

УДК 004.946

РАЗРАБОТКА ПОДСИСТЕМЫ ИНТЕРАКТИВНОГО ВЗАИМОДЕЙСТВИЯ В СОСТАВЕ ТРЕНАЖЕРНОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМА ICP РЕГИСТРАЦИИ ОБЛАКОВ ТОЧЕК И ПЛАТФОРМЫ MICROSOFT KINECT

Бабков В.С., Соболев Е.Г.

*Донецкий национальный технический университет
кафедра прикладной математики и информатики
yevgeniysobolev@gmail.com*

Статья описывает подсистему интерактивного взаимодействия пользователя с виртуальной средой в составе тренажерного обучающего комплекса с использованием платформы Microsoft Kinect. Рассмотрены существующие системы интерактивного взаимодействия реального и виртуального пространств. Разработана независимая система распознавания жестов на основе платформы Microsoft Kinect. Рассмотрены методы построения виртуальных моделей с помощью трехмерного сканирования и, в частности, алгоритм ICP регистрации облаков точек.

Введение

В современном мире автоматизация производства затрагивает почти все сферы человеческой деятельности. В связи с этим весьма актуальными являются тренажерные обучающие системы, которые позволяют не только ускорить подготовку соответствующего персонала, но и помогают в моделировании и отработке, как обычных рутинных действий, так и различных аварийных ситуаций.

Существует два принципиально разных способов построения таких систем. Первый – это воссоздание рабочей обстановки персонала в реальности (различные управляемые механизмы, пульта и пр.). Данный подход далеко не всегда является успешным, так как весьма трудно точно воссоздать реальное рабочее место сотрудника. Второй способ – создание рабочей обстановки персонала в виртуальной реальности является более эффективным, так как позволяет при меньших затратах получать более детальную рабочую обстановку и большую свободу для отработки различных действий.

Таким образом, применение современных платформ виртуальной реальности является перспективным и актуальным направлением в области построения тренажерных обучающих комплексов. В данной работе описывается разработанная подсистема интерактивного взаимодействия пользователя с виртуальной средой в составе тренажерного комплекса «Виртуальная шахта» с использованием платформы Microsoft Kinect.

Общая постановка проблемы

На сегодняшний день основной проблемой при построении систем виртуальной реальности является создание интерфейса интерактивного взаимодействия человека

и виртуальной среды. До появления платформы Microsoft Kinect [1] данная проблема решалась путем облачения пользователя во всевозможные сенсоры (специальные костюмы, очки, перчатки и т.п.), что не всегда является удобным. Именно избавление от данной необходимости и стало главной особенностью Kinect (рис. 1).



Рисунок 1. Контроллер Microsoft Kinect

Microsoft Kinect – игровой «контроллер без контроллера» первоначально был представлен для консоли Xbox 360 в ноябре 2010 года. Устройство оказалось очень удачным – в течение первых трёх месяцев после запуска было продано более 10 миллионов систем Kinect для Xbox 360 [2].

Kinect позволяет пользователю взаимодействовать с виртуальной средой без помощи игрового контроллера через устные команды, позы тела и показываемые объекты или рисунки.

Контроллер Kinect возможно также использовать в качестве трехмерного сканера. В то время как камеры глубины не являются концептуально новыми, Kinect сделал такие устройства доступными для всех. Качество глубины сканирования, учитывая низкую стоимость и характер устройства, являются существенным преимуществом, и делают датчик чрезвычайно популярным среди ученых и энтузиастов.

Архитектура тренажерной системы

На данный момент система состоит из следующих разработанных подсистем:

- модуль интерактивного взаимодействия «человек – система»
- описания 3D моделей (скелетов, текстур, артефактов)
- серверное хранение подсистемы 3D моделей
- клиентская подсистема для визуализации модели

Работа системы выглядит следующим образом:

- пользователь перед экраном выполняет необходимые действия;
- контроллер Kinect распознает человека, строит его скелет и отслеживает все его действия. Затем полученные данные передаются на промежуточное программное обеспечение;
- промежуточное программное обеспечение преобразует язык жестов тела пользователя в высокоуровневые команды (шаг вперед – движение вперед, поворот корпуса – поворот камеры в виртуальном мире и т.д.);
- клиент воспроизводит сцену с человеком, и соответственно выполняет все перемещения, повороты и т.д.;
- за визуализацию сцен и взаимодействие сценариев отвечает сервер.

Подсистема интерактивного взаимодействия

Главной задачей при разработке подсистемы интерактивного взаимодействия пользователя с виртуальной средой было обеспечение нативных, интуитивно понятных и удобных движений для управления аватаром в виртуальном пространстве.

На данный момент система реализует такой набор команд управления:

- перемещение ноги вперед – перемещение вперед;
- перемещение ноги назад – перемещение назад;
- перемещение ноги вправо – перемещение вправо;
- перемещение ноги влево – перемещение влево;
- наклон туловища вперед – поворот камеры вниз;
- наклон туловища назад – поворот камеры вверх;
- поворот туловища влево – поворот камеры влево;
- поворот туловища вправо – поворот камеры вправо;
- поднятие левой руки выше головы – переход (выход) в (из) режим от первого лица.

Стоит отметить, что система позволяет осуществлять повороты и перемещения как поэтапно (на определенное значение), так и постоянно (пока пользователь не вернется в исходную позицию). Принцип работы подсистемы приведен на рис. 2.

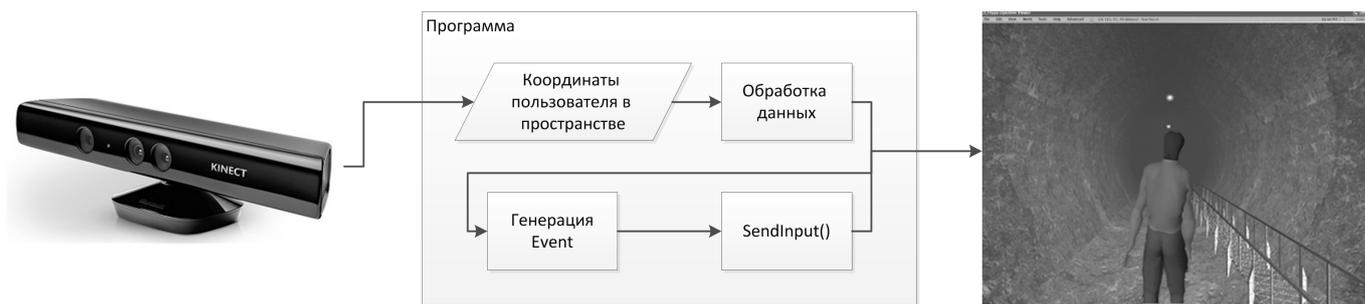


Рисунок 2. Структура подсистемы интерактивного взаимодействия

При любом перемещении пользователя Kinect генерирует новый фрейм с указанием текущих координат отслеживаемых частей тела. Разработанная подсистема на основе этих данных вычисляет углы наклона рук, ног, туловища и далее, сопоставляя текущие значения со значениями на предыдущем фрейме, либо генерирует событие о том, что произошло одно из определенных движений либо нет. Также, стоит заметить, что пороговые значения срабатывания движений являются настраиваемыми, что делают систему комфортной для любого пользователя.

При запуске программы создается слушатель, который обрабатывает сгенерированные события: в зависимости от типа события в систему посылаются соответствующие команды (нажатие клавиш клавиатуры, перемещения мыши) с помощью метода `SendInput()`. Данный подход является очень гибким и универсальным, так как позволяет передавать команды в любое активное приложение. Иными словами разработанную подсистему интерактивного взаимодействия можно использовать для управления любым приложением всего лишь изменяя положение частей тела пользователя.

Использование Kinect в качестве трехмерного сканера

Следующим шагом, который в полной мере раскрывает огромный потенциал использования Kinect, является применения его в качестве трехмерного сканера. Это позволяет, например, создавать свои собственные аватары на основе объектов реального мира.

Камера Kinect создает в реальном времени карты глубины, содержащие дискретные измерения диапазона физической сцены. Эти данные могут быть отражены как набор дискретных 3D-точек (или облака точек). Процесс обработки данных для построения трехмерной модели представлен на рис. 3.



Рисунок 3. Схема процесса обработки данных

Kinect создает два типа изображения: RGB изображения и карты глубины. Каждый пиксель в карте глубины дает 3D положение точки в сцене, в то время как RGB-изображение содержит соответствующую информацию о текстурах.

Облако точек может быть построено с помощью следующего процесса.

Пиксель в глубине изображение проецируется в 3D пространстве, используя внутренние параметры камеры глубины. Это дает 3D положение точки в системе координат камеры глубины.

Далее, используя преобразование системы координат между камерами глубины и RGB, определяется соответствующая точка в системе координат камеры RGB. Наконец, преобразованная точка в системе координат камеры RGB отображается на изображение RGB, чтобы получить цвет точки. Созданные облака точек должны быть отфильтрованы, согласно процессу регистрации.

Регистрация облаков точек является процессом приведения двух облаков точек с перекрытием областей к одному облаку, благодаря определению систем координат преобразования между ними. Для регистрации облаков точек может быть использован итеративный алгоритм ближайших точек как наиболее распространенный алгоритм для регистрации облаков точек.

Итеративный алгоритм ближайших точек (Iterative Closest Point - ICP) - алгоритм, используемый для сведения к минимуму разницы между двумя облаками точек. ICP часто используется для восстановления двумерных (2D) или трехмерных (3D) поверхностей с разных фреймов сканирования [9].

Алгоритм концептуально простой и часто используется в режиме реального времени. Он итеративно применяет преобразования (сдвиг, вращение), которые необходимы для сведения к минимуму расстояния между точками из двух необработанных фреймов сканирования.

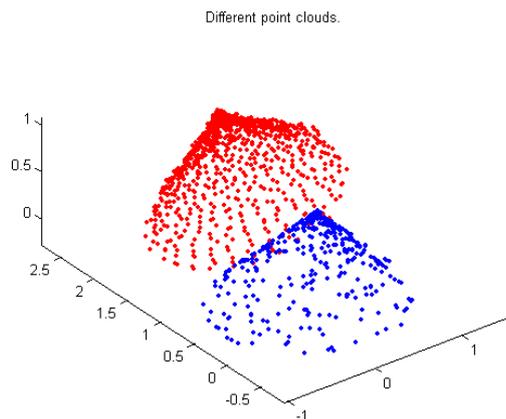


Рисунок 4. Облака точек

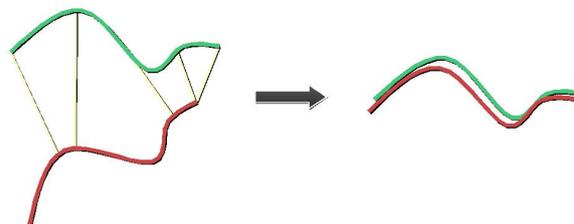


Рисунок 5. Результат работы ICP алгоритма

Входами алгоритма есть точки из двух необработанных сканов, первичная оценка трансформации, критерии для остановки итерации.

Результат: совершенное преобразование.



Рисунок 6. Алгоритм ICP

Выводы

В статье рассмотрена подсистема интерактивного взаимодействия пользователя в составе тренажерной обучающей системы «Виртуальная шахта». Для реализации данной подсистемы была выбрана платформа Microsoft Kinect. Рассмотрен алгоритм ICP регистрации облаков точек как наиболее универсальный для построения трехмерной модели объекта. Разработанная версия предоставляет интерактивное бесконтактное взаимодействие и визуализацию исполняемых жестов. В процессе проектирования и разработки была изучена библиотека Microsoft Kinect for Windows SDK для взаимодействия с контроллером. В дальнейшем предполагается увеличение числа распознаваемых жестов, улучшение их качества и усложнением сценариев симуляции.

Список источников

- [1] Kinect for Windows [Electronic resource] / Интернет-ресурс. – Режим доступа: <http://www.microsoft.com/en-us/kinectforwindows/>
- [2] Kinect Sales Surpass Ten Million. [Электронный ресурс]. Режим доступа: <http://www.xbox.com/en-US/Press/archive/2011/0308-Ten-Million-Kinects>
- [3] Kinect for Windows SDK [Electronic resource] / Интернет-ресурс. - Режим доступа: www.microsoft.com/en-us/kinectforwindows/develop/overview.aspx
- [4] Kinect for Windows Programming Guide [Electronic resource] / Интернет-ресурс. – Режим доступа : msdn.microsoft.com/en-us/library/hh855348.aspx
- [5] В.С. Бабков, Н.В. Гузий, А.С. Подлинный. Разработка тренажерного комплекса «виртуальная шахта» на базе платформы Microsoft Kinect. Материалы 4-й международной научно-технической конференции “Моделирование и компьютерная графика–2011”. Донецк, 5-8 октября 2011 г. С. 30-34.
- [6] Kinect for Windows Blog [Electronic resource] / Интернет-ресурс. - Режим доступа: <http://blogs.msdn.com/b/kinectforwindows/>
- [7] ICP алгоритм – [Electronic resource] / Интернет-ресурс. - Режим доступа: http://en.wikipedia.org/wiki/Iterative_closest_point
- [8] Использование ICP алгоритма и PCL – [Electronic resource] / Интернет-ресурс. – Режим доступа: http://pointclouds.org/documentation/tutorials/iterative_closest_point.php