

СЕКЦИЯ 06
Прикладные задачи обработки информации в управлении

ОСОБЕННОСТИ ПРЕДСТАВЛЕНИЯ ПРОСТРАНСТВЕННЫХ ДАННЫХ, ПОЛУЧЕННЫХ В РЕЗУЛЬТАТЕ 3D-СКАНИРОВАНИЯ

А.Ю. Аксенов, В.В. Александрова, А.А. Зайцева

*Федеральное государственное бюджетное учреждение науки
Санкт-Петербургский институт информатики и автоматизации РАН (СПИИРАН)
Санкт-Петербург, 199178, 14-я линия В.О., дом 39.
Тел.: +7-(812)-323-51-39. E-mail: cher@iias.spb.su*

Ключевые слова: 3D-моделирование, 3D-сканирование, 3D-прототипирование, облако точек, 3D-модель

Статья посвящена анализу особенностей представления растровых пространственных данных (облако точек), таких, как точность и ее зависимость от разрешающей способности 3D-сканера, влияние постобработки, плотность заполнения, артефакты, вызванные ошибками сканирования.

Предложен метод эффективного представления, учитывающий вышеперечисленные особенности, предназначенный для повышения компактности хранения 3D-моделей в цифровых хранилищах и упрощения передачи объемных данных по каналам связи в рамках цифровой программируемой технологии.

Введение

Начиная с Р.Декарта и Г.Монжа сформировались парадигмы прямоугольного представления координат в пространстве и способы описания объектов средствами начертательной геометрии.

Позднее, при начале доминирования цифровой техники над аналоговой отраслевым стандартом стало «пиксельное представление» объектов, в отличие от аналогового «векторного представления».

В настоящее время 3D-технологии в самых различных областях человеческой деятельности стали обыденностью. Развитие цифровых программируемых технологий привело к возможности непосредственной цифровизации трехмерных объектов, компьютерного 3D-моделирования и репликации объектов.

Работа является продолжением серии исследований по 3D-технологиям и когнитивному программированию [1], [2], [3], [4], а также развитием методов компрессии пространственных данных [5]. Как отмечалось в [1], [4], развитие программируемых цифровых технологий привело к возможности быстрого и высококачественного сканирования, т.е. непосредственной оцифровки сложных художественных фигур и пространственных трехмерных объектов.

3D-данные могут быть получены различными способами: моделирование, сканирование, применение методов томографии, восстановление 3D-данных по нескольким изображениям.

В работе проводятся эксперименты с данными, полученными в результате 3D-сканирования.

3D-сканирование (трёхмерное сканирование)

3D-сканирование (трёхмерное сканирование) — это процесс перевода физической формы реального объекта в цифровую форму, то есть получения компьютерной модели объекта.

Трёхмерное сканирование активно используется при решении задач реверс-инжиниринга, проектирования приспособлений, оснастки, запасных частей при отсутствии оригинальной компьютерной документации на изделие, при необходимости перевода в цифровой вид поверхностей сложной формы, в том числе художественных форм и слепков. Также 3D-сканирование играет важную роль в медицинских задачах для протезирования [6].

Бесконтактные 3D-сканеры в основном используют двойную систему ввода координат тела. Многие устройства совмещают лазерные датчики и цифровую фотокамеру, что позволяет получить модели объектов с наложенными текстурами.

В некоторых случаях вместо лазерных датчиков пространства могут применяться и более сложные системы. Например, появились системы 3D-сканирования на базе ультразвуковых установок, преимуществом которых перед конкурентами является режим сканирования тел с внутренней структурой или тел, погруженных в однородную среду.

Активно ведутся разработки магнитных сканеров, использующих для определения пространственных координат объекта изменение его пространственного магнитного поля. Многообразие столь сложных устройств говорит о том, что без хорошего программного обеспечения и вмешательства человека данные, получаемые 3D-сканерами, все равно остаются малоприменимым набором цифр [7].

В настоящее время известно множество методов представления трехмерных объектов и связанных с ними методов визуализации, в том числе многомасштабных.

В зависимости от способа получения 3D-данных, занимаемый ими объем битового представления будет различным (таблица 1).

Таблица 1

Ориентировочное сравнение объемов битового представления различных 3D-объектов

Вид объекта	Способ получения	Размер, байт
Барельеф	3D-сканирование	108 265 982
Подшипник	3D-сканирование	137 843 202
Вертолет (модель)	синтезирован в 3D Studio Max	5 505 893
Стереопара в HD-качестве (фильм)		12 441 600

В таблице 1 представлены сравнительные объемы исходного (несжатого) битового представления различных 3D-объектов. Из нее видно, что для эффективного хранения 3D-объектов необходимы форматы представления этих объектов в сжатом виде, ориентированные на конкретный класс представления.

Исходные данные для экспериментов

Для проведения экспериментов использовалось полученное с помощью 3D-сканера Artec Spider облако точек (рисунок 1). Под облаком точек будем понимать набор вершин в трёхмерной системе координат, которые предназначены для представления внешней поверхности объекта.

Параметры сканирования: процесс сканирования производился в «реальном времени» с использованием программного обеспечения Artec Studio версии 9.2.3.15. Облако точек получено за один проход сканирования.

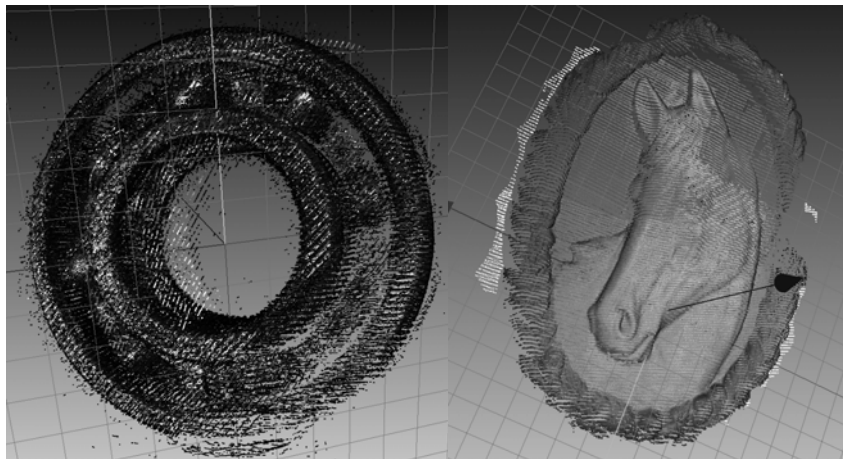


Рис. 1. Исходное облако точек различных 3D-объектов

Описание метода

В процессе работы 3D-сканер формирует наборы изображений с нескольких камер, которые с помощью программной обработки преобразуются в набор пространственных точек с известными расстояниями от них до сканера с общей системой координат.

Традиционным представлением сканированных данных являются форматы: ply, STL, хуз, WRML, трехмерное облако точек (asc), aop, obj, ptx, x, xyzrgb, e57.

В них в текстовой или бинарной форме содержится матрица координат ненулевых точек \mathbf{M} . Для эффективного представления матрицы \mathbf{M} предлагается алгоритм, блок-схема которого представлена на рисунке 2.

Используем тот факт, что облако точек является трехмерной битовой матрицей \mathbf{M} , которая при конечном разрешении, обеспечиваемым сканером, является разреженной и может быть достаточно эффективно скомпрессирована.



Рис. 2. Общая схема алгоритма компрессии

Основным этапом сжатия является преобразование трехмерной матрицы \mathbf{M} в линейную бинарную последовательность (в терминологии работы [8] этот этап является нормализацией, ориентированной на семантику данных). Особенности матрицы \mathbf{M} являются компактные области точек пространства, соответствующие поверхностям сканируемого объекта, соответственно для этапа нормализации (преобразования $3D \rightarrow 1D$) (рисунок 2) целесообразно использовать алгоритм обхода точек, сохраняющий локальные особенности, например, алгоритм заполняющей пространство кривой (ЗПК). Способ построения ЗПК для применения в процедуре кодирования описан в [9].

Визуализация ЗПК для этого случая представлена на рисунке 3.

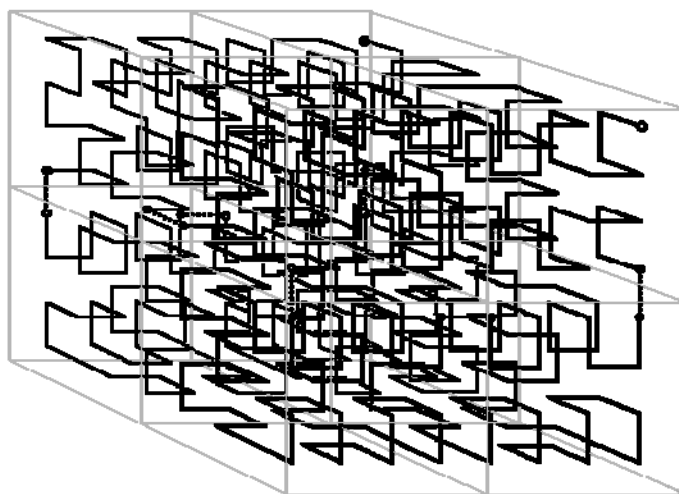


Рис. 3. Заполняющая пространство кривая для трехмерного случая с размерностью $8 \times 8 \times 8$

Групповое кодирование. После преобразования матрицы \mathbf{M} из 3D в 1D в сформированной битовой последовательности присутствуют длинные последовательности нулей, которые при использовании группового кодирования типа RLE позволяет достичь уменьшения битового объема.

Алгоритм RLE (Run-Length Encoding) осуществляет групповое кодирование длинных последовательностей нулей. При этом формируются пары вида <пропустить, число>, где «пропустить» является счетчиком пропускаемых нулей, а «число» — значение, которое необходимо поставить в следующую ячейку. Так, вектор 42 3 0 0 0 -2 0 0 0 1 ... будет свернут в пары (0,42) (0,3) (3,-2) (4,1)

Вторичное сжатие. В качестве заключительного этапа вторичного сжатия может быть использован один из методов энтропийного сжатия, например, арифметическое сжатие [10] или сжатие Хаффмана [11]. Эти алгоритмы компрессии позволяют уменьшать среднюю длину кодового слова для символов алфавита.

Результаты

В таблице 2 приведены результаты применения предлагаемого метода для различных сложных пространственных форм, полученных в результате сканирования.

При компрессии все пространство разбивалось на трехмерные ячейки размером $256 \times 256 \times 256$, каждая из которых сжималась независимо от других, а результаты сжатия объе-

динялись в единый выходной поток. В таблице приводятся результаты компрессии ячейки $256 \times 256 \times 256$ для разных объектов. Приводится сравнение неkomпрессированного битового объема облака точек, заданного в виде координат (RAW-файл), объема RAW-файла, сжатого алгоритмом Deflate, а также результаты применения предложенного алгоритма компрессии с использованием обхода пространства по ЗПК и стандартного построчного преобразования $3D \rightarrow 1D$.

Таблица 2

Результаты применения метода

Объект	Количество точек в облаке	Размер элемента 3D-куба	Размер файла с облаком до сжатия, байт	Размер после сжатия методом Deflate, байт	Размер после сжатия стандартным сканированием, байт	Размер после сжатия предложенным методом, байт
Horse (барельеф)	21312	256	127872	44152	14486	10891
Подшипник	11198	256	67188	21991	12251	5519

Для экспериментальной проверки предлагаемый метод реализован программно в виде кроссплатформенного приложения на языке Java, представляющего собой кодер/декодер компрессированного потока 3D-данных (рисунок 4). В режиме кодера исходный формат представления (облако точек или набор полигонов) при помощи конвертера преобразуется во внутреннее представление матрицы \mathbf{M} в виде массива координат точек. Отдельный программный формирователь ЗПК задает порядок обхода матрицы \mathbf{M} для последующего группового кодирования и вторичного сжатия. При необходимости выходной поток может быть сохранен в одном из стандартных контейнеров (XML, RIFF и т.д.) или стать основой сетевого протокола. Впоследствии кодер/декодер может быть выполнен в виде модуля (plugin) для программных продуктов, работающих с данными в 3D-формате для непосредственной работы с этим форматом. Декодер работает в противоположном направлении и использует тот же программный формирователь ЗПК.

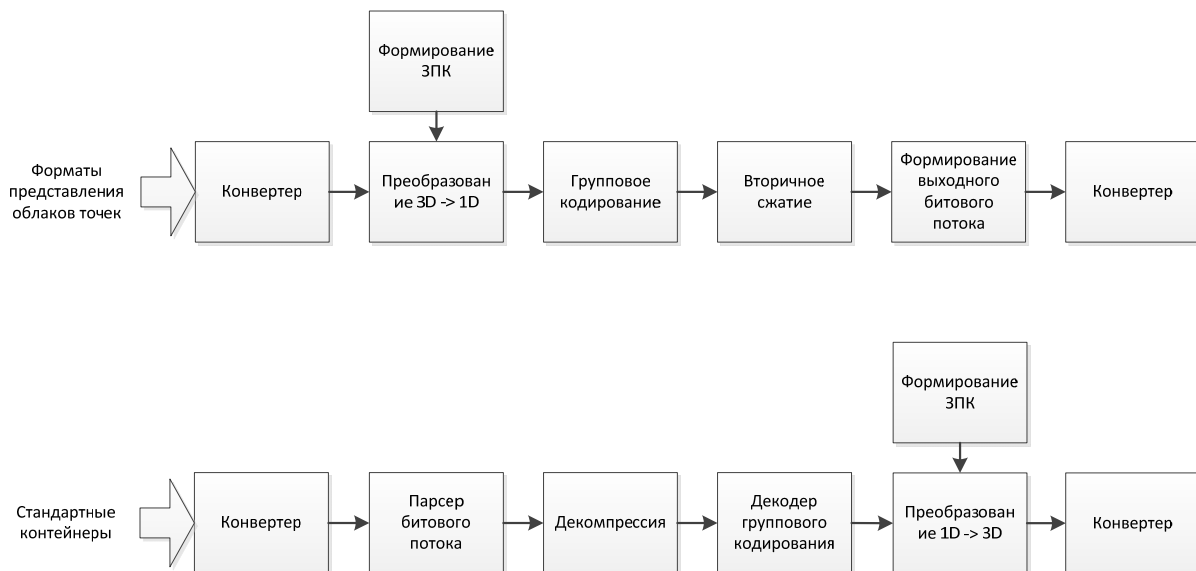


Рис. 4. Схема программной реализации алгоритма компрессии 3D-данных

Заключение

В статье предложен метод переупорядочивания облака точек в битовый поток, имеющий упорядоченную структуру, отличающийся способностью сохранять локальные области пространств, что позволяет эффективно применять методы группового кодирования для уменьшения битового объема.

Разработан алгоритм сжатия 3D-данных, отличающийся применением заполняющей поверхности кривой для переупорядочивания облаков точек и не требующий для работы восстановленной поверхности объекта (набора полигонов).

В связи с тем, что облако точек, полученное в результате 3D-сканирования, представляет собой компактные области точек пространства, соответствующие поверхностям сканируемого объекта, то применение ЗПК дает положительный эффект.

Предложенные методы могут быть использованы в программном обеспечении, предназначенном для работы с 3D-сканерами, пакетами пост-обработки сложных пространственных форм для повышения эффективности использования систем хранения 3D-данных, для создания библиотек 3D-объектов.

ЛИТЕРАТУРА

1. **Александров В.В., Сарычев В.А.** Цифровые программируемые технологии. // «Информационно-измерительные и управляющие системы», №11, т.8, 2010. С. 3–9.
2. **Александрова В.В., Зайцева А.А.** 3D технология и когнитивное программирование. // Информационно-измерительные и управляющие системы. 2012, т.10, №5. – С. 61–64.
3. **Александрова В.В., Зайцева А.А., Брысковский Г.В.** Цифровые программируемые 3D технологии // Материалы конференции «Информационные технологии в управлении» (ИТУ–2012) 09–11 октября 2012. С. 547–553
4. **Александрова В. В., Зайцева А. А., Тыжненко Д. А.** Сканирование и редактирование 3D-объекта для прототипирования на 3D-принтере // Информационно-измерительные и управляющие системы. 2013, т.11, №9. С.53–57.
5. **Кулешов С.В.** Формат представления реальных трехмерных сцен для объемного телевидения (True3D Vision) // Информационно-измерительные и управляющие системы. №4, т.7, 2009. С. 49–52.
6. 3D-сканирование. // [Электронный ресурс] — Доступ: <http://rostecps.ru/3d-skanirovanie>
7. [Электронный ресурс] — Доступ: <http://www.comprice.ru/articles/detail.php?ID=40134>
8. **Кулешов С.В.** Критерий оценки энергетической эффективности компрессии видеопотока. // Информационно-измерительные и управляющие системы. №11, т.8, 2010. С. 16–18
9. **Александров В.В., Поляков А.О., Лачинов В.М.** Рекурсивная алгоритмизация кривой, заполняющей многомерный интервал // Техническая кибернетика. № 1. 1978.
10. **Huffman, D.A.** A method for the construction of minimum redundancy codes. // In Proceedings IRE, vol. 40, 1962, pp. 1098–1101.
11. **Witten I., Neal R.M., Cleary G.** Arithmetic coding for data compression // Comm. ACM. – 1987, V.30 – No 6. pp. 520–540