

УДК 004.92

МЕТОД ГЕОМЕТРИЧЕСКОГО УПРОЩЕНИЯ
3D ПОЛИГОНАЛЬНЫХ ОБЪЕКТОВ

В.Т. Тозик, А.В. Меженин

В статье рассматривается метод геометрического упрощения 3D полигональных моделей объектов с целью сокращения объема данных, что приводит к повышению скорости процессов передачи и визуализации. Предлагаемый метод упрощения трехмерных полигональных объектов использует операции свертывания ребер. Дополнительно выполняется проверка, направленная на устранение эффекта вырожденной полигональной сетки. Для оценки качества упрощения полигональной сети предлагается использовать размерность Хаусдорфа.

Ключевые слова: многомасштабное представление, упрощение сеток.

Введение

Одной из основных проблем передачи и отображения трехмерных данных является передача больших объемов данных, необходимых для описания сложных трехмерных моделей в условиях ограничения пропускной способности каналов передачи данных, таких как Интернет, космическая связь и т.д. [1]. Упрощение сложных 3D объектов является важным направлением исследований в компьютерной графике. Уменьшение уровней детализации Level of Detail (LOD) упрощает не только задачу передачи данных, но и задачу визуализации, особенно для небольших дисплеев мобильных устройств и специальной аппаратуры.

Большинство итерационных алгоритмов упрощения полигональных моделей можно разделить на три категории: прореживание вершин, свертывание ребер, прореживание граней [2, 3] (рис. 1).

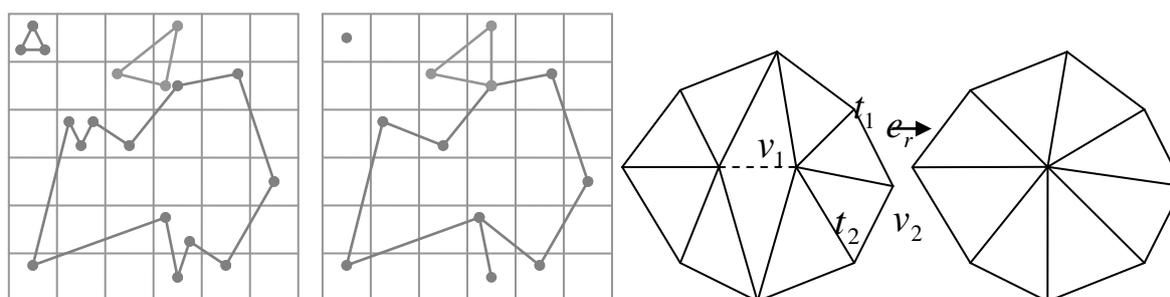


Рис. 1. Прореживание вершин и свертывание ребер

Алгоритм кластеризации вершин относится к категории прореживания вершин и состоит в объединении вершин, находящихся в одном кластере пространства, в одну. Его достоинство – высокая скорость, а недостаток в некоторых случаях – нарушение топологии получаемых моделей. Более точно топологию модели передает алгоритм прореживания вершин, выполняемый в несколько проходов. На каждом этапе происходит удаление вершин, расположенных на расстоянии меньше заданного от усредненной плоскости соседних вершин. Одна из проблем, возникающая при реализации алгоритма – исчезновение граней, принадлежащих удаляемым вершинам. При этом могут образоваться разрывы, которые необходимо заполнить, выполняя операции триангуляции, что требует дополнительного времени.

Алгоритмы, основанные на свертывании ребер, можно рассматривать как частный случай удаления вершин, однако они не требуют выполнения дополнительных операций триангуляции. Поэтому их реализация дает хорошие результаты с точки зрения качества и быстродействия. Свертывание ребра представляет собой слияние двух вершин, образующих ребро, в одну. При этом в общем случае происходит удаление двух треугольных ячеек. Последовательность свертывания ребер определяется мерой ошибки, которая отражает локальное геометрическое отклонение ячеек. Способы вычисления этой ошибки определяют различие между алгоритмами этого класса.

Известны следующие меры ошибки, используемые для выбора стратегии при свертывании ребра:

- определение среднего расстояния между новыми треугольниками и типовыми точками на исходной модели;
- нахождение величины допуска как выпуклой комбинации сфер, расположенных в каждой вершине упрощения. Выбор граней основывается на самой малой длине, а новая вершина позиционируется таким образом, что новая поверхность будет гарантированно лежать в пределах этого допуска;
- поддержка связи между точками исходной модели и соответствующими окрестностями на упрощенной модели.

Метод кластеризации ограничивающего объема и операции свертывания ребер

Предлагаемый авторами подход к задаче упрощения трехмерных полигональных объектов и организации прогрессивной сетки основан на кластеризации ограничивающего объема и операциях свертывания ребер. Очевидно, что в первую очередь необходимо обрабатывать только те вершины, которые не влияют существенно на форму объекта. Вершины, в которых поверхность объекта существенно меняется, будем называть опорными. Такие вершины выделяются на стадии предварительной обработки и постоянно присутствуют при синтезе объекта. Выделение опорных вершин производится следующим образом.

Шаг 1. Строится ограничивающий объем, представляющий собой параллелепипед, стороны которого параллельны осям координат. Координаты вершин параллелепипеда находятся как значения минимальных и максимальных координат вершин объекта по осям:

$$\begin{aligned} x_{\min} &= \min(x_1, x_2, \dots, x_n), \quad x_{\max} = \max(x_1, x_2, \dots, x_n), \\ y_{\min} &= \min(y_1, y_2, \dots, y_n), \quad y_{\max} = \max(y_1, y_2, \dots, y_n), \\ z_{\min} &= \min(z_1, z_2, \dots, z_n), \quad z_{\max} = \max(z_1, z_2, \dots, z_n) \end{aligned}$$

Шаг 2. Координаты ограничивающих объемов находятся из предыдущих значений следующим образом:

$$x_{c_k} = \frac{x_{\min} + x_{\max}}{2k_x}, \quad y_{c_k} = \frac{y_{\min} + y_{\max}}{2k_y}, \quad z_{c_k} = \frac{z_{\min} + z_{\max}}{2k_z},$$

где k_x, k_y, k_z – количество разбиений по осям X, Y, Z .

Шаг 3. Определяется принадлежность вершин к каждой области разбиения путем проверки условий

$$x_{\min} \leq x_i \leq x_{\max}, \quad y_{\min} \leq y_i \leq y_{\max}, \quad z_{\min} \leq z_i \leq z_{\max}.$$

Шаг 4. Определяется квадрат расстояния от вершин области до центров областей разбиения:

$$d_i^2 = (x_{c_{k+1}} - x_i)^2 + (y_{c_{k+1}} - y_i)^2 + (z_{c_{k+1}} - z_i)^2.$$

Шаг 5. Находятся минимальные и максимальные квадраты расстояний. Вершины, расположенные на минимальном и максимальном расстояниях, удаляться не будут.

Дальнейшее упрощение основано на свертывании ребер. В качестве критериев используются угол между нормальными векторами граней и изменение площади свертываемой грани. На каждом шаге производятся следующие действия.

- Вычисляется показатель ошибки для каждой операции свертывания ребра и выполняется упорядочивание по приоритетам.
- Выбирается ребро $\vec{e}_r(v_0, v) \mapsto v$ с минимальной величиной ошибки и заменяется на v . Во время этой операции треугольники, имеющие общие ребра с e_r , становятся сингулярными и удаляются. Оставшиеся грани, инцидентные вершине v_0 , модифицируются таким образом, что все, что было связано с v_0 , заменяется на v .
- Пересчитывается величина ошибки свертки ребра $\vec{e}_r(v_1, v) \mapsto v$. Добавляется стоимость ошибки схлопывания $\vec{e} = (v_0, v_1)$ в ту часть очереди, которая связана с вершиной v_1 , и модифицируется очередь по приоритетам. В результате формируется прогрессивная структура сети.

Когда происходит свертывание произвольного ребра $\vec{e}_r(v_0, v) \mapsto v$, две грани, примыкающие к ребру, становятся выродившимися. Свертывание ребра $e_r = \{v_0, v\}$ приводит к отображению треугольника $t = (v_0, v_1, v_2)$ в треугольник $t' = (v, v_1, v_2)$. Ребра $\{v_0, v_1\}$ и $\{v_0, v_2\}$ треугольника $t = (v_0, v_1, v_2)$ вырождаются (рис. 2).

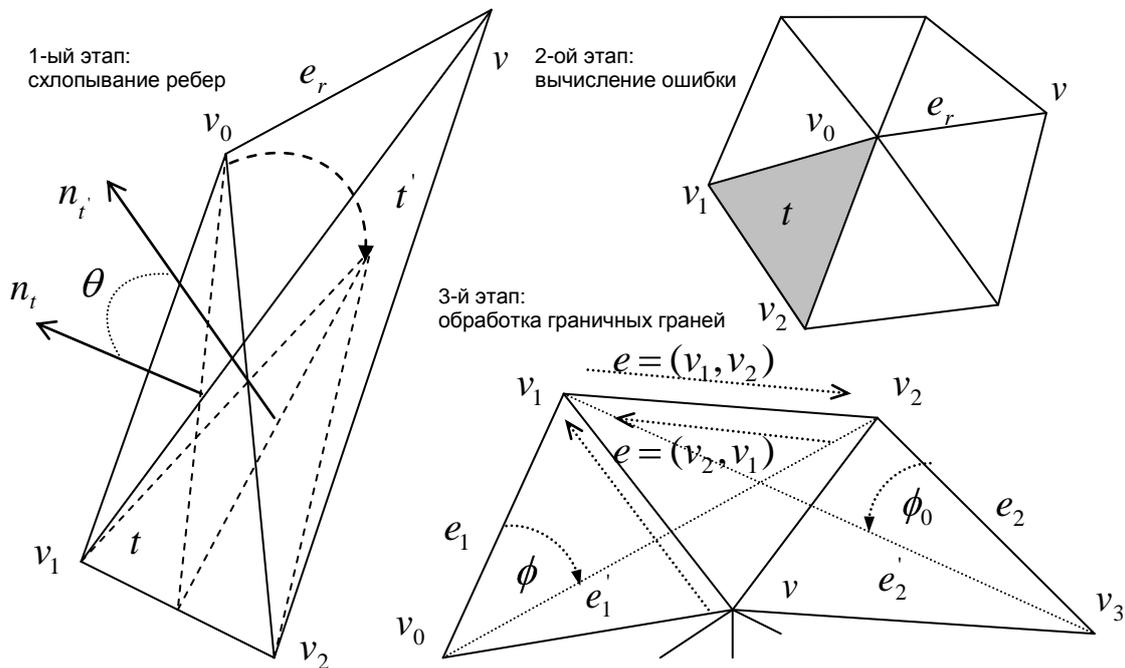


Рис. 2. Свертывание ребер

В результате смещения оставшихся граней происходит изменение геометрической формы поверхности. Величина этого изменения может характеризоваться, с одной стороны, углом поворота грани $t = (v_0, v_1, v_2)$, с другой стороны, площадью треугольника $t'' = (v_0, v, v_1)$. Чем больше та или другая величина, тем больше изменение формы

поверхности. Таким образом, для грани $t = (v_0, v_1, v_2)$ величина ошибки может характеризоваться произведением этих величин:

$$Q_t = A\theta,$$

где $A = \frac{1}{2} \|\mathbf{a} \times \mathbf{b}\|$, $\mathbf{a} = \mathbf{v}_0 - \mathbf{v}_1$, $\mathbf{b} = \mathbf{v}_1 - \mathbf{v}_2$, θ – угол между векторами нормалей двух треугольников. Суммарная величина ошибки $\vec{e}_r(v_0, v) \mapsto v$ может быть определена как сумма ошибок сверток всех граней, примыкающих к v_0 : $Cost(\vec{e}_r) = \sum_{t \in (v_0, v) \mapsto v} Q_t$.

Граничные грани требуют специальной обработки для сохранения формы модели. Ребра, расположенные вдоль границы, делим на две категории: ребра, которые имеют одну вершину на границе $\vec{e} = (v, v_1)$ и ребра, имеющие две вершины на границе $\vec{e} = (v_1, v_2)$. Ребро $e_1 = \{v_0, v_1\}$ поворачивается на угол ϕ при свертывании $\vec{e} = (v_1, v_2)$. Соответственно, если свертывается $\vec{e} = (v_2, v_1)$, то ребро $e_2 = \{v_2, v_3\}$ повернется на угол ϕ_0 , при этом получается $\phi_0 < \phi$. В результате схлопывания ребра $e = \{v_0, v\}$ треугольник $t = (v_0, v_1, v_2)$ может перекрыть грань (рис. 2).

Свертывание ребра $\vec{e} = (v_s, v_t) \mapsto v_t$ с конечной вершиной, находящейся на границе, не требует специальной обработки. Однако, если начальная вершина находится на границе, то свертывание ребра исказит границу, и это необходимо учитывать.

Ребро $e = \{v_0, v_1\}$ может быть свернуто в вершины v_1 или v_2 , но лучший результат будет при сворачивании в v_1 . Для достижения этого необходимо измерить угол ϕ и ϕ_0 между ребрами e_1 и e_1' и e_2 и e_2' соответственно.

Результаты апробации алгоритма

На рис. 3 представлены результаты апробации предлагаемого метода. Исходная полигональная модель содержит 3679 граней, т.е. 100% (а), а результаты упрощения – 2856 граней, т.е. 80% (б), и 2441 грань, т.е. 60% (в). На рис. 3 видно сохранение топологии, отсутствие разрывов и вырожденных граней.

Для оценки качества работы алгоритма и сравнения с другими известными алгоритмами предлагается вычислять ошибку аппроксимации между двумя полигональными моделями, которая базируется на вычислении расстояния аппроксимации (approximate distance) и может быть определена следующим образом. Дана точка p и поверхность S . Определим расстояние $e(p, S)$, и расстояние между двумя поверхностями S_1, S_2 как $e(p, S) = \min_{p' \in S} d(p, p')$, $E(S_1, S_2) = \max_{p \in S_1} e(p, S_2)$, где $d()$ – евклидово расстояние между двумя точками в E^3 . Необходимо отметить, что это расстояние несимметрично. Существуют поверхности такие, что $E(S_1, S_2) \neq E(S_2, S_1)$. Хаусдорфово расстояние (двустороннее) может быть получено как максимум $E(S_1, S_2)$ и $E(S_2, S_1)$. Среднее (mean) E_m расстояние между двумя поверхностями можно определить как интеграл по поверхности, определенный областью S_1 :

$$E_m(S_1, S_2) = \frac{1}{|S_1|} \int_{S_1} e(p, S_2) ds.$$

На рис. 4 приведены графики средней ошибки (расстояния аппроксимации) при уменьшении количества граней тестовой полигональной модели для разных алгоритмов упрощения [2, 3]. Разработанный алгоритм показал лучшие результаты.

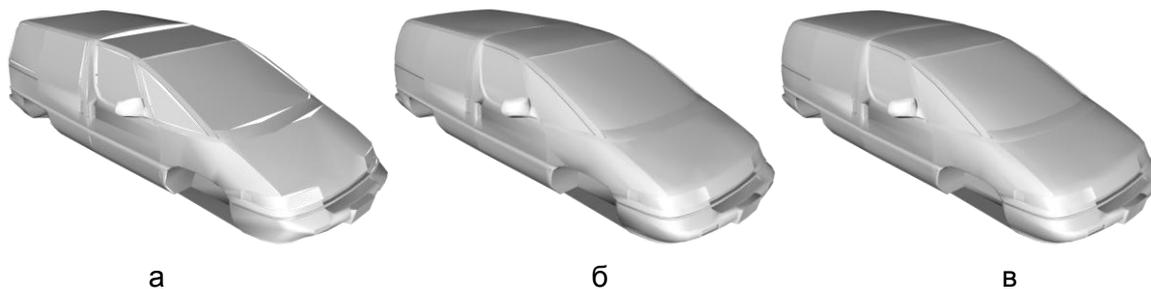


Рис. 3. Результат работы алгоритма

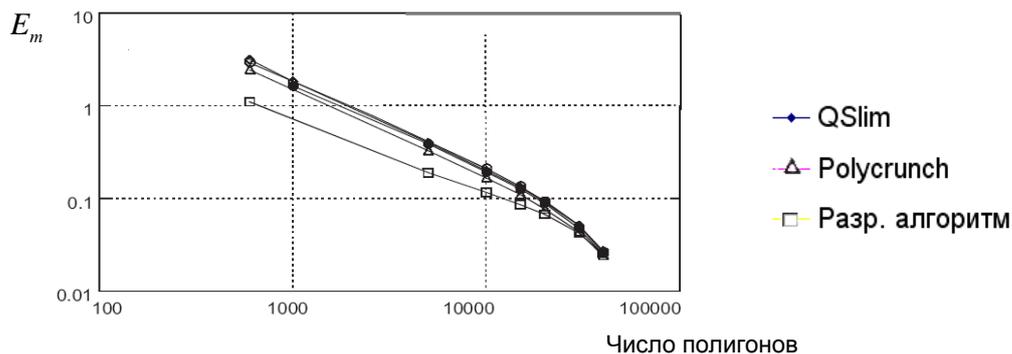


Рис. 4. Графики средней ошибки

Заключение

- В работе предложен алгоритм, обладающий следующими характеристиками.
- Алгоритм алгоритмически сохраняет визуально важные особенности полигональной модели лучшим образом по сравнению с известными алгоритмами упрощения.
 - Предложенный способ накопления ошибки не требует сохранения истории геометрических изменений, что позволяет эффективно упрощать большие полигональные модели, не требуя больших объемов памяти.
 - Алгоритм в вычислительном отношении показывает лучшие результаты по сравнению с известными алгоритмами, основанными на свертывании ребер.
 - Алгоритм автоматически предотвращает появление искажений типа сверток.

Литература

- 1.Парамонов П.П., Видин Б.В., Меженин А.В., Тозик В.Т. Методы представления сложных полигональных моделей в графических системах, работающих в режиме реального времени // Изв. вузов. Приборостроение. – 2006. – Т. 49. – № 6. – С. 17–19.
2. Hoppe H. Progressive Meshes // Computer Graphics. – Vol. 30. –SIGGRAPH 96.
- 3.Garland M. Heckbert P. Multiresolution Modeling for Fast Rendering // Proceedings of Graphics Interface '94. – 1994.

- Тозик Вячеслав Трофимович** – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, заведующий кафедрой, tozik@mail.ifmo.ru
- Меженин Александр Владимирович** – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, mejenin@mail.ru