САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Математико-механический факультет

Кафедра системного программирования

Система анализа реконструктивных хирургических операций при помощи Microsoft Kinect

Дипломная работа студента 545 группы

Ромашкина Амира Сергеевича

Научный руководитель . . . . . . . . . . . . . . . . . . . . аспирант

/подпись/ Петров А.Г.

Рецензент . . . . . . . . . . . . . . . . . . . . старший преподаватель

/подпись/   Антипов И.Г.

“Допустить к защите” . . . . . . . . . . . . . . . . . . . . д.ф.-м.н. А. Н. Терехов А.Н.  
заведующий кафедрой, /подпись/

Санкт-Петербург

2012

SAINT PETERSBURG STATE UNIVERSITY

Mathematics & Mechanics Faculty

Software Engineering Chair

In-surgery analytics in reconstructive cases using Microsoft Kinect

by

Romashkin Amir

Graduate paper

Supervisor . . . . . . . . . . . . . . . . . . . . graduate student

A. G. Petrov

Reviewer . . . . . . . . . . . . . . . . . . . . senior lecturer

I. G. Antipov

“Approved by” . . . . . . . . . . . . . . . . . . . . Professor A. N. Terekhov

Head of Department

Saint Petersburg

2012

# Оглавление

[Оглавление 3](#_Toc325929605)

[1. Введение 5](#_Toc325929606)

[2. Обзор технологии Microsoft Kinect 8](#_Toc325929607)

[2.1 Состав контроллера Kinect 9](#_Toc325929608)

[2.2. Программная поддержка Kinect 13](#_Toc325929609)

[3. Постановка задачи 17](#_Toc325929610)

[3.1. Просмотр 2D моделей 17](#_Toc325929611)

[3.2. Просмотр 3D моделей 18](#_Toc325929612)

[3.3. Аналитика: вычисление численных показателей сцены 18](#_Toc325929613)

[5. Описание решения 22](#_Toc325929614)

[5.1. Инструментарий 22](#_Toc325929615)

[5.2. Архитектура верхнего уровня 22](#_Toc325929616)

[5.3. Альтернативы реализации 24](#_Toc325929617)

[5.4. Конечная реализация 28](#_Toc325929618)

[5.4.1. Реализация модуля просмотра 2D моделей 28](#_Toc325929619)

[5.4.1.1. Алгоритм манипуляции 2D моделями 29](#_Toc325929620)

[5.4.2. Реализация модуля просмотра 3D моделей 30](#_Toc325929621)

[5.4.2.1. Алгоритм манипуляции 3D моделями 31](#_Toc325929622)

[5.4.3. Реализация модуля аналитики 31](#_Toc325929623)

[5.4.3.2. Функциональность модуля аналитики 31](#_Toc325929624)

[5.4.3.1. Алгоритм распознавания пальцев 34](#_Toc325929625)

[5.4.3.3. Алгоритм вычисления объема слоя 37](#_Toc325929626)

[5.4.3.4. Анализ погрешности вычислений 38](#_Toc325929627)

[6. Итоги 40](#_Toc325929628)

[6.1. Достигнутые результаты 40](#_Toc325929629)

[6.2. Возможности развития 42](#_Toc325929630)

[7. Заключение 43](#_Toc325929631)

[8. Список литературы 44](#_Toc325929632)

# 1. Введение

Медицина - одна из древнейших и важнейших сфер жизнедеятельности. Она существовала на самых ранних этапах становления человечества.

Сегодня IT все глубже проникает во все мыслимые и немыслимые сферы жизни и, конечно же, медицина, и особенно хирургия, находятся среди первых в очереди за технологическими новшествами.

В данной работе предлагается двигаться в ногу со временем и внести в сферу хирургии новые возможности, которые открываются благодаря появлению технологии Microsoft Kinect. Главное ее достоинство - возможность бесконтактного взаимодействия с компьютером.

Люди привыкли использовать мышь и клавиатуру для работы с программами, но это не всегда удовлетворяет специфике сферы деятельности.

Одной из ключевых особенностей хирургии как сферы использования ПО - это требование стерильности, которое обычно сложно было выполнять из-за устройств ввода/вывода, которые требуется заново тщательно стерилизовать после каждой операции. При использовании компьютера в операционной врачи используют помощников, которые взаимодействуют с компьютером и не участвуют в процессе самой операции. Это усложняет весь процесс, а также лишние люди в операционной могут мешать.

В связи с этим в большинстве случаев хирурги стараются свести к минимуму взаимодействие с компьютером во время операции, хотя оно бывает очень полезно.

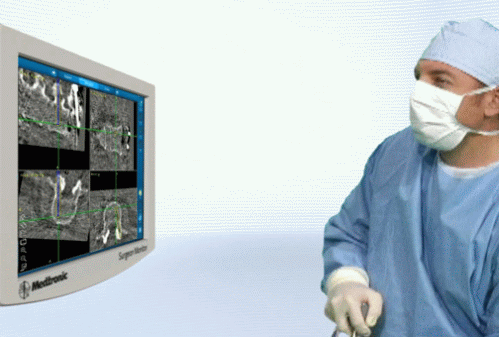


рис.1.1. Снимок томографии рис.1.2. Пример использования системы просмотра МРТ и КТ снимков

Во время операции хирургу часто необходимо знать детали внутреннего строения органов пациента, чтобы не допустить ошибку, проводя операцию вслепую. Для этого используется программное обеспечение, позволяющее в реальном времени просматривать визуальные модели внутренних органов во время операции (рис. 1.2). Эти модели бывают 2-х типов: 2D модели и 3D модели.

К наиболее часто используемым 2D моделям относят фотографии пациента и снимки МРТ (Магнитно-резонансной томографии) или КТ (Компьютерной томографии).

Также часто используются 3D модели (рис. 1.2). Они могут либо отражать реальную структуру органов конкретного человека, если использовалось сложное ПО, оцифровавшее органы, либо какие-нибудь идеальные общеизвестные модели органов или частей тела, которые могут использоваться как справочный материал для хирурга.

Общей целью данной работы является разработка программной системы, которая позволит хирургам и их помощникам просматривать и манипулировать хирургическими моделями, а также производить разного типа анализ поверхности тела пациента.

В данной работе описывается система, которая позволит, наряду с другой функциональностью, манипулировать такого рода моделями во время операции без контакта с мышью или клавиатурой.

Также данная работа ставит перед собой цель решения следующей проблемы.

Часто перед реконструктивной операцией (или во время нее) нужно определить некоторые численные характеристики поверхности тела пациента: например, объем снимаемой покровной ткани с пациента или расстояние между двумя точками на теле.

Реализуемая система позволит это делать в режиме реального времени так, что от хирурга (или его помощника) потребуется лишь отметить два конца отрезка для вычисления расстояния или провести пальцем по контуру области для вычисления объема слоя. Программа посчитает и отобразит результат.

В итоге, благодаря реализуемым модулям использование данной системы сможет увеличить степень удобства операций, ускорить сам процесс операции, снизить риск врачебной ошибки, уменьшить количество помощников в операционной и улучшить качество операций в целом.

# 2. Обзор технологии Microsoft Kinect



рис.2.1. Контроллер Microsoft Kinect

Для начала перед анализом проблемной области стоит познакомиться с ключевым устройством и технологией, используемыми в данной работе, которое известно под названием Microsoft Kinect.

Kinect – это интерактивный игровой бесконтактный контроллер, первоначально представленный для консоли Xbox 360 (1 июня 2009). 1 февраля 2012 вышел новый контроллер Kinect For Windows (вместе с одноименной библиотекой), который отличается от своего “игрового” коллеги тем, что способен вычислять глубину сцены начиная от 40см.

Xbox 360 - это игровая приставка от Microsoft, которая конкурирует с Sony PlayStation 3 и Nintendo Wii в качестве игровой системы седьмого поколения. Ее пользователи - миллионы геймеров по всему миру. По некоторым данным это самая популярная игровая приставка в мире.



рис.2.2. Xbox 360 вместе с Kinect в действии

Свободно распространяемое программное обеспечение, поставляемое вместе с контроллером, позволяет отслеживать в реальном времени скелет пользователя и отличать фон от пользователя переднего плана, осуществляет полное 3-х мерное распознавание движений тела, мимики лица и голоса.

Kinect открывает новую страницу для всего машинно-пользовательского взаимодействия и выпуск контроллера Kinect For Windows, позволяющего воспринимать близко находящиеся объекты сцены, только подтвердил амбиции Microsoft на этот счет. Идея, что компьютер может взаимодействовать с пользователем при помощи естественных жестов, поражает и приближает нас к будущему.

## 2.1 Состав контроллера Kinect

Физически, Kinect — это горизонтально расположенная коробка на небольшом круглом основании, которую помещают выше или ниже экрана. Размеры — примерно 23 см в длину и 4 см в высоту. Состоит из двух сенсоров глубины, цветной видеокамеры и микрофонной решетки (рис. 2.3).

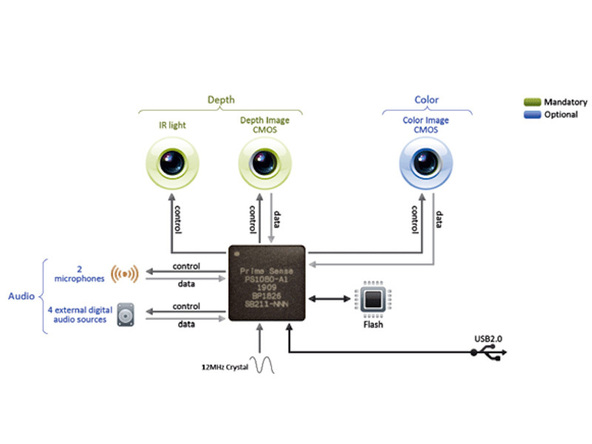


рис.2.3. Схема устройства Kinect

Сердцем сенсора Kinect, является 3D-технология от компании PrimeSense, которая использует структурированный свет [1], инфракрасные камеры и специализированный процессор для измерения расстояния от камеры до сцены.

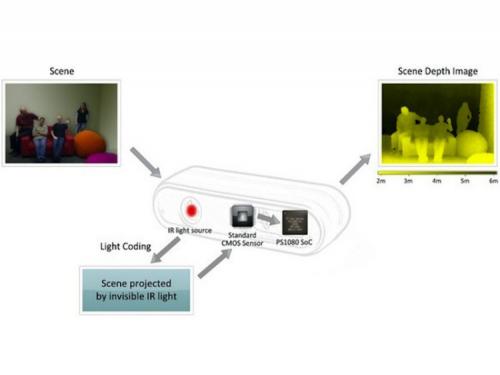


рис.2.4. Схема работы Kinect

Эти измерения производятся по всему полю зрения сенсора. В результате, получается облако точек, состоящее из 307200 измерений расстояний между сенсором и сценой.

Точки проецируются на заданных углах от лазера. Лазерные точки отражаются от объектов сцены и фиксируются ИК-камерой. Объекты сцены, расположенные близко к камере отражаются в точки изображения, которые находятся близко друг к другу, а объекты, расположенные далеко от камеры — отражаются в точки изображения, которые находятся дальше друг от друга [15].

Из-за того что ИК-излучатель находится не в одной точке с камерой, появляется эффект “тени” (рис.6). Не все пиксели в зоне видимости камеры глубины видны из излучателя. Из-за этого при работе с близкими объектами могут появляться невычисленные дальности в карте глубины, что может приводить с сильным погрешностям в вычислениях.

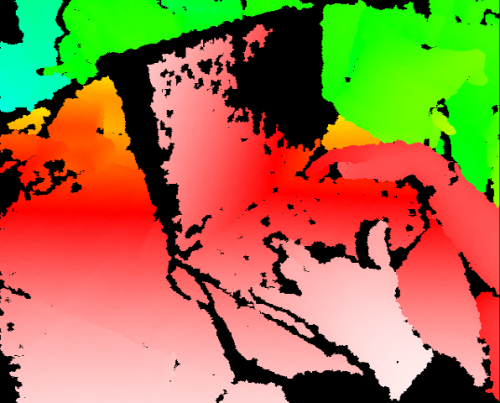


рис.2.6. Карта глубины сцены

У этой сравнительно молодой технологии есть недостатки: при довольно хорошем распознавании скелета, максимальное разрешение камеры и сенсора глубины 640x480, обусловленное пропускной способностью USB 2.0, ограничивает точность распознавания тонких жестов.

Microsoft также заявляет о выходе в будущем устройства Kinect 2 (рис. 2.7), который, по словам компании, будет очень мощным и точным в распознавании. По различным слухам релиз ориентируется на 2013 год[17].



рис.2.7. Логотип и примерный вид сенсора Kinect 2

## 2.2. Программная поддержка Kinect

Устройства Kinect само по себе передает пользователю данные из RGB камеры и камеры глубины. Работать с чистыми потоками данных было бы очень затруднительно и, конечно же, существуют библиотеки, которые предоставляют более высокоуровневый интерфейс для работы с Kinect.

Подробное сравнение библиотек приводится в разделе “5.3 Альтернативы реализации”. В данном разделе лишь кратко описывается, библиотека, которая была использована в работе. После сравнения их характеристик было принято решение об использовании библиотеки KinectForWindows [4].

KinectForWindows предоставляет высокоуровневый API для работы с одним или несколькими сенсорами Kinect, позволяет одновременно распознавать скелеты двух пользователей и примерно находить расположение 6-ти пользователей.

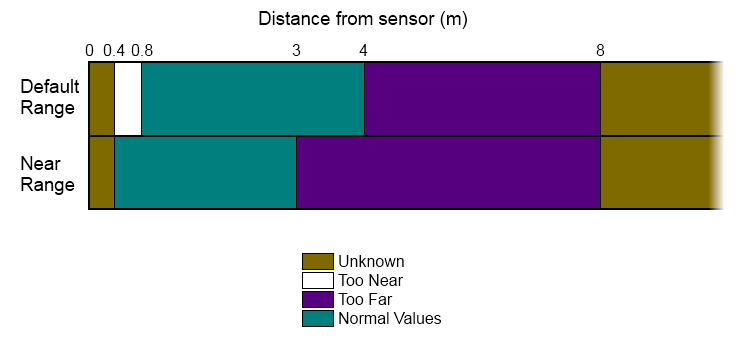


рис. 2.8. Два режима работы KinectForWindows:

1. Режим по умолчанию: доступный диапазон глубин от 80 см до 400 см  
   Доступен в сенсорах Kinect For Xbox и в Kinect for Windows
2. Ближний режим: доступный диапазон глубин от 40 см до 400 cм  
   Доступен в сенсоре Kinect for Windows, возможно будет доступен в Kinect 2.

Библиотека KinectForWindows предоставляет доступ к 20 точкам на теле человека. На рис. 9 изображены эти точки. Также SDK позволяет посчитать любую точку сцены в 3D координатной системе, началом которой будет Kinect, а точнее, камера глубины. Соответственно пользователь легко может получить реальные данные о положении точек в пространстве.

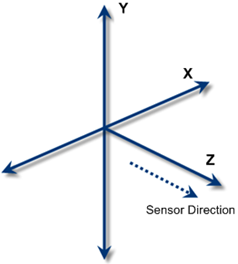
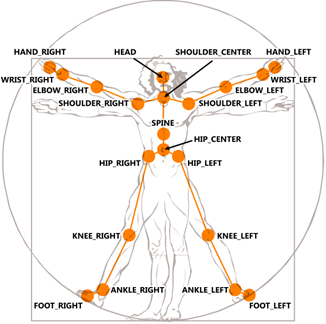


рис. 2.9. Схема получаемых точек на теле рис.2.10. Реальные координаты

Благодаря всему этому мы получаем мощный инструмент для разработки приложений, основанных на бесконтактном принципе взаимодействии с программным обеспечением.

## 2.3. Аналоги Kinect

Успех контроллера Microsoft Kinect означает, что аналогичные по функциональности устройства будут выпущены и другими производителями, желающими приобщиться к "вкусному" рынку интерактивных развлечений. Так, компания ASUS официально представила приспособление под названием WAVI Xtion (произносится как "wavy action"), предназначенное для беспроводного подключения телевизора к персональному компьютеру посредством интерфейса HDMI и укомплектованное Kinect-подобным датчиком распознавания движений [18], подключаемым к ПК по USB2.0.

Рис. 2.11. Система Asus WAVI Xtion

Комплект системы состоит из трех устройств - двух приемо-передатчиков, транслирующих данные с ПК на HD-телевизор, и собственно сенсора, воспринимающего информацию о движении пользователя.

По характеристикам он мало отличается от Kinect, так как использует ту же технологию от компании PrimeSense и работают через одинаковый интерфейс USB 2.0. Тем не менее, XTion стоит почти в полтора раза дороже.

Однако судя по официальному описанию, контроллер от Asus поставляется с уже встроенным инструментом распознавания жестов.

Существуют и другие неигровые сенсоры глубины, которые позволяют считывать глубину в большем разрешении, чем Kinect и Xtion, но в общем случае решения довольно дорогие. Низкая цена Kinect и Xtion обуславливается тем, что компании целенаправленно занижают их стоимость, чтобы занять свободную на рынке нишу и потом получить большую прибыль при продаже игр и других приложений на основе бесконтактных контроллеров.

Также отдельного упоминания заслуживает компания "Системы Компьютерного Зрения" (дочерняя компания фирмы ЗАО "ЛАНИТ"). Компания получила грант от Фонда "Сколково" на разработку пассивной системы распознавания жестов под названием ViEye [19].

Эта система работает по концептуально иной технологии, чем большая часть ее конкурентов: 3D координаты высчитываются при помощи 2х цветных камер, а не ИК-лазера. В этом ее достоинства и недостатки.

У этой системы есть несколько серьезных преимуществ над “Kinect-driven” распознаванием:

* Малая стоимость: достаточно 2х простых цветных камер.
* Широкий спектр применения: от небольших дальностей (в пределах комнаты) до существенных - до 200 м.

С другой стороны, по имеющейся информации о системе существует единственный, но серьезный недостаток по сравнению с Kinect – это точность распознавания 3D координат точек.

* Насколько бы ни были хорошими алгоритмы воссоздания 3D сцены по двум потокам 2D картинок, вряд ли они могут соперничать с реальными данными с 3D сканера.
* Плохая освещенность (или полное отсутствие света) может влиять на качество распознавания, что в работе с Kinect не является серьезной проблемой.
* Такая система сложно масштабируется, когда как новые более точные 3D сканеры будут позволять распознавать все более тонкие жесты (пальцев, плеч, рта и т.д.)

По субъективному мнению автора работы, сфера использования ViEye немного отличается от возможной сферы использования систем с Kinect.

# 3. Постановка задачи

Для достижения поставленных целей требуется система, которая должна упростить хирургам и другим медицинским работникам их профессиональную деятельность.

Данная система должна состоять из 3 модулей:

* Модуль просмотра 2D моделей
* Модуль просмотра 3D моделей
* Модуль аналитики: вычисление численных показателей на теле пациента

Далее сформулируем, каким требованиям должен отвечать каждый из модулей.

## 3.1. Просмотр 2D моделей

Первым модулем системы должно являться приложение, позволяющее загружать и просматривать 2D изображения.

Среди поддерживаемых форматов должны быть стандартные форматы растровых изображений (.bmp, .jpg, .png и т.д.), а также специальный формат файлов DICOM [7].

DICOM - стандарт создания, хранения, передачи и визуализации медицинских изображений и документов обследованных пациентов. Этот формат наиболее часто используют для хранения и передачи снимков МРТ и компьютерной томографии.

Функциональность приложения должна совпадать с функциональностью простой фото-галереи, за исключением того, что взаимодействие с пользователем происходит через Kinect.

Пользователь жестами рук должен иметь возможность:

* Перелистнуть на кадр вперед\назад
* Быстро перелистывать вперед\назад (это нужно, так как количество МРТ снимков обычно большое и снимки практически одинаковы)
* Увеличить\уменьшить изображение, производить навигацию внутри масштабированного изображения.

## 3.2. Просмотр 3D моделей

Система должна включать приложение, позволяющее загружать и просматривать 3D модели.

Среди поддерживаемых форматов должны быть стандартные форматы 3D объектов (.obj, .xaml, .ply и т.д.)

Функциональность приложения должна совпадать с функциональностью простого визуализатора 3D моделей, позволяющего осуществлять классические манипуляции, за исключением того, что взаимодействие с пользователем происходит через Kinect.

Пользователь жестами рук должен иметь возможность:

* Увеличить\уменьшить модель
* Повернуть модель

## 3.3. Аналитика: вычисление численных показателей сцены

Последним модулем системы должно являться приложение, позволяющее оценить объем покровной ткани пациента, внутри ограниченного контура и вычислить расстояние от одной точки до другой на теле пациента.

Пользователь должен иметь возможность зафиксировать изображение пациента и далее производить над изображением аналитические действия.

Должна быть возможность выбора между двумя режимами: “Вычисление глубины слоя” или “Вычисление длины отрезка”.

При выборе первого режима пользователь может ввести с клавиатуры глубину слоя снимаемой ткани, обвести пальцем контур на изображении, приложение должно показать обведенный контур и посчитать объем слоя снимаемой ткани.

При выборе второго режима пользователь может отметить две точки на сохраненном изображении, и приложение должно посчитать длину кривой, получившейся проекцией точек отрезка с изображения на 3D поверхность сохраненной сцены (скорее всего, тела пациента).

# 4. Обзор проблемной области

В данной главе мы изучим актуальность данной проблемы, а также проекты в медицине, где уже сейчас применяется Kinect.

Kinect сегодня используется во многих сферах жизнедеятельности и в том числе в медицине. В промо-ролике к выходу Kinect от Microsoft было красочно показано, как теоретически можно использовать технологию в лечебных целях.

Корпорация Microsoft годами билась за то, чтобы медицинские организации пользовались ее технологиями, и Kinect, по ее мнению, имеет все шансы на успех в данной сфере. На саммите Pacific Healthcare Summit 2011 в Сиэтле, посвященном здравоохранению, директор по стратегиям и исследованиям Крейг Манди (Craig Mundie) показал, каким образом сенсорные технологии способны улучшить их работу.

В одном из сценариев пациенты могли виртуально посетить групповые терапевтические приемы. Чтобы не выдавать внешность пациентов, Kinect использует аватары, улавливающие движения рук, плеч и даже бровей - всё это позволяет сохранять анонимность в группе.

Во время демонстрации Манди показал докторам способ использования Kinect для взаимодействия с данными о пациентах. Стоя перед экраном с подключенным Kinect, он называл имена клиентов, и на экране появлялись записи о них. С помощью жестов Манди пролистал историю пациентки и отметил, что у неё диабет. В электронной записи были закладки с уровнем глюкозы в крови, и эти данные сразу отправлены на веб-сайт. Электронная система также показала, что пациентка недавно растянула связки, что вполне бы могло стать полезной информацией для докторов.

Манди заявил, что компания желает изменить способ взаимодействия людей с компьютерами. Это главная, по его словам, задача компании.

Также интересно, что на данной конференции Манди подчеркнул, что Microsoft занимается разработкой прототипа, который позволит управлять МРТ снимками без мышки, т.е. Microsoft вскоре выпустит продукт, дублирующий первый модуль разрабатываемой в данной работе системы [10].Однако даже примерные сроки выпуска на данный момент неизвестны.

В интернете много статей о том, как может быть полезен Kinect в медицине, о том в каких ситуациях его можно применять, но при этом практически нет ссылок на реальные проекты.

На данный момент в Российской медицине не используются системы, которые при помощи Kinect позволяли бы совершать манипуляции с хирургическими моделями типа DICOM или простыми 3D моделями. Проекты по анализу количественных характеристик сцены при помощи Kinect также не было найдено.

Такая ситуация обусловлена относительной молодостью технологии, и я уверен, что в ближайшее время много проектов в мире подобных данной работе увидят свет в промышленном масштабе и, возможно даже, в России.

# 5. Описание решения

В данном разделе мы опишем процесс решения поставленных выше задач.

## 5.1. Инструментарий

В качестве главной технологии была использована платформа .NET 4 и язык C#. Визуальные компоненты, в том числе работа с 3D, были реализованы по технологии WPF.

Перечисленные технологии были выбраны исходя из того, что требования кроссплатформенности изначально не стояли, библиотека KinectForWindows поставляется для совместимых с .NET языков (C#, C++ и др.), и также на WPF можно строить сложные UI компоненты, в том числе и для высокопроизводительной работы с 3D графикой.

## 5.2. Архитектура верхнего уровня

На верхнем уровне структура компонент выглядит следующим образом:

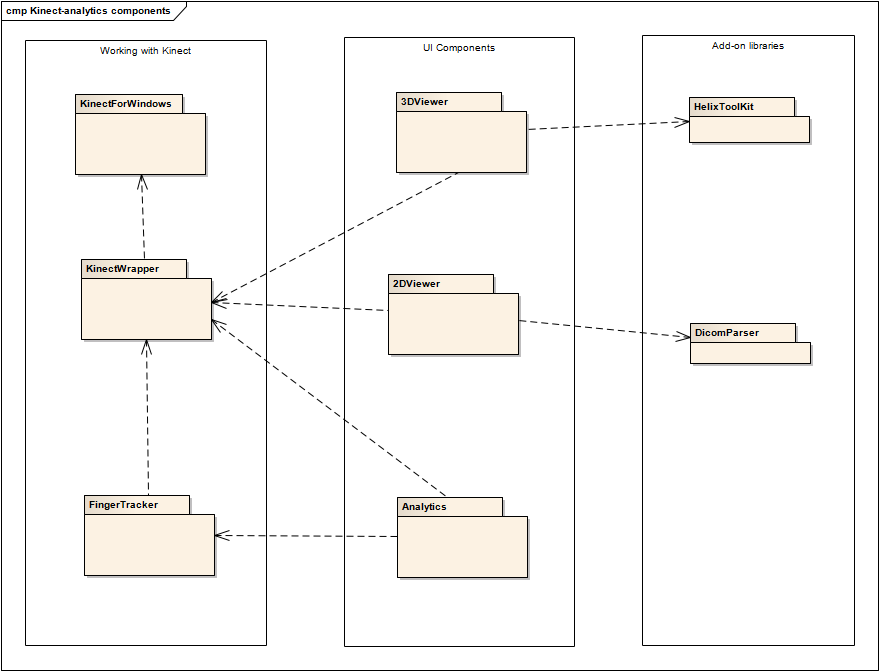
****

рис. 5.1. Структура модулей системы и зависимости между ними

Система состоит из 8 логических (реально их больше, но оставшиеся затрагивают лишь детали) сборок-компонент.

Распишем каждую компоненту подробнее:

* KinectForWindows: библиотека от Microsoft, предоставляющая доступ к данным от Kinect
* KinectWrapper: обертка над KFW, все взаимодействие с KinectForWindows происходит через нее
* FingerTracker: компонента, отвечающая за распознавание пальцев
* 2DViewer, 3DViewer - UI компоненты, позволяющие манипулировать соответственно 2D и 3D моделями
* Analytics - UI компонента, позволяющая высчитывать численные показатели сцены
* HelixToolKit - сторонняя библиотека для взаимодействия с 3D моделями
* DicomParser - компонента преобразование DICOM-файла в набор изображений, использующая сторонний инструмент конвертации изображений ImageMagick [16]

Такое выделение обуславливается требованием возможности подмены одних модулей другими без существенной перестройки системы. Например, новые версии библиотеки KinectForWindows будут выходить, возможно, каждые полгода. Также не должно быть сложно подменить библиотеку распознавания пальцев другой.

Далее мы рассмотрим некоторые компоненты подробнее.

## 5.3. Альтернативы реализации

Одним из основополагающих решений был выбор библиотеки взаимодействия с Kinect.

На данный момент известны 3 наиболее популярные библиотеки:

* Libreenect [2]
* OpenNI [3]
* KinectForWindows [4]

В таблице 5.1. приведено сравнение этих библиотек.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Libfreenect | OpenNI | KinectForWin |
| Первый релиз | 11.2010 | 11.2010 | 06.2011 |
| Open Source | + | + | - |
| Свободное распространение | + | + | + |
| Низкоуровневый доступ к камерам | + | + | + |
| Доступ к микрофонному массиву | - | - | + |
| Распознавание скелета | - | + | + |
| Кроссплатформенность | + | + | - |
| Управление телеметрией | + | - | + |
| Поддержка нескольких Kinect устройств | Неизвестно | +-  (по слухам, плохая) | + |
| Распознавание пальцев | - | - | - |
| Встроенная калибровка камер глубины и цвета | - | + | + |

Таблица 5.1. Сравнение библиотек для работы с Kinect.

Первая библиотека libfreenect предоставляет самый низкоуровневый доступ к Kinect: в частности, только цветное и глубинное изображения. После краткого сравнительного анализа становится ясно, что libfreenect не подходит за отсутствием поддержки распознавания скелета.

При сравнении OpenNI и KinectForWindows итоговый выбор пал на официальную библиотеку от Microsoft по следующим причинам [5]:

* Лучшее распознаванию конечностей
* Поддержка записи звука с Kinect – значит, есть возможность управления голосом при помощи технологии Microsoft Speech [6].
* Поддержка нескольких сенсоров Kinect
* Прямая поддержка контроллеров Kinect For Windows, Kinect 2

Стоит уточнить, что данная работа велась при помощи устройства Kinect For Xbox, который изначально планировался для работы с Xbox на расстоянии от 80 см.

Для каждой из трех подсистем проведем обзор существующих решений компонент, которыми можно было бы воспользоваться при реализации.

Просмотр плоских изображений относительно прост. Существует много реализаций фотоальбомов на основе Kinect. Тем не менее, нет такого инструмента, который бы при этом мог читать DICOM-формат.

Манипуляция с 3D моделями - это классическая проблема, решение которой требуется, конечно, не только в медицинской сфере. Любая 3D аналитика “на ногах”, требующая отсутствия человека за компьютером в данный момент, нуждается в таком приложении.

Тем не менее, из-за молодости технологии Kinect и стремительного роста технологий обработки 3D моделей готовых решений для Kinect найдено не было. Однако, было найдено несколько инструментов, позволяющих эффективно просматривать 3D объекты классическими способами.

Мной были использованы следующие свободные библиотеки:

* 3DTools [8]
* HelixToolkit [9]

Библиотека 3DTools просто устроена, легка во внедрении, но она позволяет читать 3D объекты только в формате XAML.

HelixToolkit позволяет читать разные форматы, также обладает очень качественным интерфейсом управления моделями.

Последним модулем является модуль аналитики. Эта компонента не имеет аналогов в России, так как решаемая ей задача довольно специфична сама по себе. За рубежом, возможно, этим занимаются, но подобных проектов найдено не было.

Однако для разработки данного модуля требовалась отдельная серьезная библиотека - инструмент для распознавания пальцев. Ни одна из существующих библиотек для работы с Kinect не позволяет распознавать движения пальцев из-за относительно низкого разрешения карты глубины сцены – 640x480 и из-за помех и артефактов в потоке данных от устройства. Из-за этого сама компания Microsoft отказаться от распознавания пальцев.

По словам представителей Microsoft Kinect 2 будет настолько точным, что сможет распознавать даже мимику лиц, а и в том числе губ. Соответственно можно быть уверенным, что о распознавании пальцев корпорация позаботится точно. Но на данный момент существуют некоторые решения для распознавания пальцев от разработчиков-энтузиастов, но не от серьезных компаний.

Существует несколько инструментов для слежения за пальцами под OpenNI, но в итоге было принято решение работать с библиотекой KinectForWindows, для которой было найдена лишь одна: библиотека под названием CandescentNUI [11].

Алгоритм распознавания в этой библиотеке основан на изначальной кластеризации изображения на блоки примерно одинаковой глубины. Далее среди этих блоков выявляются те, что могут оказаться руками при помощи различных алгоритмов.

В процессе распознавания не используется информация о расположении рук и скелета человека, что может быть полезно с практической точки зрения: в такой ситуации человеку не обязательно стоять перед Kinect в полный рост для вычисления скелета.

После внедрения этой библиотеки и некоторого времени работы с ней, от нее пришлось отказаться из-за низкой точности распознавания и большого количество ложных срабатываний алгоритма.

Соответственно, было принято решение реализовать инструмент распознавания пальцев с нуля.

## 5.4. Конечная реализация

После ознакомления с существующими решениями можно приступить к описанию самой реализации для каждой из подсистем.

### 5.4.1. Реализация модуля просмотра 2D моделей

На первых этапах разработки системы был реализован просмотр изображений в стиле IPhone, листание в котором производилось при помощи жестов руки. После уточнения задачи и знакомства со снимками DICOM было приятно решение отказаться от уже реализованного функционала и написать более простой статичный просмотр изображений.

Причина упрощения реализации в том, что плавное перелистывание в стиле мобильных устройств пагубно сказывается на восприятии пользователем практически одинаковых картинок, отличающихся в деталях.

Также при этом эффект просмотра видео в случае быстрого листания вперед и назад практически полностью терялся.

В итоге была реализована простая фотогалерея, с возможностью:

* Загрузки растровых изображений и DICOM-снимков
* Перелистывания вперед\назад на один кадр
* Быстрого перелистывания вперед\назад
* Масштабирования
* Навигации внутри увеличенного изображения

Пользователь может загрузить в приложение, как растровые изображения, так и DICOM-файлы. Для распознавания DICOM-файлов используется стороннее консольное приложение ImageMagick [16], позволяющее конвертировать изображения разных форматов в другие. В данном проекте происходит конвертация DICOM-файлов в изображения в формате .png.

#### 5.4.1.1. Алгоритм манипуляции 2D моделями

Подробнее рассмотрим жесты рук, которые используются для совершения упомянутых выше действий.

* Листание на один кадр вперед/назад происходит, если произведены шаги. Он интуитивно схож с жестомЖест должен совершатся резко:
  + Правая или левая рука отводится на расстояние >35 см от груди вперед.
  + Рука производит резкое движение на расстояние 30 см в сторону в течение отрезка времени в 200мс - 800мс с момента отведения руки от груди на расстояние > 35 см.
* Быстрое листание вперед/назад начинается, если произведены два шага:
  + Рука отведена в первоначальное положение на расстояние >35 см от груди.
  + Рука зафиксирована немного правее/левее от первоначального положения, быстрое листание со скоростью около 2 кадров в секунду начинается через 2 секунды.
* Масштабирование (zoom in\out) происходит, когда обе руки пользователя находятся на расстоянии >35 см от груди и пользователь разводит/сводит руки. Величина масштабирования зависит от величины изменения расстояния между руками
* Навигация внутри увеличенного изображения происходит при перемещении руки на расстоянии >35 см от груди.

Численные значения настраивались опытным путем для наиболее устойчивого управления.

### 5.4.2. Реализация модуля просмотра 3D моделей

Единственной проблемой данной подсистемы являлся выбор и внедрение библиотеки работы с 3D моделями. Изначально была внедрена библиотека 3DTools [8], которая позволяла читать лишь один формат 3D моделей - xaml. Из-за малой распространенности XAML пришлось отказаться от этой библиотеки в пользу более серьезной и сложной библиотеки HelixToolkit.

Данная библиотека читает наиболее распространенные на сегодняшний день форматы 3D моделей - .obj, .stl, .3ds, .lwo.

Приложение позволяет пользователю загружать 3D модель совершать с ней стандартные манипуляции.

#### 5.4.2.1. Алгоритм манипуляции 3D моделями

Опишем жесты рук, при помощи которых происходят эти манипуляции:

* Поворот происходит при движении руки пользователя на расстоянии >30см, вектор поворота вычисляется исходя из направления движения руки и ее скорости
* Масштабирование (zoom in\out) происходит аналогично алгоритму, описанному выше при работе с 2D моделями.

### 5.4.3. Реализация модуля аналитики

Данный модуль нужен, чтобы хирурги и их помощники получали различную информацию о поверхности пациента во время операции. Теоретически его можно использовать во всех ситуациях, когда нужно определить характеристики сцены в интервале глубины от 60 до 200 см.

Данный модуль - единственный, в котором на данный момент используется распознавание пальцев, потому как в остальных компонентах жесты рук вполне удовлетворяли требованиям удобства, когда как в этом модуле требует более точная локализация отмечаемых точек.

#### 5.4.3.2. Функциональность модуля аналитики

Система может предоставить пользователю требуемый функционал:

* вычисление расстояния между 2 точками
* вычисления объема слоя среза на теле пациента

Для начала опишем общие моменты для всех вариантов поведения.

После изучения различных вариантов реализации взаимодействия хирурга и системы в процессе анализа тела пациента было приятно решение использовать 2 контроллера Kinect. На рисунке 5.2 изображена схема расположения сенсоров и участников операции:

* “Kinect 1” висит непосредственно над пациентом.
* “Kinect 2” стоит рядом с монитором в отдалении от операционного стола, и когда требуется провести анализ поверхности хирург или его помощник подходит к экрану и требуемый анализ.

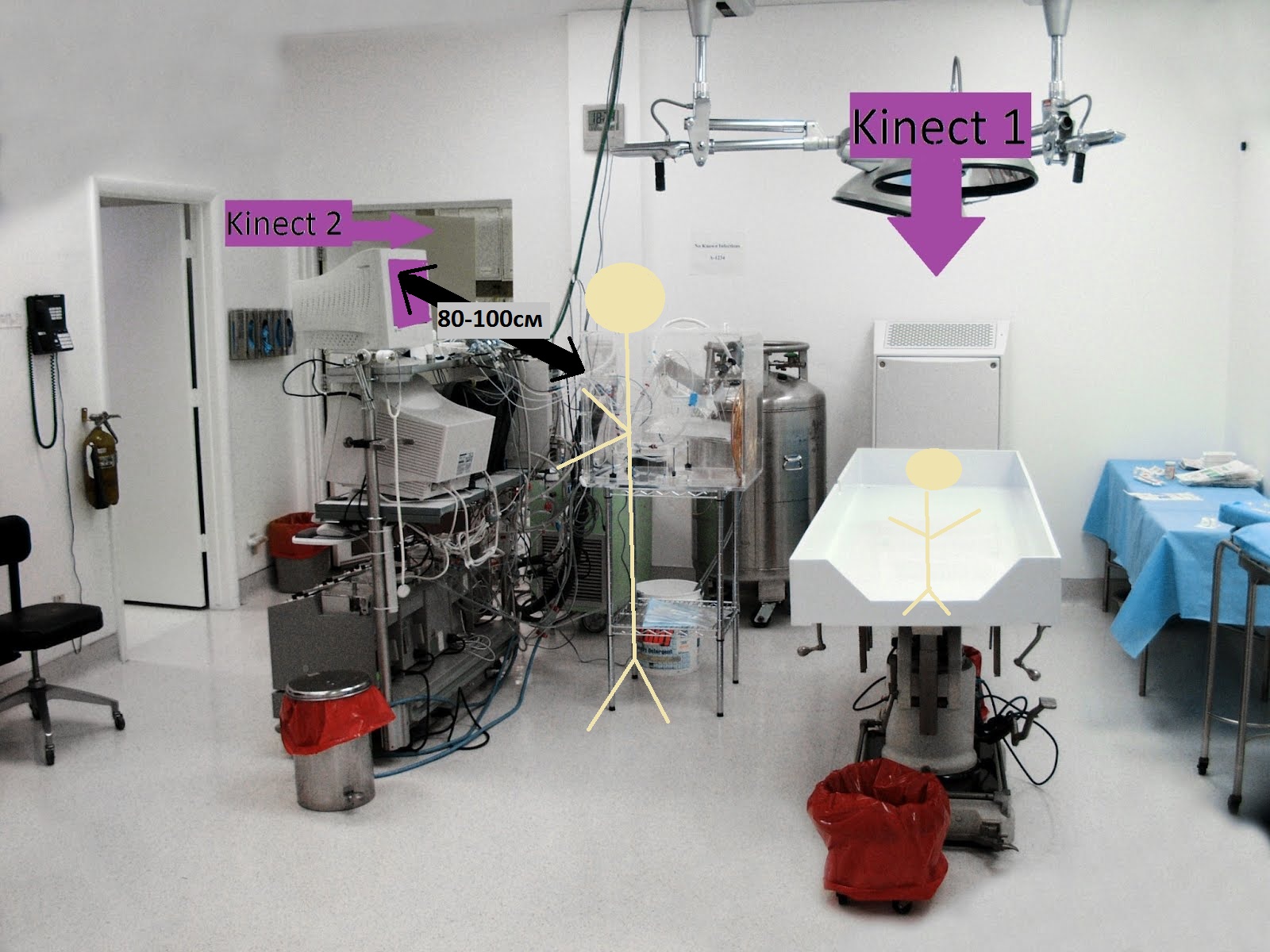


рис. 5.2. Примерная схема расположения Kinect сенсоров в операционной и участников операции

На первом шаге по определенной команде системы сенсор “Kinect 1” «фотографирует» пациента, лежащего на операционном столе, запоминая цветную фотографию и карту глубины сцены.

Далее хирург подходит к компьютеру и встает напротив “Kinect 2” на расстоянии 80-100см и анализирует поверхность тела пациента.

Такой подход имеет несколько преимуществ, по сравнению с подходом, при котором используется 1 Kinect сенсор, который одновременно следит за пациентом и хирургом:

1. Взаимодействие с пациентом сводится к минимуму.
2. Операция может вообще не прерываться во время процесса вычисления характеристик поверхности тела, если анализом занимается помощник.
3. Не требуется осуществлять поиск “идеального” угла, при котором будут видны и хирург, и пациент.
4. Снижается шумовое влияние других участников операции на процесс анализа, так как “Kinect 2” можно поставить где угодно.

Для всех вариантов анализа поверхности требуется возможность отмечать точку на фотоснимке. Это действие реализуется через следующий жест пользователя:

1. На правой руке загнуты все пальцы кроме одного.
2. Расстояние от левой руки до груди > 30 см. Если требуется одиночная отметка точки, а не продолжительное отмечание, то жест считается совершенным, когда значение расстояния до груди становится >30см.

Далее рассмотрим каждый вариант поведения в отдельности.

С точки зрения пользователя, вычисления расстояния между 2-мя точками выглядит следующим образом:

1. Пользователь отмечает 2 точки на фотографии пациента, полученной от “Kinect 1”
2. На экран выводится отрезок, соединяющий эти две точки, и длина соединяющей их линии.

В данном контексте под “линией” понимается не отрезок, соединяющий две точки в 3D пространстве, а ломаная, полученная дроблением этого отрезка на некоторое количество малых отрезков, которые, соединившись вместе, аппроксимируют кривую, проведенную по поверхности сцены от одной точки к другой. Проекция этой кривой на плоскость камеры глубины Kinect сенсора будет совпадать с отрезком.

С точки зрения пользователя, вычисление объема снимаемой ткани с пациента выглядит следующим образом:

1. Пользователь вводит значение толщины покрова с клавиатуры.
2. Пользователь обводит на сохраненной фотографии замкнутый контур.
3. На экран выводится контур, отмечаются внутренние точки контура и показывается объем 3D слоя, замкнутого по введенному контуру.

Для вычисления объема требуется сначала вычислить площадь 3D поверхности, затем умножить на введенную глубину.

Далее подробнее рассмотрим компоненту, отвечающую за распознавание пальцев и то, как оно работает.

#### 5.4.3.1. Алгоритм распознавания пальцев

Алгоритм распознавания пальцев интуитивно несложный. Его можно разбить на несколько шагов:

1. На вход поступает точка центра ладони (получаемая от Kinect)
2. Любым алгоритмом типа FloodFill [12] происходит “заливка” всех точек вокруг центра ладони до резкого “прыжка в глубину”
3. Для каждой точки, на которой остановился алгоритм FloodFill, находим предыдущую просмотренную и отмечаем ее кодом “Точка контура”
4. Вычисляется выпуклая оболочка “Точек контура”
5. Остается небольшое количество точек, отсортированных по часовой стрелке, которые можно отсекать различными эвристиками. Эвристики, которые использовались в работе:
   1. Отсекание ближних точек: отсекаются точки, достаточно близкие к центру ладони, это позволяет не распознавать в сжатом кулаке пальцы
   2. Отсекание запястья: точка отсекается, если угол относительно центра ладони между ней и точкой запястья (получаемой от Kinect) достаточно мал
   3. Прореживание: из каждой группы точек близких друг к другу берется лишь одна точка, самая дальняя от центра ладони

На выходе алгоритма получаем несколько точек, в общем случае хорошо определяющие положения пальцев пользователя.

Отдельно стоит обсудить некоторые пункты алгоритма.

Например, пункт применения алгоритма типа FloodFill: существуют различные алгоритмы заполнения произвольной замкнутой области. На первом этапе разработки использовался самый простой чисто рекурсивный алгоритм:

1. Начинаем с некоторой точки
2. Если точка не была уже посещена и не является граничной точкой, то:
   1. считаем точку посещенной
   2. повторяем пункт 1 для соседних 4 (или 8) точек.

Данный алгоритм в силу своей рекурсивной природы очень жаден к стеку вызовов. Соответственно при большой площади заполнения, могло происходить переполнение стека.

В итоге был использован другой алгоритм, который работает через цикл и не использует рекурсию [13].

Алгоритм примерно состоит в следующем:

1. Начинаем с некоторой точки, кладем ее в очередь
2. Если очередь пуста, заканчиваем алгоритм, иначе достаем из очереди точку и считаем ее текущей
3. Если текущая точка была посещена, переходим к пункту 2.
4. Если текущая точка находится на границе, добавляем ее в список граничных точек и переходим к пункту 2.
5. Если текущая точка не принадлежит границе, идем вправо и влево от текущей точки пока не дойдем до границ области, получаем соответственно значения Xleft и Xright
6. Все точки, для которых выполнены условия:
   * 1. Y-координата отличается от Y-координаты текущей точки на 1
     2. X-координата находится в интервале (Xleft,Xright)

добавляем в очередь.

1. Переходим к пункту 2.

Используемый алгоритм хорошо справляется с задачей нахождения контура ладони.

Далее, после получения списка точек на границе происходит построение выпуклой оболочки, которое было реализовано по алгоритму Грехема [14], который справляется с задачей за время O (N log N) от числа точек контура ладони.

Оставшиеся эвристики не являются алгоритмически сложными, и их реализация не составляет труда.

#### 5.4.3.3. Алгоритм вычисления объема слоя

Алгоритм нахождения площади ограниченной контуром поверхности можно описать так:

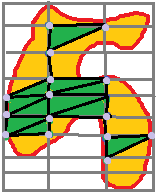
* Поверх контура кладется воображаемая сетка. Размер ячейки определяет точность определения площади. Чем меньше размер, тем больше точность.
* Рассматриваются только те узлы сетки, которые лежат внутри минимального прямоугольника, содержащего контур.
* Для каждого узла сетки вычисляется, принадлежит ли он контуру или нет. На рис. 5.3 точками изображены внутренние узлы.

Рис. 5.3. Определение площади слоя

* + Узел с координатами (x,y) принадлежит контуру тогда и только тогда, когда отрезок [(x,y),(0,y)] пересекается нечетное количество раз.
* Каждая ячейка делится на два треугольника.
* Для каждого треугольника, который полностью оказался внутри контура, вычисляется площадь.

В итоге, общая сумма площадей треугольников будет примерно равна реальной площади 3D поверхности, ограниченной контуром.

#### 5.4.3.4. Анализ погрешности вычислений

Для определения точности выходных данных при оценке объема слоя и расстояния между точками производилось тестирование на дальности 80-100 см.

Для тестирования расстояния между точками были использованы 4 различные поверхности. В таблице отражены первые замеры для каждой поверхности:

В таблице 5.1 приведено по одному замеру для каждой поверхности.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Тело | Изгиб гитары | Сильно согнутый листок бумаги | Прямая плоскость | Средняя погрешность |
| Длина отрезка (см) | Real: 23  Out: 23 | Real: 32  Out: 34 | Real: 25  Out: 24 | Real: 54  Out: 52 | 2-4% |
| Площадь внутренности контура (см2) | Real: 606  Out: 612 | - | Real: 162  Out: 150 | Real: 1419  Out: 1501 | 2-6% |

Таблица 5.1. Результаты тестирование на различных поверхностях. Обозначения:

*Real* - истинное значение

*Out* – результат работы программы

Во время тестирования было выявлено, что погрешность больше зависит от качества нанесения меток пальцами, чем от дальности или кривизны поверхности.

Погрешности в нанесении меток могут быть следствием неаккуратности движений пользователя или секундной ошибке Kinect-а в распознавании скелета.

Распознавание скелета становится наиболее чувствительным местом в системе, так как при различных углах поворота тела пользователя или сильном перекрещивании рук контроллер иногда «теряется» и сбивает модель скелета, что в данной реализации сбивает расположение пальцев.

После выхода Kinect 2 данное слабое место, скорее всего, перестанет быть актуальным из-за повышенной точности сенсора и обновленным алгоритмам распознавания скелета.

Полученная погрешность вычислений длины отрезка лежит в рамках от 2% до 4%, что очень мало, притом, что погрешность в несколько сантиметров - это естественная человеческая погрешность при нанесении меток на фото пальцами.

При вычислении площади погрешность, бывает, достигает 7-8%. Тем не менее, в среднем получаемая погрешность вполне удовлетворяет требованиям точности.

# 6. Итоги

В итоге была проделана серьезная работа. Требуемая функциональность реализована.

## 6.1. Достигнутые результаты

Результатом данной работы является работающая система, состоящая их 3-х функциональных частей:

1. Просмотр классических растровых изображений и DICOM-снимков при помощи Kinect, в который входит:
   1. Листание на один кадр вперед/назад
   2. Быстрое листание кадров вперед/назад
   3. Масштабирование

Рис. 6.1.1. Пример работы модуля просмотра 2D моделей

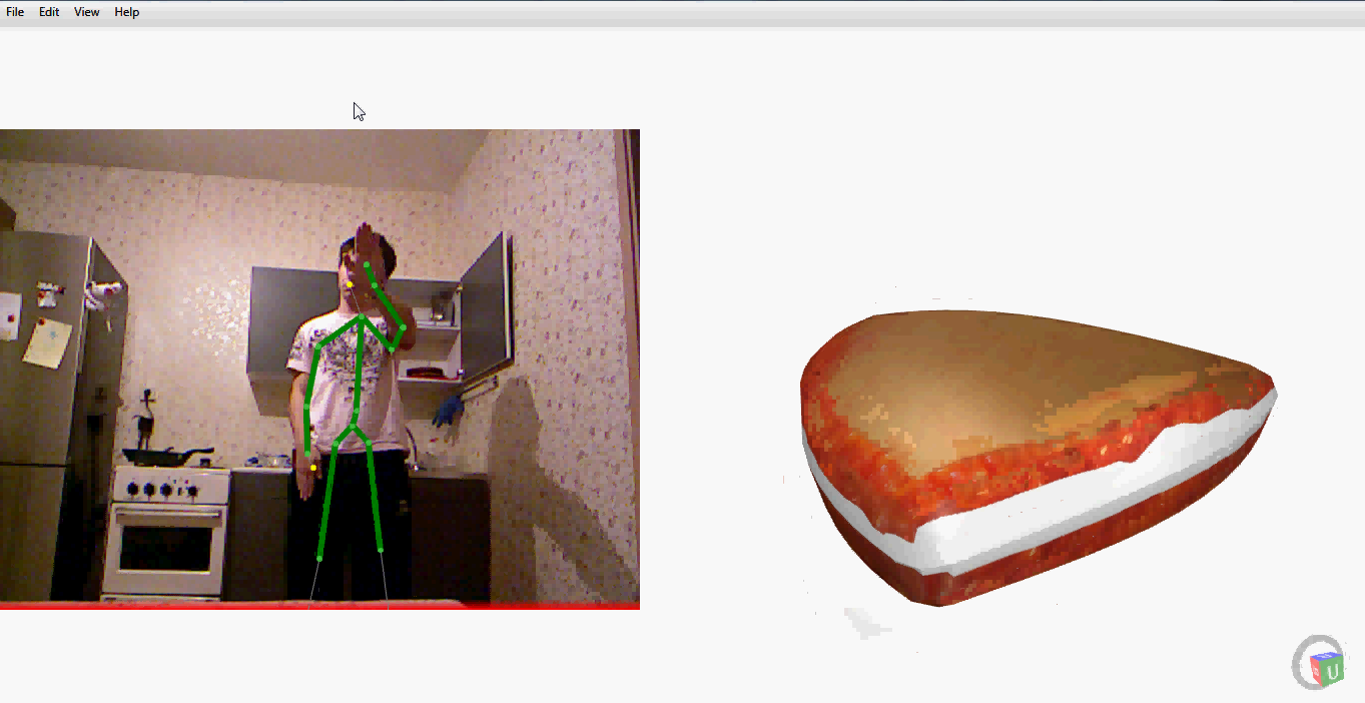
1. Просмотр классических 3D моделей:
   1. Поворот
   2. Масштабирование

Рис.6.1.2. Пример работы модуля просмотра 3D моделей

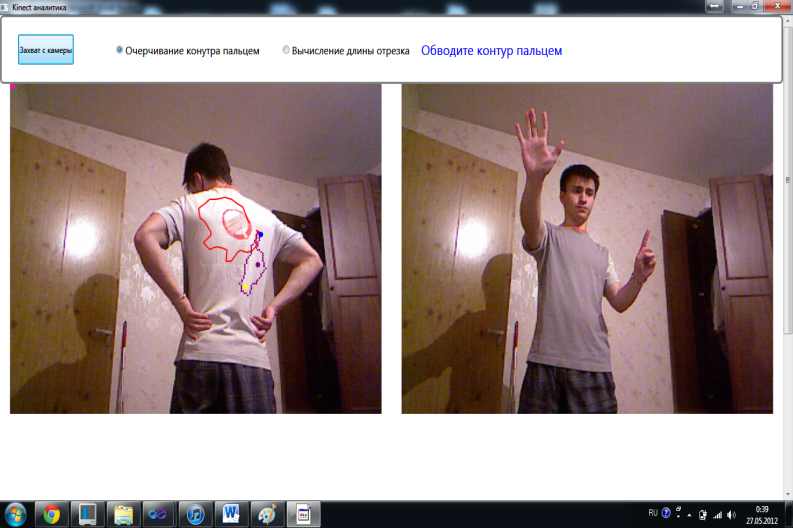
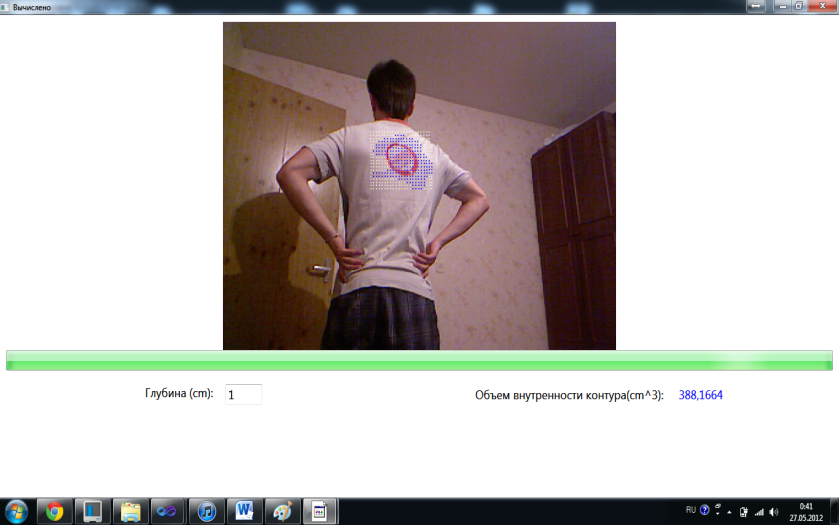
1. Анализ количественных характеристик поверхности сцены:
   1. Вычисление расстояния между двумя точками на поверхности
   2. Вычисление глубины слоя, ограниченного контуром

рис. 6.1.3. Пример работы модуля аналитики. Обведение контура и вычисление объема.

Погрешность вычислений при анализе поверхности вполне приемлема для использования данной системы в реальных условиях для примерной оценки расстояния между точками и объема ограниченного слоя.

Разработанная система функционирует и после некоторых доработок в UI готова для внедрения.

## 6.2. Возможности развития

Данная система имеет некоторые слабые места, которые можно усилить при развитии данной идеи.

1. Алгоритм распознавания пальцев можно улучшить так, чтобы происходило более гладкое перемещение пальцев без резких “прыжков”
   1. Например, можно запоминать несколько предыдущих значений положений пальцев, и высчитывать новое положение исходя из них.
2. Вычисление длины отрезка зависит от угла наклона камеры Kinect к сцене. Соответственно, при наличии сложной сцены и при плохой постановке Kinect над пациентом отрезок может совпадать с тем, которого ожидает пользователь.
   1. Возможно, для этой цели стоило реализовать вычисление длины ломаной, которую провел пользователь.
3. Для полной стерильности в операционной стоит перевести все взаимодействие с программой на бесконтактную основу. Для этого потребуется:
   1. Внедрение распознавания речи для различных команд
   2. Введение глубины слоя при помощи Kinect

# 7. Заключение

Медицина не стоит на месте, соперничество между различными медицинскими клиниками выливается в ежедневное внедрение новых технологий с целью повышения качества медицинского обслуживания и собственной привлекательности.

Ведущие IT-компании борются за лидерство в сфере медицины с помощью создания новых систем или адаптации старых [10].

Данная работа представила очередной шаг в пути внедрения технологии Microsoft Kinect в медицину, и, в особенности, хирургию. Система позволит малыми затратами улучшить качество хирургических операций, поднять престиж клиники.

Существует также много других всевозможных применений технологии Kinect в медицине. Наверняка на данный момент в мире в рамках различных научных или коммерческих проектов разрабатываются сотни подобных продуктов, которые будут внедрены в реальные клиники и больницы.

Можно быть уверенным, что рано или поздно технология Kinect приживется и будет принята работниками сферы здравоохранения по всему миру.

# 8. Список литературы

1. [Понятие структурированного](http://en.wikipedia.org/wiki/Structured-light_3D_scanner) света;   
   <http://en.wikipedia.org/wiki/Structured-light_3D_scanner>
2. Библиотека Libfreenect; http://openkinect.org/wiki/Main\_Page
3. [Библиотека OpenNI](http://75.98.78.94/default.aspx); http://75.98.78.94/default.aspx
4. Библиотека [KinectForWindows](http://kinectforwindows.org); http:/kinectforwindows.org
5. [Сравнение KinectForWindows Beta и OpenNI](http://www.brekel.com/?page_id=671); <http://www.brekel.com/?page_id=671>
6. [Технология](http://en.wikipedia.org/wiki/Microsoft_Speech_API)  [Microsoft](http://en.wikipedia.org/wiki/Microsoft_Speech_API) Speech; <http://en.wikipedia.org/wiki/Microsoft_Speech_API>
7. DICOM формат файла; <http://ru.wikipedia.org/wiki/DICOM>
8. Библиотека 3DTools; <http://3dtools.codeplex.com/>
9. Библиотека HelixToolKit; <http://helixtoolkit.codeplex.com/>
10. Статья о притязаниях Microsoft на сферу медицины;  
    <http://www.thevista.ru/page.php?id=14452>
11. Библиотека распознавания движения пальцев Candescent NUI;  
    <http://candescentnui.codeplex.com/>
12. Алгоритмы FloodFill закрашивания произвольной области;  
    <http://en.wikipedia.org/wiki/Flood_fill>
13. Различные алгоритм заливки областей;  
    <http://algolist.manual.ru/graphics/fill.php>
14. Алгоритм Грехема нахождения выпуклой оболочки;  
    <http://algolist.manual.ru/maths/geom/convhull/graham.php>
15. Статья посвященная обзору контроллера Kinect;  
    <http://robocraft.ru/blog/technology/641.html>
16. Инструмент конвертации изображений в различные форматы;  
    <http://www.imagemagick.org/script/index.php>
17. Статья о Kinect 2;  
    <http://www.eurogamer.net/articles/2011-11-25-kinect-2-so-accurate-it-can-lip-read>
18. Описание контроллера Asus WAVI Xtion; http://www.thg.ru/technews/20110603\_084500.html
19. Система ViEye; http://www.comvisionsys.com/product/