

## ПРАКТИЧЕСКОЕ СРАВНЕНИЕ МЕТОДОВ КОДИРОВАНИЯ КООРДИНАТ ВЕРШИН ТРЁХМЕРНЫХ МОДЕЛЕЙ

*Буртыкин Дмитрий Алексеевич*

*Студент*

*Факультет МиКН СГОАН, Самара, Россия*

*E-mail: dmitrii\_burtykin@nayanova.edu*

Самым популярным видом отображения типов поверхностей и размеров трехмерных моделей является полигональная сетка. На сегодняшний день почти в каждом персональном компьютере, телефоне или планшете присутствует графический процессор, отвечающий за отображение двухмерных изображений из пол. сеток.

Со временем потребность в точности отображения данных в областях, использующих пол. сетки (медицина, развлечения, электронная коммерция), только увеличивается; это ведёт к повышению детализации модели и, как следствие, к расширению пространства, занимаемого пол. сеткой одной 3D-модели. Одним из способов сжатия модели для использования меньшего пространства является преобразование пол. сетки в список состоящих из треугольников-полигонов "вееров" с последующим кодированием энтропии полученной модели.

Исследование применимости существующих методов сжатия к 3D-видео данным было рассмотрено в [1]. Обзор и сравнение методов сжатия трехмерных моделей проведен в [2]. В зависимости от задач визуализации и передачи данных существуют различные подходы, как например [3].

В процессе решения конечной задачи были реализованы алгоритмы определения начальной вершины для каждого веера, свёртки вершин в веер и сохранения информации о каждом веере с наименьшими потерями. К конечным данным для сравнения эффективности сжатия применяются алгоритмы Хаффмана и арифметического кодирования.

Входными данными нашей задачи является файл формата .obj - открытого формата файлов описания геометрии. Формат файлов OBJ — это простой формат данных, который содержит только 3D геометрию, а именно, позицию каждой вершины, связь координат текстуры с вершиной, нормаль для каждой вершины, а также параметры, которые создают полигоны [4]. Применение подхода с учетом геометрии модели является целесообразным. В работе рассмотрена

модификация исходной модели по методу, предложенному в [5].

Модель преобразуется в набор вееров, задаваемых следующими свойствами: веер степени  $d$  есть упорядоченный набор  $d$  треугольников, обозначаемый

$$(t_j), j \in 0, \dots, d - 1, \quad (1)$$

и определяемый последовательностью  $(d+2)$  вершин

$$(v_0, v_1, \dots, v_{d+1}) \quad (2)$$

таких, что

$$\forall j \in \{0, \dots, d - 1\}, t_j = \{v_0; v_{j+1}; v_{j+2}\}. \quad (3)$$

По определению, треугольники веера TF удовлетворяют следующим свойствам:

- (P1): каждые два последовательно следующих треугольника веера имеют общую грань,
- (P2): все треугольники веера TF имеют одинаковое направление обхода вершин,
- (P3): все треугольники веера TF имеют общую вершину  $v_0$ , называемую центром веера TF.

Результатом предварительного преобразования исходных моделей в список вееров, в зависимости от типа и формы фигуры, послужило уменьшение объема конечного файла. Применение алгоритмов сжатия к полученному набору данных позволило получить дополнительное уменьшение объема конечного файла. Реализация данного подхода позволяет получить максимальную степень сжатия для исходных данных трехмерных моделей, за счет увеличения времени обработки и требований к вычислительным ресурсам. Изучение вопросов оптимизации предложенного подхода представляется на данный момент актуальным.

### Литература

1. Аксенов А. Ю. Исследование применимости существующих методов сжатия к 3D-видео данным. // Труды СПИИРАН. 2013. Вып. 4(27). С. 73-80 (из перечня ВАК)
2. Maglo A., Lavoue A., Dupont F., Hudelot C. 3D mesh compression: survey, comparisons and emerging trends. ACM Comput. Surv. 9, 4, Article 39 (September 2013), 40 p.

3. Sungyul C., Junho K., Haeyoung L., Seungyong L. Random Accessible Mesh Compression Using Mesh Chartification // IEEE Trans Vis Comput Graph, 2009 Jan-Feb, 15(1):160-73.
4. Спецификация формата Object Files (.obj). [http://www.cs.utah.edu/~boulos/cs3505/obj\\_spec.pdf](http://www.cs.utah.edu/~boulos/cs3505/obj_spec.pdf)
5. Mamou K., Zaharia T., Preteux F. TFAN: A low complexity 3D mesh compression algorithm. Comp. Anim. Virtual Worlds (2009). [www.interscience.wiley.com](http://www.interscience.wiley.com)