

Использование нескольких датчиков оптического трекинга Leap Motion для построения моделей в среде виртуального окружения*

В. А. Киселев, В. А. Пигарёв, М. А. Хламов, К. В. Чувилин

{ kiselev.va | vladislav.pigarev | khlamov | kirill.chuvilin }@phystech.ru

Московский физико-технический институт (Государственный университет)

В работе рассматривается задача организации человеко-машинного взаимодействия посредством оптического трекинга жестов. Показываются возможности датчиков Leap Motion, и объясняется потребность в одновременном использовании нескольких датчиков одновременно. Исследуется возможность реализации такого подхода, которую не предоставляют стандартные средства разработки. Предлагается система синхронизации и обмена данными, получаемыми с разных датчиков, и структура для хранения и обработки таких данных. Приводятся результаты применения разработанного ПО, реализующего предложенные подходы, а также описываются трудности, возникшие в процессе его создания.

Ключевые слова: визуализация, виртуальная реальность, интерфейс пользователя, оптический трекинг, человеко-машинное взаимодействие, Leap Motion, Unity3D.

The use of multiple Leap Motion optical tracking sensors to construct models in a virtual environment*

Kiselev V. A., Pigaryov V. A., Khlamov M. A., Chuvilin K. V.

Moscow Institute of Physics and Technology (State University)

The research deals with the task of organizing human-machine interaction using optical tracking of gestures. The capabilities of the Leap Motion sensors are shown, and the need to simultaneously use multiple sensors at the same time is explained. The possibility of implementing such an approach, not provided by standard SDK, is studied. A system for synchronizing and exchanging data, obtained from different sensors, and a structure for storing and processing such data are proposed. The applications of the developed software, implementing the proposed approaches, are given. The difficulties encountered in the process of its creation are described.

Keywords: human-machine interaction, Leap Motion, optical tracking, Unity3d, user interface, UX, virtual reality, VR, visualization.

Введение

Одной из наиболее актуальных задач при использовании систем виртуальной или дополненной реальности является организация человеко-машинного взаимодействия. Это подразумевает, в том числе, возможность передачи команд пользователями контроллеру. Наиболее естественным способом взаимодействия является отслеживание жестов и действий пользователей, для чего используются устройства оптического трекинга [1, 2].

Изучаемые нами задачи связаны с анализом жестов и других движений пользователей [3, 4, 5]. На основе проведенного анализа для использования в исследованиях при анализе мелкой моторики был выбран датчик человеко-машинного взаимодействия,

использующий технологию Leap Motion [6], которая позволяет передавать положение рук пользователя в виртуальное окружение.

Рынок датчиков такого типа довольно ограничен. Поэтому на практике происходит следующее: коммерческие компании производят аппаратное обеспечение, а сторонние исследователи предлагают программные решения на их основе. Например, инструментарий, предложенный французскими инженерами из Университета Падуйи [7] анализирует данные с устройств разработанных конкурирующими компаниями: Kinect (Microsoft) и Leap Motion. Их подход, в том числе, использует метод опорных векторов и позволяет получить более информативное описание руки, превосходящее по своим характеристикам штатный программный инструментарий, предлагаемый обеими компаниями. Сообще-

Работа выполнена и опубликована при финансовой поддержке РФФИ, гранты 17-57-52011, 16-07-01267, 18-07-20102.

Международная конференция «Ситуационные центры и информационно-аналитические системы класса 4i для задач мониторинга и безопасности» (SCVRT2018), ЦарьГрад, 21-23 ноября 2018 гг.

International Conference on Situational Centers and Information-Analytical System 4i Class for Monitoring and Security Tasks, November 21-23, 2018, Tzargrad

ство исследователей во главе с командой из южно-калифорнийского института креативных технологий развивает программное решение [8], облегчающее интеграцию датчиков глубины, совместимых с OpenNI [9], с приложениями виртуальной реальности. В настоящее время поддерживаются датчики PrimeSensor и Microsoft Kinect. В работе [10] предлагаются различные 3D-интерфейсы для облегчения работы с анимацией. Они сочетают работы с мышью и устройством Leap Motion, которое обеспечивает взаимодействие в трёх измерениях, но, как правило, обеспечивают меньшую точность. В той же работе приведено исследование удобства использования, которое показало, что пользователи предпочитают Leap Motion компьютерной мыши, поскольку предлагаемый жестовый интерфейс существенно уменьшает количество необходимых операций и время, необходимое на реализацию задач, особенно для начинающих пользователей.



Рис. 1. Устройство датчика Leap Motion

Датчик Leap Motion представляет из себя устройство размера 13мм × 13мм × 76мм, включающее несколько инфракрасных светодиодов, которые освещают рабочую область, и две камеры для получения изображения (см. рис. 1). Сопровождающее программное обеспечение анализирует отражённый волновой фронт и предоставляет API, позволяющее получить характеристики положения пальцев, ладоней и запястий рук пользователя.

Стандартный функционал позволяет распознавать некоторые жесты, такие как пе-

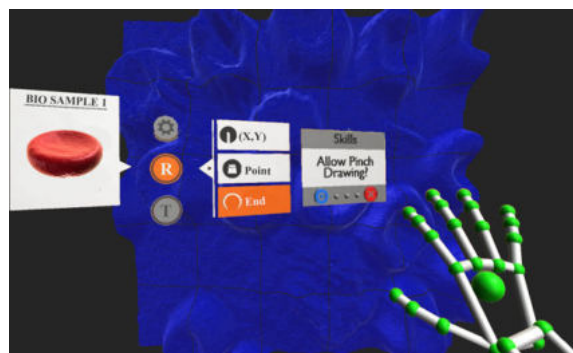


Рис. 2. Использование данных датчика Leap Motion для управления виртуальной сценой.

ремещения, смахивания, щипки и вращения. Это позволяет закреплять за разными жестами определённый функционал. Но для реализации более гибкого взаимодействия требуется анализировать движения отдельных суставов. API позволяет это делать, предоставляя информацию о некоторых точках ладони и запястья, включающую их положение в пространстве, скорость и ускорение в каждый момент времени. Частота обновления данных контроллера варьируется от 20 до 200 герц, в зависимости от настроек пользователя и доступной вычислительной мощности ПК, с которым работает датчик. Эта информация оказывается достаточно, например, при распознавании статических жестов для управления просмотром данных, получаемых методами цифровой голографической микроскопии [11]: жесты позволяют перемещать точку наблюдения в пространстве, выбирать направление угла обзора, масштабировать модели микрообъекта и измерять расстояния между точками, изменять кадрирование и детализацию, управлять освещением в сцене, взаимодействовать с виртуальным меню (см. рис. 2). Таким образом, разработчикам предоставляется возможность создавать человеко-машинные интерфейсы, позволяющие пользователям наиболее естественным образом управлять процессами в виртуальной сцене. Более того, это может быть использовано и в «обычных» задачах взаимодействия с компьютером [12].

Однако использование одиночных датчиков оптического трекинга имеет ряд недостатков. Главный из них заключается в том, что принципиально устройство основывается на визуальном контакте с объектом захвата, то есть запястья, ладони и все пальцы должны быть в непосредственной видимости для камер датчика. На практике часто пальцы перекрывают друг друга, а иногда и вовсе одна рука полностью заслоняет другую. Другим недостатком является нестабильность контроллера Leap Motion при длительной работе в условиях комнатной температуры: после двух часов работы, датчик нагревается, что приводит к тому, что частота получаемых данных уменьшается, а уровень зашумлённости данных увеличивается. Эти же проблемы относятся не только к Leap Motion, но и аналогичным устройствам для оптического трекинга.

Сопровождающее программное обеспечение реализует некоторый прогноз расположения невидимых фаланг пальцев, однако иногда этого недостаточно. Предсказанные координаты захватываемых объектов, вообще говоря, отличаются от истинных, что может повлечь неудобства в работе, а иногда и невозможность выполнения действий, требующих повышенной точности.

В случае с человеческим зрением, мы получаем лучшее и более точное представление об объекте, осмотрев его с разных сторон. Аналогичный подход удобно использовать и для решения рассматриваемой проблемы. Установив несколько устройств Leap Motion и направив их под разными углами на рабочую область, можно получить больше данных для большей определённости. Так, например, используя два датчика, нацеленных в одну и ту же область под ортогональными направлениями, можно непрерывно регистрировать одну кисть руки почти под любым углом её расположения: если один датчик видит руку, повернутую ребром, то второй увидит, например, ладонь.

Возможности, которые может предоставить такой программно-аппаратный комплекс:

- создание надёжной системы из устройств — при выходе одного устройства из строя или появления помехи остальные будут работать;
- увеличение точности регистрируемых данных;
- расширение рабочей области.

Увы, производители устройств не предоставляют штатных средств одновременного использования нескольких датчиков. Поэтому и появилось данное исследование, целью которого является реализация программно-аппаратной системы, позволяющей это делать на основе предоставляемого API.

Постановка задачи

Для взаимодействия с API, предоставляемым Leap Motion SDK, мы используем движок и среду разработки Unity3D [13], описывая логику на языке программирования C#.

SDK устроен так, что одновременно можно использовать только один датчик. Поэтому мы исходим из предположения, что к каждому датчику подключается свой экземпляр ПО, работающий в отдельном окружении.

Таким образом, требуется:

1. разработать структуру данных, которая позволяла бы хранить и обрабатывать информацию, приходящую с нескольких датчиков оптического трекинга;
2. разработать систему синхронизации данных, получаемых с разных сенсоров в разных окружения;
3. разработать и реализовать программный комплекс, позволяющий получать и использовать данные оптического трекинга с нескольких датчиков.

При синхронизации данных в данной работе мы предполагаем, что положения устройств известны и не меняются со временем. Это позволяет получать абсолютные координаты на каждом устройстве.

Работа с данными датчиков в среде Unity3D

В текущей реализации, применяемой для исследований, для хранения данных используются структуры и сущности Unity 3D. Важным понятием является *префаб* — шаблон, позволяющий создавать в сцене копии сложных объектов, структура которых описывается заранее.

Каждый подключённый датчика в системе представляется объектом, являющимся экземпляром префаба *Device*, представляющего из себя совокупность объектов и скриптов, которые отвечают за ввод информации об объектах, регистрируемых подключённым устройством, в виртуальную среду. За основу этого префаба взят префаб, поставляемый Leap Motion, который при помощи библиотек Leap Motion обрабатывает данные, полученные с камеры, строит структуру объектов, описывающую скелет кисти руки, и даёт пользователю графическое представление о физическом состоянии руки.

Так как объекты префаба *Device* предоставляют информацию лишь одного устройства, за которым они закреплены, то соответствующую модель виртуальной кисти не требуется отображать, поскольку она является промежуточным этапом обработки данных. Поэтому из оригинального префаба убрана визуальная составляющая. По аналогичным причинам из него удалены колайдеры (компоненты, которые обозначают формы объектов для физического движка) и твёрдые тела (компоненты, которые определяют поведение объекта для физического движка).

Для обработки и передачи полученной с датчика информации к префабу *Device* добавлен скрипт *Connector*. Он находит в сцене объект, содержащий в себе скрипт *TotalDataHandler*, обрабатывающий данные, полученные от всех устройств, и в который передаётся информация о новом подключённом датчике.

Совокупность объектов, представляющая кисть руки, регистрируемую устройством,

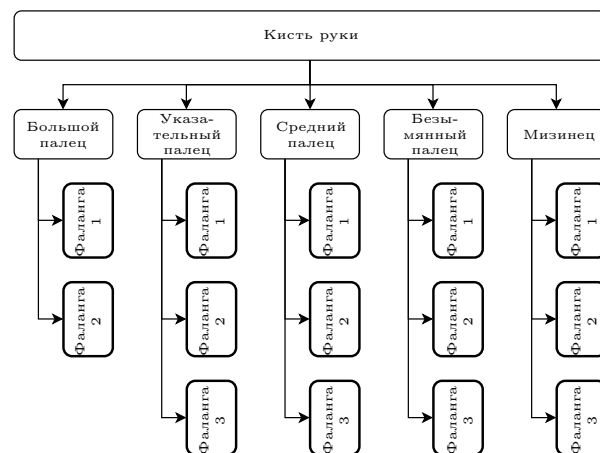


Рис. 3. Представление кисти руки в виде древовидного графа.

имеет структуру древовидного графа (см. рис. 3). Корневым узлом является объект, хранящий в себе скрипт, отвечающий за руку целиком. В нём содержится информация о том, какая рука регистрируется (правая или левая), указатели на объекты ладони и предплечья, на объекты их точек привязок и на объекты пальцев.

Объекты пальцев являются дочерними для объекта кисти руки и хранят в себе скрипты, отвечающие за представление пальца в целом, в них содержится информация о том, какой именно палец они представляют, а также указатели на объекты соответствующих фаланг и объекты точек их привязки, несущие логический смысл физических суставов.

Третий уровень дерева заполняют объекты фаланг пальцев. Именно эти объекты несут ключевую информацию, а именно: позиции и повороты костей регистрируемой кисти руки. Эту информация, в основном, и используется для дальнейшей обработки. Для удобства значения позиции и поворота хранятся в относительных координатах.

При обращении скрипта *Connector* к скрипту *TotalDataHandler* передаются указатели на объекты фаланг. Для более корректного построения модели мы используем и другие объекты кисти руки, поэтому указатели на них также передаются. Упорядоченный набор таких указателей для

каждого устройства входит как отдельный массив в список `Sources`.

Перед каждым кадром (новым набором данных с датчиков) скрипт `TotalDataHandler` проходит по всем массивам из списка `Sources` и совмещает данные в соответствии с выбранной стратегией. На данный момент реализована простейшая стратегия, по которой относительные позиции и повороты объектов усредняются. Однако на этом этапе допустима реализация и более сложных алгоритмов, включающих, в том числе, фильтрацию данных.

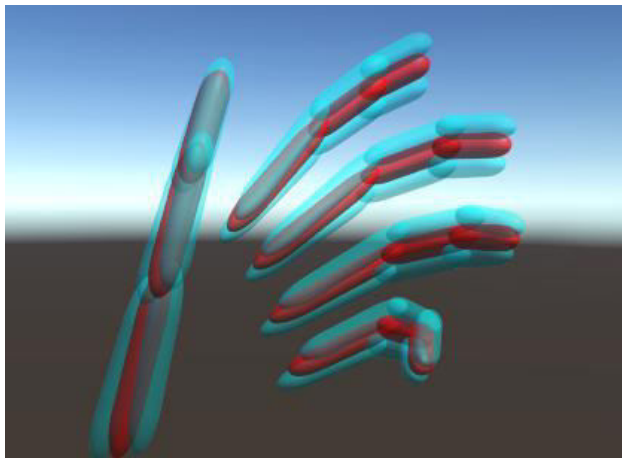


Рис. 4. Пример построенной модели кисти руки. Голубым цветом изображены данные двух отдельных устройств. Красным цветом представлены усреднённые данные.

Результат обработки записывается в структуру `FinalModel`, схожую с теми, которые изначально приходят с датчиков. Эта конструкция является конечным результатом для кадра, содержащим как физические составляющие, так и объекты визуализации (см. рис. 4).

Сбор данных с нескольких датчиков

В текущей реализации каждый датчик подключается к отдельному компьютеру, а синхронизация производится по TCP/IP. Для этого в ПО внедрены необходимые конструкции.

Во-первых, реализован объект `NetworkObject`, содержащий в себе базовые сетевые компоненты Unity: `NetworkManager` (управляет сетевой частью) и `NetworkManagerHUD` (со-

здаёт простейший пользовательский интерфейс для создания сервера или подключения к нему по IP-адресу).

Во-вторых, объекты, которые создаются из префаба `Device`, создаются на компьютерах, непосредственно связанных с датчиками, а не сразу на одной машине. Созданные объекты передаются по сети благодаря компоненту `Unity NetworkIdentity` (сетевой идентификатор объекта) и реализованному скрипту `NetworkSkeleton`, который обеспечивает синхронизацию данных между устройствами в сети. В частности, *центральный компьютер*, отвечающий за обработку данных всех датчиков, получает всю информацию, которая необходима скрипту `TotalDataHandler`. Наблюдатели также могут получать данные со всех устройств, что может быть полезно в целях научной визуализации.

В-третьих, структура объектов `FinalModel` тоже наделяется свойствами, позволяющие передавать ей по сети. Это сделано для того, чтобы центральный компьютер мог отправлять результаты обработки наблюдателям в формате, аналогичном тому, который принимает сам.

В итоге мы получили программную систему, способную обрабатывать информацию с нескольких устройств Leap Motion и передавать результаты внешним наблюдателям (см. рис. 5).

Выводы

В работе рассматривается задача, связанная с повышением точности результатов оптического трекинга за счёт использования нескольких устройств. Её актуальность обусловлена общим интересом к системам виртуальных окружений и способам управления, а текущий рынок устройств позволяет использовать их как в практических целях, так и для развлечений.

В ходе исследования предложены новые подходы и реализован прототип системы, которая позволяет одновременно использовать данные с нескольких устройств Leap Motion. Разработанное приложение следует запускать на нескольких компьютерах,

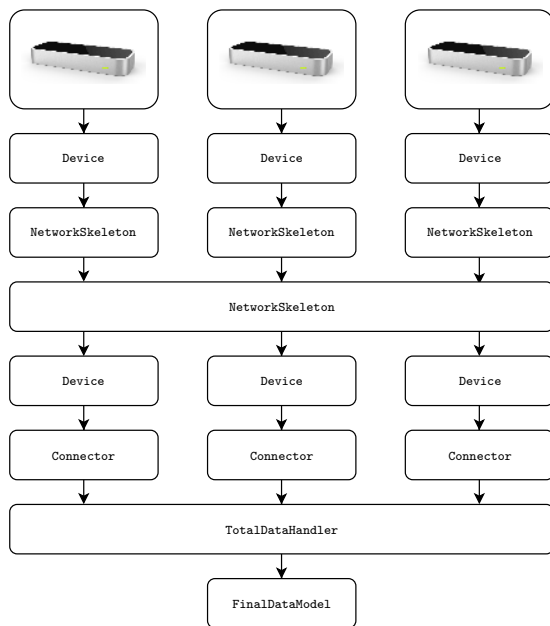


Рис. 5. Схема обмена данными.

к каждому из которых подключен датчик, и которые соединены в сеть. Датчики следует направить примерно в одном направлении, а центральному компьютеру, получающему данные со всех устройств, сообщить положение и направление.

В текущей реализации получается и усредняется информация о положениях фаланг пальцев левой руки. Однако предусматривается возможность использовать более сложные алгоритмы, что должно, в том числе, позволить повысить устойчивость системы к ошибкам работы отдельных датчиков.

При разработке программной системы был обнаружен ряд технических ограничений API, предоставляемых производителями устройств. Основным из них является то, что программное обеспечение Leap Motion не только не позволяет обрабатывать информацию с нескольких устройств в одном проекте, но и не поддерживает запуск нескольких камер на одном компьютере. В текущей реализации мы используем несколько компьютеров, передавая данные по сети. Но такая структура критически усложняет масштабируемость системы. Альтернативно можно использовать виртуальные машины, но это только в несколь-

ко раз уменьшит количество физических машин, одновременно увеличив суммарные требования к вычислительным мощностям. При этом исходная проблема останется не решена. Поэтому требуются модификации предоставляемого SDK, в том числе, быть может, внесение собственных правок.

Планируется продолжить исследование, чтобы избавиться от указанных проблем. Результатом программной реализации станет пакет в формате UnityPackage, который можно будет импортировать в любой проект Unity 3D, чтобы получить в нём поддержку работы с несколькими устройствами Leap Motion.

Литература

- [1] Киселев В. А. и др. Современные устройства трекинга для систем виртуального окружения // Труды Международной научной конференции по физико-технической информатике (СРТ1617). Москва–Протвино, 2017.
- [2] Хворов Н. С., Чувильин К. В. Трекинг мелкой моторики в реальном времени // Труды Международной научной конференции СРТ2015. Протвино: АНО ИФТИ. — 2016. — С. 237–240.
- [3] Каленков С. Г. и др. Визуализация цифровых голограмм биообъектов в среде виртуального окружения // Труды Международной научной конференции «Ситуационные центры и ИАС4i для мониторинга и безопасности» SCVRT1516, 21–24 ноября 2015–2016 г. Протвино–Москва, 2016: Изд. ИФТИ. — С. 291–294.
- [4] Киселев В. А., Пигарёв В. А. Мультиинструментальный интерфейс для работы с цифровыми голограммами в виртуальном окружении (виртуальной реальности) // Труды Международной научной конференции по физико-технической информатике СРТ2018, Москва–Протвино, 2018: Изд. ИФТИ. — С. 73–78.
- [5] Хламов М. А., Чувильин К. В. Использование системы трекинга лыжного тренажёра для определения уровня спортсмена // Труды Международной научной конференции СРТ2015. Протвино: АНО ИФТИ. — 2016. — С. 233–236.
- [6] Leap Motion Developer. URL: <https://developer.leapmotion.com>
- [7] M. Giulio, F. Dominio, P. Hand gesture recognition with leap motion and kinect devices // Image Processing (ICIP), 2014 IEEE International Conference on. IEEE. — 2014. — doi: 10.1109/ICIP.2014.7025313.
- [8] Flexible Action and Articulated Skeleton Toolkit (FAAST). URL: <http://projects.ict.usc.edu/mxr/faast/>

- [9] OpenNI 2 Downloads and Documentation | The Structure Sensor. URL: <https://structure.io/openni>
- [10] *Kyto M., et al.* Improving 3D character posing with a gestural interface // IEEE Computer Graphics and Applications. — 2017 Jan-Feb. — Pp. 70–78. — doi: 10.1109/MCG.2015.117.
- [11] *Киселев В. А., Клименко С. В., Михайлюк М. В.* Исследование и разработка жестового интерфейса для работы с инструментом визуализации цифровых голограмм в виртуальном окружении // Труды Международной научной конференции по физико-технической информатике (СРТ1617). Москва-Протвино, 2017
- [12] Introducing the Leap Motion. URL: https://www.youtube.com/watch?v=_d6KuiuteIA
- [13] Unity. URL: <https://unity3d.com/ru>