

СОЗДАНИЕ ИНТЕРАКТИВНОЙ СИСТЕМЫ ВИЗУАЛИЗАЦИИ

Дусеев В.Р., Рудь М.Н., Мальчуков А.Н.
Томский политехнический университет
634050, Россия, г. Томск, пр-т Ленина, 30
E-mail: vagiz.d@gmail.com

Введение

В настоящее время одной из задач, решаемых информационным сообществом, является задача наглядной визуализации различных данных с использованием современных достижений науки и техники. Если в процессе получения новой информации человеком задействуется не только его слуховая и визуальная системы, но и кинестетическая, то её усвоение происходит гораздо продуктивнее. Одним из вариантов решения подобной задачи является создание таких технических систем, взаимодействие с которыми предоставляет человеку возможность получать информацию в интерактивном режиме, подключая к процессу перечисленные выше системы восприятия (рис. 1).

Создание интерактивной песочницы

Одна из технических систем подобной направленности была создана командой проекта «Интерактивные системы дополненной реальности» для музея занимательных наук г. Томска весной 2012 года. Данная установка представляет собой ёмкость, наполненную материалом (в данном случае, в качестве такого материала был выбран песок), на который проецируется изображение, генерируемое управляющей программой. На высоте порядка двух метров над ёмкостью расположена система, непосредственно создающая изображение на материале. Схема работы и описание составляющих системы следующая:

1. *Сенсор Kinect* – произведенный корпорацией Microsoft, имеет в своём составе датчик глубины, состоящий из инфракрасного излучателя и приёмника. В данной установке задача сенсора заключалась в определении расстояния до каждой точки поверхности материала в реальном времени, и составлении карты высот по этим данным. Эта информация по интерфейсу USB передаётся на компьютер.

2. *Компьютер* – осуществляющий обработку данных, полученных с Kinect. На основании карты высот, программа придает каждой точке поверхности цвет, соответствующий её высоте. Самые нижние точки приобретают цвет моря, точки, занимающие среднее положение – цвет равнин, самые высокие – цвет горных вершин (в реальной установке уровней намного больше [1]). Таким образом, изменяя расположение песка в области проекции, мы получаем ландшафт, соответствующий нашим действиям.

3. *Мультимедийный проектор* – непосредственно создающий изображение на песке.



Рис. 1. Система визуализации во время работы

Программная реализация

Алгоритм программы состоит из нескольких стадий:

1. Получение карты высот от сенсора Kinect.
2. Обработка и фильтрация карты высот.
3. Передача карты высот в графический процессор.
4. Построение трехмерной модели по данным из карты высот.
5. Раскрашивание каждой вершины в модели в соответствии с её значением высоты.
6. Проецирование полученного изображения на песок.

Прототип алгоритма был реализован с помощью фреймворка XNA [2] на языке C#. Результат испытаний данного алгоритма на стенде представлен на рисунке 1. Выбор фреймворка XNA позволил в короткие сроки реализовать алгоритм, но низкая производительность фреймворка при работе с 3D моделями, изменяемыми в реальном времени, не позволила остановиться на этом решении.

Таким образом, для реализации были выбраны язык программирования C++ и платформа Microsoft Windows. В качестве готовых инструментов использовались такие программные библиотеки как *freeglut* [3], реализующая управление графикой OpenGL, создание и управление окном и устройствами ввода [4], и *OpenNI* [5], осуществляющая подключение и управление сенсором Kinect. Выбор в пользу описанных компонент, платформ и языков обоснован в первую очередь соотношением простоты разработки и производительности готового решения.

Испытания построенной системы визуализации выявили необходимость добавления дополнительной стадии пост-обработки изображения в алгоритм работы программы.

Данный этап устраняет несовпадение границ реального проецируемого изображения с границами изображения, получаемого сенсором Kinect.

В стадии пост-обработки границы проецируемого изображения растягиваются до момента совпадения их с границами изображения с Kinect. Для этого на проецируемое изображение накладывается искажающая сетка (рис. 2).

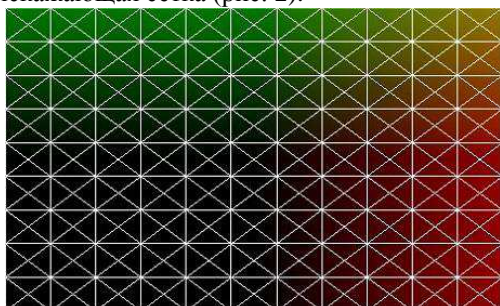


Рис. 2. Искажающая сетка, наложенная на проецируемое изображение

Крайние вершины искажающей сетки утягиваются к границам изображения с сенсора, а соседние с ними вершины «следуют» за крайними по линейному коэффициенту, зависящему от расстояния до удаляющейся вершины (рис. 3).

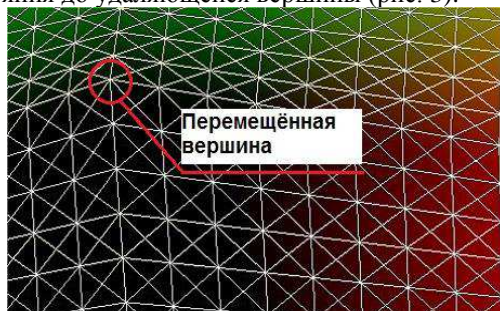


Рис. 3. Одна из вершин смещена. Остальные вершины следуют за ней по линейному коэффициенту

Результат работы данного этапа алгоритма при запуске на испытательном стенде системы интерактивной визуализации представлен на рисунке 4.

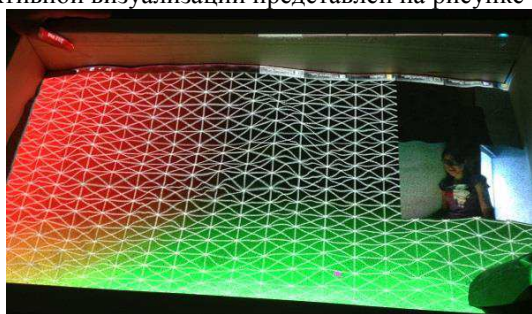


Рис. 4. Пост-процессинг вершин проецируемого изображения

Фильтрация данных с сенсора Kinect

Получаемые от сенсора Kinect данные зачастую плохо пригодны к использованию в обычных алгоритмах обработки данных. В ситуации создания интерактивной системы визуализации ландшафта – требования к стабильности и сглаженности полученных данных выходят на первое место. Однако изображения, получаемые от сенсора Kinect, такими качествами не обладают. На рисун-

ке 5 представлено изображение, полученное с сенсора без предварительной обработки.



Рис. 5. Изображение с сенсора Kinect без обработки

Зашумленные области (белые участки на рисунке 5), скрыты от инфракрасной камеры или поглощают излучение IR проектора.

Чтобы устранить нестабильность и мерцание в результате такой зашумленности была реализована группа программных фильтров, обрабатывающих данные сенсора Kinect. В основу группы фильтров положено совместное использование алгоритмов закрашивания и сглаживания. Результат применения реализованных алгоритмов представлен на рисунке 6.



Рис. 6. Отфильтрованное изображение с сенсора

Заключение

Данная установка имеет, в большей степени, развлекательную и обучающую направленность, поскольку несёт в себе возможности создания различных обучающих программ, которые могли бы использовать метаморфозы ландшафта.

Литература

1. StudyAndPlay.ru [Электронный ресурс] – Режим доступа: <http://www.studyandplay.ru/>, свободный.
2. XNA Game Studio [Электронный ресурс] – Режим доступа: <http://msdn.microsoft.com/en-us/aa937791.aspx>, свободный.
3. OpenGL Shading language: third edition, Dave Shreiner, изд. Addison Wesley, 2009 – 804 с.
4. OpenGL Red book, изд. Addison-Wesley; 5th edition (August 11, 2005), – 896 с <http://www.opengl.org.ru/>, свободный.
5. OpenNI SDK [Электронный ресурс] – Режим доступа: <http://www.openni.org/openni-sdk/>, свободный.