

Разработка инструментария для показа стерео-презентаций из файлов в KML-формате

Александр Бобков, Илья Казанский, Станислав Клименко

Факультет общей и прикладной физики,

Московский физико-технический институт, Москва, Россия

alexbobkov@list.ru, kazansky@icpt.su, stanislav.klimenko@gmail.com

Аннотация

Статья посвящена разработке инструментария для показа интерактивных стерео-презентаций, содержащих геопривязанные данные. Стереопрезентация — это презентация с возможностью визуализации данных в стереоскопических системах виртуальной реальности. Для хранения данных используется формат KML. Для подготовки данных используется геоинформационное приложение Google Earth. Оно обеспечивает просмотр спутниковых снимков и работу с геопривязанной информацией. Имеется возможность добавления пользовательских данных: меток, объектов, изображений, виртуальных путешествий. Эти данные экспортируются в файлы в KML-формате.

В рамках данной работы разработано приложение, которое обеспечивает работу со стерео-презентациями на базе данных в KML-формате и показывать эти презентации на установке виртуальной реальности. Для работы с KML-файлами используется библиотека libkml, для стерео-визуализации — фреймворк AVANGO.

Keywords: виртуальное окружение, стереопрезентации, Google Earth, KML.

1. ВВЕДЕНИЕ

В настоящее время увеличивается популярность стереоскопических систем: снимаются 3d-фильмы, на рынке становятся доступны бюджетные системы активного стерео. Стереоскопические технологии проникают в обычные виды деятельности. Например, возникает понятие стереопрезентации. Стереопрезентация — это презентация, которая обеспечивает стереоскопический просмотр и взаимодействие с данными. Стереопрезентации опираются на технологию виртуального окружения. Виртуальное окружение — современная технология человеко-машинного взаимодействия, позволяющая “погружение” человека в трехмерную среду предметной области и более эффективное усвоение информации.

В данной работе рассматриваются стереопрезентации, содержащие географическую информацию. Под этим понимаются следующие вещи: географические карты, спутниковые снимки, трехмерный ландшафт, трехмерные модели, текстовая, звуковая и видеоинформация, привязанная к географическим координатам, история изменений этой информации во времени.

Чтобы определить способы и методы работы с геопривязанной информацией, имеет смысл рассмотреть понятие “неогеография”. Этот термин был введен в 2006 году Эндрю Тернером и обозначает совокупность новых

методов и средств для работы с геопространственной информацией. Неогеографию отличают следующие принципы: использование растровых данных, как основного представления данных; использование географических координат, вместо картографических; использование открытых форматов данных.

Один из таких открытых форматов — формат KML (Key-hole Markup Language), который является также родным для приложения Google Earth. В настоящее время это открытый стандарт, который поддерживается Открытым геопространственным консорциумом (Open Geospatial Consortium). KML позволяет хранить ссылки на внешние данные (звуки, фото, видео, гиперссылки) и задавать географическую привязку для этих данных. Также KML позволяет задавать векторные данные (точки, маршруты, контуры), добавлять временную метку к данным и указывать способ обзора данных (точку обзора, направление обзора, облет по заданному маршруту).



Figure 1: Устройство стереопрезентации

Использование методов неогеографии и формата KML видится разумным для работы со стереопрезентациями, содержащими геопривязанные данные. Можно выделить следующие концепции представления данных:

- Трехмерная модель Земли. Модель должна содержать спутниковые снимки поверхности и трехмерный ландшафт;
- Географические и другие карты, наложенные на трехмерную модель Земли;
- Текстовая, звуковая, фото и видеоинформация, прикрепленная к модели Земли в заданных точках;
- Векторная информация, наложенная на модель Земли;

- Просмотр информации в исторической перспективе
- Интерактивное перемещение по трехмерной модели Земли и взаимодействие с геопривязанной информацией;
- Воспроизведение записанных виртуальных “путешествий”.

Цель данной работы — разработка программного инструментария для показа стерео-презентаций, который обладает вышеприведенными возможностями. Все возможности можно разбить на 2 группы: интерактивная трехмерная модель Земли и работа с геопривязанной информацией из KML-файлов. Устройство стерео-презентации изображено на рис. 1. В статье рассматривается только работа с KML-файлами. Разработка интерактивной модели Земли выходит за рамки этой статьи.

Разработанный инструментарий может использоваться для широкого круга задач, связанных с географией. Это могут быть стерео-презентации в школах, вузах, музеях, выставках и конференциях. Стерео-презентации могут использоваться для бизнес-планирования и принятия решений. На базе инструментария могут быть разработаны ситуационные центры, авто и авиа-тренажеры нового поколения.

Новизна подхода заключается в использовании формата KML и фреймворка AVANGO. Формат KML обеспечивает возможность интеграции инструментария с другими приложениями, прозрачный способ хранения и обмен данными. Фреймворк AVANGO обеспечивает возможность использования различных стереосистем, устройств ввода и обеспечивает быструю модификацию инструментария для конкретных задач.

2. ПОХОЖИЕ СИСТЕМЫ

2.1 Present3D

Present3D — приложение для показа стерео-презентаций. Оно разрабатывается Робертом Осфилдом, который также является разработчиком графического инструментария OpenSceneGraph. Present3D использует OpenSceneGraph в качестве графической подсистемы.

Презентация в Present3D состоит из набора слайдов. Present3D позволяет вставлять в слайды 3d-модели и анимировать их. Для задания слайдов используется формат XML.

Наш инструментарий отличается тем, что он нацелен на более узкую область презентаций — презентаций, которые содержат геопривязанную информацию. Инструментарий позволяет размещать такую информацию на виртуальном глобусе — интерактивной модели планеты Земля — в заданных геокоординатах. И, в отличие от Present3D, он использует формат KML для описания данных.

2.2 Google Earth

Google Earth — геоинформационное приложение, которое содержит трехмерную модель Земли со спутниковыми снимками и трехмерным ландшафтом высокой детализации. Google Earth также содержит различную геопривязанную информацию: названия

стран, населенных пунктов, рек, географические достопримечательности.

Google Earth позволяет загружать из KML-файлов пользовательские данные и просматривать их. Google Earth позволяет также создавать свои собственные геопривязанные данные: ставить метки, накладывать изображения, отмечать маршруты и области, добавлять 3d-модели зданий и других объектов. Google Earth позволяет записывать “путешествия” и записывать звуковое сопровождение. И в этом смысле его можно рассматривать как средство создания и показа географических презентаций.

Недостатком Google Earth является его закрытость и проприетарность. Google Earth не позволяет писать свои расширения. В том числе, невозможна стереоскопическая визуализация.

Наш инструментарий отличается тем, что он использует открытые библиотеки и язык Python для высокоуровневого программирования. Это позволяет обеспечить стереоскопическую визуализацию, а в перспективе и другие возможности, которые нет у Google Earth, например, синтез русской речи для воспроизведения описаний геообъектов.

Но Google Earth остается удобным средством для ввода данных. Его можно использовать для первичной подготовки данных, затем экспортировать их в формате KML и использовать с нашим инструментарием.

3. ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ

3.1 KML

Рассмотрим формат KML[1]. Этот формат был разработан и начал использоваться в приложении Google Earth. Сейчас это открытый стандарт[2], и его поддерживает различное программное обеспечение. KML-файлы часто распространяются в виде ZIP-архива с расширением .kmz. В этом случае архив может также содержать дополнительные файлы: изображения, звуковые файлы, 3d-модели.

Формат KML основан на XML, и поэтому файлы в KML-формате представляют собой иерархическое дерево элементов. На рис. 2 изображена схема элементов KML. Все элементы наследуются от базового элемента Object. Object имеет свойство id — идентификатор элемента. Это позволяет всегда найти в KML-файле нужный элемент по его идентификатору.

Главный структурный элемент KML-файла — Feature. Производные элементы — Document и Folder служат для группировки других элементов Feature. Другие производные элементы — листья дерева элементов — называются визуальными, т.к. они содержат данные, которые видит пользователь. К ним относятся элемент Placemark и элементы-наложения GroundOverlay, ScreenOverlay и PhotoOverlay, содержащие разные виды изображений. Все остальные элементы KML являются вспомогательными и задают свойства структурных элементов.

Элемент Placemark содержит геометрию — один из производных элементов от Geometry. Геометрия может быть следующих типов: точка (Point), путь (LineString), многоугольник (Polygon), 3d-модель (Model). Элемент Model содержит координаты местоположения модели и ссылку на внешний файл с моделью в формате COLLA-

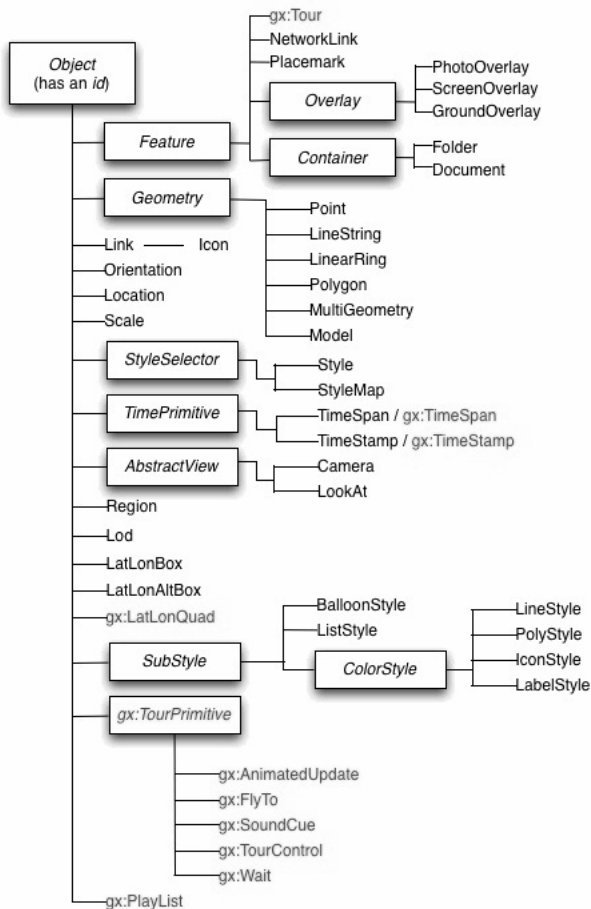


Figure 2: Схема KML

DA.

Элементов-наложения существует 3 типа: GroundOverlay, ScreenOveraly и PhotoOverlay. Все они содержат ссылки на внешние изображения. Отличаются они способом показа этого изображения.

GroundOverlay — изображение, которое накладывается на трехмерный ландшафт поверх других изображений. Изображение вписывается в четырехугольную рамку. Географические координаты углов рамки задаются через свойство LatLonBox.

ScreenOverlay — изображение, которое накладывается на экран. Можно задавать размеры изображения и положение на экране.

PhotoOverlay — позволяет помещать фотографии в любую точку Земли и задавать параметры просмотра. Это могут быть как обычные плоские фотографии, так и панорамные.

Во всех вышеприведенных элементах используются для задания координат точек используются географическая система координат: долгота от -180° до $+180^\circ$, широта от -90° до $+90^\circ$ и высота в метрах. При этом многие элементы имеют дополнительное свойство altitudeMode, которое может принимать 3 значения: clampToGround — игнорировать высоту, поместить точку прямо на трехмерный ландшафт, relativeToGround — считать высоту как высоту над ландшафтом, absolute — считать

высоту, как высоту над уровнем моря.

Внешний вид геометрии может задаваться с помощью стилей. Как видно из рис. 2, существуют стили для линий, для полигонов, для меток и других элементов.

В KML-файле можно задавать положение точки наблюдения и направление наблюдения с помощью двух элементов: LookAt и Camera (рис. 3). LookAt позволяет задать положение относительно рассматриваемого объекта. Camera позволяет задать абсолютное положение обзора. Элемент LookAt можно использовать, например, вместе с элементом Placemark для задания оптимального обзора для наблюдения данного объекта.

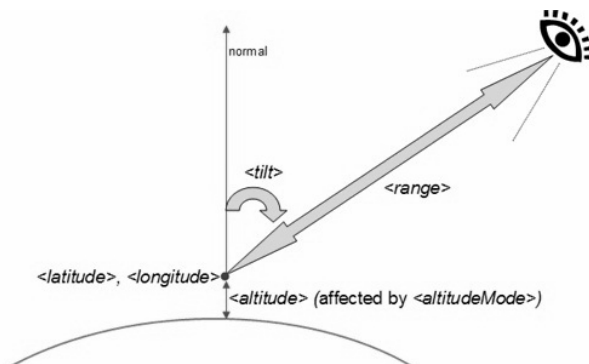


Figure 3: Задание положения камеры

В KML-файле можно задавать “путешествия” (Tour). Путешествие — полет камеры, который сопровождается звуками и изменениями объектов из файла. Путешествие состоит из последовательности примитивов. Примитивы бывают нескольких видов: полет камеры к заданной точке (FlyTo), пауза (Wait), запуск звука (SoundCue), запуск изменения (AnimatedUpdate).

Элемент FlyTo содержит конечное положение камеры, заданное с помощью LookAt или Camera, и продолжительность движения из текущего положения в конечное.

Элемент AnimatedUpdate позволяет изменять свойства элементов из KML-файла. Каждый элемент может иметь идентификатор id. AnimatedUpdate ссылается на идентификатор и указывает свойство, которое нужно изменить. Например, это можно использовать для управления видимостью элементов.

3.2 libkml

С KML-файлами можно работать как с обычными XML-файлами. Но для более удобного программного доступа к KML-файлам компанией Google разработана библиотека libkml[3]. Библиотека написана на языке C++, и имеет привязки к языкам Java и Python.

libkml состоит из нескольких составных частей. KML DOM — программный интерфейс, который обеспечивает доступ к иерархической структуре KML-файла. Он позволяет проходить по дереву элементов, осуществлять прямой доступ к свойствам элементов, в том числе вносить изменения в KML-файл. KML Engine — программный интерфейс, расширяющий функционал KML DOM. Он позволяет читать KML и KMZ-файлы и разных источников, разрешать ссылки на другие KML-файлы, делать поиск стиля по идентификатору

(несколько геометрий могут иметь общий стиль, поэтому стили обычно хранятся отдельно, а геометрии содержат ссылки на нужные стили), работать с размерами элементов.

3.3 AVANGO

Для создания стереоскопической визуализации применяется кроссплатформенный фреймворк AVANGO[4]. AVANGO поддерживает различные виды дисплеев и устройств ввода. AVANGO обеспечивает быстрое создание интерактивных графических приложений. На нижнем уровне AVANGO опирается на графическую библиотеку OpenGL. Для написания программ используется язык Python[5]. AVANGO состоит из нескольких подсистем:

- Подсистема поддержки различных видов дисплеев и многоэкранных конфигураций;
- Подсистема поддержки различных устройств ввода (джойстиков, рулей, манипуляторов);
- Подсистема поддержки звука;
- Подсистема поддержки распределенных приложений. Она позволяет синхронизировать данные между разными компьютерами;
- Настройка над библиотекой OpenGL[6]. Она добавляет к классам OpenGL возможность быть узлами графа потоков данных;

Главная особенность AVANGO — это граф потоков данных[7]. Это структура данных, которая позволяет обеспечить гибкий обмен данных между разными объектами. Состояние каждого объекта — узла в графе потоков данных — описывается набором полей. Поле может хранить некоторое значение и быть связанным односторонней связью с другими полями других объектов. В случае изменения значения поля-источника это значение пересылается по связи в поле-приемник. У объекта, которому принадлежит это поле-приемник, вызывается специальный метод, где он может прореагировать на изменение[8].

Такой граф потоков данных можно формировать по ходу выполнения приложения. Можно добавлять и удалять узлы и связи. Граф потоков данных позволяет организовать взаимодействие между разными подсистемами приложения. Данные от устройств ввода по связям поступают в объекты-вычислители. Объекты-вычислители передают данных по связям к объектам OpenGL[9], которые уже отрисовываются на экране. Данные могут пересылаться по сети для синхронизации между экземплярами приложения, запущенными на разных компьютерах. Связи могут использоваться для анимации трехмерных объектов.

4. РАЗРАБОТКА ИНСТРУМЕНТАРИЯ

4.1 Архитектура инструментария

Весь функционал инструментария можно разбить на 3 большие группы:

- Визуализация данных из KML-файла;

- Взаимодействие с данными:

- Запуск воспроизведения видео;
- Запуск воспроизведения звуков;
- Показ изображений;
- Показ описаний объектов;
- Скрытие/показ самих объектов;
- Просмотр исторической информации;

- Управление движения камерой;

- Ручное управление с помощью мыши или джойстика;
- Автоматическое управление — воспроизведение “путешествий” из KML-файла.

Чтобы данные могли быть визуализированы, они должны быть организованы в структуру данных графа сцены. Граф сцены управляется библиотекой OpenGL. Эта структура данных удобна для визуализации, но не удобна для взаимодействия с данными. Данные в KML-файла также не удобны для прямой работы с ними. Поэтому удобно ввести промежуточное хранилище данных. Данные из KML-файла помещаются в хранилище данных, а затем из них формируется граф сцены для визуализации.

На основе данных из хранилища можно организовать систему меню. С помощью меню можно осуществлять взаимодействие с данными, как перечислено в вышеприведенном списке. В том числе из меню можно управлять графом сцены, а значит и визуализацией.

KML-файл может также содержать записанные “путешествия”. Модуль путешествий может управлять движение камеры по трехмерной сцене. Как было указано в описании формата KML, путешествия могут также содержать изменения данных. Например, можно автоматически скрывать или показывать отдельные объекты.

Схему работы инструментария можно увидеть на рис. 4. Далее все модули будут рассмотрены подробнее.

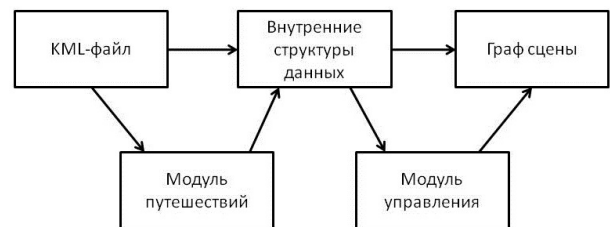


Figure 4: Схема работы инструментария

4.2 Подготовка данных

Рассмотрим сначала хранилище данных. Основные сущности, которые необходимо хранить: метки (точки), контуры, сложную геометрию, 3d-модели, изображения, стили.

В хранилище каждая сущность состоит из двух частей: это “сырые” данные взятые из KML-файла (они характеризуются географическими координатами)

и преобразованные данные, готовые в визуализации, т.е. узлы графа сцены (они характеризуются декартовыми координатами). Каждая сущность имеет также свой идентификатор, который берется из KML-файла.

В хранилище сущности хранятся в виде дерева. Граф сцены — это направленный ациклический граф, но для простоты его тоже можно считать деревом. Эти деревья структурно не совпадают, т.к. организованы по разным принципам. Структура дерева хранилища определяется KML-файлом. Данные в нем сгруппированы по логическому принципу: данные, близкие друг другу по смыслу или по типу объектов, находятся в хранилище рядом. В графе сцены же данные организованы по принципу эффективности визуализации. Данные, близкие пространственно, находятся в графе сцены рядом. Это необходимо для быстрого отбрасывания данных, которые выходят за объем видимости. Например, граф сцены может представлять собой структуру данных quadtree, когда пространство рекурсивно делится на 4 квадрата.

Инициализация инструментария состоит из нескольких этапов: разбор KML-файла и помещение данных в хранилище, построение узлов графа сцены для каждой сущности в хранилище и объединение всех узлов в граф сцены. Рассмотрим эти этапы подробнее.

Для разбора KML-файла применяется библиотека libkml. Происходит проход по иерархической структуре KML-файла, во время которого для обработки разных типов элементов применяются специальные функции-обработчики. Например, функция HandleContainers осуществляет обработку элементов Folder и Container, производит спуск по дереву элементов и вызывает другие функции-обработчики: HandlePlacemark, HandleGroundOverlay, HandleScreenOverlay, HandleTour. Функции обработчики помещают данные в хранилище.

Далее происходит создание узлов графа сцены для каждой сущности. При этом все географические координаты точек (долгота, широта, высота) преобразуются в декартовы координаты в системе координат, связанной с центром Земли.

Для меток создаются сферы, расположенные по координатам метки. Рядом со сферой помещается текстовая подпись с названием метки. Чтобы сфера и подпись были видны с любого расстояния, над сферой в графе сцены помещается объект AutoTransform. Он автоматически вычисляет необходимую матрицу масштабирования.

Пути и многоугольники напрямую преобразуются в стандартные геометрические примитивы OpenGL. Пути строятся двумя способами: с использованием геометрических примитивов LINE_STRIP и QUAD_STRIP. Первый вариант удобен, если путь должен выглядеть тонкой линией. Второй — если нужно показать толщину. OpenGL поддерживает только выпуклые многоугольники, поэтому сложные многоугольники из KML-файла предварительно разбиваются тесселятором на простые примитивы.

Накладываемое на землю изображение преобразуется в четырехугольник с текстурой.

Сложнее обстоит дело с экранными наложениями. Экранные наложения должны быть зафиксированы относительно камеры, а не трехмерной сцены. Для этого к корню графа сцены прикрепляется

специальный узел Camera. Камера содержит единичную матрицу вида и ортогографическую матрицу проекции. Камера настраивается так, чтобы отрисовывать свой подграф после всех остальных объектов. Перед отрисовкой очищается буфер глубины. К этой камере прикрепляются подузлы, содержащие экранные наложения. Настройки камеры обеспечивают, что наложения будут всегда видны на экране. Матрица проекции не зависит от того, для левого или правого глаза отрисовываются экранные наложения. Поэтому для обоих глаз экранные наложения совпадают на экране. Это улучшает восприятие изображений.

После создания всех узлов начинается построение графа сцены. При этом для каждого узла сверху добавляет дополнительный узел Switch, который позволяет управлять видимостью объекта.

4.3 Взаимодействие с данными

Далее на основе хранилища можно создать меню для взаимодействия с данными. 3d-меню создается средствами библиотеки AVANGO. Для каждой сущности в хранилище можно добавить свою кнопку: для запуска видео, звука, для показа изображения, для показа/скрытия меток, контуров, объектов.

Сущностей может быть очень много, поэтому для каждой делать свою кнопку не целесообразно. Для каждой отдельной стерео-презентации разработчик может сам указать, какие кнопки ему нужны. При этом он может использовать стандартные классы кнопок для осуществления стандартных операций: показ/скрытие объектов, запуск видео, звуков.

Например, рассмотрим показ/скрытие объектов. Мы помним, что для каждого объекта в графе сцены создан специальный узел-переключатель Switch, который умеет управлять видимостью объекта. Существует специальный класс SwitchButton, который содержит ссылки на переключатель и кнопку меню, а также поля ButtonPressed и VisibleChild, которые связаны с соответствующими полями переключателя и кнопки меню. При нажатии на кнопку происходит изменение видимости объекта и изменение надписи на кнопке.

Помимо основного меню, существует слайдер, который соответствует шкале времени. При перемещении ползунка слайдера изменяется текущее время приложения. В зависимости от наличия у объектов временных меток показывают только те объекты, которые соответствуют текущему времени.

Также для взаимодействия с метками разработан специальный класс MarkPointBehavior. Этот класс позволяет выводить при нажатии на метку контекстное меню, описание метки и проигрывать звуковой файл с описанием метки (если такой существует). Создается единственный экземпляр этого класса и прикрепляется ко всем меткам с помощью стандартного механизма библиотеки AVANGO. При нажатии правой кнопкой мыши на метку вызывается функция show_context_menu() этого класса, и в качестве аргумента передается выбранная метка.

4.4 Управление камерой

Для полноценной работы со стерео-презентацией пользователь должен иметь возможность перемещаться по трехмерной сцене, либо самостоятельно управляя

камерой, либо автоматически, с помощью “путешествий” из KML-файла.

Для ручного режима разработан специальный класс-манипулятор, который умеет работать как с мышью, так и с джойстиком SpaceNavigator[10]. Главная особенность этого манипулятора в отличие от стандартных манипуляторов — учет кривизны Земного шара и учет трехмерного ландшафта. Манипулятор позволяет отдаляться от поверхности Земли и приближаться. При этом скорость зависит от высоты. Манипулятор не позволяет проваливаться сквозь землю. Манипулятор позволяет перемещаться вдоль поверхности Земли.

Рассмотрим теперь автоматический режим. Как мы помним, каждое путешествие состоит из набора элементов: FlyTo, Wait, SoundCue, Animated Update. Воспроизведение путешествия заключается в последовательном выполнении всех элементов. Для этого создан специальный класс CameraPath с единственным экземпляром. Для запуска путешествия в этот экземпляр загружается список примитивов путешествия. Каждый кадр происходит вызов функции evaluate и проход по списку примитивов. Если текущий примитив — полет камеры к заданной точке, то аппроксимируется текущее положение камеры между начальным положением на момент запуска этого примитива и конечным положением. Вычисленное значение помещается в выходное поле MatrixOut. Если текущий примитив — запуск звукового файла, то этот файл загружается и воспроизводится. Если текущий примитив — изменение, то производится поиск в хранилище объекта, который нужно изменить и вносятся необходимые изменения. Выходное значение MatrixOut соединяется связью с матрицей вида камеры.

Для каждого загруженного путешествия заводится своя кнопка меню. При нажатии на кнопку запускается соответствующее путешествие.

5. РЕЗУЛЬТАТЫ

На основе инструментария была создана система визуализации для проекта “Виртуальный Оболенск”. Проект содержит модели 18ти зданий в формате COLLADA с фотографическими текстурами (рис. 5). Ссылки на модели и их геокоординаты хранятся в соответствующем KML-файле. Также проект содержит данные об эпидемиях болезней в Оболенске с 2004 года. Каждому факту появления болезни соответствует метка в KML-файле. Метка обладает геокоординатами, которые указывают на дом и квартиру, где болезнь была обнаружена, и датой обнаружения болезни (рис. 6). Инструментарий позволяет просматривать эти метки в хронологическом порядке.

6. ЗАКЛЮЧЕНИЕ

Разработан инструментарий для показа стереопрезентаций, содержащих геопривязанные данные. Инструментарий позволяет использовать данные из файлов в формате KML и совмещать их с интерактивной моделью Земли. Инструментарий позволяет взаимодействовать с данными: запускать привязанные звуковые и видео файлы, показывать текстовую и графическую информацию. Инструментарий позволяет воспроизводить виртуальные “путешествия”, записанные в приложении Google Earth.

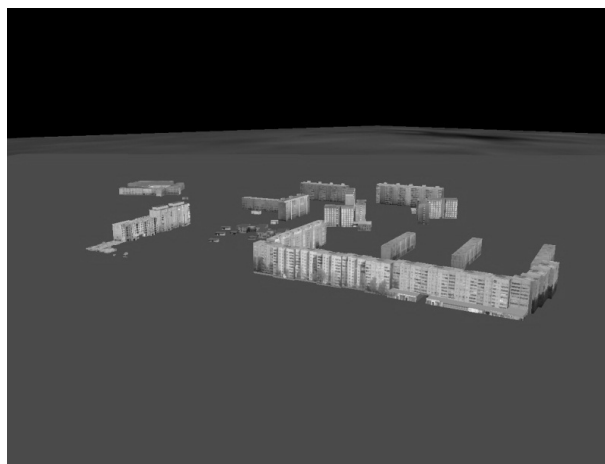


Figure 5: Виртуальный Оболенск

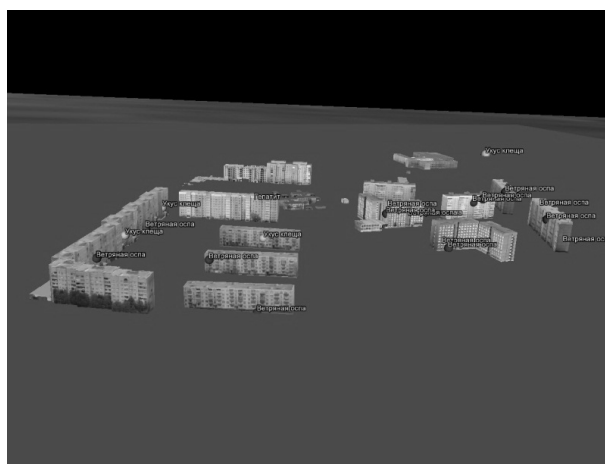


Figure 6: Информация об эпидемиях

Инструментарий опирается на открытые библиотеки и использует высокоуровневый язык Python. Это позволяет легко интегрировать его в сторонние приложения и создавать самые разнообразные географические презентации.

7. БЛАГОДАРНОСТИ

Авторы выражают благодарность Российскому фонду фундаментальных исследований (РФФИ) за частичное финансирование по грантам 08-07-00362-а и 09-07-00401-а.

Авторы выражают благодарность Еремченко Е.Н. за поддержку и предоставленные геоданные.

8. СПИСОК ЛИТЕРАТУРЫ

- [1] “Kml reference,” <http://code.google.com/apis/kml/documentation/kmlreference.html>.
- [2] “Kml specification,” <http://www.opengeospatial.org/standards/kml/>.
- [3] “libkml,” <http://code.google.com/p/libkml/>.

- [4] Henrik Tramberend, "Avocado: A distributed virtual reality framework," in *VR '99: Proceedings of the IEEE Virtual Reality*, Washington, DC, USA, 1999, p. 14, IEEE Computer Society.
- [5] Roland Kuck et al., "Improving the avango vr/ar framework — lessons learned," *Virtuelle und Erweiterte Realität: 5. Workshop der GI-Fachgruppe VR/AR*, pp. 209–220, 2008.
- [6] Robert Osfield, "Openscenegraph: Past, present and future," in *SIGGRAPH 2006 OpenSceneGraph Birds of Feather session*, 2006.
- [7] Л.Д. Никитина С.В. Клименко, И.Н. Никитин, *AVANGO. Система разработки виртуальных окружений*, Москва-Протвино, 2006.
- [8] Henrik Tramberend, *Avocado: A Distributed Virtual Environment Framework*, Ph.D. thesis, Technische Fakultät at Universität at Bielefeld, 2003.
- [9] Paul Martz, *OpenSceneGraph Quick Start Guide*, Skew Matrix, 2007.
- [10] "The economic payback of 3d mice for cad design engineers," Tech. Rep., Technology Assessment Group, 2008.

Об авторах

Александр Бобков — аспирант Кафедры системной интеграции и менеджмента Факультета общей и прикладной физики Московского физико-технического института. Email: alexbobkov@list.ru

Илья Казанский — ассистент Кафедры системной интеграции и менеджмента Факультета общей и прикладной физики Московского физико-технического института. Email: kazansky@icpt.su

Станислав Клименко — профессор, доктор физ.-мат. наук, директор Института физико-технической информатики Email: stanislav.klimenko@gmail.com

Abstract

The article is about development of toolkit for displaying stereo presentations, containing georeferenced data. Stereo presentation is a presentation with the possibility of stereoscopic visualization. KML is used for data storage. Google Earth is used for data preparation. Google Earth allows to view satellite imagery and to work with georeferenced information. Toolkit allows to add custom data: tags, objects, images, virtual tours. This data is exported in KML-format.

The application is developed that allows work with stereo presentations based on data in KML-format and show these presentation on the installation of virtual reality. Libkml library is used for reading KML-files, AVANGO framework - for highlevel development.