*Рис. 6.* Реконструкция сцены 2 по 11 видам.

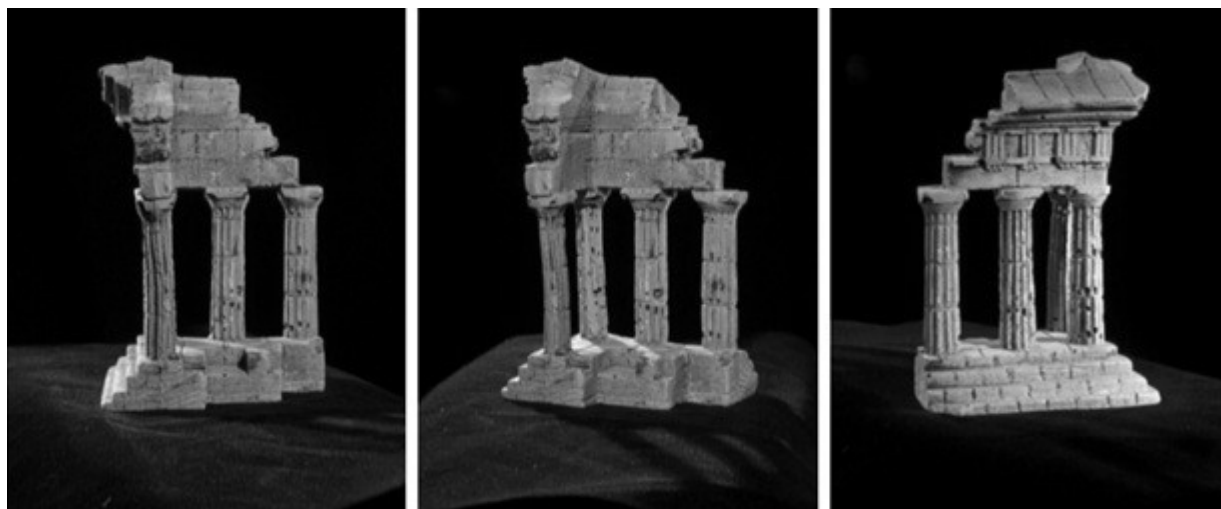


Рис. 7. Исходные виды сцены 3 «Temple».

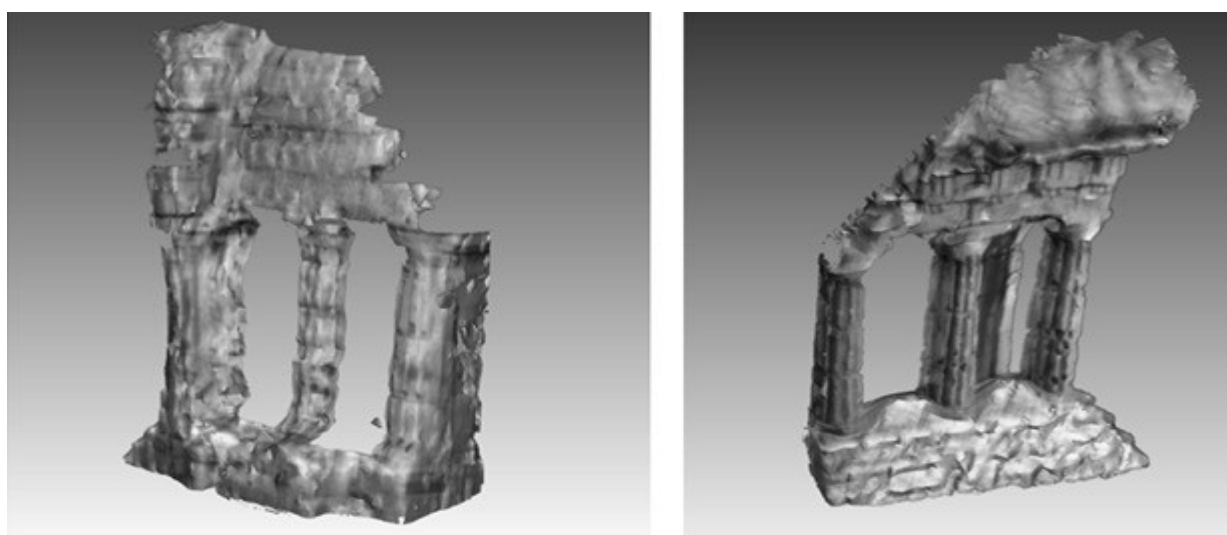


Рис. 8. Реконструкция сцены 3 по трем (слева) и пяти (справа) видам.

| Сцены                     | Размерность воксельного пространства | Кол-во треуг. | Количеств видов  |                  |                  |                   |                       |
|---------------------------|--------------------------------------|---------------|------------------|------------------|------------------|-------------------|-----------------------|
|                           |                                      |               | 1                | 2                | 3                | 6                 | 60                    |
| Модельная сцена 1         | 128 <sup>3</sup>                     | 8 тыс.        | 58;3.8;<br>0.6   | 66;4.3;<br>0.9   | 114;6.9;<br>1.2  | 145;9.3;<br>1.5   | 641;42;<br>3.4        |
|                           | 256 <sup>3</sup>                     | 8 тыс.        | 308;14.7;<br>2.8 | 413;21.2;<br>3.4 | 685;32.4;<br>4.9 | 1034;49;<br>6.2   | 4082;<br>186;<br>15.4 |
| Модельная сцена 2 «Город» | 128 <sup>3</sup>                     | 22 тыс.       | 96;6.3;<br>0.8   | 132;8.5;<br>1.1  | 158;9.4;<br>1.5  | 194;12.7;<br>2.4  | 894;59;<br>4.2        |
|                           | 256 <sup>3</sup>                     | 22 тыс.       | 434;21.4;<br>3.2 | 619;31.2;<br>3.9 | 987;49.3;<br>4.8 | 1438;61.5;<br>6.7 | 5865;225;<br>18.7     |
| Реальная сцена 3 «Temple» | 128 <sup>3</sup>                     | 16 тыс.       | 84;5.5;<br>0.7   | 114;7.5;<br>0.8  | 128;8.2;<br>1.0  | 161;10.7;<br>1.4  | 764;52;<br>3.3        |
|                           | 256 <sup>3</sup>                     | 16 тыс.       | 406;19.7;<br>2.9 | 589;28.4;<br>3.4 | 905;43.2;<br>4.2 | 1308;55.5;<br>5.9 | 5143;205;<br>13.8     |

## Заключение

В работе представлен алгоритм построения триангуляционной 3D-модели сцены по множественному набору пространственных данных, основанный на воксельном подходе. Предложенная структура данных с использованием индексации треугольников позволила существенно повысить производительность алгоритма в сравнении с аналогами. Дополнительное ускорение получено за счет организации параллельных вычислений на графических процессорах средствами языка CUDA.

## ЛИТЕРАТУРА

1. *Real-time* video-based reconstruction of urban environments / P. Mordohai, J.-M. Frahm, A. Akbarzadeh, B. Clipp, C. Engels, D. Gallup, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, H. Towles, G. Welch, R. Yang, M. Pollefeys, and D. Nister. - ISPRS Working Group V4 Workshop 3D-ARCH 2007: 3D Virtual Reconstruction and Visualization of Complex Architectures, (ETH Zurich, Switzerland). – 2007. – P. 121-131.
2. *Brian Curless and Marc Levoy*. A Volumetric Method for Building Complex Models from Range Images // Computer Graphics (SIGGRAPH '96 Proceedings). – 1996. – P. 303-312.
3. *W.E. Lorensen and H. E. Cline*. Marching cubes: A high resolution 3D surface construction algorithm // In Computer Graphics (SIGGRAPH '87 Proceedings). –1987. – Vol. 21. – P. 163-169.
4. *M. Goesele, B. Curless, S. M. Seitz*. Multi-View Stereo Revisited // Computer Vision and Pattern Recognition, IEEE Computer Society Conference on. – 2006. – Vol. 2. – P. 2402-2409.
5. *A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms* / Steve Seitz, Brian Curless, James Diebel, Daniel Scharstein, Richard Szeliski, Vision and Pattern Recognition. // IEEE Computer Society Conference on. – 2006. – Vol. 1. – P. 519-526.

Статья представлена к публикации членом редколлегии Е.А. Нурминским

*E-mail:*

Бобков В.А. – [bobkov@iacp.dvo.ru](mailto:bobkov@iacp.dvo.ru);

Кудряшов А.П. – [kudryashova@dvo.ru](mailto:kudryashova@dvo.ru).

**ИММ УрО РАН (Екатеринбург),  
ИПМ им. М.В.Келдыша РАН (Москва),  
ЮГИНФО ЮГУ (Ростов-на-Дону)  
с 10 по 16 сентября 2012 года проводят VI Всероссийскую конференцию  
"Актуальные проблемы прикладной математики и механики",  
посвященную памяти академика А.Ф. Сидорова.**

Тематика конференции:

аналитические методы в задачах газовой динамики и механики сплошной среды;  
оптимизация процессов в задачах тепломассопереноса;  
численные методы решения задач математической физики;  
проблемы построения оптимальных сеток.

Для участия в конференции необходимо до 15 июня 2012 года заполнить регистрационную форму на сайте конференции [afsid.imm.uran.ru](http://afsid.imm.uran.ru) или выслать заявку на адрес [afsid@imm.uran.ru](mailto:afsid@imm.uran.ru)

Контактные телефоны: (343)375-34-80, (343)375-34-94, факс (343)374-25-81.

Вместе с заявкой до 15 июня 2012 года следует выслать краткие тезисы докладов.