

Секция «Методологии разработки программного обеспечения»

В.И. Амозов, Е.И. Авдеев, Е.А. Доможиров
Научный руководитель к.т.н., доц. Н.П.Мольков
Муромский институт (филиал) Владимирского государственного университета
602264 г. Муром Владимирской обл., ул. Орловская, 23
sapres@mivlgu.ru

Программа для начального изучения принципов работы АЛУ восьмиразрядного микропроцессора

Современное развитие электроники и цифровой техники требует глубокого изучения принципов работы цифровых схем и логических элементов, основой которых они являются. Несмотря на большое количество программ различного уровня сложности, облегчающих процесс освоения цифровой техники, трудности изучения, особенно на начальном этапе, остаются, что подталкивает к необходимости разработки новых программ.

Предлагаемая работа посвящена разработке программы эмулирующей работу арифметического логического устройства восьмиразрядного микропроцессора. Особенностью предлагаемой программы является наглядный, графический интерфейс, который позволяет по шагам проследить процесс выполнения арифметических и логических действий. Кроме того, имеется возможность в ходе работы контролировать умение правильно переводить числа из одной системы в другую. Программа особенно полезна для начального изучения основ микропроцессорной техники давая возможность понять логику работы арифметического логического устройства при выполнении арифметических и логических операций.

Программа разработана для постановки лабораторного курса по дисциплине "Основы цифровой схемотехники". В отличие от существующих промышленных, платных программ, требующих значительных временных затрат на освоение сложного интерфейса, предлагаемая программа позволяет сосредоточиться на принципах работы изучаемого АЛУ. Несложный интерфейс, наглядность, отсутствие платы за пользование делают её доступной для широкого круга пользователей, приступающих к изучению сложной микропроцессорной техники.

Программа использовалась в лабораторных работах по курсу "Основы цифровой схемотехники" и может быть полезна и в других курсах связанных с изучением цифровой техники.

Современное состояние систем слежения за объектами по видеопоследовательности

Современный мир трудно представить без компьютерных технологий. Быстро развиваются цифровые технологии: фото и видео устройства. В связи с этим развиваются и алгоритмы позволяющие анализировать информацию, полученную с цифровых устройств. Например, распознавание образов на изображении, или слежение за объектами на видео. Существует множество различных алгоритмов, которые позволяют оценивать положение одного или нескольких целевых объектов на кадрах видеопоследовательности. Данные алгоритмы входят в многочисленные приложения компьютерного зрения в робототехнике и видеонаблюдении [1].

В последнее десятилетие были разработаны методы обнаружения и распознавания на основе внешности объектов (appearance-based), использующие для описания объектов их локальные ключевые точки (key points) в аналогии с подходами распознавания, встроенными в систему человеческого восприятия. Эти методы показали удовлетворительные робастность и инвариантность к геометрическим преобразованиям и изменению освещенности. Однако основным недостатком этих методов является их вычислительная сложность, что мешает их применение в задачах слежения за объектами в реальном масштабе времени. Вследствие выше рассмотренных обстоятельств разработка эффективных методов обнаружения объектов на изображениях видеопоследовательности и увеличение робастности существующих методов с целью проектирования информационно-измерительных систем видеослежения за движущимися объектами являются важными и необходимыми задачами для развития человеческого общества, и представляют собой актуальные научно-исследовательские задачи.

Объект исследования - автоматизированные системы сопровождения и видеослежения за движущимися объектами с использованием поворотной видеокамеры.

Предмет исследования - методы и алгоритмы обнаружения, видеослежения и измерения параметров движущихся объектов на изображениях видеопоследовательности, показывающие инвариантность и робастность к геометрическим преобразованиям.[2]

Решить задачу с целью видеонаблюдения и слежения за одним объектом, можно с помощью многочисленных алгоритмов, основными из которых являются:

1. Алгоритмы управления движением наклонно-поворотной платформы, обеспечивающие автоматическое слежение за движущимся объектом;
2. Компьютерное моделирование процесса обнаружения и слежения за движущимся объектом и процесса вычисления его параметров;
3. Исследование и разработка инвариантных и робастных к аффинному преобразованию методов и алгоритмов, позволяющих устойчиво обнаружить объекты и оценивать их параметры на статических изображениях и в кадрах видеопоследовательности;

В случае изменения объекта с течением времени процесс слежения (трекинг) может осуществляться за ложными объектами, которые не представляют интерес точки зрения эксперта. Необходимо определить моменты времени, когда трекер дает сбой. Для этого необходимо ввести критерии, характеризующие надежность трекинга. В современных системах обычно используются следующие:

1. Forward-Backward ошибка;
2. Аналог постоянной Липшица;
3. Критерий устойчивости Ляпунова

В настоящее время я считаю актуальность данной темы слежение по видеопоследовательности за одним объектом. Хотя уже разработано достаточное количество алгоритмов слежения, но на этом разработка их не заканчивается, мир идет вперед техника развивается, а следовательно стоит вопрос о разработке новых алгоритмов, которые будут более проще и легче в применение.

Литература

1. www.lib.tpu.ru/fulltext/c/2012/C21/208
2. <http://mm-dsp.com/index.php>
3. <http://wiki.technicalvision.ru/index.php>

Методы одновременной навигации и построения карты

В данной исследовательской работе будет проведён сравнительный анализ методов SLAM (Simultaneous Localization And Mapping).

SLAM — Simultaneous Localization And Mapping — это метод одновременной навигации и построения карты, используемый роботами и автономными транспортными средствами для построения или обновления карты в пространстве.

Главной особенностью SLAM, является то, что он не располагает знаниями о среде. Все решения строятся только на результатах измерений датчиков. Одним из главных моментов является то что – среда считается статичной.

Рассмотрим основные методы слам алгоритмов:

- ЕКМ – Extended Kalman Filter
- Loop closure
- iSam
- TORO
- DP-SLAM

ЕКМ - Extended Kalman Filter.

В данном методе у нас имеется на каждом шаге, набор ранее полученных особенностей и поступивших данных с датчиков. На основе метода визуальной однометрии мы определяем смещение робота и предсказываем его новую позицию. С другой стороны, из нового кадра, мы можем выделить местоположение особенностей и вычислить положение робота относительно их. На основе разницы между двумя этими оценками позиций робота обновляются вероятности/веса для всех особенностей и корректируются позы-траектория движения робота. В качестве структуры для хранения информации о мире и траектории движения робота – используется разрастающаяся со временем матрица ковариации, содержащая на каждом шаге исчерпывающую информацию о нашем текущем представлении мира.

Loop closure - обнаружение петель.

Главной задачей метода отслеживание ситуации когда робот возвращается туда, где уже был. Одно из решений – составление корзины «слов». Каждому кадру ставится в соответствие дескриптор, вычисляемый на основе визуальных особенностей изображения. Для хранения информации о изображениях, на основе данных для обучения, формируется словарь в виде дерева, содержащий “слова” для представления дескрипторов и их веса (отражающих насколько часто они встречались в наборе изображений для обучения). Для формирования “слова” производится поиск визуальных особенностей на основе данных для обучения и их последующая группировка. Каждый последующий уровень дерева получается путем повторения данной операции с дескриптором родительского узла. В результате, при анализе ново-поступивших данных производится быстрое разложение идентификатора текущего кадра в линейную комбинацию узлов дерева, где в роли коэффициентов выступают их веса – вероятности.

iSam.

В данном методе используется фильтрация с применением последовательных методов Монте-Карло. В качестве расширения могут быть использованы узлы-якоря – особенности наблюдаемые разными роботами или особенности которые видит один и тот же робот в течении нескольких независимых путешествий. Ядро системы производит оптимизацию данного графа выравнивая все узлы на основе новых данных. Возможна работа и без особенностей и без узлов-якорей – только на основе результатов визуальной однометрии.

TORO.

Данный метод определяет конфигурации графов при которой вероятность наблюдаемых особенностей будет максимальна. В основу TORO лег метод стохастического градиентного

спуска. Особенностью данного метода является последовательная оптимизация всего графа для каждого ограничения (взаимного расположения особенностей и робота). Для объединения данных оптимизации с нескольких ограничений используются специальные коэффициенты для корректировки значений остатка. Ключевое достижение данного метода – представление данных для SGD – в TORO используется дерево параметров в качестве индексов для линейного уравнения поз робота.

DP-SLAM.

Данный метод является принципиально другим подходом к представлению окружающей среды. Робот находится в среде с препятствиями сложной формы, в которых не возможно выделить отдельные ориентиры. Вместо положения ориентиров строится полная детализированная карта и происходит фильтрация с применением последовательных методов Монте-Карло.

В ходе сравнительного анализа мы выявили, что самым оптимальным методом является TORO, его оптимальность заключается в быстром получении результатов, жертвуя точностью вычислений.

Литература

1. Smith, R.C.; Self, M.;Cheeseman, P. (1986). "Estimating Uncertain Spatial Relationships in Robotics" UbiComp 2009. Orlando, Florida, USA:
2. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo, (2011). "A random-finite-set approach to Bayesian SLAM.". Retrieved 2010-07-30.
3. Wikipedia.ru. [Электронный ресурс]: Свободная энциклопедия. – Электро. дан. – М., Режим доступа: <http://ru.wikipedia.org>

О. С. Конихина, А. М. Докукин
Научный руководитель – к.т.н., доцент каф. САПР Д. Г. Привезенцев
Муромский институт Владимирского государственного университета
602264 г. Муром, Владимирской обл., ул. Орловская, д. 23
Email- konikhina_olesya@mail.ru

Современное состояние интеллектуальных систем видеонаблюдения

В современном обществе проблематика интеллектуального видеонаблюдения, видеослежения является актуальной, поскольку в отличие от обычных систем, изначально спроектированных и предназначенных для «ручного» наблюдения, интеллектуальная система видеонаблюдения не требует постоянного вмешательства оператора и оптимизирована под быструю обработку событий.

Система видеонаблюдения — это программно-аппаратный комплекс, предназначенный для организации видеоконтроля, как на локальных, так и на территориально-распределенных объектах. К функциям видеонаблюдения относится не только защита от преступников, но и наблюдение за работниками, посетителями в офисе, на складе или в магазине, контроль деятельности в любом помещении. [3]

Таким образом, система видеонаблюдения обеспечивает:

- визуальный контроль ситуации на охраняемом объекте — предоставление информации на пост наблюдения в режиме реального времени;
- запись видеоинформации на цифровой видеорегистратор, что позволяет документировать события, происходящие на объекте;
- выполнение функций охранной сигнализации через детекторы движения видеокамер или внешних охранных датчиков и информированность оператора системы о возникновении тревоги в контролируемой зоне. [1]

Системы интеллектуального видеонаблюдения предназначены не только для визуального контроля или автоматического анализа изображений, например, автоматическое распознавание лиц, государственных номеров автомобилей, а так же для видеоконтроля, используемого для определенных, конкретных целей.

Одной из важнейших функций интеллектуальных систем видеонаблюдения является отслеживание пути наблюдаемого объекта, при этом происходит его обнаружение, выделение, классификация.

Анализ объектов видеонаблюдения – это полезный инструмент:

- для видеонаблюдения за поведением объектов, например, за интенсивностью потока в определенной зоне и определенное время;
- для подсчета объектов видеонаблюдения: к примеру, людей в очереди к кассе в крупном магазине;
- для отслеживания путей объекта видеонаблюдения: оставленного без присмотра багажа в оживленном аэропорту или праздношатающихся людей около банкоматов.

Для обнаружения движущихся объектов система видеонаблюдения анализирует все пиксели кадра, сравнивая их с опорными кадрами. При выделении объектов система извлекает обнаруженные движущиеся объекты и присваивает им свойства: цвет, размер, направление и время движения. Классифицируя объекты, система распределяет выделенные объекты по разным типам – человек, автомобиль, сумка и др. Описывающая объекты информация называется метаданными. Получив метаданные об объектах, система начинает сравнивать их с неким набором критериев. Например, метаданные могут показывать, что человек идет в неправильном направлении, или оставленную без присмотра сумку, или автомобиль, заезжающий в запретную зону; в таких ситуациях система может подать сигнал тревоги или начать записывать видео на носитель для возможности использовать его в качестве доказательства. [2]

Когда система видеонаблюдения обнаруживает и классифицирует объект, эту информацию можно использовать разнообразными способами. Одно из важнейших применений видеонаблюдения (особенно в розничной торговле) – это подсчет людей; технология позволяет получить исчерпывающую информацию для оптимизации расположения товаров и обслуживания клиентов.

Благодаря возможности считать количество людей в определенных областях, система может посылать предупреждения при наличии большого количества людей, близкого к критическому, или в ситуациях, когда что-то мешает свободному проходу. Анализирующее объекты программное обеспечение позволяет определить границу или область, доступ в которую будет контролироваться. [2]

Как и все в этом мире, интеллектуальные системы видеонаблюдения не совершенны, но, при условии следования рекомендациям по использованию и квалифицированному подбору, их эффективность достигает 90 – 98 %.

Литература

1. Герман Кругль Профессиональное видеонаблюдение. Практика и технологии аналогового и цифрового CCTV. — «Секьюрити Фокус», 2010. — 640 с.
2. <http://os-info.ru/>
3. Wikipedia.ru. [Электронный ресурс]: Свободная энциклопедия. – Электро. дан. – М., Режим доступа: <http://ru.wikipedia.org>

А.М. Кузьмин, И.С. Нагичев, М.В. Пшеничкин
Научный руководитель к.т.н., доц. Н.П.Мольков
Муромский институт (филиал) Владимирского государственного университета
602264 г. Муром Владимирской обл., ул. Орловская, 23
sapres@mivlgu.ru

Программа для изучения и разработки цифровых логических схем

Разработка цифровых логических схем представляет собой довольно сложный, многоэтапный процесс связанный с составлением логических уравнений, их оптимизации и дальнейшей схемотехнической реализацией. Процесс освоения приемами проектирования может значительно затягиваться. На начальном этапе проектирования, бывает полезно сразу оценить во что схемотехнически выливается то или иное логическое уравнение.

Предлагаемая работа посвящена разработке программного эмулятора цифровых логических схем. Особенностью программы является возможность мгновенной схемотехнической реализации устройства, заданного логическим уравнением. Графический интерфейс делает работу программы удобной и наглядной, что позволяет сократить время на разработку цифровых схем и дает возможность проработать большее количество вариантов решений за счёт легкости внесения изменений в исходное логическое уравнение и контроля результата этих изменений.

Программа разработана для постановки лабораторного курса по дисциплине "Основы цифровой схемотехники". В отличие от существующих промышленных, платных программ, требующих значительных временных затрат на освоение сложного интерфейса, предлагаемая программа позволяет сосредоточиться на принципах составления логических уравнений и их схемотехнической реализацией.. Несложный интерфейс, наглядность, отсутствие платы за пользование делают её доступной для широкого круга пользователей, приступающих к изучению сложной микропроцессорной техники.

Программа использовалась в лабораторных работах по курсу "Основы цифровой схемотехники" и может быть полезна и в других курсах связанных с изучением цифровой техники.

Разработка управляющего алгоритма системы сбора экологических данных

В городах, имеющих промышленные предприятия, степень загрязнения атмосферного воздуха может превысить санитарно-гигиенические нормативы. Для обеспечения требуемой чистоты атмосферного воздуха в регионе, важно знание временного и пространственного изменения концентрации вредных веществ. Основой для контроля за экологической обстановкой служат стационарные посты наблюдения. От точности и оперативности наблюдений зависит своевременность и эффективность воздухо-охранных мероприятий.

При проектировании и построении систем экологического мониторинга необходимо учитывать различные источники выбросов [1]. Основные источники выбросов загрязняющих веществ являются: промышленные предприятия, автотранспорт, теплоэнергетика, сельское хозяйство, а также выбросы вредных веществ переносимы с ветром с соседних регионов. Главные загрязнители атмосферного воздуха, образующиеся в процессе производственной и иной деятельности человека — диоксид серы (SO₂), оксид углерода (CO) и твердые частицы. На их долю приходится около 98% в общем объеме выбросов вредных веществ [2].

При использовании стационарных постов для достижения высокой точности и широты охвата подконтрольной территории необходимо увеличивать количество точек наблюдения. Увеличение количества стационарных постов требует больших финансовых затрат. Решением данной проблемы может стать создание нескольких мобильных постов наблюдения.

В данной работе рассматривается структура автономной мобильной системы экологического мониторинга. Основа системы – микроконтроллерный блок, предусматривающий подключение различных датчиков в зависимости и типа контролируемых вредных веществ. Это в основном датчики концентрации оксида углерода, диоксида азота, диоксида серы и пыли. В качестве основы для разработки была выбрана платформа Arduino. Arduino — аппаратная вычислительная платформа, основными компонентами которой являются простая плата ввода-вывода и среда разработки.

Для передачи полученных с датчиков показаний в соответствии с разработанной структурой системы, используются различные способы беспроводной связи. Выбор типа связи осуществляется при предварительной настройке системы путем установки соответствующего модуля и загрузки программы. Для связи со стационарным постом экологического контроля используются следующие модули: GSM/GPRS, Wi-Fi, Bluetooth, 433 Mhz RF-Link

При использовании GPRS-канала для передачи данных к системе подключается модуль ECom GPRS/GSM Shield. Для получения текущих координат используется специализированный модуль GPS Shield, являющийся компактным и высокопроизводительным с низким энергопотреблением на базе чипа SiRF Star III, который может отслеживать до 20 спутников одновременно.

Благодаря наличию беспроводной передачи данных, показания получаемые с датчиков устройства можно принимать на мобильные устройства в непосредственной близости от системы или передавать на сервер для обработки и хранения. Такое представление информации позволяет проводить всесторонний статистический анализ измерений и формировать отчеты заданного вида. Кроме того, существует возможность наложения измеренных данных на электронную карту территории, что позволяет представить результаты измерения в удобном для пользователя виде, проводить оперативный мониторинг загрязнения местности и отслеживать характер изменения уровня загрязнения в зависимости от времени измерений и географических характеристик местности.

Предлагаемая автономная система может быть расположена на городском экологически чистом транспорте, таком как трамваи или троллейбусы. Питание может осуществляться как от бортовой сети транспорта, так и от собственного источника питания.

Применение данной системы в экологическом мониторинге позволит увеличить не только зону контроля, но и повысить точность построения модели распространения загрязнения воздуха. Применение стандартных компонентов существенно удешевляет систему и позволяет более гибко конфигурировать ее под текущие потребности.

Литература

1. Рощупкин Э.В. Система распределенного автоматизированного мониторинга загрязнения атмосферного воздуха промышленных регионов// Безопасность жизнедеятельности. №5. 2010. С.46-52
2. Качурин Н.М., Левкин Н.Д., Комиссаров М.С. Геоэкологические проблемы угледобывающих регионов: монография. Тула: Изд-во ТулГУ, 2011.

Сравнение возможностей инструментов разработки игровых приложений

Развлекаться человек любил всегда. Сегодня, вне зависимости от возраста и своих физических возможностей, любой человек может примерить на себя любую маску и осилить любое испытание. Мир видео игр стал настолько реалистичным, что многие из нас предпочитают виртуальное развлечение реальному.

Игровой движок — это центральный программный компонент компьютерных и видеоигр или других интерактивных приложений с графикой, обрабатываемой в реальном времени[1]. Он обеспечивает основные технологии, упрощает разработку и часто даёт игре возможность запускаться на нескольких платформах. Несмотря на специфичность названия, игровые движки часто используются в других типах интерактивных приложений, требующих графику в реальном времени, таких как рекламные демо-ролики, архитектурные визуализации, обучающие симуляторы и среды моделирования.

Наиболее известным для новичков является движок Unity за свою простоту при создании игровых приложений. Так же, наиболее известными игровыми движками являются: Unreal Engine, Frostbite Engine, CryEngine, которые будут рассмотрены ниже.

Unity - мультиплатформенный инструмент для разработки двух- и трёхмерных приложений и игр. Созданные с помощью Unity приложения работают под операционными системами Windows, OS X, Windows Phone, Android, Apple iOS, Linux, а также на игровых приставках Wii, PlayStation 3 и Xbox 360. Игровой движок позволяет прямо в редакторе испытывать игру. Интерфейс редактора настраиваемый, осуществлена система наследования объектов. Но самое главное в unity - наличие бесплатной версии с некоторыми ограничениями[2]. Во время мероприятия Game Developers Conference 2014, стартовавшего 17 марта в Сан-Франциско, компания Unity Technologies представила пятое поколение своего популярного игрового движка Unity. Одной из важнейших особенностей, отличающих его от предыдущей редакции, станет новая система глобального освещения в реальном времени Enlighten, внедренная при участии специалистов британской компании Geomerics[3].

Преимущества: кроссплатформенность, полный набор инструментария в одной среде, быстрая разработка, быстрое изучение.

Недостатки: закрытый исходный код и проприетарная модель, каждую новую игру вы начинаете писать по сути с нуля

Unreal Engine - игровой движок, разрабатываемый и поддерживаемый компанией Epic Games. Написанный на языке C++, движок позволяет создавать игры для большинства операционных систем и платформ. На данный момент является самым востребованным игровым движком став основой более, чем для 300 игр. Некоторые версии данного движка являются бесплатными[4].

Преимущества: большое количество обучающих статей и видео по созданию игр, весьма «дружелюбен» с новичками.

Frostbite Engine - разработан компанией EA Digital Illusions CE. Технология адаптирована в первую очередь для игр в жанре шутера от первого лица и работает на персональном компьютере под управлением Microsoft Windows, а также на игровых приставках Xbox 360 и PlayStation 3. Движок поддерживает технологии DirectX 11. Технология способна обрабатывать разрушаемость ландшафта и окружения (например, построек, деревьев, автомобилей). Поддерживается динамическое освещение и затенение. Максимальный размер локации составляет ограничение в 32 × 32 километра отображаемой площади и 4 × 4 километра игрового пространства. Помимо этого, утверждается о том, что максимальная дистанция прорисовки позволяет увидеть уровень вплоть до горизонта[5]. Данный движок является очень сложным при разработке игровых приложений и предназначен для группы разработчиков.

Основным отличием от предыдущих двух движков является полностью платное распространение.

Недостатки: дорогое содержание движка, не подходит для малых групп разработчиков, сложный при создании приложений, не предназначен для мобильных устройств.

CryEngine - Игровой движок, разработанный немецкой компанией с ограниченной ответственностью Crytek. Изначально является кроссплатформенным движком. Ориентирован на разработку массовых многопользовательских онлайн-игр. является полностью коммерческим, и основным акцентом при его разработке является его лицензирование другими компаниями. Шейдеры в CryEngine 3 пишутся единоразово на языке программирования высокого уровня, а затем автоматически компилируются под каждую платформу. Вывод шейдеров оптимизируется через настройки художника и под ту трёхмерную среду, в которой данный шейдер будет использоваться. Благодаря этому становится возможным создание таких эффектов, как: «эффект невидимости», мокрые, грязные и замороженные поверхности, которые могут быть «наслоены» одна на другую и объединены с другими шейдерами, симулирующими такие эффекты, как металл, стекло и другие. Так же поддерживается попиксельное освещение реального времени, отражения, преломления, эффекты объёмного жара и анимированные текстуры для симуляции окон, поверхностей с солнечными бликами и много других эффектов.[6] Очень похож на движок Frostbite Engine и имеет такие же недостатки.

Современные игровые движки — одни из самых сложных в написании приложений, зачастую, состоящие из десятков различных компонентов, каждый из которых можно настраивать по отдельности под нужды игры. Честь из движков является простой, другая сложной при разработке игровых приложений. К примеру: при создании игры на Unity можно обойтись и одному, но используя движок Frostbite Engine требуется целая компания разработчиков. Поэтому каждый выбирает для себя, какой продукт в конечном итоге он хочет получить.

Литература

- 1) http://ru.wikipedia.org/wiki/%C8%E3%F0%E2%E9_%E4%E2%E8%E6%EA [Электронный ресурс]: Wikipedia - Электрон. Текстовые дан., [2014].- Режим доступа: ru.wikipedia.org, свободный.-Загл. с экрана
- 2) [http://ru.wikipedia.org/wiki/Unity_\(%E8%E3%F0%E2%E9_%E4%E2%E8%E6%EA\)](http://ru.wikipedia.org/wiki/Unity_(%E8%E3%F0%E2%E9_%E4%E2%E8%E6%EA)) [Электронный ресурс]: Wikipedia - Электрон. Текстовые дан., [2014].- Режим доступа: ru.wikipedia.org, свободный.-Загл. с экрана
- 3) <http://www.3dnews.ru/812083> [Электронный ресурс]: Новостная газета - Электрон. Текстовые дан., [2014].- Режим доступа: 3dnews.ru, свободный.-Загл. с экрана
- 4) http://ru.wikipedia.org/wiki/Unreal_Engine [Электронный ресурс]: Wikipedia - Электрон. Текстовые дан., [2014].- Режим доступа: ru.wikipedia.org, свободный.-Загл. с экрана
- 5) http://ru.wikipedia.org/wiki/Frostbite_Engine [Электронный ресурс]: Wikipedia - Электрон. Текстовые дан., [2014].- Режим доступа: ru.wikipedia.org, свободный.-Загл. с экрана
- 6) http://ru.wikipedia.org/wiki/CryEngine_3[Электронный ресурс]: Wikipedia - Электрон. Текстовые дан., [2014].- Режим доступа: ru.wikipedia.org, свободный.-Загл. с экрана

Разработка программного обеспечения для интерактивных многофункциональных часов

21й век — это век бурно развивающихся технологий, особенно в сфере информационных технологий. Современный мир предоставляет широчайший спектр информационных продуктов для потребителей и огромные возможности для разработчиков этих продуктов. Поэтому, сейчас, везде можно встретить устройства, помогающие упростить жизнь людей. Современные гаджеты содержат в себе огромные возможности и функционал. Практически все эти устройства являются более сложными в своей конструкции, чем обыватель может представить при первом взгляде. Сегодня современные устройства не являются чем-то необыкновенным и уникальным. Они настолько внедрены в жизнь людей, что часто образуют симбиоз между собой, дополняя свои возможности. Даже самые обыкновенные часы могут нести огромный функционал и возможности, подстраиваясь под повседневные потребности современных людей.

Сейчас общество является очень динамическим, у людей множество задач, работы, людям важно поддерживать связи между друг другом. Поэтому людям важно максимально эффективно использовать своё время.

Практически каждый человек использует электронную почту и социальные сети. Иногда проверка почты может занять некоторое время. Часто обычная проверка почты превращается во многочасовое прохождение времени в интернете. Поэтому можно автоматизировать некоторые ежедневные рутинные задачи и создать «умное», функциональное и решающее эти задачи устройство.

Создаваемое устройство представляет собой компактную панель 8x32 пикселей, которая обеспечит человеку простой, не требующий временных затрат, доступ к самой часто используемой информации – это текущее время; количество писем, заголовок или отправитель письма в социальной сети или электронной почте; окружающая температура.

Для того чтобы отображать время достаточно просто включить устройство, но время можно корректировать и с компьютера. Для отображения писем нужно запустить на компьютере специально разработанную программу и с помощью неё подключить устройство к компьютеру, через Bluetooth. Также компьютер должен быть подключен к интернету, и во всё той-же программе должна быть произведена авторизация в электронной почте и/или должен быть подключен аккаунт социальной сети. Теперь система готова к функционированию.



Рис. 1. Схема взаимодействия устройств в разработанной системе

Система использует следующие отдельные устройства, соединенные в единый блок: Arduino Mega 2560 является управляющим микроконтроллером; Bluetooth; электронное табло Shield MaTrix 8x32, табло состоит из четырёх подключаемых секций, каждая имеет разрешение 8x8; датчик температуры. (Рисунок 1)

Каждые 30 секунд компьютер подключается к почтовым серверам, с помощью специальной программы, и получает информацию о почте. К микроконтроллеру подключен модуль Bluetooth. Компьютер подключается к микроконтроллеру по средствам Bluetooth и передаёт необходимые данные. Микроконтроллер выполняет специально написанную программу, которая читает данные с подключенному к нему Bluetooth. Программа обрабатывает полученные данные и отображает соответствующую информацию на табло. Кроме того, микроконтроллер опрашивает датчик температуры и также отображает эту информацию на табло. Когда программа посылает данные о почте, то табло отображает линию шириной в 1 пиксель, если отключен режим попеременного отображения. Если включен режим попеременного отображения, то такая информация как количество почты, заголовок почты или имя отправителя; температура отображаются попеременно оставаясь на экране до 40 секунд.

На самом табло, кроме дисплея, есть кнопка и датчик интенсивности окружающей освещенности. Поэтому созданное устройство может обеспечить ряд дополнительных возможностей таких как: регулировка яркости экрана в зависимости от окружающей освещенности, отключение экрана, например, в ночные промежутки времени, когда человек спит, устройство также легко сможет вывести любую другую информацию на свой экран, или играть роль будильника. Также на табло присутствует кнопка, с помощью которой можно задавать режимы работы. Кнопка находится на табло, но она напрямую соединена с микроконтроллером, поэтому с помощью кнопки можно выполнять дополнительные действия. Одно нажатие соответствует включение или отключение режима - попеременного отображения разной информации: температура, почта, время. Двойное нажатие может отключить или включить табло. Тройное нажатие заставляет микроконтроллер прекратить опрашивать Bluetooth.

В дальнейшем к контроллеру вместо Bluetooth можно подключить модуль Wi-Fi, и контроллер сам сможет обращаться к почтовым серверам за нужными данными без использования компьютера. Также на основе данных от таких датчиков как датчик интенсивность освещенности, температуры, влажности можно производить дополнительные действия. Возможно, сделать отправку информации, для пользователя, с датчиков на электронную почту или в социальную сеть

Созданное устройство, является не только полезным, многофункциональным продуктом, но и может быть очередной инновацией в мире современных гаджетов, которые предлагают новые возможности людям. Устройство может стать дополнением к существующему семейству «умных» вещей. Оно поможет людям качественнее распоряжаться временем и возможно поможет уменьшить время прохождение в интернете, что иногда является для многих проблемой.

Мессенджер с поддержкой собственного API

Интернет-мессенджеры – это программные системы, которые позволяют легко общаться с нужными людьми. Сейчас общество становится более «мобильными», и людям необходимо поддерживать дешёвую и эффективную связь с другими людьми. Интернет мессенджеры стали набирать популярность вместе с массовым распространением Интернета и ростом вычислительных мощностей мобильных устройств. В 2013 году было продано 722,5млн смартфонов в 2014 прогнозируется увеличение продаж до миллиарда. Статистика, опубликованная в конце прошлого года Международным союзом электросвязи, говорит, что в мире 6,8 млрд. пользователей мобильной связи, что примерно эквивалентно 96% населения земли. [1] Поэтому из выше приведенных фактов можно сказать, что 20 программ соответствующей направленности, размещенные в каталогах App Store и Google Play, не способны покрыть потребности такого количества людей. Поэтому в связи с бурным развитием данного направления, было решено создать мессенджер, который будет способен решить некоторые недостатки существующих решений. Существующие мессенджеры, такие как WhatsApp, Line – обеспечивают хороший функционал для общения, но их недостатком можно считать то, что с помощью них можно поддерживать связь только с людьми, которые находятся в записной книге его смартфона.

Разрабатываемый мессенджер должен обладать следующими функциями: авторизация пользователя; редактирование личной информации; фильтруемый поиск других пользователей; добавление пользователей в друзья; отправка личных сообщений открыто - адресат будет знать кто ему прислал сообщение; отправка личных сообщений анонимно - адресат не будет знать кто ему прислал сообщение; просмотр случайной переписки двух пользователей при их согласии; установка личного статуса.

Серверная часть мессенджера разрабатывается на языке C# с использованием технологии создания Web-приложений - ASP.NET WebForms. Серверная часть мессенджера работает на сервере Internet Information Services.

Такая архитектура позволит разработать собственный API, что позволит разработать клиент мессенджера для любого устройства. API - это набор готовых методов, клиент может отправить GET или POST запрос на сервер, при помощи API клиент может выполнять некоторые функции и получать ответ от сервера. С использованием API клиент может получать данные от сервера и обрабатывать их; изменять содержимое базы данных, на серверной стороне; и так далее. Ответом от сервера может быть произвольная строка, строка JSON или строка в формате XML. Ответом от сервера на запрос, содержащий методов API могут быть: полезные данные с которыми клиент будет работать, отчет об ошибке, либо данные характеризующие какое-то состояние данных на сервере.

Серверная часть мессенджера, работающая на сервере будет отправлять клиенту данные преимущественно в формате JSON. Данные JSON имеют преимущество над остальными в том, что они могут содержать древовидную структуру как XML, JSON в чистом виде легко читается человеком, несет в себе меньше бесполезной информации, существует множество готовых решений, помогающих работать с этими данными. Разрабатываемый мессенджер может сериализовать любые данные в формат JSON, также может десериализовать данные JSON в объект какого-либо класса и работать с ним используя методологии языка C#. [3]

Для взаимодействия клиента с сервером используется технология Long Polling, если клиентом является Интернет-браузер.

Long Polling (длинные опросы) - позволяет установить клиенту "постоянное" соединение с сервером. Так как клиент с сервером обменивается сообщениями с использованием протокола HTTP, а он не поддерживает постоянное соединение, то клиент вынужден переподключаться не менее 1 раза в 30 секунд. Когда клиент отправляет запрос серверу, сервер не сразу отправляет

клиенту ответ, а ожидает появления новых данных. Дождавшись ответа от сервера, клиент обрабатывает данные, затем снова устанавливает соединение с сервером и ждет ответа от него.

В дальнейшем можно обеспечить мессенджер новым API и новым рядом функций. В мессенджер будут добавлены функции, которые позволят поддерживать связь с наибольшим количеством людей. Приложение будет способно идентифицировать пользователя, при авторизации, используя: его номера телефона; учётную запись социальной сети, используя соответствующие API социальных сетей, для работы с ними. Приложение предоставит всех людей: из телефонной книги пользователя, всех, кто является его друзьями в социальных сетях, это позволит держать контакт со всеми близкими людьми. Преимущество будет в том, по сравнению с обычными социальными сетями, будет в том, что даже при интеграции мессенджера с социальными сетями, личные данные не будут передаваться на сервер социальных сетей. Это обеспечит дополнительные функции, например, человек сможет, анонимно отправлять сообщения, другим пользователям из социальных сетей, не оставляя истории своих действий в самих социальных сетях. Так же, в дальнейшем, можно обеспечить возможность назначать время отображения отправленного сообщения, по истечению которого оно уничтожится.

Литература

1. The World in 2013: ICT Facts and Figures: [Электронный ресурс]. Режим доступа: <http://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx>
2. Зиборов Виктор Владимирович, Visual C# 2010 на примерах, БХВ-Петербург
3. Освоение Ajax: Часть 11. JSON на стороне сервера: [Электронный ресурс]. Режим доступа: <https://www.ibm.com/developerworks/ru/library/wa-ajaxintro11/>

Программная система дополненной реальности – описание, применение

Вместе с развитием мощностей вычислительных систем, появляются всё новые задачи, требуемые всё больших вычислительных ресурсов. Одно из таких направлений это компьютерное зрение.

С появлением компьютерного зрения[1] стали развиваться технологии дополненной реальности[2]. Дополненная реальность - среда с прямым или косвенным дополнением физического мира цифровыми данными в режиме реального времени при помощи компьютерных устройств — планшетов, смартфонов и инновационных гаджетов. Данная технология бурно развивается и применяется, примерами этой технологии могут служить: игровая индустрия, телевидение, реклама, робототехника, образование, мобильные технологии, системы безопасности, переводы, военные конфликты, навигация, биологии, архитектуре и т.д.

Актуальность данного направления увеличивается, с быстрым развитием информационных технологий и человеческой деятельности, поэтому было решено создать программную систему, основанную на технологиях компьютерного зрения с добавлением дополненной реальности. Программа реализована на языке программирования C# в среде разработки Visual studio 2012. Программа работает на операционной системе Windows. Программа при помощи web-камеры может обнаружить QR-код на кадре - QR-код служит маркером – и выполнить с ним ряд действий. Программа отслеживает местоположение QR-кода и накладывает на него заранее определенное или любое выбранное пользователем изображение. Изображение может менять размеры в соответствии с изменением размера самого QR-кода при его удалении или приближении относительно экрана. Так как QR-код хранит в себе информацию, то пользователь может связать любой QR-код с любым изображением и использовать связку в дальнейшем, это действие называется обучение программы. Программа может извлечь информацию которая храниться в QR-коде и показать непосредственно её – это может быть изображение которое располагается в URL-ссылке, заголовок интернет-страницы или текст, эта информация отображается не статически, а перемещается в след за перемещением QR-кода относительно кадра. Тем самым обеспечивается подобие дополненной реальности. Данная программа также может выполнять ряд дополнительных действий. Это – слежение за QR-кодом, и в случае его перемещения на заданное количество пикселей можно вывести какое-то оповещение; Обычное распознавание QR-кодов и вывод информации в виде обыкновенных текстовых данных.

Работу алгоритма программы можно разделить на несколько основных этапов:

1. Обнаружение и подключение Web-камеры.
2. Ожидание входных данных, таких как: нажатие кнопок, некоторые параметры настроек программы.
3. Получение кадра через Web-камеру.
4. Попиксельно чтение кадра, обнаружение трёх, или более, маркеров на QR-коде.
5. Определение координат QR-кода
6. Чтение данных содержащихся в QR-коде.
7. При необходимости, в зависимости от режима работы, наложение на кадр дополнительных данных.
8. При включенном соответствующем режиме, отслеживание смещение QR-кода и вывод оповещения; вывод информации зашифрованной QR-кодом.

Для работы с камерой и получения кадра на устройстве используется библиотека OpenCV[1] – версия для C#. Для чтения QR-кода использовалась библиотека zxing[2].

Версии данных библиотек распространяются бесплатно и практически для всех вычислительных устройств, имеющих возможность работы с камерой. Поэтому созданную программную систему можно, написать для мобильных устройств и применять, например, в

рекламных целях; пользователь с установленным приложением сможет отсканировать какой-либо товар и оперативно получить информацию о нём или даже трёх мерный образ продукта.

Обычно в системах дополненной реальности используются заранее определенные, жестко закрепленные маркеры. В случае использования QR-кода пользователь может легко создать нужный ему QR-код, связать код с любой информацией и каждый сможет легко воспользоваться программой для получения информации

Литература

1. Компьютерное зрение. Лекция для Малого ШАДа Яндекса: [Электронный ресурс]. Режим доступа: <http://habrahabr.ru/company/yandex/blog/203136/>
2. Что такое дополненная реальность? [Электронный ресурс]. Режим доступа: <http://arnext.ru/dopolnennaya-realnost>
3. OpenCV: [Электронный ресурс]. Режим доступа: <http://opencv.org/>
4. zxing GitHub: [Электронный ресурс]. Режим доступа: <https://github.com/zxing/zxing>

Д.Е. Нужнов
Научный руководитель: к.т.н., декан ФИТ, М.В. Усачев
Муромский институт (филиал) Владимирского Государственного университета
602264, г. Муром, Владимирская обл., ул. Орловская, 23
sergashcx75@mail.ru

Разработка алгоритмов преобразования графов больших размеров

Теория графов зародилась ещё в 1736 году, родоначальником которой считается Леонард Эйлер.[1] Но широкое практическое применение обрела только с развитием электронных сетей. Собственно термины «сеть» и «граф» в некоторых источниках употребляются как синонимы. Поэтому разработка систем, моделирующих графы, сегодня очень востребованы.

Допустим мы имеем текстовый файл содержащий произвольное количество связей между узлами графа (ребер). Для начала мы должны представить их в удобном виде для дальнейшей обработки. В результате мы получим бинарный файл из потока чисел, каждое из которых представляет собой совмещение двух номеров узлов путем побитового сдвига. Это дает нам преимущество для сортировки.

Сортировка ребер графа осуществляется методом «быстрой сортировки». Так как ребро у нас представляется как одно число, то сортировка производится с очень большой скоростью и в результате получим сортированный не только по первому узлу ребра, но и по второму тоже.

При 11098 изменений графа программа сортировки затратила 1108 мс..

Для дальнейшего удобства обработки графа создадим вспомогательный файл представляющий собой поток чисел являющимися «ссылками» на первое ребро, первый узел которого отличен от предыдущего. Тем самым получим ссылку на поток ребер исходящих от одного узла, окончание данного потока является значение следующей ссылки или конца файла.

В имеющемся графе могут отсутствовать некоторые узлы. Добавить их можно путем прохода по потоку ребер, проверяя пропуски по первым узлам ребер, и добавляя отсутствующие в конец файла. По окончании операции граф снова становится несортированным, поэтому сортируем его.

При добавлении 3850 ребер затрачено 28 мс..

Так как исходный граф является ориентированным, нужно сделать его неориентированным. Проходим по всем узлам, для каждого из них проверяем обратную связь с связующими узлами. Добавляем отсутствующие ребра.

При добавлении 2843 ребер программа затратила 77 мс..

Поиск кратчайшего пути от узла к узлу методом волнового алгоритма.

Литература

1. Графы на поверхностях и их приложения, МЦНМО, 2010 г., Звонкин А.К., Ландо С.К.

Сравнительный анализ сред разработки мобильных приложений.

В настоящее время стремительно растет количество создаваемых мобильных приложений. Для их создания используются различные среды их разработки. Сейчас этих сред настолько много что рассказывать о всех можно очень большое количество времени. Рассмотрим наиболее популярные среды разработки.

AppGyver - работающие прототипы мобильных приложений за минуты без знания кода.

Существует множество конструкторов мобильных приложений, причем большинство из них либо требуют хоть каких-нибудь навыков программирования, либо представляют собой решения типа drag&drop, с помощью которых можно создавать простейшие приложения или мобильные веб-сайты. Новый стартап AppGyver предлагает другой подход. AppGyver представляет собой сервис для создания работающих на смартфоне прототипов мобильных приложений и предназначен для тех, кто хочет быстро набросать макеты дизайнов в ходе процесса разработки.

Данный сервис в первую очередь предназначен для дизайнеров и менеджеров по продукту – если у вас есть готовые наброски, Photoshop-файлы, то вы легко можете превратить их с помощью AppGyver в интерактивный прототип. И этот процесс займет считанные минуты. Инструмент также дает доступ к аппаратной ускоренной анимации, нативной пользовательской навигации, камере и другим функциям смартфона.

Appscend создает возможность межплатформенной разработки в облаке и содержит в своем арсенале поддержку push-сообщений, технологии для управления и анализа приложениями. IgniteMarkup позволяет развертывать приложения на основе единой кодовой базы.

Appscend предоставляет все инструменты, необходимые для создания и управления удивительный B2C & B2E родные приложения.

Tabris от EclipseSource - это инструментальный набор Java-разработчиков, предназначенный для создания межплатформенных нативных мобильных приложений. Благодаря Eclipse RAP (удаленная платформа для разработки приложений) приложения для Android и iOS можно писать на основе единой базы кода.

Сервисная платформа DreamFactory. Эта программа с открытым исходным кодом призвана упростить разработку благодаря своей серверной платформе для разработки приложений на HTML5. Кроме того, предоставляются услуги NoSQL хранилища и управления пользователями. Сервисную платформу DreamFactory можно установить в любом дата-центре или на облачном компьютере.

Embarcadero RAD Studio XE4 поддерживает разработку программ для смартфонов, планшетов и ПК в стандартной среде Delphi и C++. Поддержку разработки под Android обещают в этом году.

Embarcadero RAD Studio XE4 Architect предназначена для разработчиков, создающих традиционные клиент/серверные приложения, интенсивно взаимодействующие с базами данных и облачными сервисами, а также Web-приложения в рамках корпоративных систем. RAD Studio Architect сочетает в себе технологию быстрой разработки приложений с возможностями визуального моделирования данных, что помогает разработчикам проектировать наиболее критичные структуры данных в корпоративной системе. RAD Studio Architect включает всё, что есть в редакции Enterprise.

Данная среда разработки позволяет выполнять такие операции как:

Обратное проектирование (reverse-engineer), анализу и оптимизации баз данных;

Автоматическая генерация кода для создания баз данных на основе моделей для прямого проектирования (forward-engineering);

Создание логической и физической модели на основании информации из базы данных и файлов со скриптами;

Генерация ясных и понятных диаграмм с возможностью лёгкой навигации по ним;

Продвинутые возможности сравнения и слияния с двунаправленным процессом сравнения и слияния структур моделей и баз данных.

Embarcadero RAD Studio XE4 — среда быстрой разработки приложений (RAD) для Microsoft Windows фирмы Embarcadero Technologies. Текущая версия Embarcadero RAD Studio XE4 объединяет Delphi XE4 и C++ Builder XE4 в общую интегрированную среду разработки. Существует возможность компилировать под различные платформы, в частности, кроме win32 можно компилировать под win64 и под Mac OS X (визуальный интерфейс предлагается создавать на базе новой кроссплатформенной библиотеки FireMonkey).

Среда разработки Intel HTML5. Intel HTML5 представляет собой межплатформенную среду разработки, тестирования и развертывания приложения на различных мобильных устройствах. Она основана на веб-стандартах и была приобретена корпорацией Intel у appMobi.

HTML5 — это новый тренд в разработке приложений. По мнению Intel, очень важно дать опытным разработчикам все возможности для перехода на этот многоплатформенный стандарт и помочь начинающим воспользоваться новыми функциями, чтобы они могли быстро создавать приложения и игры, совместимые практически со всеми современными платформами. Данная среда позволяет разработчикам создавать приложения HTML5 и PhoneGap для iOS, Android, Amazon, Nook, Facebook и других магазинов приложений. Используя стандарт HTML5, интерфейсы игровых приложений дополняют возможности визуального представления объектов, обеспечивая многоканальное звучание, ускорение отображения физических параметров и визуальных эффектов для более реалистичного моделирования и качества игровых процессов, добиваясь при этом исключительной схожести объектов с реальностью и несравненной производительности.

Xamarin Studio является интегрированной средой разработки для проектирования, разработки, отладки и развертывания мультиплатформенных нативных мобильных приложений. Сервис потоковой музыки Rdio использовал Xamarin Studio и технологии разработки под Android iOS для расширения поддержки мобильных устройств. В Xamarin 2.0 есть возможность отладки мобильных приложений для iOS прямо из Visual Studio. Xamarin предоставляют возможность создания библиотек, которые могут быть использованы (в виде кода или готовых сборок) сразу для нескольких платформ.

Eclipse — свободная интегрированная среда разработки модульных кроссплатформенных приложений. Наиболее известные приложения на основе Eclipse Platform — различные «Eclipse IDE» для разработки ПО на множестве языков (например, наиболее популярный «Java IDE», поддерживавшийся изначально, не полагается на какие-либо закрытые расширения, использует стандартный открытый API для доступа к Eclipse Platform).

Каждая из этих сред поможет реализовать программисту те или иные задачи в создании мобильных приложений. Для новичков в программировании и для создания дизайнера приложения пригодится AppGyver. При программировании приложений на разных устройствах пригодятся облачные среды Appscend, Intel HTML5 и DreamFactory. Если создаваемое приложение должно работать и на Android и iOS, то лучше остановить свой выбор на Tabris от EclipseSource. Embarcadero RAD Studio XE4 понравится тем, кто привык программировать в Delphi и C++.

Рынок сред разработки мобильных приложений широк и разнообразен. Но каждая из сред имеет свои преимущества и недостатки. Поэтому при выборе среды для программирования нужно их учитывать, а так же делать выбор исходя из задач, которые будет выполнять готовое приложение и на каком устройстве.