



Петрозаводский государственный университет
Кафедра информатики и математического обеспечения



Баженов Н.А.

Научный руководитель: Корзун Д.Ж.

Возможности использования повседневных мобильных видеокамер в условиях Интернета вещей

70-я Всероссийская научная конференция обучающихся и молодых ученых
25.04.2018, Петрозаводск, Россия

Цель работы

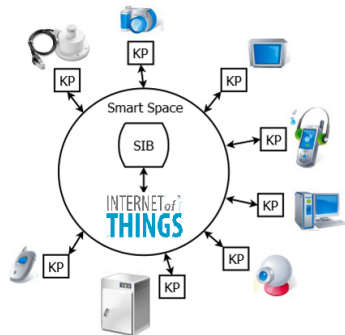
Мобильное распределенное управление потоком видеоданных с использованием технологий Интернета вещей.

Задачи

- Исследовать имеющиеся сервисно-ориентированные концепции видеокамер и интеллектуальных пространств
- Разработать примеры классов сервисов с точки зрения выполняемой роли
- Предложить и реализовать прототип программной системы

Интеллектуальные пространства

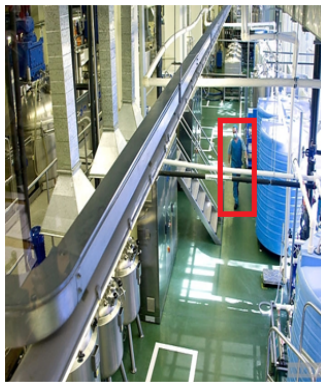
- Многоагентная база знаний
- Условия вычислительных окружений
Интернета физических устройств
- Программная инфраструктура:
 - ▶ семантический информационный брокер (SIB)
 - ▶ процессор знаний (КР)
- Возможность разработки
сервисно-ориентированных систем
- Платформа Smart-M3



Прикладные области(1)

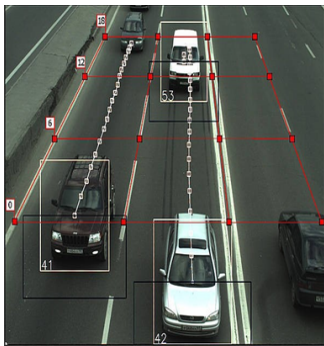


(a) Медицина



(b) Мониторинг оборудования

Прикладные области(2)



(a) Техническое зрение



(b) Распознавание лиц

Требуемые свойства:

- Доступность (выполнение на массовых устройствах)
- Интеллектуальность (эффективные алгоритмы обработки)
- Распределенность (привлечение сторонних устройств)

Взаимодействие с интеллектуальными пространствами

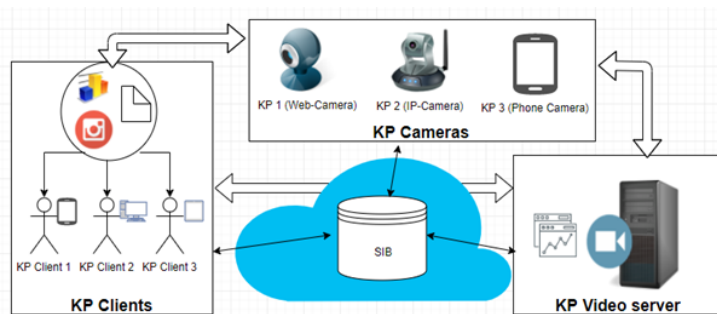


Рис.: Концептуальная модель SIB

Введение посредника **SIB**, обеспечивающего:

- Упрощенные операции взаимодействия с устройствами
- Нетривиальность обработки и моделей данных

Онтология базового подключения к камере (1)

Необходимость создания онтологии ввиду сложных и неочевидных взаимодействий:



Онтология базового подключения к камере (2)

Необходимые знания:

- Клиент:

- ▶ Сервер для подключения
- ▶ История предыдущих адресов, соединений и управлений камерой

- Камера:

- ▶ Доступность в сети для возможности подключения любого авторизованного клиента
- ▶ Подключение с параметрами (протокол подключения, разрешение изображения, кол-во кадров в секунду)

- Видеосервер:

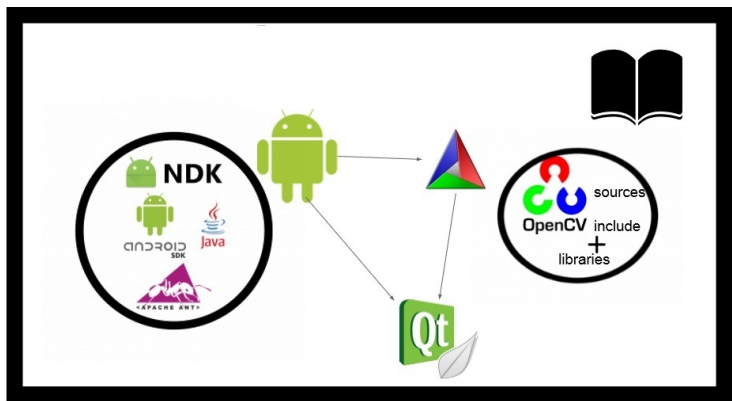
- ▶ Список клиентов, камер, истории соединений клиентов с камерами
- ▶ Аутентификация клиентов для подключения к камере
- ▶ Хранение IP-адресов, логинов и паролей камер

Сценарии работы сервисов

Могут быть выделены различные сценарии, описывающие взаимодействие компонентов:

- 1** Сценарий 0 (подключение к камере в сети при помощи ввода IP-адреса или на смартфоне):
 - ▶ Запуск приложения и выбор способа подключения к камере или видео
 - ▶ Подключение к камере
- 2** Сценарий 1 (получение видеопотока с возможностью записи и снимка с последующим сохранением на устройство):
 - ▶ Просмотр трансляции в режиме реального времени
 - ▶ Запись видео с камеры
 - ▶ Снимок с видеокамеры
- 3** Сценарий 2 (анализ видеоизображения и получение результатов анализа):
 - ▶ Загрузка видео в приложение
 - ▶ Обработка и анализ видео в течение некоторого времени
 - ▶ Построение графиков и моделей по видео

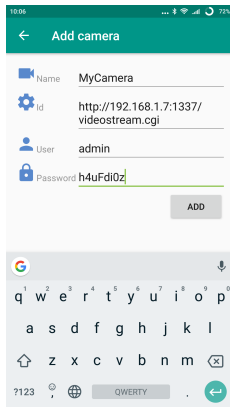
Архитектура прототипа



Использование компонентов:

- 1 IDE: Qt Creator
- 2 Разработка под мобильные ОС: Android
- 3 Компилятор CMake (обработка видеoinформации: OpenCV)

Возникающие трудности



- 1 Qt Creator: трудность переноса на мобильную платформу
- 2 Android: взаимодействие среды с библиотеками OpenCV
- 3 OpenCV: сложность реализации некоторых алгоритмов

Работа прототипа (1)

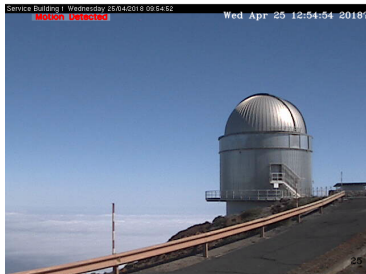
Сервис множественного доступа для ресурсов видеозахвата в среде
Интернета вещей



- 1 Подключение через IP-адрес или напрямую к камере устройства
- 2 Возможность выбора разрешения изображения
- 3 Встроенное в сервис получение текущего времени
- 4 Возможность снимка/записи видео из потока
- 5 Поддержка онтологии в базовом варианте

Работа прототипа (2.1)

Отслеживание объектов для сервисов обнаружения и анализа движения: детектирование движения

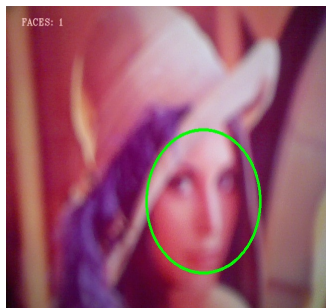


Фильтрация изображения:

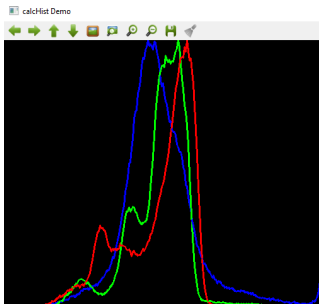
- 1 GaussianBlur
- 2 Threshold
- 3 findContours

Работа прототипа (2.2)

Отслеживание объектов для сервисов обнаружения и анализа движения: распознавание лица и глаз на изображениях



(a) Распознавание лиц



(b) Построение графиков

Haar Cascade Classifier:

- 1 haarcascade_frontalface_alt.xml
- 2 haarcascade_eye_tree_eyeglasses.xml

| Прототип | Языки | Строки кода |
|---------------|----------|-------------|
| Прототип 1: | C, C++ | 238 |
| Прототип 2.1: | C++ | 92 |
| Прототип 2.2: | C++, XML | 162 |
| Конфигурация | .pro | 70 |

Заключение

Результаты и задачи на будущее:

- Концепция сервисов на основе интеллектуальных пространств
- Архитектура для реализации этой концепции для условий IoT
- Сценарии, поддерживаемые архитектурой
- Онтология для представления семантики взаимодействия и обеспечения управления
- Планируется дальнейшее вовлечение приложения в интеллектуальные пространства и развитие функционала

Спасибо за внимание!
github.com/nikitawow1337
bazhenov@cs.karelia.ru