

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем

Системное программирование

Платонов Владислав Николаевич

Framework для применения дополненной реальности в медицинских процедурах

Выпускная квалификационная работа

Научный руководитель:
ст. преп. Смирнов М. Н.

Рецензент:
инженер-программист
ООО "Системы КМ" Менькин С. А.

Санкт-Петербург
2018

SAINT-PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems
Software Engineering

Vladislav Platonov

Framework for using augmented reality technology in medicine

Graduation Project

Scientific supervisor:
senior lecturer Mikhail Smirnov

Reviewer:
software engineer
CM Systems LLC Sergei Monkin

Saint-Petersburg
2018

Оглавление

Введение	4
1. Постановка задач	6
2. Обзор существующих решений и технологий	7
2.1. Методы совмещения	7
2.2. Технологии визуализации	9
2.3. Существующие системы дополненной реальности для операционных	10
2.4. Используемые технологии	12
3. Архитектура фреймворка	13
3.1. Выработка требований и сценарии использования	13
3.2. Компоненты системы	14
4. Реализация	17
4.1. Совмещение 3D-моделей	17
4.2. Загрузка и визуализация КТ-слоёв на пациенте	18
4.3. Проведение интраоперационных измерений	19
4.3.1. Получение 3D-модели сцены	19
4.3.2. Функции интраоперационных измерений	20
Заключение	23
Список литературы	24

Введение

Технологии дополненной реальности - совмещения виртуальных объектов с объектами реального мира - находят применение в медицине. При планировании медицинских процедур и операций врачи часто руководствуются проведёнными заранее исследованиями пациента, такими как КТ-сканирование. Такие данные позволяют, в частности, определить местоположение, строение и структуру различных тканей, спланировать расположение медицинских инструментов. При этом для достижения лучшего результата требуется точно и наглядно представить эти данные врачу во время операции. Использование технологий дополненной реальности позволяет наложить предварительно сделанную поверхностную или КТ-модель на получаемое в реальном времени изображение пациента. К задачам, решаемым в ходе разработки систем дополненной реальности, относятся отслеживание положения пациента, наложение на пациента заранее подготовленных 3D-моделей, визуализация отдельных органов, КТ-снимков и дополнительной информации, а также проведение интраоперационных измерений.

Для реконструкции сцены и отслеживания в ней положения пациента могут применяться различные устройства. До недавнего времени для этой цели использовались в основном камеры и трёхмерные сенсоры, такие как Microsoft Kinect и Intel RealSense. Они оставались неподвижными в ходе операции и позволяли выводить изображение построенной сцены на экран монитора. С появлением очков дополненной реальности, таких как Microsoft HoloLens, стало возможным создание более удобных приложений, которые строят сцену вокруг перемещающегося пользователя, отправляют изображения на экран перед глазами пользователя и предоставляют возможность управления с помощью жестов. При этом возможности очков дополненной реальности также ограничены точностью сканирования. Эти устройства ориентированы на построение модели всего помещения, в котором они находятся, и эта модель может быть недостаточно подробной для задач точного измерения расстояний и объёмов. К тому же очки, реализующие трекинг

и позиционирование объектов с высокой точностью, всё ещё не ориентированы на широкий круг пользователей. Поэтому использование камер и трёхмерных сенсоров, как в сочетании с очками, так и без них, остаётся актуальным.

Дополненная реальность может быть использована при проведении различных процедур. Методы отслеживания пациента и визуализации виртуальных объектов также зависят от используемого аппаратного обеспечения. При этом такие приложения различными методами решают общие задачи. Таким образом, представляет интерес задача разработки фреймворка для применения дополненной реальности в медицинских процедурах.

1. Постановка задач

Целью работы является разработка фреймворка для создания приложений, использующих дополненную реальность для планирования и проведения медицинских процедур.

Для достижения этой цели были поставлены следующие задачи:

- выполнить обзор существующих решений и технологий;
- разработать архитектуру фреймворка;
- реализовать функциональность фреймворка:
 - реализовать совмещение 3D-модели с пациентом в реальном времени;
 - реализовать загрузку и визуализацию КТ-слоёв на пациенте;
 - реализовать функции получения модели и интраоперационных измерений на ней.

2. Обзор существующих решений и технологий

2.1. Методы совмещения

Для всех виртуальных объектов, загруженных в сцену, построенной в реальном времени 3D-модели сцены, а также сканирующего устройства определены координаты и углы поворота относительно осей координат. Реконструируемая в реальном времени модель пациента и загруженная 3D-модель могут быть совмещены применением к загруженной модели евклидова преобразования, которое может быть выражено комбинацией поворота и переноса [5]. Таким образом, для того, чтобы получаемая в реальном времени модель пациента была совмещена с загруженной моделью и другими данными, нужно, чтобы их координаты и углы поворота совпадали. Для этого нужно распознать в сканируемой сцене пациента и определить его координаты.

Многие методы, используемые в дополненной реальности для распознавания объектов в сцене описаны в [4]. Один из них использует двумерные маркеры. На видеопотоке детектируется заранее заданное изображение. Зная координаты пикселей, соответствующих углам маркера, а также его размеры, можно рассчитать матрицу гомографии и таким образом определить положение камеры относительно маркера в трёхмерном пространстве. Этот метод реализован в библиотеках дополненной реальности Vuforia¹ и ARToolkit². Его преимуществами являются простота реализации и использования. Кроме того, он может быть применён на любой платформе, на которой доступен видеопоток с камеры, и не требует дополнительного аппаратного обеспечения. Однако этот метод подразумевает получение координат только четырёх точек на поверхности пациента, что не позволяет проводить интраоперационные измерения. При этом расположение маркера должно быть выбрано таким образом, чтобы его углы не смещались по отношению к пациен-

¹Библиотека Vuforia, <https://www.vuforia.com> (дата обращения 22.04.2018)

²Библиотека ARToolkit, <https://github.com/artoolkit> (дата обращения 22.04.2018)

ту в ходе операции. Для стабильного распознавания маркеры должны быть достаточно большими, что также затрудняет их использование в операционной.

Другим решением задачи совмещения может быть распознавание в сцене объёмных маркеров. Библиотека Vuforia предлагает в качестве маркеров задавать трёхмерные объекты и детектировать их по видеопотоку. Задание модели пациента в качестве маркера могло бы упростить наложение виртуального объекта на реальный, так как не пришлось бы определять положение объекта относительно маркера. Однако данный метод, применимый для простых объектов, сложно использовать для распознавания лица пациента, так как маркер должен представлять собой жёсткую модель, геометрия которой не меняется при сканировании. В силу подвижности тканей лицо не является таким объектом, кроме того форма и объёмы отдельных областей могут меняться в ходе операции.

В качестве маркеров могут быть использованы ключевые точки на лице пациента. В библиотеке для работы с сенсором Kinect 2 Kinect for Windows SDK v2 [3] реализованы функции распознавания лица, которые, используя данные камеры и сенсора глубины, позволяют определить координаты более тысячи точек на лице. Выбрав любые три из них и зная координаты соответствующих точек на модели, можно совместить модель с лицом пациента.

Для проведения безмаркерной регистрации требуется реконструировать 3D-модель сцены. Такая реконструкция может проводиться с использованием камер глубины, доступных на 3D-сенсорах. При отсутствии доступа к данным глубины может быть использована реконструкция модели по видеопотоку [1]. После построения поверхностной модели сцены её требуется совместить с заранее подготовленной. Для этого может быть использована одна из реализаций итеративного алгоритма ближайших точек³. Использование этого алгоритма может быть вычислительно сложным на высокодетализированных моделях. Кроме

³Итеративный алгоритм ближайших точек, Besl P., McKay H. / A method for registration of 3-d shapes // IEEE Transactions on Pattern Analysis and Machine Intelligence. - 1992. - Vol. 14, no.2.-Pp.239-256.

того, для качественного совмещения требуется отфильтровать геометрию реконструируемой модели, удалив из неё полигоны, не относящиеся к пациенту.

Разработанный фреймворк позволяет проводить совмещение с использованием любых маркеров. При разработке использовалось распознавание ключевых точек на лице, так как данный метод предоставляет достаточно точные координаты точек на поверхности лица пациента и не требует больших вычислительных ресурсов. Для платформ, не предоставляющих доступ к картам глубины, использовалась регистрация с помощью двумерных маркеров Vuforia.

2.2. Технологии визуализации

При использовании камер и трёхмерных сенсоров вывод видеопотока и наложенной на него информации производится на экран монитора. С появлением очков дополненной и виртуальной реальности появилась возможность выводить эту информацию на экран перед глазами пользователя и таким образом сделать процесс использования приложения удобнее.

Для разработки приложений для очков дополненной реальности выбрана платформа Windows Mixed Reality (WMR)⁴. Существует несколько устройств, поддерживающих эту платформу. Одним из них является Microsoft HoloLens. Это устройство обладает набором датчиков и камер, что позволяет ему строить 3D-модель окружающего пространства. С их помощью также производится отслеживание положения самих очков в сцене, что позволяет привязывать голограммы к окружающему миру. Кроме того, HoloLens обладает прозрачным дисплеем, что позволяет пользователю напрямую видеть реальный мир, дополненный виртуальными моделями.

Недостатком платформы Windows Mixed Reality является отсутствие доступа приложения к данным карты глубины. Вместо этого приложению доступна построенная модель сцены, которая не являет-

⁴Платформа Windows Mixed Reality, <https://www.microsoft.com/en-us/windows/windows-mixed-reality> (дата обращения 22.04.2018)

ся достаточно детализированной для задачи измерений длин и объёмов на поверхности пациента. Поэтому актуальным остаётся использование отдельных 3D-сенсоров. В исследовании [2] совместно с очками используется сенсор Intel RealSense для получения карт глубин.

С учётом преимуществ работы с очками дополненной реальности, а также необходимости в некоторых случаях использовать другие устройства, описываемый в данной работе фреймворк предусматривает работу с использованием камер, сенсоров, предоставляющих доступ к картам глубин, а также устройств, работающих на платформе Windows Mixed Reality.

2.3. Существующие системы дополненной реальности для операционных

В настоящее время разрабатываются и уже реализованы различные проекты, направленные на ассистирование врачам во время операций (табл. 1). Перечислим некоторые из них:

- Scopis Holographic Navigation Platform⁵ - платформа для планирования операций и помощи врачу во время их проведения, работающая на очках дополненной реальности HoloLens. Во время операции система помогает врачу правильно расположить инструменты, проецируя их модели и дополнительную информацию в очки.
- Novarad OpenSight⁶ - платформа, позволяющая накладывать на изображение пациента заранее подготовленные 3D-модели, созданные по МРТ и КТ-снимкам. Благодаря визуализации анатомии пациента врачи могут планировать операции, проводить исследования и использовать эти модели для обучения.

⁵Платформа Scopis Holographic Navigation, URL: <https://holographic.scopis.com> (дата обращения 21.12.2017)

⁶Платформа Novarad OpenSight, URL: <https://www.novarad.net/products/opensight> (дата обращения 22.04.2018)

- Исследователи Университета Дьюка разработали приложение, использующее HoloLens для визуализации моделей органов и детектирования хирургических инструментов⁷. При этом отсутствует отслеживание положения пациента и модель позиционируется пользователем вручную.
- Исследователи Microsoft⁸ создали систему дополненной реальности для ассистирования хирургу с использованием сенсора Kinect. В ней реализовано совмещение моделей органов пациента с его изображением. При этом пользователю необходимо перемещать сенсор, чтобы сначала построить модель пациента, а затем совместить модели.
- В [6] описывается приложение дополненной реальности для операционных Holographic Human, работающее на очках HoloLens. Реализуется функциональность визуализации отдельных органов и отслеживания положения инструментов, которое проводится с помощью маркеров Vuforia. Модели органов перемещаются с помощью жестов.

Эти примеры показывают применимость дополненной реальности в задачах ассистирования врачу во время операций. Интерес вызывает задача совмещения голограмм с пациентом, так как не во всех примерах эта задача решается автоматически. Кроме того, данные решения не заявляют функциональности измерений поверхности пациента в ходе операции. В разрабатываемом решении реализуются отслеживание пациента, совмещение с ним КТ-модели, визуализация КТ-снимков, а также функции измерения поверхности пациента.

⁷Проект Neurosurgical HoloLens Adaptation, URL: <https://sites.duke.edu/braintool/2016/03/31/neurosurgical-hololens-adaptation/> (дата обращения 22.04.2018)

⁸Kinect + Brain Scan = Augmented Reality for Neurosurgeons, URL: <https://spectrum.ieee.org/automaton/robotics/medical-robots/microsoft-kinect-fusion-augmented-reality-neurosurgeons>

	Оборудование	Отслеживание позиции че- ловека	Наложение голограмм на пациента
Scopis Navigation Platform	HoloLens, ИК-камера	+	+
Novarad OpenSight	HoloLens	+	+
Duke University	HoloLens		+
Cambridge Microsoft	Kinect	+	+
Holographic Human	HoloLens		+

Таблица 1: Существующие решения

2.4. Используемые технологии

Для разработки использовался игровой движок Unity⁹. Он позволяет создавать кроссплатформенные графические приложения. Его использует библиотека MixedRealityToolkit-Unity¹⁰, применяемая для работы с WMR-устройствами. Для детектирования двумерных маркеров и позиционирования объектов с их помощью используется библиотека Vuforia. Для распознавания особых точек на лице и получения карт глубины с сенсора Kinect используется сборка Kinect for Windows SDK для Unity. Для загрузки 3D-моделей в формате OBJ и данных КТ-сканирования в формате DICOM применяются библиотеки Runtime OBJImporter¹¹ и OpenDICOM.NET¹². Описываемый фреймворк собирается под универсальную платформу Windows¹³, что позволяет разрабатывать приложения, работающие на Windows-устройствах, в том числе использующих платформу Windows Mixed Reality.

⁹Игровой движок Unity, URL: <https://unity3d.com/ru> (дата обращения 22.04.2018)

¹⁰Библиотека MixedRealityToolkit-Unity, URL: <https://github.com/Microsoft/MixedRealityToolkit-Unity> (дата обращения 22.04.2018)

¹¹Библиотека Runtime OBJ Importer, <https://assetstore.unity.com/packages/tools/modeling/runtime-obj-importer-49547> (дата обращения 30.05.2018)

¹²Библиотека OpenDICOM, <https://http://opendicom.sourceforge.net> (дата обращения 30.05.2018)

¹³Документация по универсальной платформе Windows, <https://docs.microsoft.com/ru-ru/windows/uwp/index> (дата обращения 27.05.2018)

3. Архитектура фреймворка

3.1. Выработка требований и сценарии использования

При проектировании были выработаны требования к фреймворку. Он должен предоставлять пользователю следующие возможности:

- видеть пациента, лежащего на столе,
- видеть КТ и МРТ-слои, совмещённые с пациентом,
- видеть трёхмерную поверхностную модель, совмещённую с лежащим пациентом,
- измерять объёмы поверхностей.

Согласно основному сценарию использования, работа пользователя с приложением разделена на несколько шагов:

- загрузить 3D-модель и КТ-снимки,
- определить на модели точки, соответствующие маркерам,
- выбрать требуемые КТ-слои и установить прозрачность модели,
- выделить область на изображении пациента,
- провести измерения.

Этим действиям соответствуют состояния программы, изображённые на рис. 1. Состояние `Idle` является состоянием ожидания действий пользователя. В состоянии `ImportOBJ` и `importDICOM` программа входит при загрузке 3D-моделей и КТ-слоёв. В состоянии `TrackPatient` программа находится всё то время, когда распознаётся пациент в сцене. При этом программа может находиться в состоянии выделения точек на модели (`SpecifyPoints`), измерения объёмов (`MeasureVolume`), визуализации КТ-снимков (`ShowCT`) и их скрытия (`HideCT`). Наложённая модель визуализируется, и её прозрачность может регулироваться пользователем.

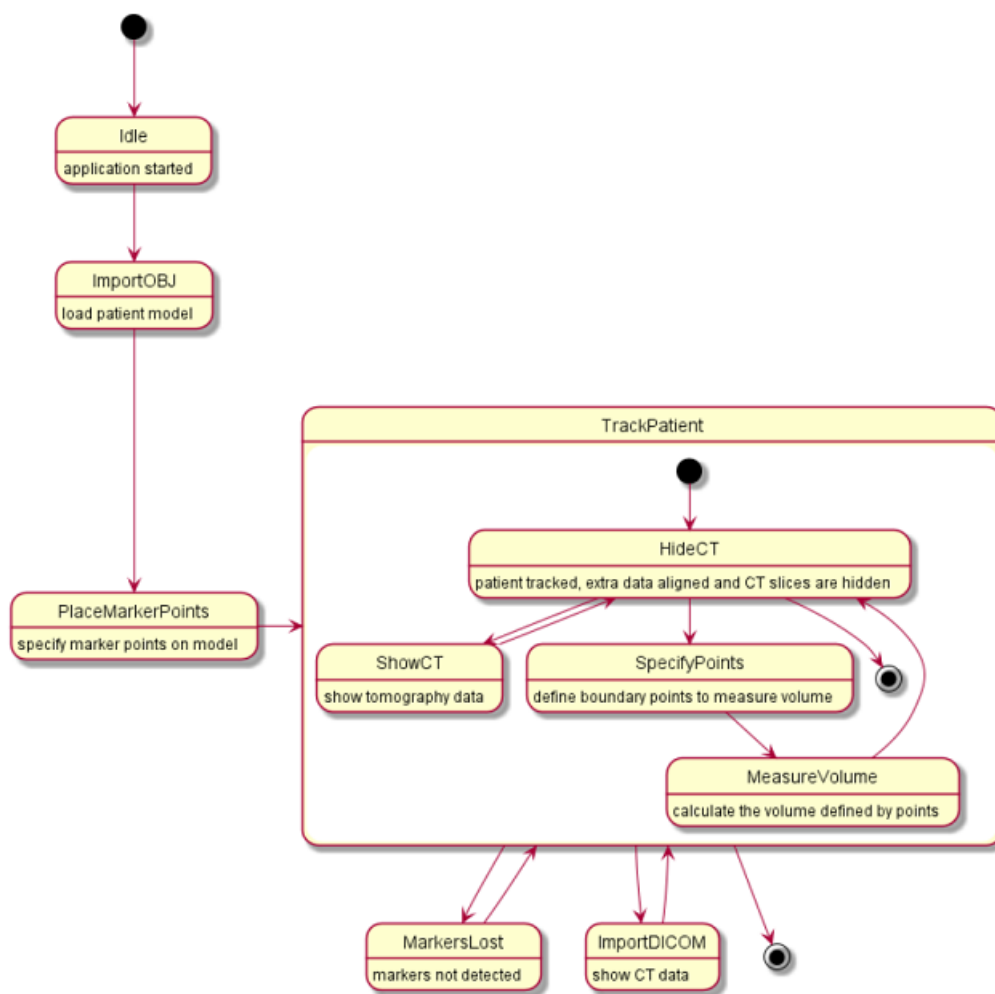


Рис. 1: Диаграмма состояний

3.2. Компоненты системы

Структурно фреймворк состоит из нескольких компонентов, отвечающих за решение отдельных задач. В зависимости от платформы и выбранных алгоритмов их реализации могут отличаться, но интерфейс взаимодействия между компонентами остаётся неизменным. Диаграмма компонентов изображена на рис. 2.

Компонент Scene отслеживает объекты находящиеся в сцене, отслеживает состояние, в котором находится система и обрабатывает пользовательский ввод. Компонент Patient объединяет все данные, относящиеся к пациенту, в том числе совмещаемые 3D-модель и КТ-снимки. Компонент Recognition реализует распознавание положения пациента в сцене.

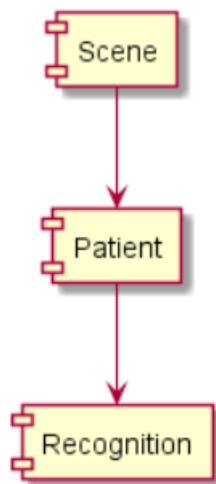


Рис. 2: Диаграмма компонентов

На рис. 3 представлена диаграмма классов. При запуске приложения создаются объекты классов `SceneManager` и `OnGUI`. Класс `OnGUI` содержит функции, которые вызываются при нажатии на элементы пользовательского интерфейса. `SceneManager` хранит состояние программы, генерирует уведомления и может породить остальные объекты сцены, в частности объект класса `Patient`, который содержит все данные, относящиеся к пациенту, а также управляет их визуализацией. В нём есть функции загрузки моделей в формате OBJ, которые вызывают библиотеку `ObjLoader`, а также данных томографии в формате DICOM. При загрузке DICOM-файлов используется библиотека `openDICOM`. Класс `Alignment` хранит координаты маркерных точек на модели и, получив координаты этих маркеров в сцене, выполняет преобразование координат объекта класса `Patient`. Функциональность получения маркерных точек сцены содержится в абстрактном классе `PatientRecognition`. Эта функциональность реализована в классе `KinectFaceRecognition`, использующем `Kinect SDK` и получающем ключевые точки на лице пациента и классе `VuforiaFaceRecognition`, получающем координаты маркера, сдетектированного библиотекой `Vuforia`. Также разработчик приложения может добавлять собственные компоненты распознавания маркерных точек. Объекты классов `Client` и `Server` создаются при необходимости передачи данных в очки дополненной реальности. Если приложение использует трёхмерные сенсоры,

предоставляющие карты глубины, могут быть созданы объекты классов *Reconstructor* и *Measurements*, отвечающие за построение 3D-модели сцены и измерения на ней.

Компоненту *Scene* соответствует игровой объект в Unity-сцене. Также в сцену включена система событий для управления графическим интерфейсом. Игровой объект, соответствующий компоненту *Patient*, создаётся в ходе работы приложения. Игровые объекты сцены и скрипты собираются в пакет формата *Unitypackage*, который может быть импортирован в другие unity-проекты. Для использования фреймворка разработчик приложения должен расположить на экране элементы интерфейса и может определить маркерные точки и классы, отвечающие за их распознавание.

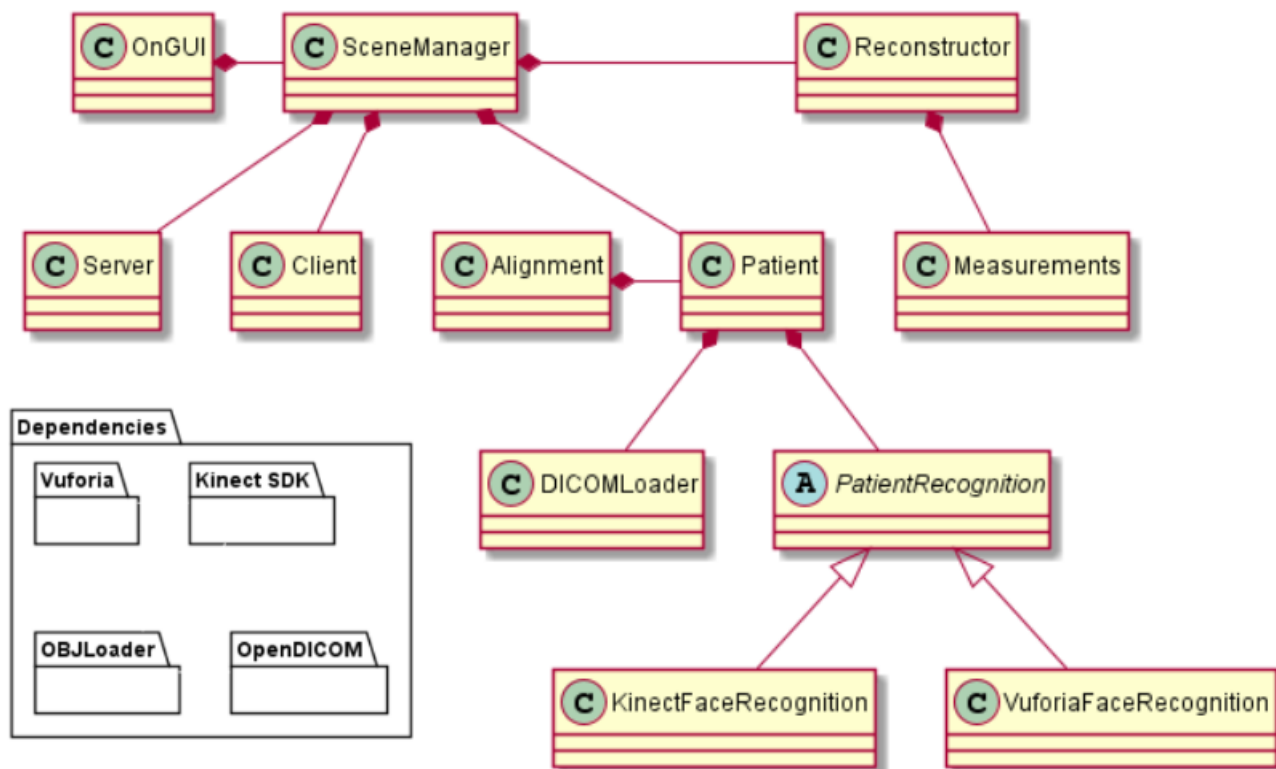


Рис. 3: Диаграмма классов

4. Реализация

4.1. Совмещение 3D-моделей

Совмещение модели пациента, полученной в реальном времени и модели, собранной из КТ-снимков, достигается совмещением распознаваемых на пациенте маркеров и точек на модели, соответствующих этим маркерам. Для этого при загрузке модели пользователь должен выделить точки на модели, по которым будет проводиться совмещение.

Пока точки на модели не выделены, её координаты не определены, однако их нужно задать для того, чтобы поместить модель в сцену. Для этого при загрузке модели определяются размеры ограничивающего её параллелепипеда. На основании угла обзора камеры и отношения ширины к высоте вычисляется дальность, на которой должна быть размещена модель для того, чтобы уместиться на экране. Координаты модели устанавливаются на главной оси камеры на соответствующем расстоянии от передней отсекающей плоскости камеры. Таким образом модель устанавливается непосредственно перед камерой, делая удобным выделение точек на ней.

Совмещение модели при отслеживании маркера реализовано в библиотеке *Vuforia*. При использовании в качестве маркеров особых точек на лице, распознаваемых библиотекой *Kinect for Windows SDK*, нужно совмещать эти модели по трём точкам.

На рис. 4 точки, соответствующие маркерам, обозначены как A , B и C . Соответствующие им точки на модели обозначены как A' , B' и C' . Сначала совмещаемая модель переносится таким образом, чтобы совместились точки A и A' , затем она поворачивается относительно этих точек на угол между векторами \vec{AC}' и \vec{AC} , чтобы совместились точки C и C' . После этого совмещённые пары точек образуют ось AC пересечения совмещаемых плоскостей (ABC) и $(A'B'C')$. Чтобы совместились точки B и B' , нужно повернуть совмещаемую модель на угол между этими плоскостями вокруг оси AC .

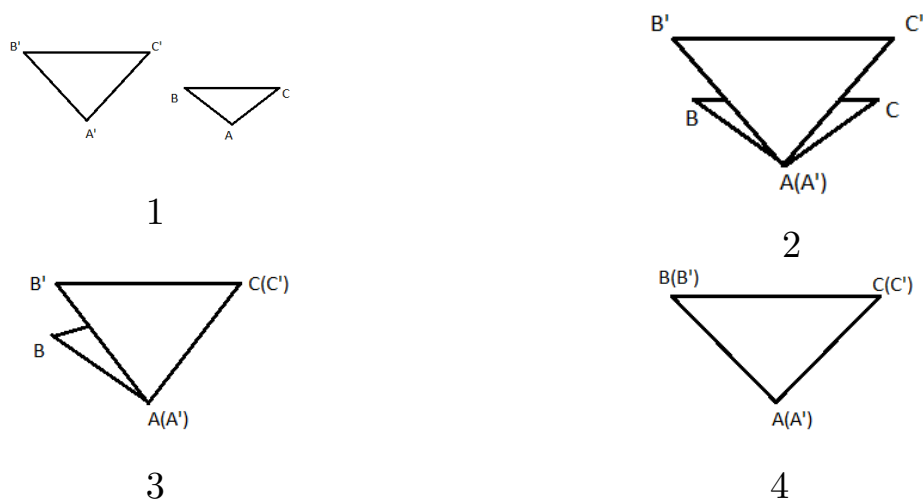


Рис. 4: Совмещение моделей по 3 точкам

4.2. Загрузка и визуализация КТ-слоёв на пациенте

Данные КТ-исследования загружаются в формате DICOM¹⁴. DICOM-файлы содержат в себе параллельные друг другу изображения, соответствующие слоям томографии, данные, позволяющие определить их положение и угол поворота в пространстве, а также метаданные, идентифицирующие исследования и содержащие информацию о них. Формат имеет теговую структуру. На рис. 5 изображены основные теги, используемые при загрузке.

Тегами, отвечающими за координаты изображения, являются Image Position (patient), определяющие координаты левого верхнего угла изображения, и Image Orientation, определяющие углы между изображением и осями координат.

Тегом Pixel Data помечен массив пикселей, каждый из которых соответствует рентгеновской плотности ткани в данной точке. Информация о параметрах изображения содержится в тегах Rows, Columns, Bits Allocated и Bits Stored, определяющих размеры изображения и количество бит, отведённых на каждый пиксель. По этим данным создаётся изображение в формате RGBA. Для пикселей, значения которых невелики, значение альфа-канала выставляется равным нулю. Таким образом не визуализируются пиксели, соответствующие пространству

¹⁴Стандарт DICOM, URL: <https://www.dicomstandard.org/current/> (дата обращения 22.04.2018)

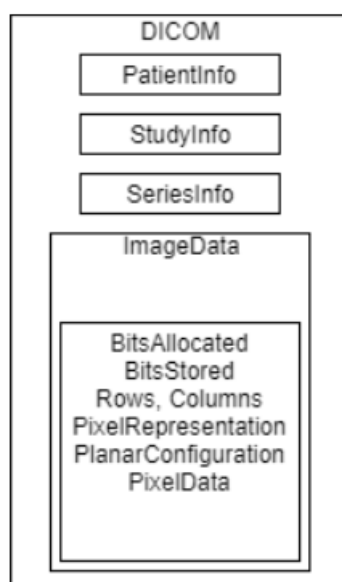


Рис. 5: Структура DICOM-файла

вне модели пациента. Прозрачность остальных пикселей регулируется пользователем.

Для каждого КТ-изображения в сцене создаётся квадратный объект. На обе стороны этого объекта накладывается полученная текстура для того, чтобы он был виден с обеих сторон. Если модель, предварительно построенная по КТ-снимкам, не была изменена, то КТ-изображения загружаются в сцену совмещёнными с моделью, так как для них определены пространственные координаты. В ходе работы приложения пользователь может выбирать визуализируемый слой (рис. 6).

4.3. Проведение интраоперационных измерений

4.3.1. Получение 3D-модели сцены

Для того, чтобы проводить измерения на пациенте, необходимо в реальном времени строить его 3D-модель. Построение 3D-модели реализуется при использовании трёхмерного сенсора, предоставляющего данные карты глубины. При разработке использовался сенсор Kinect for Windows 2. Каждый пиксель карты глубины представляет коорди-



Рис. 6: Визуализация данных КТ

нату ближайшей точки в определённом направлении по оси, перпендикулярной фокальной плоскости. Зная эту координату и углы между лучом и осью камеры, можно определить две другие координаты. Применение матрицы трансформации между камерой глубины и RGB-камерой позволяет определить также текстурные координаты точки. Таким образом, каждый пиксель карты глубины даёт вершину модели сцены. Объединив точки, соответствующие соседним пикселям карты глубины, получаем текстурированную полигональную модель сцены. Полученная таким образом модель является грубой. Для её сглаживания при определении координат вершин используется среднее значение глубины за несколько кадров.

4.3.2. Функции интраоперационных измерений

Для выделения точки на модели необходимо выпустить из камеры луч в направлении указанного пикселя. Для того, чтобы определить точку пересечения луча и модели, вычисляются расстояния между лучом и каждой точкой модели и выбирается ближайшая к лучу вершина.

Если на модели выделены несколько точек, могут быть вычислены расстояние между точками, площадь поверхности, ограниченной точками и объём выделенной области.

Расстояние между точками находится по формуле (1).

$$\rho(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2} \quad (1)$$

Для измерения площадей и объёмов определяется набор треугольников, ограниченных выбранными точками. Так как все точки получены из пикселей карты глубины, то каждой точке соответствует координата соответствующего пикселя. Таким образом, набору точек на модели соответствует многоугольник на плоскости. Критерием принадлежности точки выделенной области является принадлежность соответствующей ей точки на плоскости этому многоугольнику (рис. 7). Пусть v_1, \dots, v_n - векторы, проведённые из точки в вершины многоугольника. Тогда критерием принадлежности точки многоугольнику является сонаправленность векторов системы (2).

$$\begin{aligned} &v_1 \times v_2 \\ &\dots \\ &v_{n-1} \times v_n \\ &v_n \times v_1 \end{aligned} \quad (2)$$

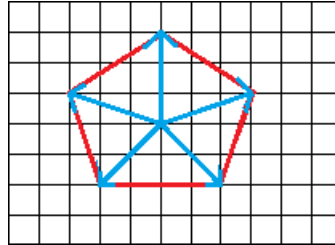


Рис. 7: Определение принадлежности точки области

Считается, что треугольник лежит в области, если все его вершины принадлежат области. Площадь выделенной поверхности вычисляется как сумма площадей треугольников области. Площади треугольников вычисляются по формуле (3), где X_1, X_2 и X_3 - координаты вершин треугольника.

$$S = \frac{1}{2} |(X_3 - X_1) \times (X_2 - X_1)| \quad (3)$$

Для вычисления объёма выбранной области выбирается точка на границе области (первая из выбранных пользователем точек) и вычис-

ляются объёмы тетраэдров, основаниями которых являются треугольники выбранной области, а вершиной - выбранная точка, по формуле (4), где X_0, X_1, X_2, X_3 - вершины тетраэдра. Сумма взятых со знаком объёмов этих тетраэдров равна объёму выделенной области.

$$V = \frac{1}{6}((X_0 - X_1) * ((X_0 - X_2) \times (X_3 - X_0))) \quad (4)$$

Заключение

В рамках работы реализован фреймворк для разработки приложений дополненной реальности, работающих на платформе Windows Mixed Reality, а также использующих камеры и трёхмерные сенсоры.

Достигнуты следующие результаты:

- проведён обзор существующих систем дополненной реальности, используемых в них технологий и методов совмещения виртуальных данных с пациентом;
- разработана архитектура фреймворка: выделены сценарии использования систем дополненной реальности и компоненты фреймворка;
- реализована функциональность фреймворка:
 - выполнено совмещение трёхмерных моделей с пациентом в реальном времени методами детектирования маркеров и особых точек на лице;
 - реализованы функции загрузки КТ-изображений, а также управления их визуализацией;
 - реализованы функции получения модели и интраоперационных измерений на ней.

Список литературы

- [1] J. Polvi, T. Taketomi, G. Yamamoto et al. SlidAR: A 3D positioning method for SLAM-based handheld augmented reality. — *Computers and Graphics*, 2017. — Vol. 55. — P. 33 – 43.
- [2] M. Garon, P.O. Boulet, J.P. Doironz et al. Real-time high resolution 3D data on the HoloLens. — *Proc. of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, 2016. — P. 189 – 191.
- [3] Microsoft. Kinect for Windows SDK 2.0. — URL: <https://www.microsoft.com/en-us/download/details.aspx?id=44561> (дата обращения: 22.04.2018).
- [4] S. Bernhardt, S.A. Nicolau, L. Soler et al. The status of augmented reality in laparoscopic surgery as of 2016. — *Medical Image Analysis* 37, 2017. — P. 66 – 90.
- [5] Szeliski R. *Computer Vision: Algorithms and Applications*. — Springer, 2010. — P. 31 – 41.
- [6] T. Itamiya, T. Iwai, T. Kaneko. The Holographic Human for surgical navigation using Microsoft HoloLens. — *EPiC Series in Engineering*, 2018. — Vol. 1. — P. 26 – 30.