

УДК 519.6

МЕТОДЫ МАНИПУЛЯЦИЙ ОБЪЕКТАМИ В ТРЕХМЕРНЫХ ВИЗУАЛЬНЫХ СРЕДАХ

В. Л. Авербух, А. В. Зырянов
(ИММ УрО РАН, УрГУ, г. Екатеринбург)

Описаны подходы к созданию средств манипуляции объектами в современных системах визуализации, в том числе построенных на базе виртуальной реальности. Рассмотрено понятие языка действий. Сделан обзор возможностей *трехмерного ввода*, включая трехмерные манипуляторы различного типа и средства *захвата движений*.

Дается описание прототипной реализации простого и удобного интерфейса для ввода жестов на базе веб-камеры и простого фонарика. Такой интерфейс может быть полезным при организации взаимодействия как в рамках систем компьютерной визуализации, так и в системах управления, использующих большие экраны.

Ключевые слова: визуализация, манипуляция объектами, виртуальная реальность.

Введение

В процессе интерпретации и анализа данных при работе в средах визуализации важную роль играет обеспечение взаимодействия с пользователем, цель которого состоит в определенном изменении виртуальной среды (будь то создание или изменение визуальных объектов, либо перемещение камеры, либо какие-нибудь иные операции). Для графических систем, основанных на двумерной графике, характерно использование таких физических устройств, как световое перо, джойстик (joystick), сенсорный экран (touch screen), трэк бол (track ball). Использование средств трехмерной графики, в том числе базирующихся на средах виртуальной реальности, естественно влечет поиск новых средств *трехмерного ввода* и построения на их основе новых систем человеко-компьютерного взаимодействия. К настоящему времени известен целый ряд подобных устройств, построенных на различных физических принципах.

Вместе с тем возникают проблемы с применением сложных систем ввода в средах визуализации, причем трудности связаны как с эксплуатацией и непосредственным использованием техники, так и с диалоговыми языками ввода и взаимодействия. Цель авторов состоит в разработке простых средств ввода в системах визуализации, адекватных задачам, стоящим перед пользовате-

лями — исследователями и/или разработчиками сложных компьютерных моделей.

Ниже рассматриваются некоторые принципы построения языков действий пользователя, приводится обзор состояния дел, связанных с развитием средств трехмерного ввода, и представляется авторская разработка прототипа простого и удобного средства ввода информации для различных визуализационных задач.

Язык действий пользователя

Для изучения рассматриваемых проблем необходимо осознать языковую природу компьютерной визуализации, понять визуализацию как язык со своим словарем, синтаксисом, семантикой и прагматикой. Можно говорить о наличии в системах компьютерной визуализации особого рода *визуальных текстов*, сочетающих вербальные и визуальные элементы, а также манипуляции пользователя с различными устройствами ввода, включая в системы виртуальной реальности само тело пользователя (или его отдельные части). Изучение языков действий пользователя в простом варианте было предметом ряда исследований [1]. Ниже приведены некоторые результаты, полезные и в настоящее время.

Введем (согласно [2]) понятие *когнитивного расстояния* интерфейса. Когнитивное расстоя-

ние измеряет усилия пользователя, необходимые для преобразования действий по вводу данных и представлений выводимой информации в операции и объекты прикладной области. Выбранные для описания модельных объектов зрительные образы должны легко подходить для этого перевода и соответствовать естественным образам моделируемой прикладной области. Точно так же действия, необходимые для манипуляций объектами, должны быть близки по смыслу к командам, применяемым при работе с модельными объектами. Важной задачей проектирования интерфейса является уменьшение его когнитивного расстояния.

В связи с этим следует обратить внимание на синтаксис языка действий и попытаться ответить на вопрос, каким образом следует реализовать связь между синтаксисом языка действий, определяемого порядком действий пользователя, применяющего различные физические и/или виртуальные устройства ввода, и семантикой операций над объектами вычислительной модели. Язык действий пользователя связывает набор операций над модельными объектами с конкретными действиями по вводу данных. В двумерном случае набор действий практически постоянен как из-за ограниченного набора устройств, так и из-за устойчивых привычек пользователей (в этом случае важна так называемая *мышечная память* пользователя). Обычный набор действий здесь состоит из перемещения пиктограмм и иных объектов из одного окна в другое или в текущем окне, совмещения пиктограмм, указания на пиктограммы или другие объекты, отображающие интерактивную деятельность, различных манипуляций с мышью и пр.

В 90-е годы выявилось еще большее сужение набора устройств ввода, используемых в массовом компьютеринге. Световое перо, джойстик, сенсорный экран, трэк бол, широко используемые в 70-х и начале 80-х годов, были чуть ли не полностью вытеснены клавиатурой и мышью, причем в начале 90-х даже наблюдалась тенденция замены реальных алфавитно-цифровых и кнопочных клавиатур виртуальными устройствами. (В настоящее время джойстик, трэк бол и особенно сенсорный экран снова заняли свое место среди устройств *двумерного ввода*, особенно в специализированных приложениях, хотя все еще уступают в этом плане мышью, по крайней мере в случае массовых персональных компью-

теров.) Казалось бы, сужение набора физических устройств ввода должно ограничить возможности пользователя, уменьшить сферу применения *безрежиссного* интерфейса (т. е. такого, который не требует от пользователя все время отслеживать, в каком режиме он работает, помнить особенности синтаксических правил, применяемых при каждом режиме работы). Однако использование большого набора пиктограмм и визуальных объектов, отображающих ввод данных, обеспечивает четкое различие команд путем их реализации на разных виртуальных устройствах. Кроме того, в связи с уменьшением числа устройств ввода увеличилась однородность интерфейса (см. ниже).

Среди набора традиционных требований к языкам действий пользователя (таких, как обратимость действий, наличие контекста, который обеспечивает поддержку пользовательской ориентации, наглядность интерфейса с минимальным доверием к памяти пользователя, наличие обратной связи почти для всех пользовательских операций, подтверждения для потенциально разрушительных действий пользователя) выделяется требование совместимости и согласованности интерфейса. Именно совместимость и согласованность позволяют использовать способности пользователя интерпретировать действия на основе предшествующего опыта. При этом должны учитываться устойчивые, привычные ассоциации между визуальными образами и манипуляциями ими в виртуальной реальности и объектами и действиями над ними в реальном мире.

Рассматриваются три уровня совместимости интерфейса:

- 1) семантический, который основывается на объектах и операциях над ними, имеющих одинаковый смысл;
- 2) синтаксический, определяющий порядок и расположение визуальных объектов;
- 3) физический, который определяется возможностями аппаратных средств.

Для оценки языков действий пользователя введем понятия диалоговой выразительности и однородности интерфейса.

Диалоговая выразительность интерфейса может быть описана следующими характеристиками:

- 1) минимальным числом необходимых для получения результата элементарных манипуляций с устройствами;

- 2) минимальным количеством необходимых для получения результата переключений с устройства на устройство;
- 3) однозначностью интерпретации действий по вводу данных;
- 4) естественностью данного действия пользователя для получения соответствующего результата.

Однородность интерфейса можно определить как степень однообразия физических манипуляций с реальными устройствами при операциях, несущих сходную смысловую нагрузку и реализованных на подобных виртуальных устройствах.

Повышение диалоговой выразительности и однородности интерфейса, так же, как и увеличение визуальной выразительности, должны служить для уменьшения когнитивного расстояния интерфейса.

Виртуальная реальность и средства трехмерного ввода

Как известно, термин *виртуальная реальность* был предложен в 1989 г. Дж. Ланьером для описания систем на базе тренажеров и симуляторов, созданных для летчиков и космонавтов. Сначала этот термин использовался исключительно для описания соответствующей аппаратуры (шлемов со встроенными экранами, перчаток, манипуляторов и пр.), но вскоре он перешел и на соответствующее программное обеспечение. Первоначально системы виртуальной реальности использовались как новое развлечение, но вскоре они стали применяться при разработке систем визуализации различного назначения.

Виртуальную реальность можно описать как ментальный опыт, который заставляет пользователя поверить, что он *там*, что он находится в виртуальном мире. Взаимодействуя с виртуальной средой, пользователь теперь не просто наблюдатель того, что происходит на экране,— он чувствует, что погружается в этот мир и принимает участие в его жизни. Это происходит, несмотря на тот факт, что в действительности и пространство этого мира, и его объекты существуют только в памяти компьютера и мозгу пользователя. В системах виртуальной реальности пользователь, "погруженный" в эту реальность, играет важную роль. Он осуществляет управление выводом информации, а также может участвовать в адаптивном управлении ра-

ботой прикладной программы. Суть виртуальной реальности состоит в содержащейся в ней зависимости между участником и виртуальной средой. Можно определить виртуальную реальность в терминах опыта человека как реальную или моделируемую среду, в которой человек, ее воспринимающий, получает впечатление *присутствия* (*immersion*). Вывод изображения осуществляется с учетом требований фотореалистичности. В литературе отмечается, что естественное, построенное на интуиции управление и взаимодействие является сутью погруженности в системах виртуальной реальности. Если управление и взаимодействие сложно или приводит к ошибкам, то иллюзия погруженности очень быстро исчезает.

Так же, как и в двумерном случае, в непривычных средах и системах, предназначенных для решения сложных задач, необходимы ясные и простые языки взаимодействия. Такие языки характеризуются сравнительно малым словарем, простым и недвусмысленным синтаксисом, семантикой и прагматикой. Можно сформулировать требования к *хорошему* интерфейсу в случае сред виртуальной реальности.

Когнитивное расстояние интерфейса может быть уменьшено при разработке таких методик манипуляций с устройством ввода, которые с какой-то степенью точности соответствуют воздействию на виртуальный объект (к примеру, поворот манипулятора приводит к аналогичному повороту объекта). В частности, согласованность интерфейса в случае сред виртуальной реальности означает требование того, чтобы манипуляции виртуальным объектом были подобными манипуляциям исходным объектом в руках человека в реальном мире.

Манипуляции в рамках систем научной визуализации часто связаны с сечениями, определениями изолиний и изоповерхностей, перемещениями в трехмерном пространстве и пр. Отсюда непосредственно вытекает, что создаваемый способ ввода должен быть трехмерным. Однородность интерфейса означает минимизацию набора виртуальных манипуляторов.

Перед разработчиком интерфейса возникают проблемы, связанные, во-первых, с выбором набора допустимых манипуляций объектами, а во-вторых, с поиском или разработкой устройств ввода, удобных для осуществления выбранных взаимодействий.

Решение может быть связано с реализацией естественных языков взаимодействия, в частно-

сти, языка жестов. Основными преимуществами языка жестов по сравнению с другими естественными языками являются универсальность (жест действия, например, вырезания объекта при помощи виртуального резца, мало зависит от национальной культуры человека), а также возможность проецировать повседневный опыт на виртуальные среды. Однако этот язык оказывается привязанным через операции к данной предметной области. Возможно использование *естественных* инструментов. Среди них могут быть реальные, например, основанные на лазерных указках и экранных сенсорах, "чувствующих" появление лазерного луча, или виртуальные, основанные на традиционной мыши, клавиатурах или трехмерных манипуляторах.

Интерфейс должен быть простым с технической точки зрения (простота установки устройства ввода и легкость его повседневного использования) и по возможности недорогим. В этой связи предпочтение следует отдать мало отвлекающим и не мешающим работать средствам, а не носимым на голове или теле громоздким и тяжелым конструкциям.

Средства трехмерного ввода достаточно разнообразны. Часть таких систем является просто расширением своих двумерных аналогов. Но существуют и по-настоящему фантастические по своим идеям реализации системы ввода, предназначенные, в том числе, и для сред виртуальной реальности. Именно к таким относятся аппаратура и программное обеспечение, позволяющие человеку управлять реальными и виртуальными объектами посредством мысли.

Системы типа *Brain-Computer Interface* (BCI) основаны на психофизиологических эффектах. Взаимодействие идет за счет анализа потенциалов мозга (электроэнцефалограмм), связанных с подготовкой движения той или иной части тела. Эти потенциалы, индивидуальные у каждого человека, оцифровываются и распознаются. На первом этапе происходит обучение программы на примерах мысленной подготовки движения пользователем. Затем достаточно уверенно проводится управление реальными объектами, объектами на экране дисплея или объектами в виртуальной реальности [3, 4]. В настоящее время созданы системы мысленного управления движением в виртуальных средах, предназначенных для обучения инвалидов.

Представляется, что ВСИ может служить в качестве средства *трехмерного перемещения* (*виртуального полета*) пользователя в средах вир-

туальной реальности, которые могут предназначаться для визуализации абстрактных математических и информационных пространств. Работы в области ВСИ успешно продвигаются в целом ряде научных и медицинских организаций, включая отечественные исследовательские центры.

Однако пока системы на базе ВСИ не могут рассматриваться в качестве близкой перспективы как средства ввода в средах компьютерной визуализации. Причин много. Одна из них — мешающая работе пользователя сложность подключения аппаратуры съема потенциалов мозга. Другая — недостаточная устойчивость мысленного управления объектами. Требуется и много дополнительных исследований, например, изучение адаптации пользователя к условиям виртуального полета. Наконец, полная цена комплекса достаточно велика. Так что пока рассмотрим менее фантастические средства в области интерфейса.

Массово применяемые в системах научной визуализации, в том числе в системах на базе виртуальной реальности, трехмерные устройства ввода можно грубо разбить на два класса:

- 1) манипуляторы, действия которых отображаются на экране, трехмерной сцене или трехмерной поверхности (трехмерная мышь, двуручные джойстики, сенсорные сферы и т. п.);
- 2) устройства типа *захвата движения* (*motion capture*) разного уровня сложности, к которым можно, в общем, отнести и перчатки для ввода данных.

Многочисленные примеры трехмерных манипуляторов описаны в опубликованных еще в прошлом десятилетии обзорах развития этой области (см., например, [5]). Интерес, в частности, представляет разделение средств ввода в рамках систем на базе виртуальной реальности на *манипуляционные* и *навигационные*. При этом отмечается, что для задач манипуляции объектами в трехмерном пространстве необходимо обеспечивать шесть степеней свободы, тогда как средства навигации в трехмерном пространстве можно реализовать в несколько упрощенном виде.

В последние годы появляются новые (достаточно экзотические) манипуляторы, в частности, многопользовательский манипулятор на базе сенсорной сферы, созданный в исследовательском центре Microsoft [6], или в чем-то схожее с ним (также сферическое) устройство

GlobeFish [7], которое рассмотрим в качестве примера.

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 в любом направлении. 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 удобно осуществлять перемещение виртуального объекта в пространстве.

К сожалению, данное устройство не подходит для осуществления полезных в системах научной визуализации манипуляций виртуальными объектами (например, сечения поверхности). Более того, для любого манипулятора можно найти такое важное в системах визуализации действие, совершать которое с помощью данного устройства оказывается крайне не удобно. Нужно найти более естественные с этой точки зрения средства интерфейса, связанные с реальными движениями, жестами и манипуляциями человека.

Средства реализации захвата движений

Как уже было указано, языки жестов являются частью общения в различных национальных и профессиональных культурах. Именно их привычность (натуральность) привлекает исследователей и разработчиков, пытающихся создать средства ввода данных в виртуальных средах. Правда, естественность жестов имеет и обратную сторону — интерфейс на их базе может иметь недифференцированный характер, из-за чего и требуется задание контекста либо режима работы.

В [8] описано использование манипулятивной метафоры *волшебной палочки* для создания интерфейса в системах с элементами виртуальной реальности. По сути, волшебная палочка оказывается системой распознавания жестов на базе отслеживания положения магнитного сенсора с заданием режима путем ввода *заклинаний* (небольшого набора легко распознаваемых устных команд). Кроме этого, существует много других подходов к реализации технологий захвата движений, которые рассмотрим далее.

Методы с использованием оптических камер и маркеров. Ввод трехмерных движений пользователя осуществляется благодаря на-

личию нескольких камер, обзоревающих его с разных углов. Сопоставляя изображения, получаемые с камер, вычислительная система определяет трехмерное положение всех закрепленных на теле пользователя маркеров, а следовательно, и положение самого тела пользователя в пространстве. Поскольку позиции и направления объективов камер известны (благодаря предварительной калибровке), для определения координат каждого маркера достаточно двух обзоревающих его камер. Однако, так как маркер может быть скрыт от камеры другой частью тела пользователя или другим человеком, требуется гораздо большее количество камер (обычно от 6 до 24, но может быть и еще значительно больше) [9].

Оптический захват движений, использующий маркеры, делится на три типа: с пассивными маркерами, с активными маркерами и с фоточувствительными маркерами. Рассмотрим каждый тип в отдельности.

В случае пассивных маркеров рядом с каждой камерой располагается направленный источник инфракрасного света. Благодаря сферической форме закрепленных на теле пользователя маркеров каждая камера "видит" световые пятна всех маркеров, находящихся в зоне прямой видимости. Специальное программное обеспечение анализирует двумерные изображения, получаемые со всех камер, и на основе информации о взаимном расположении камер вычисляет трехмерное положение каждой точки [9, 10].

В случае активных маркеров свет испускается самими маркерами, причем каждый маркер загорается в строго определенный момент времени, что облегчает идентификацию отдельных точек. В отличие от пассивных маркеров данная система более устойчива к помехам наподобие внешней засветки, но сами маркеры при этом нуждаются в постоянном электропитании [9, 11].

Фоточувствительные маркеры, наоборот, улавливают свет, который испускается расположенными вокруг инфракрасными проекторами. Свет различных проекторов отличается, что позволяет каждому маркеру определять свое трехмерное положение, а также ориентацию в пространстве [12].

К сожалению, все эти методы сложны с технической точки зрения: требуется выделенная рабочая область, вокруг которой необходимо установить множество камер и произвести их калибровку. Использование системы также затруднительно, поскольку маркеры нужно закреплять

на теле человека в строго определенных позициях. И, наконец, данные системы весьма дороги.

Безмаркерный оптический захват движений. Вокруг рабочей области располагается достаточно много оптических камер. В изображениях, получаемых с камер, выделяется контур, соответствующий телу человека (контур вычисляется как разность между текущим кадром и кадром, полученным до начала использования, т. е. без человека). На основании информации о взаимном расположении камер восстанавливается трехмерная оболочка человека. Вычисленная оболочка сопоставляется с анатомической моделью тела (задается предварительно), что позволяет получить скелетную анимацию.

Данный метод не использует маркеры или специальные костюмы, единственное требование — одежда должна достаточно плотно прилегать к телу пользователя [13]. Для этого метода также требуется выделенная рабочая область, вокруг которой необходимо установить множество камер и произвести их калибровку, однако использовать подобную систему значительно проще благодаря отсутствию маркеров. Стоимость подобного решения также существенно ниже (но по-прежнему достаточно велика).

Неоптические методы. Существует несколько разновидностей неоптических (т. е. не использующих оптические камеры) методов: инерциальный, механический, электромагнитный и ультразвуковой. В каждом из этих подходов вместо анализа изображений трехмерное положение закрепленных на теле пользователя датчиков фиксируется самими датчиками (маркерами).

Инерциальные датчики содержат в себе трехмерный акселерометр и гироскоп, что позволяет определить ориентацию датчика в пространстве и действующие на него ускорения. Зная начальную позицию (выбирается произвольно) и скорость (нуль), система может дважды проинтегрировать ускорение и вычислить текущее положение каждого датчика в пространстве [14].

Механические системы представляют собой каркас, который крепится к телу пользователя (к примеру, на руку надевается перчатка). Движение тела приводит к движениям элементов каркаса, которые захватываются при помощи простых механических датчиков. При этом захватываются не положения узловых точек, а углы между скелетными элементами каркаса.

Трехмерные позиции частей тела пользователя могут быть вычислены на основе углов, но лишь относительно друг друга. Перемещение всего тела в пространстве данная система фиксировать не способна [15, 16].

Электромагнитные системы по своей сути являются аналогами оптических систем с фоточувствительными маркерами, только вместо инфракрасных проекторов используются электромагниты. Соответственно датчики, фиксируя интенсивность электромагнитных полей, определяют свое трехмерное положение и ориентацию в пространстве. Данный метод сильно подвержен помехам и потому практически не используется [9, 16].

Ультразвуковые системы являются аналогами оптических систем с активными маркерами, только датчики испускают не свет, а звук. Соответственно вместо камер используются микрофоны [17].

По простоте использования электромагнитные и ультразвуковые системы аналогичны системам оптического захвата движений с маркерами (правда, они несколько дешевле). Инерциальные и механические системы не требуют подготовки рабочей области и за счет небольших требований к центральному компьютеру имеют сравнительно небольшую стоимость. Однако для их использования требуется закреплять специальные датчики (которые тяжелее маркеров для оптических систем), а механические каркасы, помимо прочего, еще и не универсальны (должны соответствовать росту и размеру человека).

Манипулятор для ввода жестов

Заметим, что для манипулирования виртуальными объектами не обязательно определять положение всего тела пользователя. Достаточно лишь захватить положения его рук в пространстве или даже кистей. Захват положений пальцев руки может быть как необходимым, так и необязательным — все зависит от вида взаимодействия с виртуальным объектом.

Сперва перечислим методы, которые *не* подходят для решения задачи ввода жестов. Во-первых, это безмаркерный оптический захват движений. Этот метод требует, чтобы тело человека было полностью видно, т. е. не позволяет осуществлять манипуляции, скажем, сидя за столом. Требование в виде плотно облегающей одежды также затрудняет повседневное

использование данного метода. Во-вторых, это электромагнитные системы, являющиеся аналогами оптических систем с фоточувствительными маркерами, но в отличие от последних сильно подверженные помехам со стороны других электронных устройств (например, компьютера). И в-третьих, это ультразвуковые системы, которые являются аналогами оптических систем с активными маркерами, но при этом сильнее наводят помехи друг на друга (из-за особенностей распространения звука и света).

Поскольку цель состоит лишь в захвате положения рук, установка системы ввода становится достаточно простой. Инерциальные и механические системы не требуют внешних устройств вовсе. Для оптических систем достаточно всего двух камер (или двух инфракрасных проекторов), которые можно жестко связать друг с другом, исключая процесс калибровки.

Однако, как видно из предыдущего обзора, ни одна технология не позволяет организовать пользовательский ввод просто с точки зрения ее использования. Сложность использования указанных систем ввода связана с необходимостью крепления специальных маркеров (или датчиков, или каркаса). Приклеивать маркеры непосредственно к руке или прикреплять их к одежде каждый раз слишком долго и неудобно. Использование перчаток (к которым прикреплены маркеры) порождает проблему соответствия размеров и форм. Выходом является выделение датчиков в отдельное устройство, которое пользователь будет брать в руку (или в руки). Это, кстати, исключает из рассмотрения механические системы.

Примером подобного манипулятора является Wii Remote, реализующий комбинацию инерциального и оптического методов захвата движений. Внутри устройства находится трехмерный акселерометр и оптический сенсор, воспринимающий излучение в инфракрасном диапазоне. В качестве источника излучения используется вытянутая панель, на двух концах которой находятся инфракрасные лампы. Оптический сенсор видит лампы как две яркие точки на темном фоне. Изменение видимого положения точек позволяет определить двумерное положение контроллера, изменение видимого расстояния между точками позволяет вычислить третью координату. Наклон устройства вычисляется акселерометром на основе направления вектора гравитации [18].

Wii Remote обладает шестью степенями свободы, позволяя осуществлять ввод трехмерных жестов, он прост в установке и использовании. Однако он обладает несколькими существенными недостатками:

1. Самым дорогим элементом системы является именно манипулятор. В том случае, когда нужно использовать несколько датчиков (например, захватывать положение двух рук), рациональнее сделать манипулятор как можно дешевле, а дорогую составляющую вынести во внешнее устройство.
2. Поскольку камера находится в манипуляторе, небольшой наклон устройства приводит к существенному смещению кадра. Данные о наклоне вычисляются акселерометром, который, особенно во время перемещений, работает с существенной погрешностью. Для повышения точности источник света лучше размещать внутри манипулятора, а камеру — во внешнем неподвижном датчике.

Все это было учтено при разработке собственного устройства ввода, получившего название *интерфейс фонарика*.

Интерфейс фонарика

В качестве дешевого манипулятора для ввода трехмерных жестов был выбран обыкновенный карманный фонарик. В качестве датчика света используется стандартная веб-камера, которая крепится к монитору. Фонарик, помимо низкой стоимости и легкости использования, обладает одним немаловажным преимуществом: он является не точечным, а протяженным источником света. Иными словами, камера видит фонарик не как точку, а как круг. Эта особенность позволяет 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 надевается цветная бумага (что, кстати, предотвращает ослепление фонариком веб-камеры и людей). Соответственно в полученном

изображении анализируются только пятна соответствующего цвета.

Как работает система. Для простоты понимания разобьем весь процесс на несколько этапов:

1. В каждом кадре, который приходит с веб-камеры, выделяются пиксели, отличающиеся от требуемого цвета не более чем на определенную величину. В дальнейшем будем называть такие пиксели белыми, а все остальные — черными.
2. В изображении, состоящем из белых и черных пикселей, выделяются световые пятна. Световое пятно — это область, состоящая из белых пикселей, окруженная черными пикселями или границами изображения. При этом допускается попадание одиночных черных пикселей внутрь светлой области (такие пиксели считаются белыми).
3. Производится фильтрация световых пятен по геометрическим признакам. Отсекаются слишком маленькие пятна (фактически одиночные белые пиксели), а также пятна не эллиптической формы.
4. Вычисляются координаты и вектор направления для каждого светового пятна.
 - 4.1. Поскольку размер фонарика существенно меньше расстояния до камеры, будем считать, что его светящийся круг проецируется параллельным образом на плоскость объектива камеры, т. е. в эллипс (рис. 1).
 - 4.2. Пусть D — диаметр фонарика. Тогда видимый размер V любой диагонали вычисляется по формуле $V = Dk \cos t$, где t — угол между направляющим вектором диагонали и плоскостью каме-

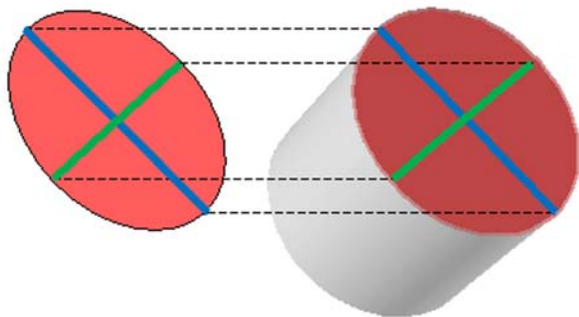


Рис. 1. Проекция светящегося круга фонарика на объектив камеры

ры; k — коэффициент пропорциональности, связанный с тем, что, чем дальше находится фонарик, тем меньше видимые размеры светового пятна.

- 4.3. В двумерном эллипсе светового пятна осуществляется поиск самой длинной (на рис. 1 темная линия) и самой короткой (светлая линия) диагоналей. Пусть их длины L и M соответственно. Поскольку самая длинная диагональ параллельна плоскости камеры, имеем $L = kD$. Так как диаметр фонарика постоянен, зная L , можно вычислить k .
- 4.4. Пусть угол между направляющим вектором самой короткой диагонали и плоскостью камеры равен q . Тогда $Dk \cos q = M$, откуда получаем q .
- 4.5. Теперь, зная углы и направляющие векторы проекций, можно вычислить трехмерные направляющие векторы длинной и короткой диагоналей. Тогда вектор направления фонарика получается как их векторное произведение. Координаты центра пятна и значение k используются для вычисления положения фонарика в пространстве.

Итог — для каждого светового пятна имеем координаты и вектор направления.

5. Производится сопоставление световых пятен и фонариков. Текущий кадр сравнивается с предыдущим (сопоставление для которого уже произведено) и выбирается такое сопоставление, чтобы суммарное перемещение всех фонариков было минимальным (иными словами, выбираются ближайшие фонарики).
6. Производится сглаживание. Дрожание рук пользователя вносит незначительный шум. Чтобы сделать захватываемые движения более плавными и облегчить ввод неподвижности, производится сглаживание каждого параметра путем усреднения значения за небольшой промежуток времени (100—150 миллисекунд, что составляет 4—5 кадров). Меньший промежуток не эффективен, больший вносит заметную для пользователя задержку.

Недостатком данного алгоритма является невозможность различать противоположные направления фонарика (противоположные по осям X, Y — например, левый верхний и правый

нижний углы). Это связано с тем, что световые пятна в обоих случаях выглядят одинаково. В случае необходимости направление фонарика может быть определено косвенным образом — на основе направления перемещения.

Поскольку анализируется пиксельное изображение, ширина и высота кадра определяют количество различных значений положения по осям X и Y . Максимально возможный диаметр светового пятна (также зависит от размеров кадра) определяет количество различных значений положения по оси Z . Точность вычисления вектора направления также зависит от длины самой большой диагонали в световом пятне, и чем дальше находится фонарик, тем меньше точность определения его направления. Таким образом, разрешающая способность камеры напрямую определяет точность ввода данных.

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 коридора. Результаты испытаний показали, что после минутного знакомства, 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 достаточно удобным.

Итак, с технической точки зрения интерфейс фонарика является хорошим устройством ввода.



Рис. 2. Мини-игра 3D Blocks

Теперь проведем анализ использованного алгоритма.

Анализ качества алгоритма. Поскольку разрабатываемая система функционирует в видимом оптическом диапазоне, фоновая засветка оказывает непосредственное влияние на ее работу. Поэтому анализ проводился в трех различных условиях: 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 выполнения алгоритма, количество световых пятен до фильтрации, количество световых пятен после фильтрации и загрузка процессора (без учета драйвера веб-камеры). Тестирование проводилось в течение минуты, полученные значения усреднялись. В кадре находился один фонарик. В условиях сильного естественного освещения пришлось произвести подстройку камеры, откорректировав яркость картинки.

Результаты приведены в таблице.

Тестирование выполнялось на ноутбуке Sony VAIO FZ31ZR:

- процессор Intel Core2 Duo T8300 2.4 GHz (программа выполняется в один поток);
- память 4096 MB;
- камера Genius Eye 312, разрешение 640×480;
- операционная система Windows Vista Ultimate 32-bit.

Тест запускался из среды разработки одновременно с выполнением множества программ (в том числе Kaspersky Internet Security 2009).

Как видно из таблицы, данный 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 выполняться в реальном времени.

Итак, авторами разработан простой манипулятор, позволяющий осуществлять ввод жестов. Однако конечная цель — произвести манипуляцию виртуальным объектом. Попробуем понять, как лучше всего это сделать.

Манипуляция виртуальными объектами. Метод взаимодействия с виртуальными объектами при помощи жестов напрямую зависит от способа ввода этих жестов. Это связано и с

Анализ качества работы алгоритма

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

полнотой вводимых данных (производится захват движений всего тела или только какой-то его части), и с точностью ввода (распознаются ли микродвижения), и даже с используемой технологией. Скажем, при использовании механической перчатки положение пальцев существенно, а при использовании манипулятора наподобие Wii Remote — нет. Следовательно, метод манипуляции виртуальными объектами должен быть индивидуальным для каждого конкретного устройства.

Цель, стоящая перед авторами, — обеспечить пользователя возможностью 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 объектами должны выполняться не непосредственно, а при помощи виртуальных инструментов.

Подобный подход является удобным для пользователя, поскольку он, основываясь на опыте реальной жизни, знает:

- как пользоваться инструментом, т. е. какой жест применить, чтобы осуществить действие;
- каких результатов следует ожидать от использования инструмента;
- какие действия с объектами вообще возможны (исходя из набора инструментов).

Пользователь может взаимодействовать с виртуальной средой без предварительного обучения. Подобный подход удобен и для разработчика, поскольку использование виртуального инструмента задает контекст действий и тем самым облегчает распознавание жестов.

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

- скоростей ввода (одна и та же траектория может состоять из разного количества точек);
- масштабов (размеры траекторий могут отличаться);
- углов (траектория может быть чуть повернута).

За основу был взят *\$1 Gesture Recognizer Algorithm*, предложенный в [19], где эти проблемы решаются следующим образом:

1. Проблема несоответствия скоростей ввода решается путем изменения набора точек. На введенной траектории равномерно располагается заранее определенное количество точек, которые и используются для дальнейшего анализа. Подобный подход также позволяет определить расстояние между траекториями как сумму по всем точкам расстояний от i -й точки базовой траектории до i -й точки пользовательской траектории.
2. Проблема несоответствия масштабов решается путем масштабирования траекторий до достижения заранее определенных габаритов. Масштабирование выполняется путем вписывания в квадрат.
3. Проблема несоответствия углов решается путем поиска наиболее подходящего поворота (т. е. осуществляется поиск минимально возможного расстояния между траекториями). 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 трех и более шаблонов точность превышает 99%. Другими важными преимуществами являются низкая вычислительная сложность и линейная зависимость от количества базовых фигур [19].

К сожалению, данный алгоритм имеет два существенных недостатка:

- 1) является "двумерным" и не способен распознавать траектории в пространстве;
- 2) не способен выделять жест в непрерывном пользовательском вводе.

Однако эти недостатки можно устранить путем достаточно простой модификации алгоритма.

Собственно алгоритм 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 проблемы несоответствия скоростей ввода и несоответствия масштабов устраняются аналогично двумерному случаю.

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

Сначала вычислим центр траектории — точку, координаты которой являются средним арифметическим координат точек траектории. От каждой точки траектории до центральной точки проведем вектор. Далее рассматривается набор попарных векторных произведений (1-го и 2-го векторов, 2-го и 3-го, ..., n -го и 1-го) и вычисляется средний вектор для этого набора (среднее арифметическое). Если угол между средними векторами двух траекторий достаточно велик, то сравнение не производится. Иначе происходит поворот траектории до совмещения средних векторов и параллельный перенос до совмещения средних точек. Далее осуществляется поиск минимума, так же, как и в двумерном случае, выполнением поворота вокруг среднего вектора.

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 совершение последовательности жестов и ухудшает погружение в виртуальную реальность.

Другой подход состоит в непрерывном анализе ввода. Каждый раз, когда появляются новые данные, предполагается, что пользователь только что завершил ввод жеста, и производится попытка его распознать. Таким образом, конец траектории — это просто последняя введенная 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, 3 секунды с момента ввода последней точки), расстояние между точками-кандидатами также ограничивается по времени (скажем, не менее 1/8 секунды). Эти меры позволяют гарантировать, что начальных точек будет не более определенного количества. Далее, с каждой точкой-кандидатом в качестве начала траектории производится попытка распознать жест. В случае успеха дальнейший поиск не производится. Если 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.

Направление дальнейших исследований

В настоящее время алгоритм распознавания жестов реализован в макетном варианте. Параллельно с завершением работы идет подготовка тестов для подстройки параметров алгоритма и проверки его эффективности. В окончательном виде система ввода должна функционировать в виде набора библиотек для использования в визуализационных приложениях. Эти приложения могут быть реализованы как на базе сред виртуальной реальности, так и в средах на базе *больших экранов*, где фонарик может применяться в качестве указателя на графические и тексто-

вые объекты. В дальнейшем планируется проведение исследований эргономичности использования интерфейса фонарика при манипуляции объектами в виртуальных средах.

Список литературы

1. *Авербух В. Л.* Визуализация программного обеспечения. Конспект лекций. Екатеринбург: УрГУ, 1995.
2. *Jacob R. J. K.* Direct manipulation // Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics. Atlanta, GA. October 14–17, 1986. N. Y. 1986. Vol. 1. P. 384–389.
3. *Guger C., Schlugl A., Neuper C. et al.* Rapid prototyping of an EEG-based brain-computer interface (BCI) // IEEE Trans. Rehab. Eng. 2001. Vol. 9 (1). P. 49–58.
4. *Guger C., Edlinger G., Leeb R. et al.* Integration of a brain-computer interface into Virtual Environments // Proc. of 8th Annual Int. Workshop on Presence (PRESENCE 2005). London, UK. September, 2005. P. 347–348.
5. *Jacob R. J. K.* Human-computer interaction: input devices // ACM Computing Surveys. 1996. Vol. 28 (1). P. 177–179.
6. *Benko H., Wilson A., Balakrishnan R.* Sphere: multi-touch interactions on a spherical display // Proc. "User Interface Software and Technology (UIST '08)". Monterey, CA. 2008. P. 77–86.
7. *Froehlich B., Hochstrate J., Skuk V., Huckauf A.* The GlobeFish and the GlobeMouse: two new six degree of freedom input devices for graphics applications // Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. Montreal, Quebec, Canada. 2006. P. 191–199.
8. *Ciger J., Gutierrez M., Vexo F., Thalmann D.* The Magic Wand // Proc. of the 19th Spring Conf. on Computer Graphics. Budmerice, Slovakia. 2003. Session: Virtual reality. P. 119–124.
9. Motion Capture. http://en.wikipedia.org/wiki/Motion_capture.
10. Optical Motion Capture Systems. <http://www.metamotion.com/motion-capture/optical-motion-capture-1.htm>.
11. *Maletsky L., Junyi S., Morton N.* Accuracy of an optical active-marker system to track the relative motion of rigid bodies // J. of Biomechanics. 2007. Vol. 40, No. 3. P. 682–685.
12. *Raskar R., Nii H., deDecker B., Hashimoto Y. et al.* Prakash: lighting aware motion capture using photosensing markers and multiplexed illuminators // ACM Transactions on Graphics (TOG). 2007. Vol. 26 (3). Proc. of the 2007 SIGGRAPH Conf. Article No. 36.
13. Stanford Markerless Motion Capture Project. www.stanford.edu/~stefanoc/Markerless/Markerless.html.
14. *Miller N., Jenkins O. C., Kallmann M., Matic M. J.* Motion capture from inertial sensing for untethered humanoid teleoperation // Proc. Int. Conf. of Humanoid Robotics. Los Angeles., CA. November, 2004. P. 547–565.
15. *Grow Ch., Gordon I., Stuart R. D., Adalja A.* Motion Capture as a Means for Data Acquisition. <http://vizproto.prism.asu.edu/datacapture/motioncapture1/>
16. *Furniss M.* Motion Capture. <http://web.mit.edu/comm-forum/papers/furniss.html>.
17. 3D Computer Mouse Designed by WPI Undergraduates Wins One of 10 Invention Awards from Popular Science. <http://www.wpi.edu/News/Releases/20067/popsciaward.html>.
18. Wii Remote. http://en.wikipedia.org/wiki/Wii_Remote.
19. *Wobbrock J. O., Wilson A. D., Li Y.* Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes // Proc. of the 20th Annual ACM Symp. on User Interface Software and Technology. Newport, Rhode Island, USA. 2007. P. 159–168.

Статья поступила в редакцию 12.12.08.