

Использование языка жестов для манипуляций с трёхмерными объектами в системах научной визуализации

А.В. Зырянов

Перед разработчиком методов манипуляции трёхмерными объектами в системах научной визуализации возникают две проблемы. Во-первых, целью визуализации довольно часто является исследование ранее не изученного объекта, т.е. невозможно заранее сказать, какие взаимодействия потребуются. Во-вторых, для проведения исследований важно, чтобы взаимодействия с виртуальными объектами выполнялись естественным образом, т.е. так, как если бы работа велась с их реальными аналогами. Предлагаемый автором метод позволяет эффективно решать обе эти проблемы. В данной статье описывается простая и удобная технология ввода и распознавания жестов, а также рассматривается легко расширяемый язык жестов, предназначенный для осуществления основных взаимодействий с трёхмерными объектами.

1. Введение

Развитие вычислительных систем позволяют проводить моделирование сложных физических процессов со всё большей точностью. Однако увеличение объема вычисляемых данных приводит к усложнению обработки полученных сведений. Научная визуализация позволяет представить результаты в виде более удобных для анализа трёхмерных объектов, но вместе с тем порождает проблему взаимодействия пользователя с виртуальной средой. «Хороший» интерфейс должен быть как можно прозрачнее, т.е. пользователь не должен задумываться, как совершить то или иное действие. Это означает, что нужно создать такой интерфейс, в котором манипуляции с устройством ввода в точности соответствуют воздействию на виртуальный объект (к примеру, поворот манипулятора приводит к аналогичному повороту объекта). Для пользователя данный подход идеален, однако перед разработчиком интерфейса он ставит две проблемы: выбор манипуляций с объектами, которые можно будет совершать, и разработка устройства ввода удобного для осуществления выбранных взаимодействий.

2. Обзор существующих решений

«Хороший» интерфейс должен удовлетворять следующим требованиям:

1. Позволять осуществлять все манипуляции с виртуальным объектом точно так же, как и если бы этот объект находился у пользователя в руках.
2. Не требовать для осуществления различных действий использовать различные манипуляторы.
3. Быть простым с технической точки зрения (простота установки устройства ввода и лёгкость его повседневного использования) и, по возможности, не дорогим.

Из первого требования непосредственно вытекает, что создаваемый способ ввода должен быть трёхмерным. Соответственно и техническое устройство, осуществляющее подобный ввод должно быть трёхмерным.

Трёхмерные манипуляторы существуют и достаточно разнообразны. В качестве примера рассмотрим GlobeFish [1]. Данное устройство представляет собой шар, который можно свободно вращать вокруг любой оси. Также, толкая шар с той или иной стороны, пользователь может осуществить ввод движения. Таким образом, GlobeFish является устройством ввода с шестью степенями свободы (три оси вращения и три оси перемещения). С помощью данного устройства очень удобно вращать виртуальный объект и осуществлять его перемещение, но для осуществления прочих манипуляций (например, разрезать предмет) данное устройство не подходит. Более того, для любого манипулятора можно найти такое действие, совершать которое с помощью данного устройства оказывается крайне неудобно. Остаётся единственный выход: отказаться от манипулятора вовсе и осуществлять ввод данных в виде трёхмерных жестов.

Различных подходов к реализации захвата движений (Motion Capture) достаточно много, но, к сожалению, все они обладают определёнными недостатками. Оптические методы требуют закрепления специальных маркеров и определенного размещения нескольких камер (для безмаркерного варианта требуется плотно облегающая одежда и отсутствие прочих движущихся объектов в кадре). Инерциальные системы также требуют закреплять датчики и, кроме того, они не способны определять абсолютное положение в пространстве без поддержки оптических систем. Механические системы, помимо прочего, достаточно тяжелы, требуют периодической калибровки и имеют привязку к конкретному размеру. Электромагнитные и ультразвуковые методы являются аналогами оптических систем, но при этом они менее устойчивы к внешним помехам [2-8].

Поскольку для манипуляций виртуальными объектами необходимо захватить положение рук в пространстве, наилучшим решением является использование оптической технологии. Сложность использования, связанную с закреплением датчиков на теле человека, можно устранить, выделив все датчики в отдельное устройство, которое пользователь будет брать в руку.

Примером подобного манипулятора является Wii Remote. Данное устройство, обладая шестью степенями свободы, позволяет осуществлять ввод трёхмерных жестов, являясь при этом простым в установке и использовании. Однако он обладает двумя существенными минусами. Во-первых, самым дорогим элементом системы является именно манипулятор. В том случае, когда нам нужно использовать несколько датчиков (например, захватывать положение двух рук), рациональнее сделать манипулятор как можно дешевле, а дорогую составляющую вынести во внешнее устройство. Во-вторых, поскольку камера находится в манипуляторе, небольшой наклон устройства приводит к существенному смещению кадра. Для повышения точности ввода источник света лучше размещать внутри манипулятора, а камеру размещать во внешнем неподвижном датчике [9].

Всё это было учтено при разработке собственного устройства ввода, получившее название “Интерфейс фонарика”.

3. Интерфейс фонарика

В качестве дешёвого манипулятора для ввода трёхмерных жестов был выбран обыкновенный карманный фонарик. В качестве датчика света используется стандартная веб-камера, которая крепится к монитору. Фонарик, помимо низкой стоимости и лёгкости использования обладает одним немаловажным преимуществом: он является не точечным, а протяжённым источником света. Иными словами, камера видит фонарик не как точку, а как круг. Эта особенность позволяет нам вычислять расстояние до объекта на основе анализа изображения всего одной камеры. Поэтому, при установке системы не требуется производить калибровку, а сама камера может быть размещена совершенно произвольно.

Поскольку веб-камера работает в оптическом диапазоне, в кадре может находиться множество других источников света, таких как окна, лампы, экраны компьютеров. Чтобы облегчить выделение светового пятна фонарика в изображении, на фонарик одевается цветная бумага (что, кстати, предотвращает ослепление фонариком веб-камеры и других людей). Соответственно, в полученном изображении анализируются только световые пятна соответствующего цвета. Также из рассмотрения исключаются слишком маленькие пятна (одиночные пиксели) и пятна неэллиптической формы.

Далее световые пятна фонариков подвергаются анализу. В эллипсе пятна осуществляется поиск самой длинной и самой короткой диагонали. Самая длинная диагональ параллельна плоскости экрана, следовательно, её длина совпадает с диаметром фонарика. Зная координаты проекции короткой диагонали и её длину, мы можем восстановить трёхмерные координаты короткой диагонали (рис.1). Тогда вектор направления фонарика получается как векторное произведение двух диагоналей. По-

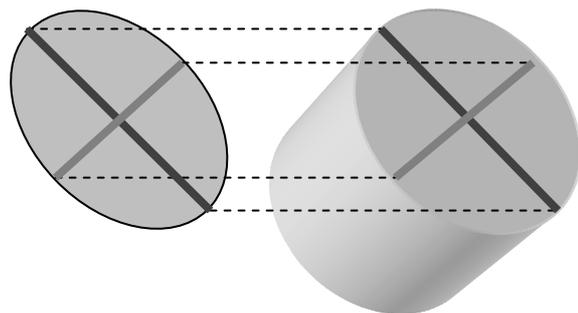


Рис. 1 – Проекция фонарика

сколько диаметр фонарика постоянен, длину диагонали можно использовать для вычисления расстояния (чем дальше находится фонарик, тем меньше световое пятно). Положение центра пятна используется для вычисления двух оставшихся координат [10].

Для оценки влияния фоновой засветки на эффективность работы, система была протестирована в трех различных ситуациях: при сильном естественном освещении (солнечный день, окно в кадре), при сильном искусственном освещении (несколько ламп в кадре) и при умеренном освещении (отсутствие источников света в кадре). В качестве анализируемых параметров были выбраны: время работы алгоритма, количество световых пятен до фильтрации, количество световых пятен после фильтрации и загрузка процессора (без учета драйвера веб-камеры). Тестирование выполнялось в течение минуты, полученные значения усреднялись. В кадре находился один фонарик. Результаты тестирования приведены в таблице 1.

Таблица 1. Анализ эффективности работы интерфейса фонарика.

	Время работы	Число световых пятен до фильтрации	Число световых пятен после фильтрации	Загрузка процессора
Сильное естественное освещение	35 мс	147	1	14%
Сильное искусственное освещение	27 мс	106	1	12%
Умеренное освещение	18 мс	1	1	11%

Как видно из таблицы, алгоритм эффективно выделяет фонарик из общей картинки в любых условиях. Интенсивность внешней освещенности влияет на время выполнения, но даже в самых сложных условиях алгоритм может работать в реальном времени.

Для оценки точности разработанной системы была создана мини-игра, цель которой – как можно дольше уклоняться от летящих навстречу кубов (рис. 2). Управление осуществляется при помощи единственного фонарика, задающего положение игрока внутри узкого коридора. Результаты испытаний показали, что после минутного знакомства с интерфейсом, пользователи могут уклоняться от кубов в течение длительного периода времени (до пяти минут и даже более). Это позволяет сделать вывод, что разработанный интерфейс обладает достаточной точностью и является удобным в использовании.



Рис. 2. – Мини-игра 3D Blocks

4. Распознавание жестов

Каждый жест представляет собой траекторию в трёхмерном пространстве. Для того чтобы распознавать жесты, нам необходимо эффективно сравнивать вводимую пользователем траекторию с набором базовых траекторий. Основными проблемами при сравнении двух траекторий являются:

1. Несоответствие скоростей ввода (одна и та же траектория может состоять из разного количества точек);
2. Несоответствие масштабов (размеры траекторий могут отличаться);
3. Несоответствие углов (траектория может быть чуть повёрнута);

В качестве основы был взят “\$1 Gesture Recognizer Algorithm”, предложенный Jacob O. Wobbrock, Andrew D. Wilson, Yang Li, где эти проблемы решались следующим образом:

1. Проблема несоответствия скоростей ввода решается путем изменения набора точек. На введенной траектории равномерно располагается заранее определённое количество точек, которые и используются для дальнейшего анализа.

2. Проблема несоответствия масштабов решается путём масштабирования траекторий к заранее определённым габаритам. Масштабирование выполняется путём вписывания в квадрат.

3. Проблема несоответствия углов решается путём поиска наиболее подходящего поворота (т.е. осуществляется поиск минимально возможного расстояния). Поворот производится в пределах тридцати градусов с точностью в один градус. В качестве результата выбирается найденный минимум.

“\$1 Gesture Recognizer Algorithm” обладает отличной точностью: при использовании единственного шаблона для каждого жеста точность достигает 97%, при использовании трёх и более шаблонов точность превышает 99%. Другим важным преимуществом является низкая вычислительная сложность и линейная зависимость от количества базовых фигур [11].

К сожалению, данный алгоритм имеет два существенных для нас недостатка:

1. Он является двумерным и не способен распознавать траектории в пространстве.
2. Он не способен выделять жест в непрерывном пользовательском вводе.

Однако эти недостатки можно устранить путём достаточно простой модификации алгоритма.

Собственно, работа алгоритма разбивается на две фазы: выполнение преобразований и вычисление расстояния. Расстояние между двумя точками может быть вычислено в любом пространстве, достаточно лишь выбрать метрику. Что касается преобразований, то проблема несоответствия скоростей ввода и проблема несоответствия масштабов устраняются совершенно аналогично.

Проблема несоответствия углов требует нового подхода. Это связано с тем, что осуществлять перебор поворотов вокруг нескольких осей слишком долго с вычислительной точки зрения. Соответственно, необходимо осуществлять поворот лишь вокруг одной из осей, и наша задача состоит в том, чтобы выбрать эту ось наилучшим образом.

Практически любой человеческий жест достаточно хорошо укладывается в плоскость. Следовательно, в качестве оси вращения удобно выбирать нормальный вектор этой плоскости. Зная нормали для пользовательской и для базовой траектории, мы можем совместить их путём поворота одной из плоскостей, после чего осуществить перебор, аналогичный двумерному случаю. Остаётся лишь вычислить уравнение плоскости.

Поскольку точки пользовательского ввода не лежат в одной плоскости (в первую очередь, из-за зашумлённости ввода), требуется вычислить некоторое приближение. В настоящее время данная задача решается при помощи метода наименьших квадратов, который обладает высокой производительностью и обеспечивает хорошую точность ввода [12].

Выделение жестов в непрерывном пользовательском вводе может выполняться двумя способами. Один из подходов состоит в требовании обозначать жест при помощи пауз в движениях. Однако данный метод требует от пользователя повышенного внимания, затрудняет совершения последовательности жестов и ухудшает погружение в виртуальную среду.

Другой подход состоит в непрерывном анализе ввода. Каждый раз, когда появляются новые данные, мы предполагаем, что пользователь только что завершил ввод жеста, и пытаемся его распознать. Таким образом, конец траектории – это просто последняя введённая точка. Начало траектории может быть определено лишь по косвенным признакам. В качестве точек-кандидатов прежде всего выбираются моменты пауз в движениях, затем точки траекторий, где скорость или направление существенно меняются и, наконец, промежуточные точки, взятые с некоторым интервалом. Длина траектории ограничивается по времени (скажем, три секунды с момента ввода последней точки), расстояние между точками-кандидатами также ограничивается по времени (скажем, не менее 1/8 секунды). Эти меры позволяют нам гарантировать, что начальных точек будет не более определённого количества. Далее, с каждой точкой-кандидатом в качестве начала траектории производится попытка распознать жест. В случае успеха дальнейший поиск не производится. Если нам удалось распознать траекторию, то вся история стирается, поскольку новый жест может начаться лишь после завершения текущего.

Непрерывный анализ требует больших вычислительных ресурсов, но его вычислительная сложность по-прежнему равняется $O(n)$, где n – число базовых фигур. Также стоит отметить, что сложность данного алгоритма не зависит от характера вводимых данных.

5. Взаимодействие с виртуальными объектами

Метод взаимодействия с виртуальными объектами при помощи жестов напрямую зависит от способа ввода этих жестов. Это связано и с полнотой вводимых данных (производится Motion Capture всего тела или только какой-то части), и с точностью ввода (распознаются ли микродвижения), и даже с используемой технологией. Скажем, при использовании механической перчатки положение пальцев должно быть существенным, а при использовании манипулятора наподобие Wii Remote – нет. Следовательно, метод манипуляции виртуальными объектами должен быть индивидуальным для каждого конкретного устройства.

Наша цель – обеспечить пользователи возможность оперировать виртуальными объектами точно так же, как если бы объект был реален. В руках у пользователя находится фонарик – манипулятор, при помощи которого осуществляется ввод. Следовательно, в «виртуальной руке» тоже должен находиться предмет, с помощью которого и осуществляется воздействие на виртуальную среду. Иными словами, все манипуляции с объектами должны выполняться не непосредственно, а при помощи виртуальных инструментов.

Подобный подход является удобным для пользователя, поскольку он, основываясь на опыте реальной жизни, знает какой жест применить, чтобы осуществить действие, и каких результатов следует ожидать. Таким образом, пользователь может взаимодействовать с виртуальной средой без предварительного обучения, либо с минимальным обучением принципам функционирования системы. Удобен подобный подход и для разработчика, поскольку использование виртуального инструмента задаёт контекст действий и, тем самым, облегчает распознавание жестов.

В качестве виртуальных инструментов лучше всего выбирать те, которые знакомы пользователю в наибольшей степени. При этом лучше не связывать инструмент с конкретным виртуальным объектом. Понятно, что кухонным ножом молекулу белка не разрезают, но реальные инструменты, использующиеся в биологии, большинству людей не известны. Конечно, можно разработать систему предназначенную именно для биологов, однако использование бытовых инструментов облегчает понимание совершаемых действий для сторонних наблюдателей.

В качестве примера рассмотрим некоторые стандартные действия с трёхмерной визуализацией результатов вычислительных процессов:

1. Перемещение и поворот. Данное действие лучше всего выполнять двумя руками, осуществляя захват объекта с противоположных сторон. В качестве инструмента можно использовать кухонные прихватки.
2. Выбор объекта. Для выбора объекта можно ткнуть в него указкой.
3. Выбор группы объектов. Захватывать множество объектов логично при помощи сети. Пользователь, держа сеть двумя руками (за два конца), обводит ею контур требуемой области. Все объекты, попавшие в контур, выделяются.
4. Сечение. Выполнять сечение, т.е. разрезать предмет можно с помощью ножа.

Данный перечень не означает, что для выполнения вышеуказанных действий должны использоваться именно эти виртуальные инструменты. Это лишь модельный пример, демонстрирующий подход к реализации взаимодействия с виртуальной средой. При создании реальной системы, в первую очередь, необходимо учитывать мнение будущих пользователей. Более точная привязка к конкретным задачам может осуществляться при помощи прилагаемого к системе инструментария позволяющего расширять или модифицировать набор взаимодействий, добавляя или изменяя виртуальные инструменты и связывая с ними соответствующие жесты.

6. Дальнейшие исследования

В настоящее время выполняется настройка параметров алгоритма распознавания жестов (выбор пороговых значений, сравнение различных промежуточных методов, отладка программы). Одновременно с этим производится проверка эффективности алгоритма на людях, не знакомых с данной системой. В дальнейшем планируется проведение опросов с целью выявления необходимых взаимодействий и наиболее знакомых пользователям инструментов, пригодных

для осуществления этих взаимодействий, а также разработка демонстрационных версий системы.

Также планируется проведение исследований на предмет эффективности использования “интерфейса фонарика” для исследования виртуальных объектов по сравнению с другими методами взаимодействия. Предполагается реализация системы в виде набора библиотек для использования в сторонних приложениях.

Литература

1. Bernd Froehlich, Jan Hochstrate, Verena Skuk, Anke Huckauf. The GlobeFish and the Globe-Mouse: Two New Six Degree of Freedom Input Devices for Graphics Applications // Conference on Human Factors in Computing Systems. 2006. P. 191-199.
2. Motion capture [http://en.wikipedia.org/wiki/Motion_capture].
3. Maletsky Lorin, Junyi Sun, Morton Nicholas. Accuracy of an optical active-marker system to track the relative motion of rigid bodies // Journal of biomechanics 2007, vol. 40, no3, p. 682-685.
4. Ramesh Raskar, Hideaki Nii, Bert deDecker... Prakash: lighting aware motion capture using photosensing markers and multiplexed illuminators // SIGGRAPH papers, 2007, article No. 36.
5. Stanford Markerless Motion Capture Project [www.stanford.edu/~stefanoc/Markerless/Markerless.html].
6. Miller Nathan, Jenkins Odest Chadwicke, Kallmann Marcelo, Matarić Maja J. Motion Capture from Inertial Sensing for Untethered Humanoid Teleoperation // In Proceedings of the IEEE International Conference on Humanoid Robots 2004.
7. Motion Capture as a Means for Data Acquisition (<http://vizproto.prism.asu.edu/datacapture/motioncapture1/>).
8. Maureen Furniss. Motion Capture [<http://web.mit.edu/comm-forum/papers/furniss.html>].
9. Wii Remote [http://en.wikipedia.org/wiki/Wii_Remote].
10. Зырянов А. В., Авербух В. Л. Интерфейс на основе жестов для взаимодействия с виртуальными средами // X международный семинар по супервычислениям и математическому моделированию, г. Саров, 29 сентября - 3 октября 2008 г. Тезисы докладов. Стр. 73-74.
11. Jacob O. Wobbrock, Andrew D. Wilson, Yang Li. Gestures without Libraries, Toolkits or Training: A \$1 Recognizer for User Interface Prototypes // Symposium on User Interface Software and Technology, 2007.
12. Зырянов А.В. Методы ввода и распознавания жестов для взаимодействия с виртуальными средами // Третья международная конференция "Информационно-математические технологии в экономике, технике и образовании" г. Екатеринбург, 20-22 ноября 2008 г. Тезисы докладов. Стр. 282-283.