

данных использовались растровые изображения шахматной доски (типа «Ches») и мозаики, полученные при генерации одноуровневого марковского поля с морфологиями «А», «С» и «D». Оценивались значения обобщенного показателя качества при различных отношениях сигнал/шум. Результаты моделирования приведены на рис. 5.

Из рисунка видно, что обобщенный критерий оценки качества алгоритмов оконтуривания изображений, позволяет получить количественные значения качества оконтуривания изображений в зависимости от типа мозаик и соотношения сигнал/шум.

СПИСОК ЛИТЕРАТУРЫ

1. Jähne B. Digitale Bildverarbeitung. – Berlin: Heidelberg, Springer-Verlag, 2010. – 650 S. ISBN 978–3–642–04951–4.
2. Liedtke C.-E., Ender M. Wissensbasierte Bildverarbeitung. – Berlin: Springer-Verlag, 1989. – 224 S. ISBN 3–540–50641–1.
3. Буймов А.Г. К статистике пальмовских полей // Автметрия. – 1981. – № 6. – С. 13–18.
4. Буймов А.Г., Ильин С.П., Мозаичное изображение с управляемой корреляцией // Автметрия. – 1987. – № 5. – С. 30–35.
5. Лаевский В.Е. Алгоритм построения одноуровневых марковских полей // Известия Томского политехнического университета. – 2006. – Т. 309. – № 8. – С. 32–36.
6. Лаевский В.Е. Методика субоптимальной оценки работы алгоритмов получения контурного рисунка изображений // Известия Томского политехнического университета. – 2009. – Т. 314. – № 5. – С. 126–131.
7. Гуткин Л.С. Оптимизация радиоэлектронных устройств по совокупности показателей качества. – М.: Советское радио, 1975. – 224 с.

Поступила 08.10.2010 г.

УДК 681.3

АЛГОРИТМЫ И МЕТОДЫ ОБРАБОТКИ ИНФОРМАЦИИ В ЗАДАЧАХ ТРЕХМЕРНОГО СКАНИРОВАНИЯ ОБЪЕКТОВ

И.В. Цапко, С.Г. Цапко

Томский политехнический университет
E-mail: irina@aics.ru

Предложены методы обработки изображений, полученных методом трехмерного сканирования. Подробно рассмотрен алгоритм совмещения отдельных сканируемых поверхностей изделия. Разработана методика формирования модели изделия в формате стереолитографии на основе первичной информации трехмерного сканера и математических методов обработки трехмерной графики. Приведены результаты апробации алгоритма построчного сканирования, использующего z-буфер применительно к задачам очистки сканированного трехмерного изображения.

Ключевые слова:

3D-сканирование, CAD-система, обработка информации, дизайн, прототипирование.

Key words:

3D-scanning, CAD-system, information processing, design, prototyping.

Достижения научно-технического прогресса в области проектирования и изготовления новых изделий позволяют до минимума сократить временные и материальные издержки, необходимые в течение жизненного цикла изделия. При изготовлении нового изделия, как правило, создают его математическую модель в любой CAD-среде. Но часто требуется воссоздать уже существующее изделие, сделать его дубликат. Если изделие имеет простую форму, то процесс проектирования занимает немного времени и требует небольших затрат.

Все становится диаметрально противоположным в случае увеличения сложности объекта, для которого требуется создать копию. Тогда создание математической модели в CAD-системе становится практически невозможно, либо требует больших временных и материальных ресурсов.

В данном случае для формирования компьютерного представления изделия можно применить технологию трехмерного сканирования. Трехмерное сканирование – это систематический процесс определения координат точек, принадлежащих по-

верхностям сложнопрофильных физических объектов (в частности, деталей) с целью последующего получения их пространственных математических моделей, которые могут модифицироваться с помощью CAD-систем. Полученную математическую модель можно использовать в качестве входных данных для станков с ЧПУ или для системы быстрого прототипирования. Быстрое прототипирование (*Rapid Prototyping*) – это процесс послойного построения физической модели (прототипа) в соответствии с геометрией математической модели.

Устройства, с помощью которых осуществляется сканирование объектов, называют трехмерными сканерами. Эти устройства не только упрощают процесс создания трехмерных моделей, но и позволяют решать эту задачу с максимальной степенью достоверности по отношению к исходному оригиналу. Созданную трехмерную модель в дальнейшем можно использовать как шаблон для формирования параметрической модели в любой CAD-среде.

Неконтактное лазерное сканирование является новейшей технологией. Несмотря на относительно небольшой срок существования, она находит применение при решении большого спектра задач, который, по мере освоения возможностей метода, продолжает расширяться. Задачи, решаемые методом неконтактного лазерного сканирования с использованием неконтактного лазерного сканера, можно условно объединить в четыре большие области применения.

1. Техническое проектирование. Включает такие задачи как:
 - индустриальный дизайн, создание вручную трехмерных моделей и их оцифровка с последующей доработкой методами машинной графики;
 - создание трехмерных моделей имеющихся штампов, пресс-форм и других изделий сложной формы, например, уникальных, в случае необходимости их изменения, ремонта или повторного воспроизведения;
 - инженерный трехмерный анализ, измерение геометрических параметров изделий, которые не могут быть измерены стандартными методами;
 - on-line контроль качества (проверка точности изготовления).
2. Архитектура. В архитектуре трехмерное сканирование применяется для:
 - реставрации, путем создания банка электронных копий (виртуального архива) и последующего изготовления или восстановления (промышленного воспроизведения) оригиналов скульптур и других рельефных изделий (сувениров, украшений, барельефов, горельефов и т. д.);
 - создания виртуальных архитектурных музеев.
3. Медицина. В медицине основное применение такие системы нашли в хирургическом плани-

ровании, ортопедии, протезировании, пластической хирургии, косметологии.

4. Археология. В последнее время системы трехмерного сканирования широко используют в археологии для решения следующих задач:
 - виртуальное сохранение произведений искусства;
 - создание документации с трехмерными изображениями объектов;
 - виртуальная реставрация, а также копирование для последующей репликации;
 - создание виртуальных музеев.

Принцип работы трехмерного сканера заключается в следующем: пучок, излучаемый лазером I класса, направляется поворотным зеркалом на объект, отражаясь от которого непосредственно через объектив, попадает снова в аппарат, где и регистрируется встроенной цифровой камерой. Согласование включения лазера с электромеханическим приводом зеркала осуществляется в автоматическом режиме. Освещенность обеспечивается самим лазерным пучком. Это очень удобно, т. к. не требуется мер для создания дополнительного освещения. Управлять процессом сканирования можно как с помощью компьютера, так и вручную.

В связи с тем, что средства трехмерного сканирования появились сравнительно недавно, технология быстрого клонирования твердотельных изделий не отработана, и существует ряд проблем, требующих решения. К таким проблемам можно отнести:

1. Наличие шумов при трехмерном сканировании. В процессе сканирования получается скан, имеющий помимо изображения самого объекта множество микрообъектов, представляющих «мусор». Данные объекты нужно убрать. В настоящее время это делается вручную, что занимает достаточно много времени.
2. При сканировании объекта всегда имеются области, которые сканер не может охватить (например, основание детали, на котором она стоит в процессе сканирования). Следовательно, требуется выполнить повторное сканирование недостающих частей изделия. В результате получается набор сканов, из которых впоследствии нужно сформировать единую трехмерную математическую модель объекта.
3. Для изготовления пресс-формы объекта по его математической модели, созданной с помощью трехмерного сканера, требуется разработка специального программного обеспечения. В данном случае в процессе сканирования можно получить полигон точек, расположенных в пространстве и формирующих поверхность (контур) изделия. Однако для инверсии в существующих CAD-системах требуется параметрическая, либо твердотельная модель объекта. При этом математическая модель, полученная с помощью трехмерного сканера, часто имеет

разрывы поверхностей, которые не позволяют использовать ее в качестве входных данных для системы быстрого прототипирования.

Для решения поставленных задач был проведен анализ существующих методов сканирования, технологий быстрого прототипирования и на их основе разработаны алгоритмы для решения задач трехмерного сканирования и формирования математической модели, используемой в задачах быстрого прототипирования.

В процессе сканирования оператор создает несколько сканов, отдельных трехмерных снимков, охватывающих часть объекта, а затем воссоздает математическую модель поверхности объекта (рис. 1, слева). При этом оператор должен произвести очистку каждого скана от «мусора». Под «мусором» понимается множество «паразитных» трехмерных объектов, выявленных на сканах. Появление данных областей обусловлено технологией лазерного сканирования, когда отражение луча от поверхности объекта создает рассеивание, а детектор сканера фиксирует наличие микрообъектов

вокруг объекта сканирования. Для реализации задачи очистки сканов от микрообъектов можно использовать ручную, либо автоматическую обработку сканов.

Ручная обработка сканов выполняется стандартными средствами любого 3D-редактора. Однако данный процесс является очень продолжительным во времени и трудоемким. Поэтому, в данной работе, предложено использовать алгоритмы автоматической обработки сканов. Существуют ряд алгоритмов, которые можно применить для решения поставленной задачи: плавающего горизонта, Робертса, Варнока, Вейлера–Азертона, построчного сканирования.

Алгоритм плавающего горизонта используется для удаления невидимых линий трехмерного представления функций, описывающих поверхность в виде $F(x,y,z)=0$. Главная идея данного метода заключается в сведении трехмерной задачи к двумерной путем пересечения исходной поверхности последовательностью параллельных секущих плоскостей, имеющих постоянные значения координат x ,

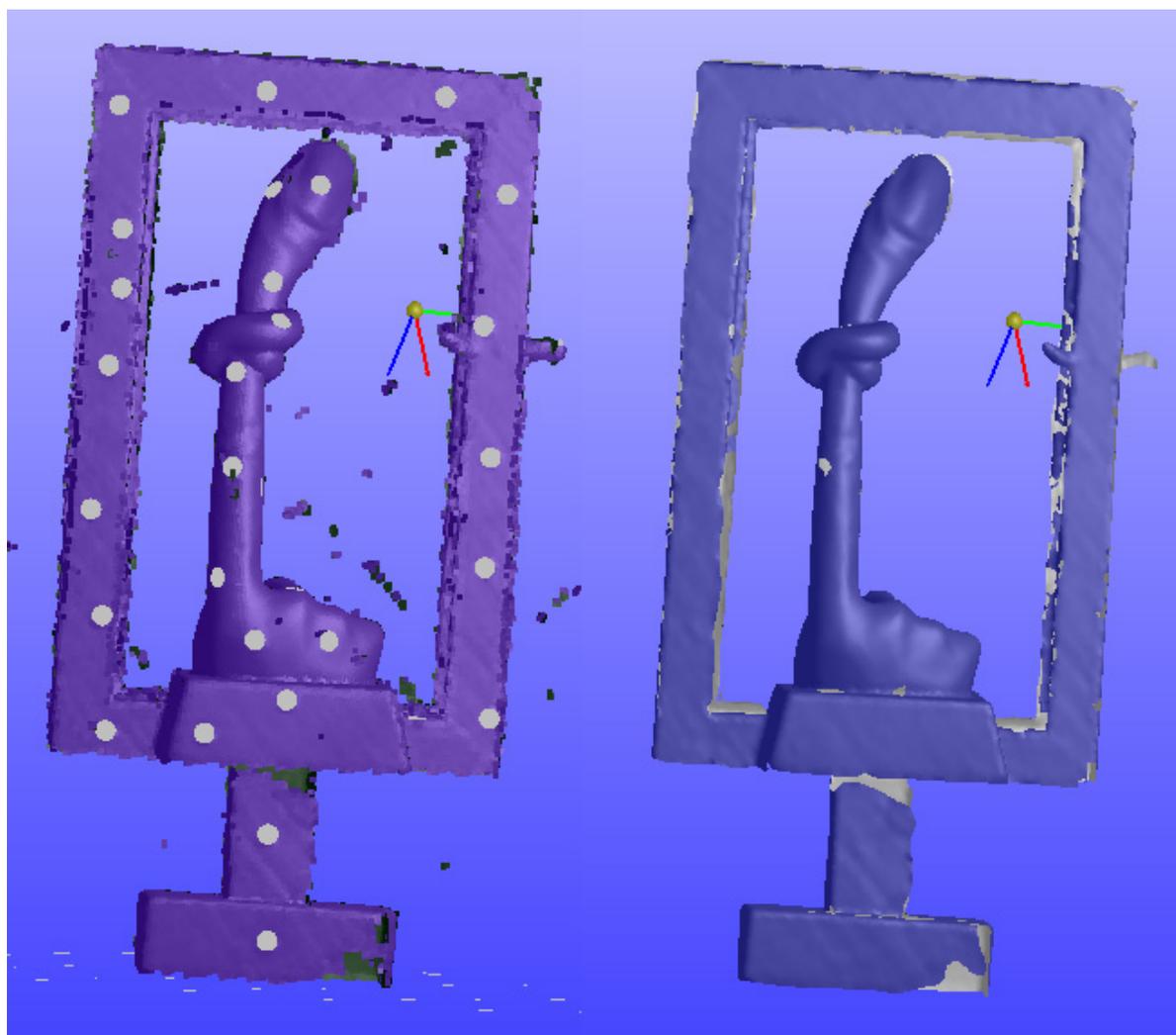


Рис. 1. Скан объекта до обработки (слева) и после обработки (справа)

у или z . Если видимость кривой меняется, то метод с такой простой интерполяцией не даст корректного результата. А это может повлечь за собой дополнительные нежелательные эффекты. В данном случае необходимо решать задачу о поиске точек пересечения сегментов текущей и предшествующей кривых.

Алгоритм Робертса удаляет из каждого тела те ребра или грани, которые экранируются самим телом. Затем каждое из видимых ребер каждого тела сравнивается с каждым из оставшихся тел для определения того, какая его часть или части, если таковые есть, экранируются этими телами. Поэтому вычислительная трудоемкость алгоритма Робертса растет теоретически как квадрат числа объектов.

В алгоритме Варнока и его вариантах делается попытка извлечь преимущество из того факта, что большие области изображения однородны. Такое свойство известно как когерентность, т. е. смежные области (пиксели) вдоль обеих осей x и y имеют тенденцию к однородности. В данном случае происходит аппроксимация картинка и края сканированных объектов отражаются с недопустимой точностью.

Алгоритм Вейлера–Азертон минимизирует количество шагов в алгоритме разбиения типа алгоритма Варнока путем разбиения окна вдоль границ многоугольника. Недостатки данного алгоритма схожи с недостатками алгоритма Варнока.

Приведенные выше алгоритмы не позволяют получить «очищенное от мусора» сканированное трехмерное изображение с заданной точностью. Авторами работы было предложено разработать программные компоненты модуля обработки сканов на базе алгоритма построчного сканирования, использующего z -буфер. Данный алгоритм является одним из универсальных алгоритмов построчного сканирования, который решает задачу удаления невидимых поверхностей. Используемое в этом алгоритме окно визуализации имеет высоту в одну сканирующую строку и ширину во весь экран. Как для буфера кадра, так и для z -буфера требуется память высотой в 1 бит, шириной в горизонтальное разрешение экрана и глубиной в зависимости от требуемой точности [4].

Концептуально этот алгоритм достаточно прост. Для каждой сканирующей строки буфер кадра инициализируется с фоновым значением интенсивности, а z -буфер — с минимальным значением z . Затем определяется пересечение сканирующей строки с двумерной проекцией каждого многоугольника сцены, если они существуют. Эти пересечения образуют пары. При рассмотрении каждого пикселя на сканирующей строке в интервале между концами пар его глубина сравнивается с глубиной, содержащейся в соответствующем элементе z -буфера. Если глубина этого пикселя больше глубины из z -буфера, то рассматриваемый отрезок будет текущим видимым отрезком. Следовательно, атрибуты многоугольника, соответ-

ствующего данному отрезку, заносятся в буфер кадра в позиции данного пикселя; z -буфер соответственно корректируется в этой позиции. После обработки всех многоугольников сцены буфер кадра размером в одну сканирующую строку содержит решение задачи удаления невидимых поверхностей для данной сканирующей строки. Он выводится на экран дисплея в порядке, определяемом растровым сканированием, т. е. слева направо. В этом алгоритме можно использовать методы устранения ступенчатости, основывающиеся как на пре-, так и на постфильтрации.

Однако практически сравнение каждого многоугольника с каждой сканирующей строкой оказывается неэффективным. Поэтому авторами предложено использовать некоторую разновидность списка упорядоченных ребер. В частности, для повышения эффективности этого алгоритма применена групповая сортировка по оси y списка активных многоугольников и списка активных ребер.

Данный алгоритм решает задачу удаления невидимых поверхностей и делает тривиальной визуализацию пересечений сложных поверхностей. Поскольку габариты пространства изображения фиксированы, оценка вычислительной трудоемкости алгоритма не более чем линейна [1].

Основной недостаток алгоритма — большой объем требуемой памяти. Если сцена подвергается видovому преобразованию и отсекается до фиксированного диапазона координат, то можно использовать z -буфер с фиксированной точностью. Информации о глубине нужно обрабатывать с большей точностью, чем координатную информацию на плоскости (x, y) , обычно бывает достаточно 20 бит. Практическая реализация данного алгоритма позволяет очистить изображение от «паразитных» шумов. Результат использования алгоритма, реализованного в программном модуле обработки сканов, приведен на рис. 1, справа.

После очистки сканов требуется объединить их. Для объединения сканов был использован алгоритм быстрого объединения поверхностей. Алгоритм учитывает множество P_A трехмерных координат p_1, \dots, p_k поверхности A и набор N_A соответствующих трехмерной поверхности нормалей n_1, \dots, n_k в этих точках. Можно выделить множество ориентированных точек (комбинация точки с её нормалью называется ориентированной точкой.) M_A поверхности A и множество ориентированных точек M_B контура поверхности B [1]

$$M_A := \{u = [p_u, n_u] \mid p_u \in P_A \text{ и } n_u \in N_A\},$$

$$M_B := \{v = [p_v, n_v] \mid p_v \in P_B \text{ и } n_v \in N_B\}.$$

Тангенциальный контакт между двумя ориентированными точками $a \in M_A$ и $b \in M_B$ означает, что координаты точек и соответствующих нормалей поверхности совпадают. В этом случае между точками a и b существует тангенциальный контакт, если $p_a = {}^A T_B p_b$ и $n_a = {}^A T_B n_b$, где ${}^A T_B$ — относительное преобразование в однородный координатный записи. Из вышеизложенного следует, что можно по-

строить гипотетическую позицию, предполагая наличие контакта между некоторыми точками на каждой поверхности. Для более точного результата четыре ориентированных поверхностных точки $a, c \in M_A$ и $b, d \in M_B$ являются достаточными, если предположить наличие тангенциального контакта между точками a и b так же, как и между точками c и d . Это предположение верно для всех степеней свободы относительного преобразования. Из рис. 2 можно выделить однородную 4×4 матрицу преобразования путем умножения двух матриц ${}^A T_B = F(a, c)^{-1} F(b, d)$, где функция $F(u, v)$ представляет собой систему координат, лежащую между ориентированными точками u и v с вектором разницы и комбинированным вектором

$$F(u, v) := \begin{bmatrix} \frac{p_{uv} \times n_{uv}}{\|p_{uv} \times n_{uv}\|} p_{uv} & \frac{p_{uv} \times n_{uv} \times p_{uv}}{\|p_{uv} \times n_{uv} \times p_{uv}\|} & p_u + p_v & \\ 0 & 0 & 2 & \\ & & & 1 \end{bmatrix},$$

где $p_{uv} = \frac{p_v - p_u}{\|p_v - p_u\|}$ – разностный вектор; $n_{uv} = n_u + n_v$ – вектор комбинированных нормалей.

Чтобы избежать одинарной матрицы, нужно быть уверенным, что векторы p_{uv} и n_{uv} не нулевые. Точный охват обеих пар точек с противоположными нормальными возможен, только если их относительные расстояния и углы идентичны

$$\text{rel}(u, v) := \begin{bmatrix} d_{uv} \\ \cos \alpha_{uv} \\ \cos \beta_{uv} \\ \delta_{uv} \end{bmatrix} := \begin{bmatrix} \|p_v - p_u\| \\ n_u \times p_{uv} \\ n_v \times p_{uv} \\ a \tan 2(n_u(p_{uv} \times n_v), (n_u \times p_{uv})(p_{uv} \times n_v)) \end{bmatrix}.$$

Чтобы проверить это ограничение, надо определить четырехмерный вектор отношения ориентированной пары точек, состоящий из Евклидова расстояния между точками d_{uv} , углов наклона между нормальными α_{uv} и β_{uv} , между нормальными n_u и n_v , линии, соединяющей p_u и p_v , а также угла поворота δ_{uv} между нормальными вокруг соединительной линии (рис. 2).

Следует отметить, что вектор отношения неизменен относительно вращения и перевода. В работе [2] показано, что подобные отношения векторов могут применяться для быстрого объединения трехмерных поверхностей. Используя эти отношения векторов, можно получить объединенную поверхность H , которая может быть выражена как:

$$H := \{(a, b, c, d) \mid \text{rel}(a, c) = \text{rel}(b, d); a, c \in M_A; b, d \in M_B\}.$$

Представленный алгоритм позволяет последовательно объединить все сканы, по два скана за одну итерацию алгоритма. На основе представленного алгоритма был разработан программный модуль объединения разрозненных сканов.

Полученная математическая модель требует дополнительной обработки, так как почти всегда существуют разрывы поверхности математической модели. Такие разрывы требуется ликвидировать и достроить поверхность. Кроме этого, как минимум требуется решить задачу триангуляции набора точек, задающих поверхность сканированного объекта. Существуют две стратегии работы с полученной моделью: можно сначала произвести последовательную триангуляцию облака точек, представляющих отдельные сканы, а затем выполнить их объединение, либо сначала объединить сканы и выполнить триангуляцию уже над всем облаком точек математической модели объекта. Использование второй стратегии позволяет применить два метода для обработки поверхности.

Первый из них основан на воксельной модели объекта. Воксельная модель – это трехмерный

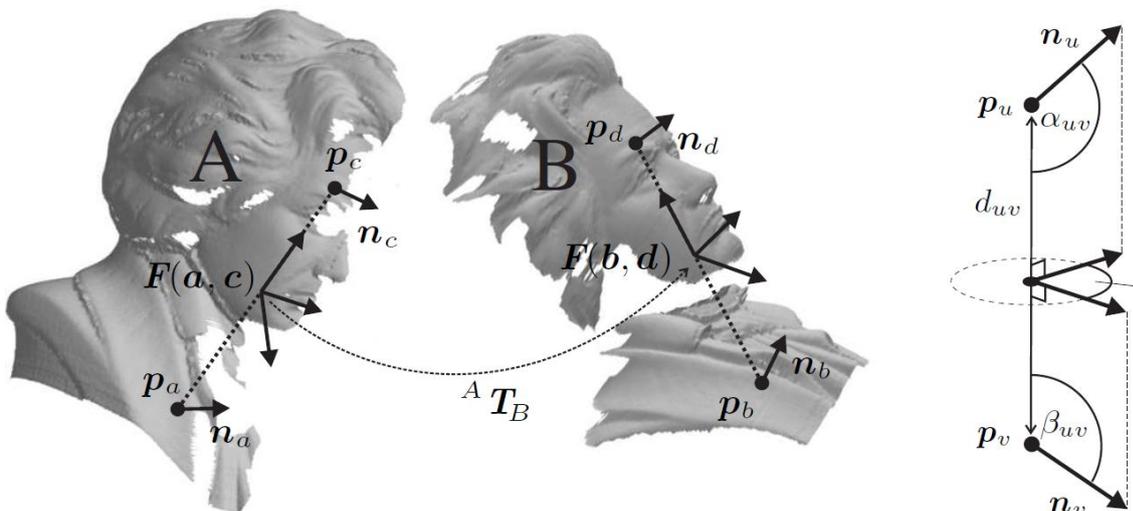


Рис. 2. Относительное преобразование ${}^A T_B$ между предполагаемыми точками контакта (левый); отношения между ориентированными точками u и v (правый)

растр. Подобно тому, как пиксели располагаются на плоскости двумерного изображения, воксели образуют трехмерные объекты в определенном объеме. Воксел – это элемент объема (*voxel – volume element*). В воксельной модели каждый воксел также имеет свой цвет и, кроме того, прозрачность. Полная прозрачность воксела означает пустоту соответствующей точки объема. При моделировании объема каждый воксел представляет элемент объема определенного размера. Чем больше вокселей в определенном объеме и меньше размер вокселей, тем точнее моделируются трехмерные объекты – увеличивается разрешающая способность.

Положительная черта воксельной модели – простота: при описании сложных объектов и сцен; при отображении объемных сцен; при выполнении топологических операций над отдельными объектами и сценой в целом. Например, просто выполняется показ разреза – для этого соответствующие воксели можно сделать прозрачными. В данном случае, для ликвидации разрывов в поверхности, вокселям, отражающим разрыв, придается цветность. Таким образом, разрыв поверхности ликвидируется.

Далее воксельная модель объекта трансформируется с помощью алгоритма триангуляции в стереолитографический формат данных. Он включает в себя описание треугольников, из которых состоит поверхность объекта. Алгоритм работает путем постоянного наращивания к текущей триангуляции по одному треугольнику за шаг. Вначале триангуляция состоит из одного ребра, по окончании работы триангуляция является триангуляцией Делоне. На каждой итерации алгоритм ищет новый треугольник, который подключается к границе текущей триангуляции. Поиск и подключение треугольника образно можно представить, как надувание двумерного пузыря, привязанного к одному из ребер границы. Если такой пузырь достигает некоторой, еще не включенной в триангуляцию точки из набора, то образуется треугольник. В случае, когда точка не встречена, обрабатывается следующее ребро границы [5].

Основное достоинство стереолитографического формата – это простота по сравнению с другими форматами представления трехмерных данных. Данный формат является основополагающим для всех систем быстрого прототипирования, а также используется всеми существующими CAD-системами.

Второй способ обработки поверхностей основан на алгоритме триангуляции полигонов. Математическая модель объекта на предыдущих этапах обработки трансформируется в стереолитографический формат данных. Однако наличие разрывов в поверхности требует применения указанного выше алгоритма. Идея заключается в расщеплении произвольного полигона P (в данном случае полигон – это разрыв поверхности модели) вдоль неко-

торой хорды и затем в рекурсивной триангуляции получившихся двух подполигонов. В конце возникает ситуация, когда подлежащий триангуляции полигон является треугольником. Алгоритм менее эффективен, чем способ, использующий расщепление полигона на монотонные части и их поочередной триангуляции, но его гораздо проще запрограммировать.

Наиболее просто можно осуществить триангуляцию выпуклого полигона. Нужно выбрать одну из n вершин полигона и соединить ее с каждой из $n-3$ не соседних вершин, получив $n-2$ треугольников [3]. Другой способ триангуляции выпуклого полигона основан на том факте, что каждая диагональ выпуклого полигона является его хордой: необходимо расщепить полигон вдоль любой диагонали и затем рекурсивно выполнить триангуляцию полученных двух выпуклых полигонов. На основе представленных рассуждений авторами работы разработан алгоритм преобразования полигона в область треугольник, представляющих поверхность, соизмеримую заполняемому полигону. Разбиение выполняется циклически с шагом дискретизации, определяемом из пропорционального соотношения площади полигона к наибольшей длине секущей плоскости. При этом точность разбиения может задаваться либо пользователем в начале выполнения алгоритма, либо автоматически на основе среднестатистической длины граней треугольников существующих поверхностей. Алгоритм был реализован программно и апробирован на моделях средней сложности. Полученная, в процессе реализации данного алгоритма, математическая модель может быть использована в задачах быстрого прототипирования, а также для формирования управляющей программы станков с ЧПУ. Это позволит создать прототип, либо дубликат отсканированного изделия.

Следует отметить, что авторами работы была предложена методика обработки трехмерных изображений сканированных изделий. На основе методики разработаны алгоритмы, реализация которых позволяет практически сформировать трехмерную модель изделия, состоящую из поверхности, представленной в виде набора треугольников. Также в работе рассмотрен вопрос выбора алгоритма триангуляции, обеспечивающего заполнение разрывов поверхности модели сканированного изделия. Данный алгоритм позволяет реализовать механизм увеличения точности (сглаженности) формируемой поверхности изделия. В работе не решена задача трансформации поверхности, представленной в стереолитографическом формате, в твердотельное изделие, но это не снижает значимость проведенных исследований, которые, в конечном счете, позволяют формировать математическую модель, отвечающую всем требованиям аппаратуры трехмерного прототипирования.

СПИСОК ЛИТЕРАТУРЫ

1. Franke K., Müller K., Nickolay B., Schäfer R. Pattern Recognition, vol. 4174 of Lecture Notes in Computer Science. – Berlin: Springer, 2006. – P. 718–728.
2. Wahl E., Hillenbrand U., Hirzinger G. Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification // Proc. 4th Intern. Conf. on 3-D Digital Imaging and Modeling. – Gent, Belgium. – 2003. – P. 474–481.
3. Ласло М. Вычислительная геометрия и компьютерная графика на C++. – М.: БИНОМ, 1997. – 304 с.
4. Роджерс Д., Адамс Дж. Математические основы машинной графики: Пер. с англ. – М.: Мир, 2001. – 604 с.
5. Романюк А., Сторчак А. Алгоритмы триангуляции // Компьютеры + программы. – 2001. – № 1. – С. 40–42.

Поступила 16.09.2010 г.

УДК 519.688

МЕТОДЫ ОПТИМИЗАЦИИ РАСЧЁТА ОСВЕЩЕНИЯ ДЛЯ ИНТЕРАКТИВНЫХ ПРИЛОЖЕНИЙ

В.В. Парубец, А.С. Огородников, О.Г. Берестнева

Томский политехнический университет
E-mail: ancora_vivo@sibmail.com

Рассматривается методика расчёта освещения, основанная на разложении функций отражённого и рассеянного света в ряд по присоединённым сферическим функциям, предлагаются возможности её оптимизации и устранения ограничений, обусловленных статичной геометрией и типом источников света.

Ключевые слова:

Обобщённая двулучевая функция распределения света, модель Кука–Торренса.

Key words:

Bidirectional reflectance distribution function, Cook–Torrance equation.

Введение

Современная компьютерная графика позволяет решать задачи от визуализации сложных физических процессов до создания интерактивных трёхмерных миров, уровень детализации и качества эффектов которых, недостижимый ещё несколько лет назад, способен превзойти все ожидания. Перед разработчиками интерактивных приложений, таких, как компьютерные игры, ставится задача, максимально увеличив реалистичность графической составляющей, сохранить скорость работы всей программы, достаточной для общей игровой динамики и полной интерактивности. Программно-аппаратные средства типа *OpenGL* и *DirectX* предоставляют широкий простор для действий разработчика в области обработки объектов сцены на каждом из этапов её визуализации, получая в итоге необходимые графические эффекты. Стоит отметить, что расширения указанных выше API отвечают не только за обработку отдельных пикселей или вертексов, но и позволяют перевести данные вычисления на мощности графических процессоров, а наличие языков высокого уровня для их программирования позволяет сосредоточиться на алгоритмической стороне задачи.

Основной составляющей «реалистичности» в современных компьютерных играх является освещение. Из-за возможной сложности сцены, различного количества источников света, их типа, а также различных типов материалов объектов сцены задача расчёта освещения, несмотря на существующие математические модели и массу методов оптимизации,

остаётся крайне нетривиальной и требующей больших вычислительных мощностей.

Математическая модель

В классическом виде расчёт освещения представлен следующей моделью [1]:

$$L(x, \vec{\omega}_0) = L_e(x, \vec{\omega}_0) + \int_S f_r(x, \vec{\omega}_i) L(x', \vec{\omega}_i) G(x, x') d\vec{\omega}_i,$$

где $L(x, \vec{\omega}_0)$ – интенсивность отражённого потока света из точки x поверхности в направлении $\vec{\omega}_0$; $L_e(x, \vec{\omega}_0)$ – интенсивность потока света, излучаемого самой поверхностью; $f_r(x, \vec{\omega}_i \rightarrow \vec{\omega}_0)$ – двулучевая функция распределения света поверхности в точке x , переводящая свет $\vec{\omega}_i$ в отражённый свет $\vec{\omega}_0$; $L(x, \vec{\omega}_i)$ – интенсивность потока света, исходящего от других объектов в направлении $\vec{\omega}_i$; $G(x, x')$ – геометрическое отношение между точками x и x' .

Для ламбертовой поверхности вид упрощается:

$$L(x) = \int_S \max((\vec{n} \cdot \vec{\omega}_i), 0) V(x, \vec{\omega}_i) d\vec{\omega}_i,$$

где $V(x, x')$ – функция видимости: принимает значение 0 или 1 в зависимости от того, освещён объект или нет; $\max((\vec{n} \cdot \vec{\omega}_i), 0)$ – количество рассеянного света, согласно закону Ламберта.

Описание методов реализации

Существует огромное количество реализаций данной модели с различными приближениями и ограничениями. Основные трудности в разработке собственной реализации модели освещения касаются в основном функции распределения све-