Что может и не может компьютерное зрение с OpenCV

http://pixdaus.com/pics/1244922167d3Z4fjf.jpg

Денис Сергеевич Перевалов

www.uralvision.blogspot.com perevalovds@gmail.com УрФУ / ИММ УрО РАН

Оглавление

Введение

О чем лекция

Эта лекция о:
- компьютерном зрении,
- библиотеке OpenCV,
- о возможностях и ограничениях компьютерного зрения, возникающих при решении прикладных задач анализа изображений.

Поэтому:

О чем лекция

Будут интересовать:

1) Алгоритмы, решающие задачи анализа изображений в (почти-) реальном режиме времени. То есть, время обработки одного кадра не должно превышать 1-10 сек.

2) Соображения и наблюдения о применимости таких алгоритмов.

НЕ будут интересовать:

1) Вопросы ускорения алгоритмов с помощью GPU.

2) Нейросети и искусственный интеллект.

Для кого эта лекция

- Для тех, кто интересуется компьютерным зрением и желает узнать о его сегодняшних возможностях и новых способах его применения.

Для кого эта лекция

- Для тех, кто не имеет опыта работы с OpenCV, но желает как можно быстрее его получить.

Для кого эта лекция

- Для тех, кто серьезно занимается компьютерным зрением, и хочет побольше узнать об узких местах и проблемах, которые могут возникнуть при использовании наилучших (на сегодняшний день) алгоритмов компьютерного зрения.

1. Что такое компьютерное зрение

Перейти к оглавлению

Определение

(из Википедии)

Компьютерное зрение — теория и технология создания машин, которые могут видеть.

http://the-gadgeteer.com/wp-content/uploads/2009/12/mr-robot-head-game.jpg

Определение

Как научная дисциплина, компьютерное зрение относится к теории и технологии создания искусственных систем, которые **получают информацию из изображений**. ...

Как технологическая дисциплина, компьютерное зрение стремится применить теории и модели компьютерного зрения к созданию **систем компьютерного зрения**. ...

http://www.spectec.net.nz/pictures/cctv%20pic.jpg

Определение

Компьютерное зрение также может быть описано как дополнение (но не обязательно противоположность) биологическому зрению.

В биологии изучается зрительное восприятие человека и различных животных, в результате чего создаются модели работы таких систем в терминах физиологических процессов. Компьютерное зрение, с другой стороны, изучает и описывает системы компьютерного зрения, которые выполнены аппаратно или программно. Междисциплинарный обмен между биологическим и компьютерным зрением оказался весьма продуктивным для обеих научных областей.

http://sobiratelzvezd.ru/wallpapers/wikimedia_23.jpg

Определение

Подразделы компьютерного зрения включают

- воспроизведение действий,

- обнаружение событий,

- слежение,

- распознавание образов,

- восстановление изображений.

1-й признак задач компьютерного зрения

Входные данные являются **двумерным массивом данных** - то есть, "изображением".

ПРИМЕЧАНИЕ
Данными также могут быть:
- видео, то есть последовательность изображений,
- 3д-данные - облака точек с 3д-сканеров или других устройств.

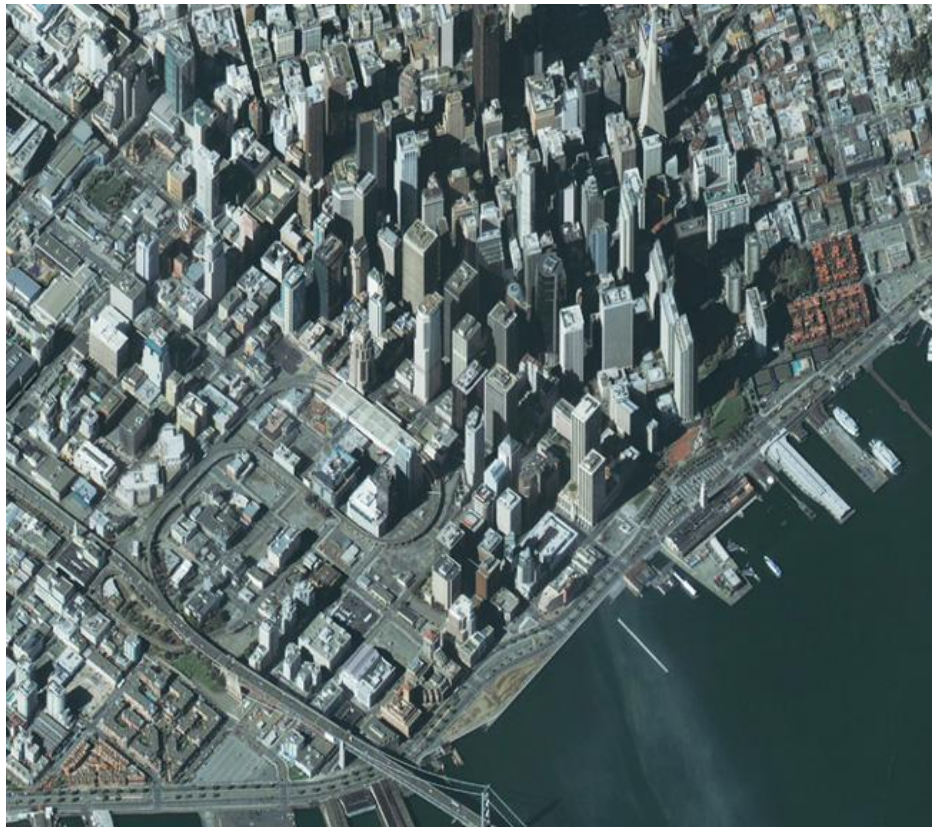
Примеры изображений

Обычный свет, радиоволны, ультразвук - все они являются источниками изображений:

1. Цветные изображения видимого спектра
2. Инфракрасные изображения
3. Ультразвуковые изображения
4. Радиолокационные снимки
5. Изображение с данными о глубине

Примеры изображений

1. Цветные изображения видимого спектра



http://rkc.oblcit.ru/system/files/images/%D0%9F%D1%80%D0%B8%D1%80%D0%BE%D0%B4%D0%B013.preview.jpg
http://imaging.geocomm.com/gallery/san_francisco_IOD032102.jpg

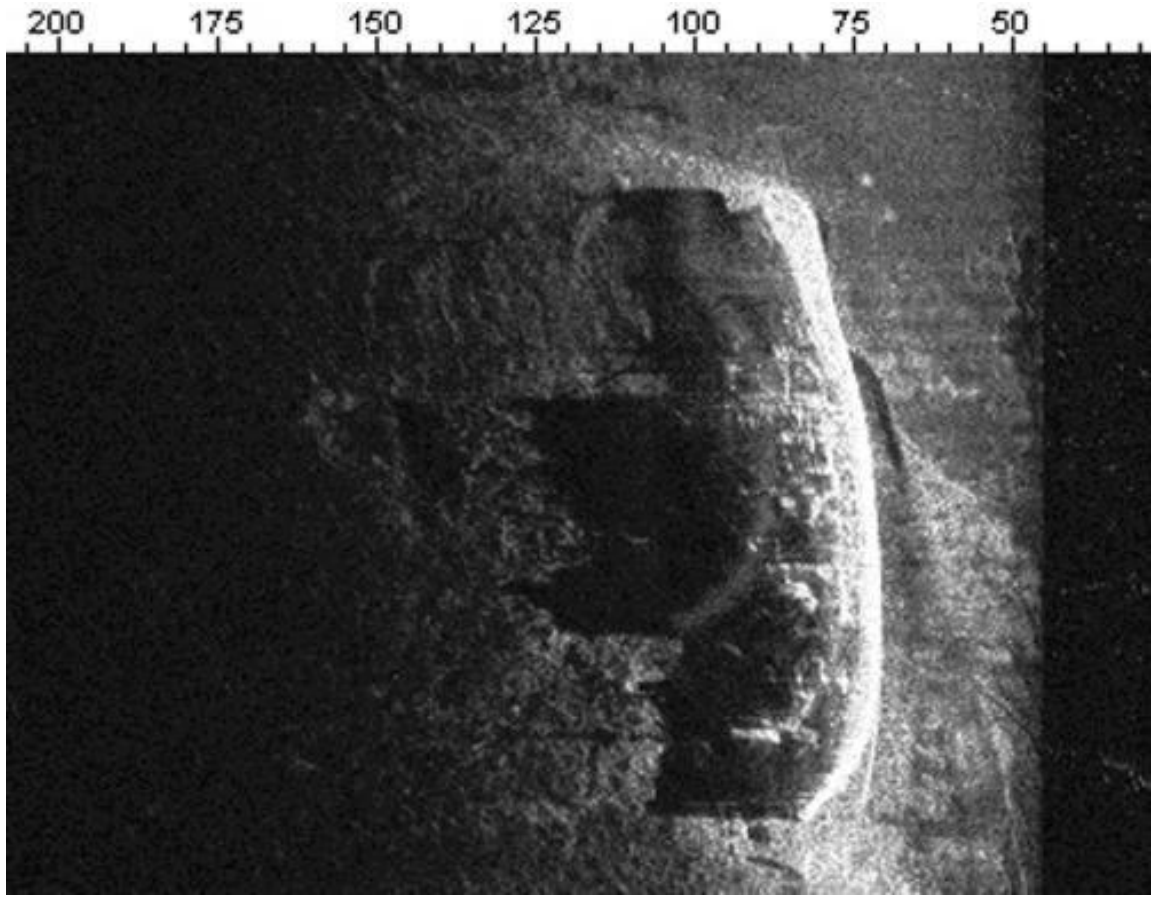
Примеры изображений

2. Инфракрасные изображения

http://lh6.ggpht.com/_Wy2U3qKMO8k/SSyB6BTdg8I/AAAAAAAACd8/Iai_3QZIjrI/Australia+5+dollars+B+se.jpg
http://i367.photobucket.com/albums/oo117/syquest/acrylic_no_filter.jpg

Примеры изображений

3. Ультразвуковые изображения
Изображение с гидролокатора бокового обзора:

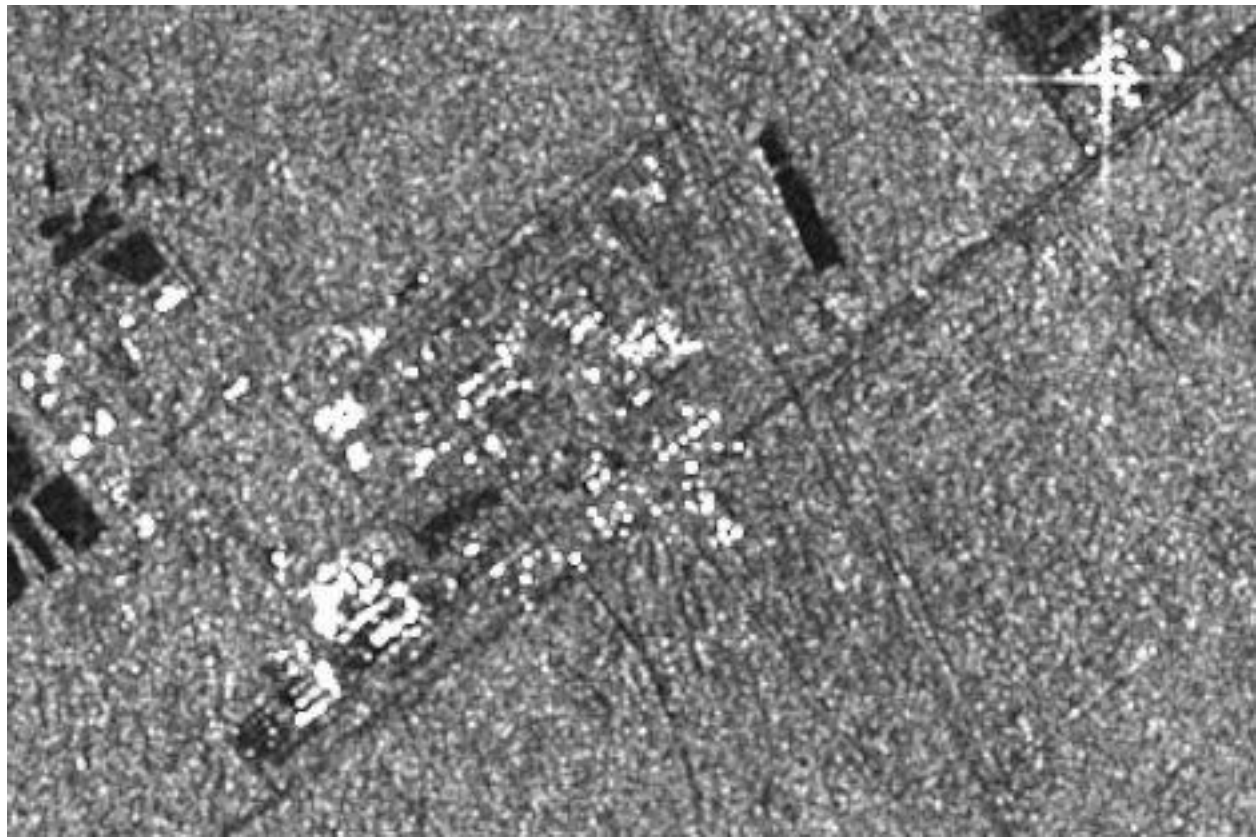


http://ess.ru/publications/2_2003/sedov/ris6.jpg

Примеры изображений

4. Радиолокационные снимки

Снимок города радаром:

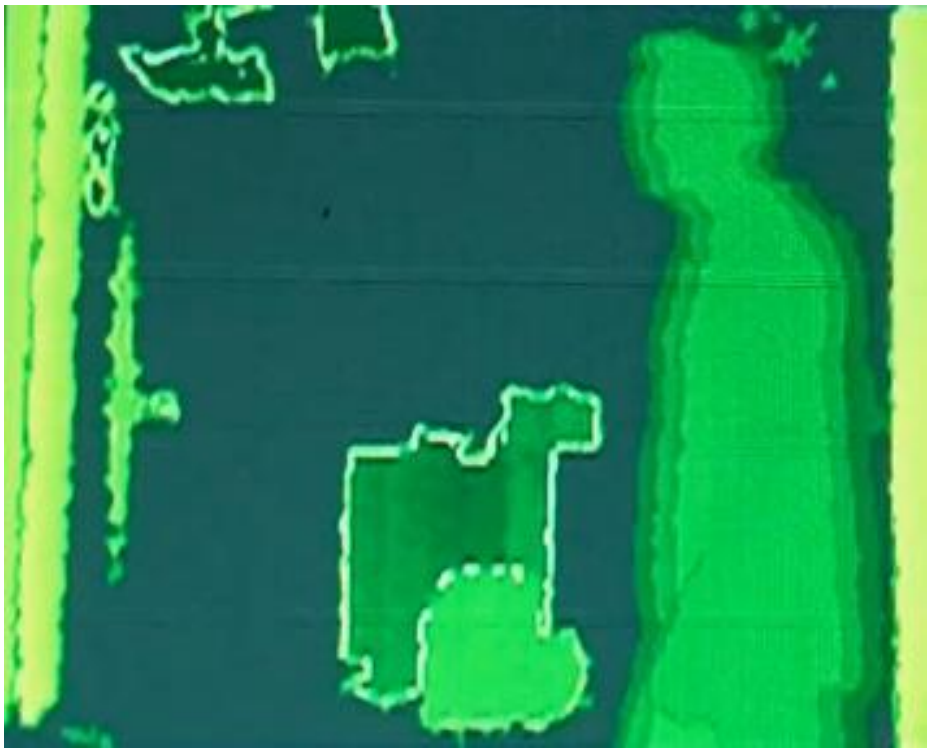


http://cdn.wn.com/pd/b1/3a/abd9ebc81d9a3be0ba7c4a3dfc28_grande.jpg

Примеры изображений

5. Изображения с данными о глубине

http://opencv.willowgarage.com/documentation/c/_images/disparity.png



Видео http://www.youtube.com/watch?v=pk_cQVjqFZ4

1-й признак задач компьютерного зрения

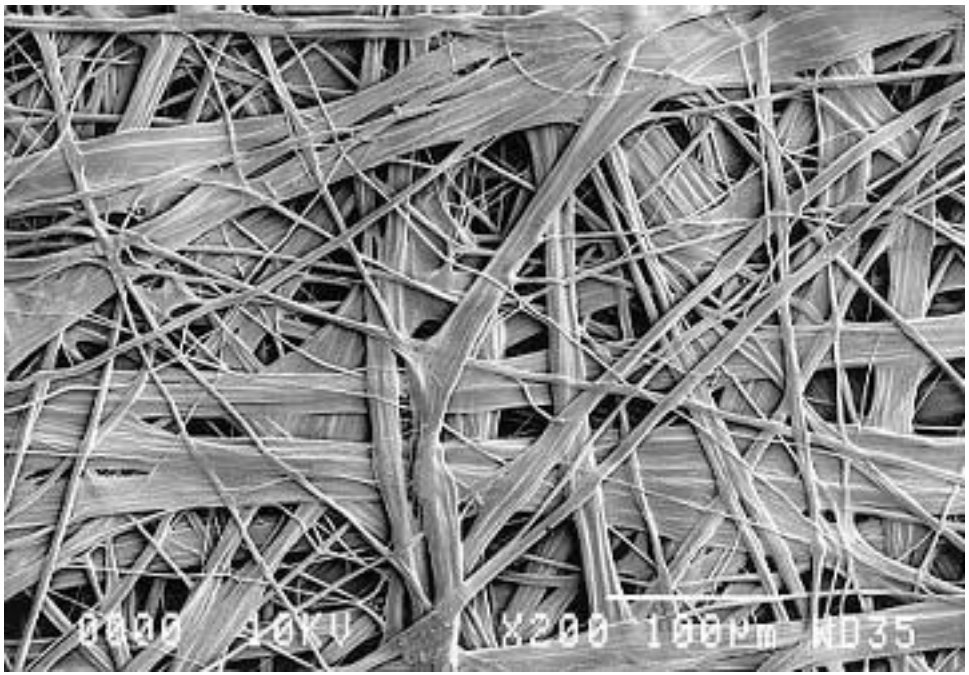
Входные данные являются **двумерным массивом данных** - то есть, "изображением".

Но двумерные массивы данных используются не только в компьютерном зрении:

Дисциплины, занимающиеся 2D-данными

2-й признак задач компьютерного зрения

Цель обработки - извлечение и использование информации о цветовых и геометрических **структурах** на изображении.



http://www.tyvek.ru/construction/images/structure.jpg

Дисциплины, занимающиеся 2D-изображениями

Дисциплины, занимающиеся 2D-изображениями

1. Обработка сигналов и изображений

Низкоуровневая обработка данных, как правило, без детального изучения содержимого изображения.
Цели - восстановление, очистка от шумов, сжатие данных, улучшение характеристик (четкость, контраст, ...)

2. Компьютерное зрение

Среднеуровневый анализ данных, заключающийся в выделении на изображении каких-либо объектов, и измерении их параметров.

3. Распознавание образов

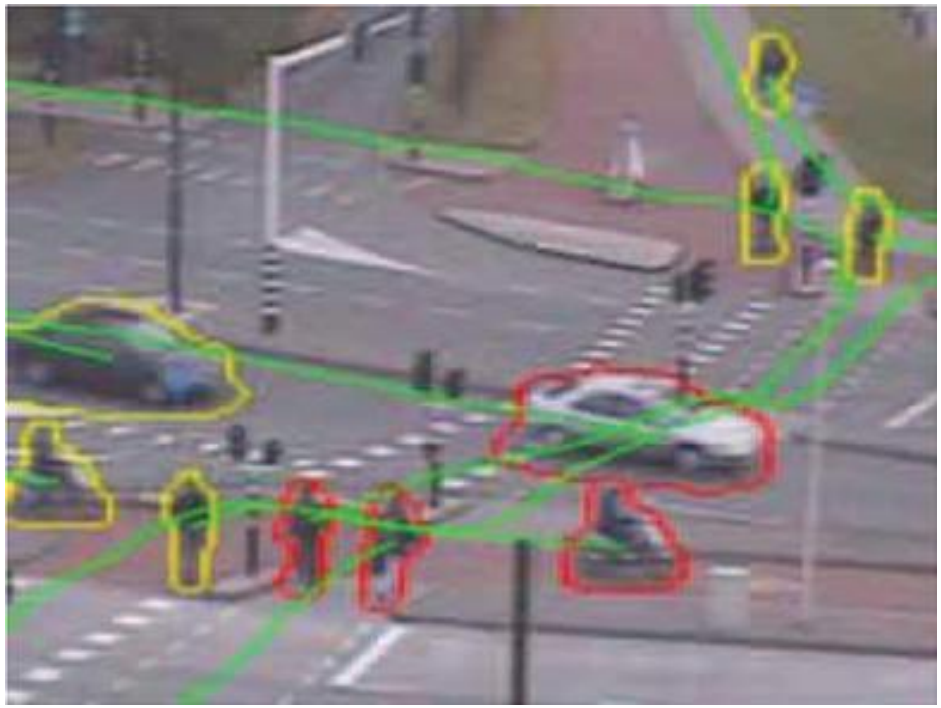
Высокоуровневый анализ данных - определение типа объекта. Входные данные, как правило, должны быть представлены в виде набора признаков. Часто для вычисления признаков применяются 1. и 2.

Примеры задач компьютерного зрения

Сегментация - разбиение изображения на "однородные" в некотором смысле области.

Примеры задач компьютерного зрения

Обнаружение интересующих объектов на изображении, и вычисление их размеров и других характеристик.



http://armi.kaist.ac.kr/korean/UserFiles/File/MMPC.JPG

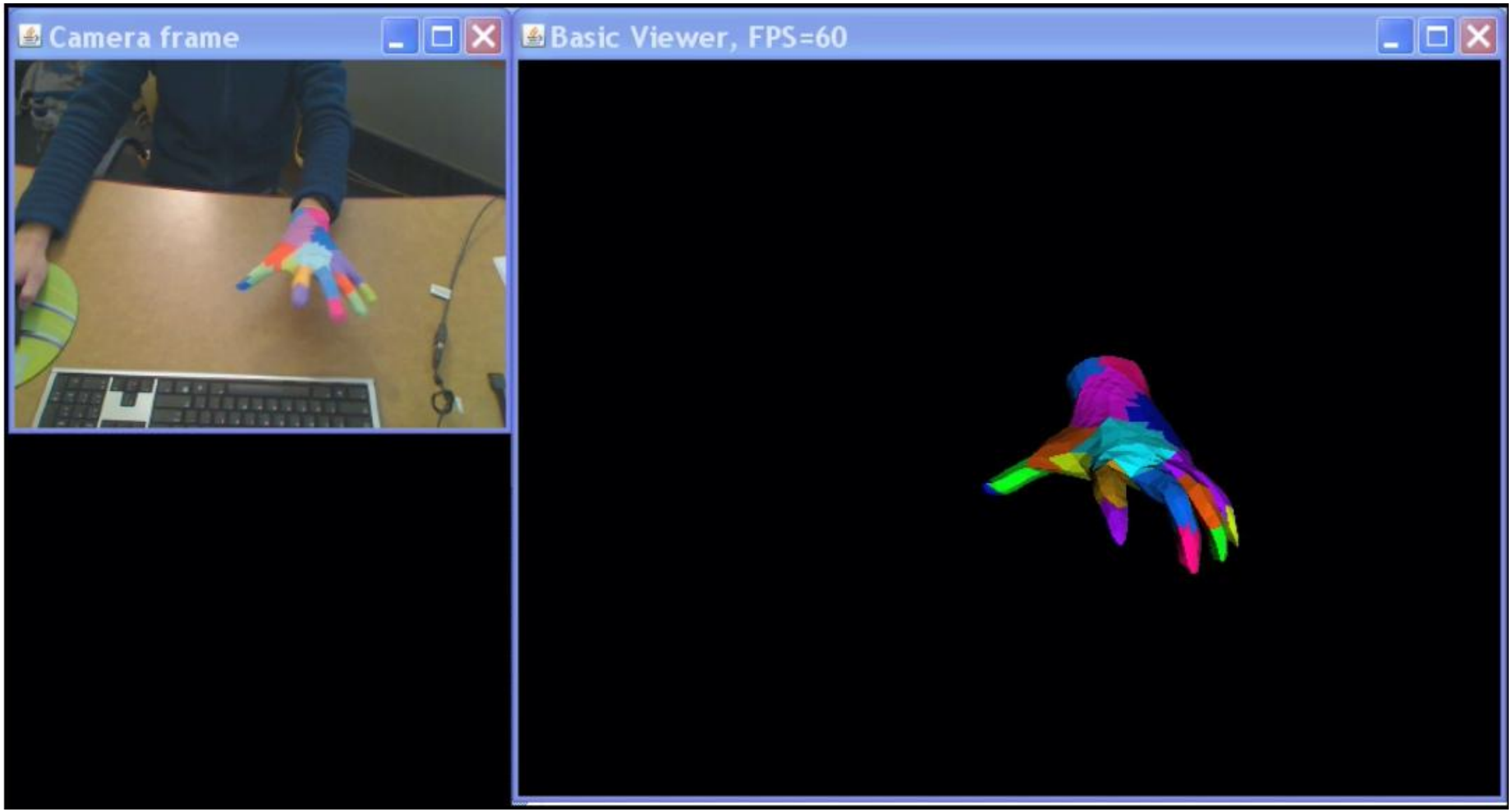
Примеры задач компьютерного зрения

Трекинг - слежение за интересующим объектом на последовательности кадров.

http://www.merl.com/projects/images/particle.jpg

Примеры задач компьютерного зрения

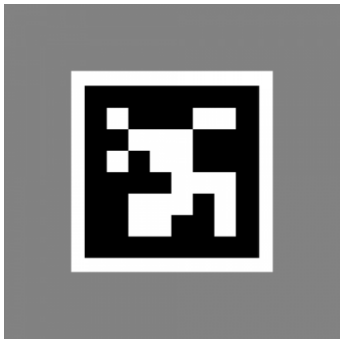
Перчатки виртуальной реальности - распознавание по цветам и модели кисти, проект MIT, прототип.



Видео http://www.csail.mit.edu/videoarchive/research/gv/hand-tracking

Примеры задач компьютерного зрения

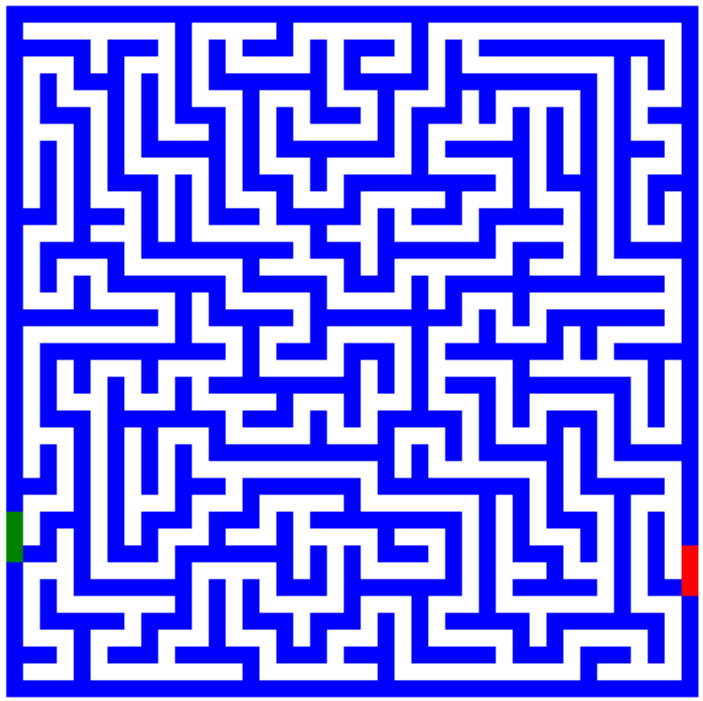
Поиск маркеров (для применения в дополненной реальности на основе маркеров).



http://www.edhv.nl/edhv/wp-content/uploads/2009/12/aug_Picture-10_no-border-450x337.jpg
http://jamiedubs.com/fuckflickr/data/web/ar-marker-BchThin_0036.png

Пример задачи НЕ компьютерного зрения

Поиск пути в лабиринте.
(Хотя входные данные - изображение, но задача - не найти объекты на нем, а решить комбинаторную задачу поиска пути).



http://www.promrds.com/chapter9/Images/NewMaze.gif

2. Камеры для компьютерного зрения

- Основные характеристики
- Примеры хороших камер

Перейти к оглавлению

Основные характеристики

Для разных задач обработки в **реальном режиме времени** нужны разные видеокамеры.

Их основные характеристики:

1. Разрешающая способность

2. Число кадров в секунду

3. Тип получаемых данных

4. Способ передачи данных в компьютер

Разрешающая способность

Это размер изображения в пикселах, получаемого с камеры.

320 x 240

точность измерения при наблюдении объекта размером 1м:
3.13 мм

размер 30 кадров:
6.6 Мб

640 x 480

точность измерения при наблюдении объекта размером 1м:
1.56 мм

размер 30 кадров:
26.4 Мб

1280 x 1024

точность измерения при наблюдении объекта размером 1м:
0.97 мм

размер 30 кадров:
112.5 Мб

http://www.mtlru.com/images/klubnik1.jpg

Число кадров в секунду

Это число картинок, получаемых с камеры за секунду.

30 к/сек

время между кадрами:
33 мсек

60 к/сек

время между кадрами:
16 мсек

150 к/сек

время между кадрами:
6 мсек

Можно использовать для музыкального инструмента

http://www.youtube.com/watch?v=7iEvQIvbn8o

Тип получаемых данных

Какие данные получаем с камеры для обработки.

Цветная или полутоновая картинка видимого спектра

Инфракрасное изображение

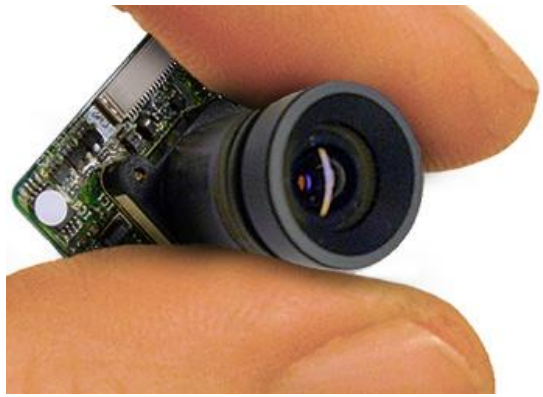
Используя невидимую глазу ИК-подсветку, такая камера будет видеть в темном помещении (на перфомансе)

Цветное изображение + глубина
(информация о расстоянии до объектов)

Способ передачи данных в компьютер

- Аналоговые
- **Веб-камеры** (USB-камеры)
- **Firewire-камеры** (камеры IEEE-1394)
- Сетевые (IP-камеры)
- "Умные" камеры (Smart cameras)

Аналоговые



Исторически появились первыми,
сигнал передается в аналоговом сигнале (формат ТВ).

(+) передают данные на большие расстояния,
хотя и с помехами (100 м)
(+) легко монтируются, малые размеры

(-) для ввода сигнала в компьютер требуется специальная плата или "ТВ-тюнер", они обычно потребляют много вычислительных ресурсов.
(-) "интерлейс", или черезстрочная развертка - очень затрудняет анализ изображения, если есть движение.
(фактически идет 2 полукадра, каждый 50 раз/сек)

Веб-камеры (USB-камеры)

Появились в ~2000г.,
передают данные через USB-протокол,
в несжатом виде, либо сжатом в JPEG.

(+) легко подключаются к компьютеру и программному обеспечению
(+) дешевые, имеются в продаже

(-) Накладные расходы - для раскодирования JPEG требуются вычислительные ресурсы.
(-) В дешевых моделях обычно плохая оптика и матрица (дает шум на изображении)
(-) Из-за ограничений пропускной способности USB нельзя подключить более 2-х камер к одному USB-хабу, но обычно на PC 2-3 USB хаба.

Firewire-камеры (IEEE-1394)

Камеры, передающие сигнал
по протоколу FireWire,
обычно в пылевлагозащитном корпусе,
обычно это камеры для промышленного применения.

(+) передают несжатое видео в отличном качестве на большой скорости
(+) можно подключать несколько камер
(+) обычно имеют отличную оптику

(-) высокая цена
(-) требуют питания, что иногда осложняет подключение к портативным компьютерам

Сетевые (IP-камеры)

Камеры, передающие данные по сетевому (проводному или беспроводному) каналу. Сейчас стремительно набирают популярность во всех сферах.

(+) простое подключение к компьютеру
(+) удобство монтажа
(+) возможность передачи данных на неограниченное расстояние, что позволяет конструировать сеть камер, охватывающих здание или район, крепить на дирижабль и т.п.
(+) возможность управления - вращать камеру, настраивать увеличение

(-) могут быть проблемы со скоростью отклика
(-) пока относительно высокая цена
(-) пока недостаточно портативны (2011 год)

"Умные" камеры (Smart cameras)

Камеры, в корпусе которых
располагается компьютер.
Такие камеры являются полнофункциональными
системами технического зрения,
передающие выходные данные об обнаруженных
объектах и т.п. по различным протоколам.

(+) компактность.
(+) масштабируемость - легко строить сети из таких камер.

(-) часто для них требуется адаптация существующих проектов.
(-) дешевые модели достаточно медленные, поэтому хорошо справляются лишь с относительно простыми задачами анализа изображений.

Отдельный тип: Инфракрасные камеры

Конструируется из обычной камеры
путем добавления ИК-фильтра
и, зачастую, ИК-подсветки.

+ ИК-лучи почти не видны человеку (в темноте видно как слабый красный цвет), поэтому часто используют для упрощения анализа объектов в поле зрения.

- специализированные ИК-камеры, подходящие для технического зрения, не являются массовым товаром, поэтому их обычно нужно заказывать.

Примеры хороших камер

Sony PS3 Eye

320 x 240 : 150 FPS
640 x 480 : 60 FPS

Типы данных:
видимый свет,
ИК (требуется удаление ИК-фильтра)

Цена: **50$**.

USB, CCD

Примеры хороших камер

Point Grey Flea3

648 x 488 : 120 FPS

Тип данных:
- видимый свет,
- ИК (?)

Цена: 600$.

Модель FL3-FW-03S1C-C
IEEE 1394b, CCD

Примеры хороших камер

Microsoft Kinect

640 x 480 : 30 FPS

Тип данных:
видимый свет + глубина

Цена: **150$**.

KINECT for XBOX 360.

(глубина - стереозрение с помощью лазернои ИК-подсветки, поэтому не работает при солнечном свете)
USB, **CMOS**

Примеры хороших камер

Point Grey BumbleBee2

640 x 480 : 48 FPS

Тип данных:
видимый свет + глубина

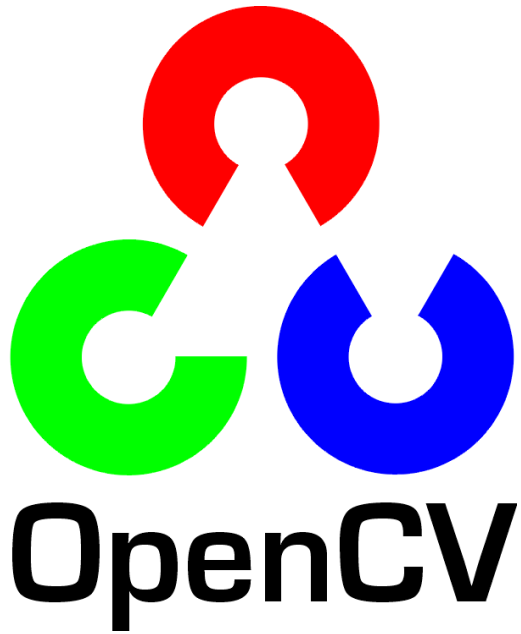
Цена: **2000$**.

(Глубина - стереозрение с двух камер)
IEEE 1394b, CCD

3. Знакомство с OpenCV

Перейти к оглавлению

Что такое OpenCV



"Open Computer Vision Library"

Открытая библиотека с набором функций для обработки, анализа и распознавания изображений, C/C++.

Что такое OpenCV

2000 - первая альфа-версия, поддержка Intel, C-интерфейс

2006 - версия 1.0

2008 - поддержка Willow Garage (лаб. робототехники)

2009 - версия 2.0, классы C++

2010 - версия 2.2, реализована работа с GPU

Первый проект на OpenCV

1. Создание проекта

Предполагаем, что Microsoft Visual C++ 2008 Express Edition
и OpenCV 2.1 уже установлены.

1. Запускаем VS2008

2. Создаем консольный проект
File - New - Project - Win32 Console Application,
в Name ввести Project1, нажать ОК.

3. Настраиваем пути
Alt+F7 - откроется окно свойств проекта
Configuration Properties - C/C++ - General - Additional Include Directories,
там ставим значение "C:\Program Files\OpenCV2.1\include\opencv";

Linker - General - Additional Library Directories, там ставим значение
C:\Program Files\OpenCV2.1\lib\

Linker - Input - Additional Dependencies -
cv210.lib cvaux210.lib cxcore210.lib cxts210.lib highgui210.lib для Release,
cv210d.lib cvaux210d.lib cxcore210d.lib cxts210.lib highgui210d.lib для Debug

Первый проект на OpenCV

2. Считывание изображения и показ его на экране

1. Готовим входные данные:

файл http://www.fitseniors.org/wp-content/uploads/2008/04/green_apple.jpg
пишем в C:\green_apple.jpg

2. Пишем в Project1.cpp:

```
#include "stdafx.h"
#include "cv.h"
#include "highgui.h"
using namespace cv;

int main( int argc, const char** argv )
{
    Mat image = imread( "C:\\green_apple.jpg" );   //Загрузить изображение с диска
    imshow( "image", image );                      //Показать изображение
    waitKey( 0 );                                  //Ждем нажатия клавиши
    return 0;
}
```

3. Нажимаем F7 - компиляция, F5 - запуск.

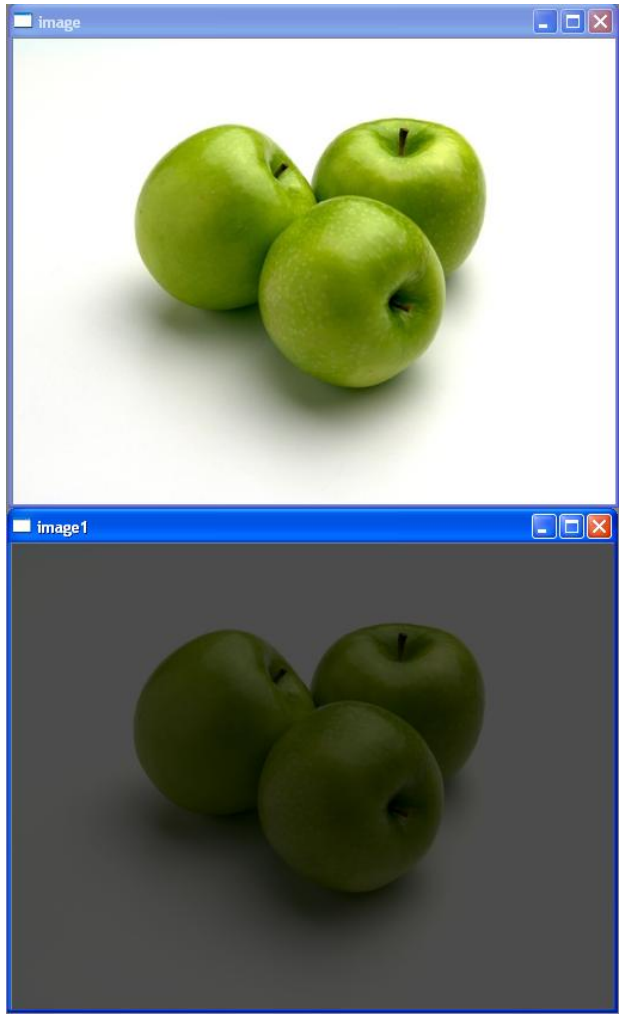
Программа покажет изображение в окне, и по нажатию любой клавиши завершит свою работу.

Первый проект на OpenCV
3. Линейные операции над изображениями

Заменяем текст в main из предыдущего примера на:

```
int main( int argc, const char** argv )
{
    Mat image = imread( "C:\\green_apple.jpg" );

    //image1 попиксельно равен 0.3*image
    Mat image1 = 0.3 * image;
    imshow( "image", image );
    imshow( "image1", image1 );
    waitKey( 0 );
    return 0;
 }
```



Первый проект на OpenCV

4. Работа с прямоугольными подобластями изображения

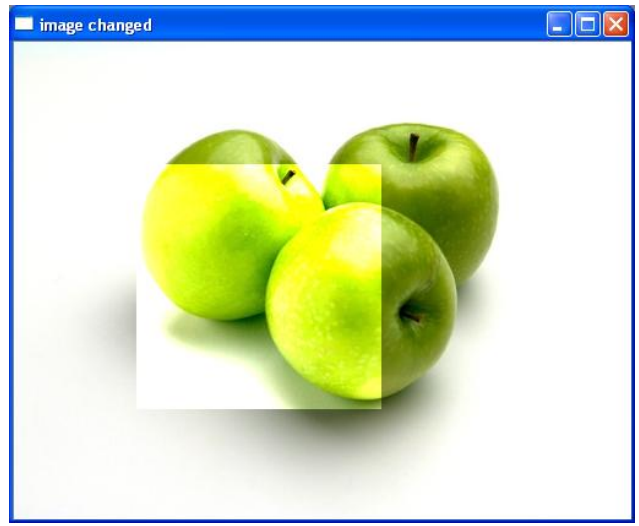
Заменяем текст в main из предыдущего примера на:

```
int main( int argc, const char** argv )
{
    Mat image = imread( "C:\\green_apple.jpg" );

    //Вырезание части картинки
    Rect rect = Rect(100, 100, 200, 200); //прямоугольник вырезания
    Mat image3;
    image( rect ).copyTo( image3 );         //копирование части image
    imshow( "image3", image3 );

    //Изменение части картинки внутри самой картинки
    image( rect ) *= 2;
    imshow( "image changed", image );

    waitKey( 0 );
    return 0;
 }
```



Класс Mat

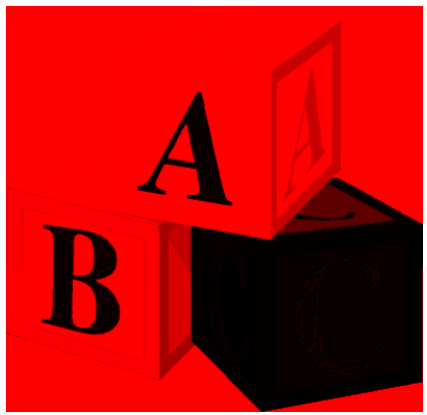
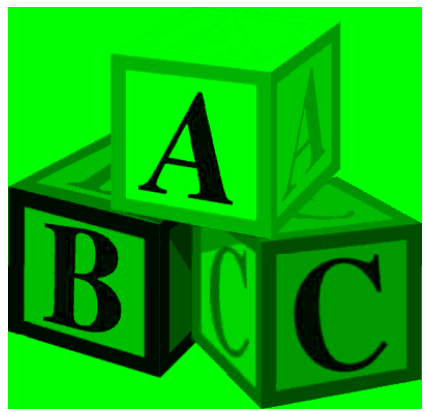
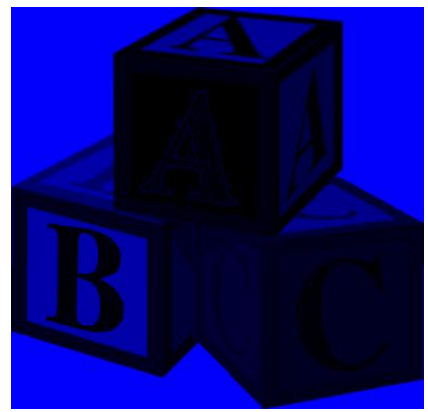
Mat - основной класс для хранения изображений OpenCV.

Класс Mat

Одно- и многоканальные изображения

Изображение является матрицей пикселов.
Каждый пиксел может хранить некоторые данные.
Если пиксел хранит векторные данные, то размерность вектора называется **числом каналов изображения**.

1-канальные изображения - называют еще полутоновыми
3-канальные изображения - обычно состоят из трех компонент (Red, Green, Blue).

Также, в OpenCV можно использовать 2- и 4-канальные изображения.

Класс Mat

Создание изображений

1) Пустое изображение без определенного типа

```
Mat imageEmpty;
```

2) Изображение **w x h** пикселов, значения 0..255
(**8U** значит "unsigned 8 bit", **C1** значит "один канал"):

```
int w=150; int h=100;
Mat imageGray( cv::Size( w, h ), CV_8UC1 );
```

Создание изображений

3) 1-канальное со значениями с плавающей точкой (32F значит "float 32 bit"):

```
Mat imageFloat( cv::Size(w, h), CV_32FC1 );
```

4) 3-канальное изображения со значениями 0..255 в каждом канале:

```
Mat imageRGB( cv::Size(w, h), CV_8UC3 );
```

Класс Mat

Управление памятью

1. Память для изображений выделяется и очищается автоматически

То есть, OpenCV сам создает изображение нужного типа и размера, если это изображение является выходным параметром некоторой функции:

```
Image imageFloat;
imageGray.convertTo( imageFloat, CV_32FC1, 1.0 / 255.0 );
```

- здесь OpenCV сам выделит память под imageFloat.
Важно, что если изображение уже нужного размера, то никаких операций по выделению памяти не производится.

2. Операции присваивания осуществляются не копированием данных (как это делает std::vector), и не путем копирования указателей, а с использованием **механизма счетчика ссылок**.

Класс Mat

Управление памятью

Механизм счетчика ссылок (в STL это shared_ptr, в Java он на всех указателях) работает так:

```
{
Mat A( cv::Size( 100, 100 ), CV_8UC1 );
//выделилась память под изображение, при этом запомнилось,
        //что эта память используется одним изображением.
{
Mat B = A;
//Тут память под изображение не выделяется, а просто
                //данные в B указывают на ту же область в памяти.
                //Поэтому, если менять B, то изменится и A.
                //Счетчик ссылок изображения увеличился, стал равным 2.
}
//Тут B вышло из области видимости, счетчик ссылок уменьшился,
//и стал равен 1.
}
//Тут A вышло из области видимости, счетчик ссылок стал равен 0,
//и память, выделенная для него, автоматически очищается.
```

Класс Mat

Управление памятью

Так как операция
Mat B = A;
не копирует изображение A в B, то для того, чтобы создать копию изображения для последующего независимого использования, нужно применять явные команды **copyTo** и **clone**:

```
image1.copyTo( image2 );
image3 = image1.clone();
```

Класс Mat

Управление памятью

Итог:
1) операция присваивания **Mat B = A;** работает очень быстро, и не осуществляет копирование данных, а настраивает специальным образом указатели на них. Это позволяет передавать **Mat** в функции прямо, без указателей и ссылок. При этом не возникнет нежелательного копирования **Mat** в стек (как это бы сталал std::vector).

Хотя, конечно, const **Mat &** будет передаваться все равно быстрее.

2) для копирования изображений нужно пользоваться явными командами **copyTo** и **clone**.

Попиксельный доступ к изображениям

В OpenCV есть несколько способов попиксельного доступа к изображениям. Они различны по степени безопасности (контроль типов и выхода за границы), по скорости работы и удобству.

Всюду, где это возможно, следует стараться избегать прямых обращений к пикселам, а вместо этого пользоваться функциями OpenCV, так как они обычно работают быстрее, а код понятнее.

Класс Mat

Попиксельный доступ к изображениям

Один из способов доступа к пикселам для изображений, у которых известен тип - использование метода **at**. Для одноканального изображения 0...255 это делается так:

```
//Взятие значения
int value = imageGray.at<uchar>(y, x);

//Установка значения
imageGray.at<uchar>(y, x) = 100;
```

Обратите внимание, что x и y в вызове переставлены местами.

Конвертация типов

Примечание

При выводе на экране изображений с плавающей точкой средствами OpenCV надо иметь в виду, что они отображаются в предположении, что их значения лежат в [0,1]. Поэтому при конвертации 8-битных изображений в изображения с плавающей точкой нужно делать трансформацию — умножение на 1.0 / 255.0.

Для конвертации типов изображений разной битности (float и unsigned char) используется член класса **convertTo**.
В нем второй параметр — тип получаемого изображения.

```
imageGray.convertTo( imageFloat, CV_32FC1, 1.0 / 255.0 );
```

Число каналов входа и выхода должно совпадать!

Конвертация типов

Для конвертации различных цветовых пространств используется функция **cvtColor**. При необходимости она способна менять число каналов.

Например, конвертация 3-канального RGB-изображения в полутоновое:

```
cvtColor( inputRGB, outputGray, CV_BGR2GRAY );
```

наоборот:

```
cvtColor( inputGray, outputRGB, CV_GRAY2BGR );
```

Класс Mat

Разбиение на каналы

Функция **split** разбивает многоканальное изображение на каналы.
Функция **merge** склеивает несколько одноканальных изображений в многоканальное.

```
void split ( const Mat& mtx,          //исходное цветное изображение
             vector<Mat>& mv          //результирующий набор 1-канальных
                                      //изображений
           )

void merge ( const vector<Mat>& mv,   //исходный набор 1-канальных
                                      //изображений
             Mat& dst                 //результирующее цветное
                                      //изображение
           )
```

Чаще всего они применяются для поканальной обработки цветных изображений, а также для различных манипуляций с каналами.

Класс Mat

Разбиение на каналы

```
int main( int argc, const char** argv )
{
    Mat image = imread( "C:\\green_apple.jpg" );

    //Разделение исходной картинки на три канала
    // - channels[0], channels[1], channels[2]

    vector<Mat> channels;
    split( image, channels );

    //Показываем каналы в отдельных окнах
    //Обратите внимание, что красный канал - 2, а не 0.

    imshow( "Red", channels[2] );
    imshow( "Green", channels[1] );
    imshow( "Blue", channels[0] );
    waitKey( 0 );
    return 0;
 }
```

Функции обработки изображений

Сглаживание

Исходное изображение

Изображение, сглаженное окном 11 x 11

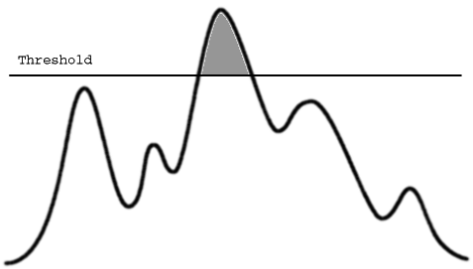
Функция **GaussianBlur** осуществляет сглаживание изображение фильтром Гаусса.

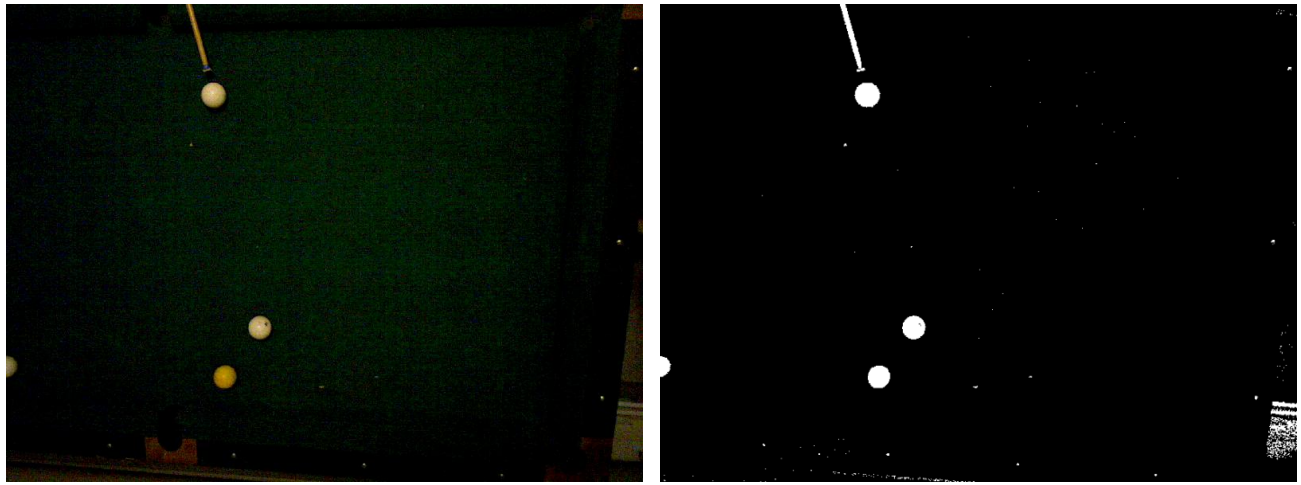
Чаще всего сглаживание применяется для устранения мелкого шума на изображении, для последующего анализа изображения. Делается с помощью фильтра небольшого размера.

http://www.innocentenglish.com/funny-pics/best-pics/stairs-sidewalk-art.jpg

Функции обработки изображений

Пороговая обработка





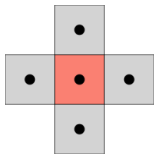
Функция **threshold** осуществляет пороговую обработку изображения.

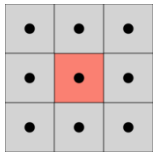
Чаще всего она применяются для выделения пикселов интересующих объектов на изображении.

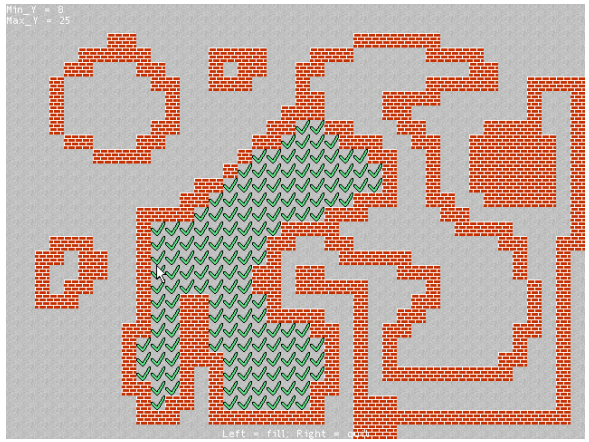
http://www.svi.nl/wikiimg/SeedAndThreshold_02.png

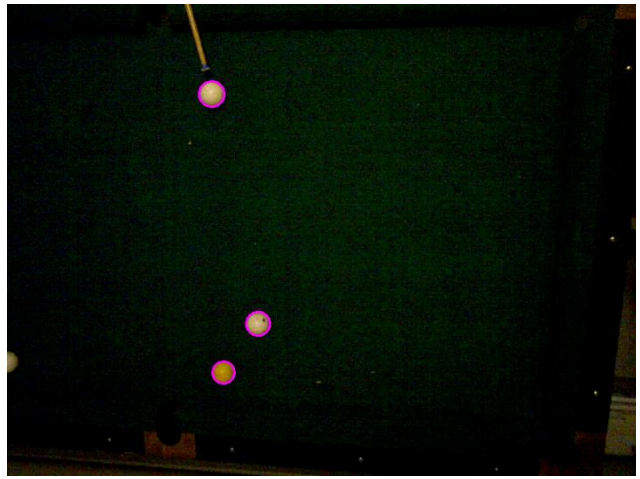
Функции обработки изображений

Заливка областей









Функция **floodFill** осуществляет заливку области, начиная из некоторого пиксела (x, y), с заданными границами останова,
используя 4- или 8- смежность пикселов.

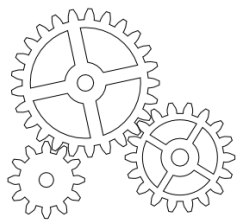
Важно: она портит исходное изображение - так как заливает его.

Чаще всего она применяется для выделения областей, выделенных путем пороговой обработки, для последующего их анализа.

http://upload.wikimedia.org/wikipedia/commons/thumb/5/5e/Square_4_connectivity.svg/300px-Square_4_connectivity.svg.png
http://tunginobi.spheredev.org/images/flood_fill_ss_01.png

Функции обработки изображений

Выделение контуров



Контур объекта - это линия, представляющая край формы объекта.
Подчеркивание точек контура - **Sobel**, выделение линий контура - **Canny**.

Применение

1. Распознавание. По контуру можно часто определить тип объекта, который мы наблюдаем.

2. Измерение. С помощью контура можно точно оценить размеры объекта, их поворот и расположение.

http:///howto.nicubunu.ro/gears/gears_16.png
http://cvpr.uni-muenster.de/research/rack/index.html

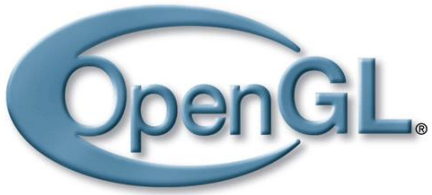
4. Интеграция OpenCV в мультимедиа-проекты

- Низкоуровневые библиотеки
- Среднеуровневые платформы
- Высокоуровневые среды

Перейти к оглавлению

Низкоуровневые библиотеки

Обработка, анализ и распознавание изображений



(Open Graphics Library)
Скоростная графика

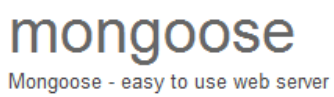
(Open Computing Language)
Распараллеливание и ускорения вычислений, в частности, средствами GPU.

(Open Audio Library)
Звук

Box2D - физический движок 2D

Bullet - физический движок 3D

[Видео 1](#) [Видео 2](#)



Веб-сервер

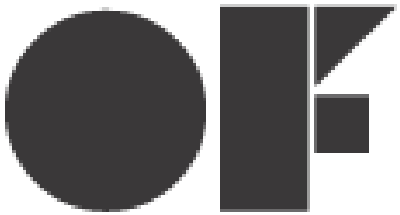
и так далее...

Среднеуровневые платформы

Это платформы для "Creative coding", включающие в себя большой набор функций и библиотек, интегрированных для удобного **программирования**.

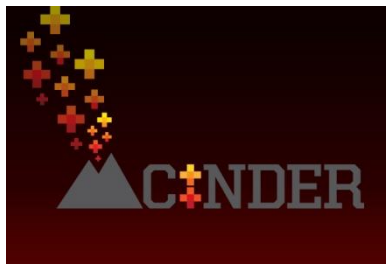
Processing

язык: Java
Для компьютерного зрения Java работает медленно.



openFrameworks

язык: C/C++



Cinder

язык: C/C++
Недавно появился, набирает популярность.

[Видео 1](#) [Видео 2](#) [Видео 3](#)

Качество результата
Сложность разработки

Высокоуровневые среды

Среды "визуального программирования", позволяющее реализовывать проекты без фактического программирования. Важно, что их можно расширять плугинами, сделанными в низкоуровневых средах.

Max/MSP/Jitter

Ориентирован на аудио.

VVVV

Ориентирован на видеоэффекты.

Unity3D

Ориентирован на качественный 3D.

5. Возможности и ограничения в **простых** задачах компьютерного зрения

Принципиальные возможности зрения

В принципе, с помощью компьютерного зрения можно измерить любые параметры физических процессов, если они выражаются в **механическом движении, изменении формы или цвета**.

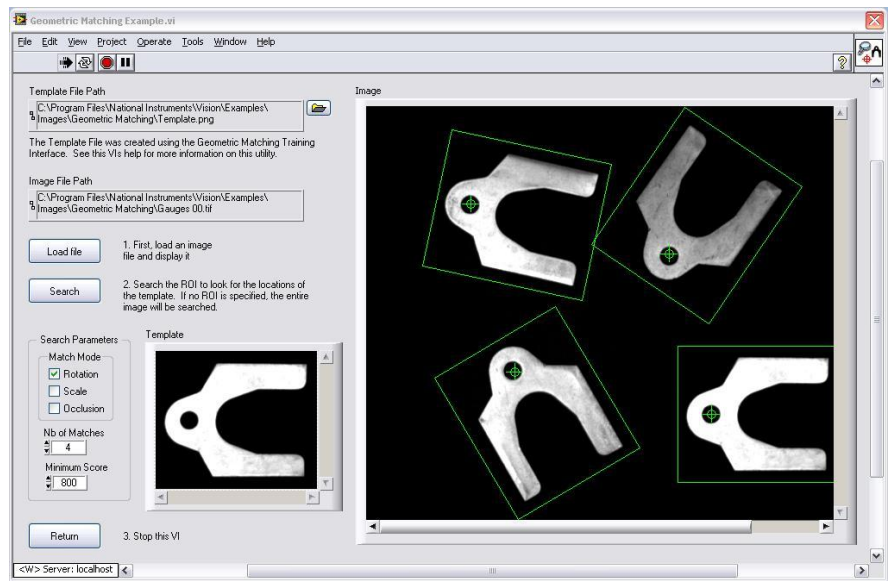
http://people.rit.edu/andpph/photofile-misc/strobe-motion-ta-08.jpg

Источник проблем

Основной источник проблем алгоритмического анализа изображений заключается в том, что нет простой связи

Значения пикселей ⟷ **Объекты в сцене**

Для простых случаев, когда такая связь есть - алгоритмы компьютерного зрения работают очень хорошо :)



http://zone.ni.com/cms/images/devzone/epd/GeometricMatchingScreenshot.JPG

"Проблема границ"

Близость цветов пикселов не значит, что они принадлежат одному объекту.

Аналогично, сильное различие цветов соседних пикселов не значит, что пикселы принадлежат разным объектам.

"Проблема границ"

Как отделить тень от дерева?

http://fineartamerica.com/images-medium/shadows-in-the-forest-gary-bydlo.jpg

"Проблема границ"

Для преодоления этой проблемы требуется строить алгоритмы, которые получают и используют контекстную информацию о расположении объектов в сцене.

"Проблема границ"

Как найти рыбу?

http://flogiston.ru/img/invisible_flounder_fish.jpg

"Проблема текстур"

На объектах бывают такие текстуры, которые не позволяют считать объекты одноцветными, но которые сложно смоделировать и описать.

"Проблема текстур"

Как выделить одну зебру?

http://dangerouswildlife.com/images/zebra-herd.jpg

"Проблема границ" и "проблема текстур" наиболее ярко проявляются при решении задачи сегментации.

Сегментация

Сегментация - разбиение изображения на "однородные" в некотором смысле области.

Цель сегментации - построить "простое" описание исходного изображения, которое можно было бы применить для последующего анализа объектов на изображении.

Сегментация

Существует множество методов сегментации,
которые используют самые различные идеи:

- "выращивание областей" по яркостному признаку,

- "метод змейки" - итерационное движение кривых,

- построение границ (например, методом "водораздела"),

- поиск разбиения на области, минимизирующий "энтропию",

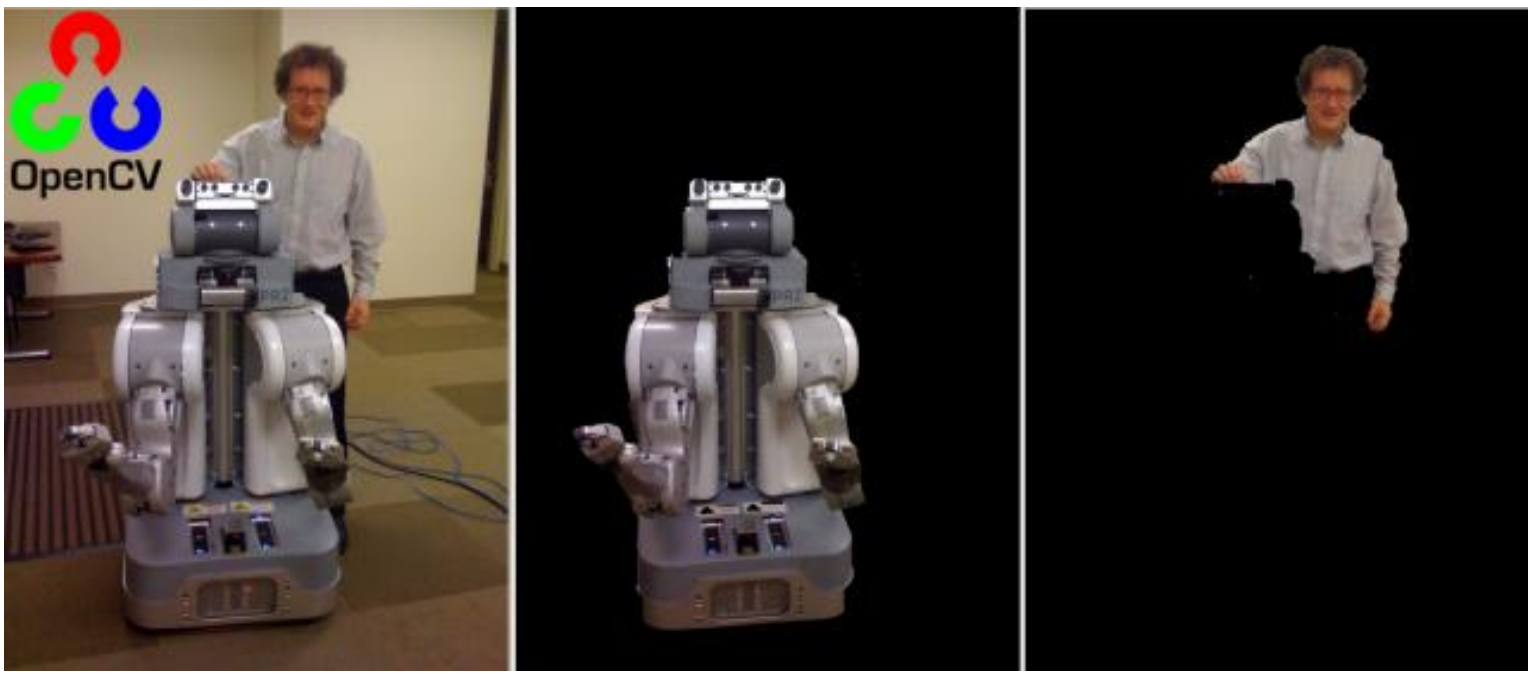
- использование априорной информации о форме области,

- использование нескольких масштабов для уточнения границ.

Сегментация

Один из лучших алгоритмов - метод **GrabCut**.
(Пока работает достаточно медленно.)

Метод основан на приближенном построении минимального разреза специального графа, который строится по пикселам изображения.



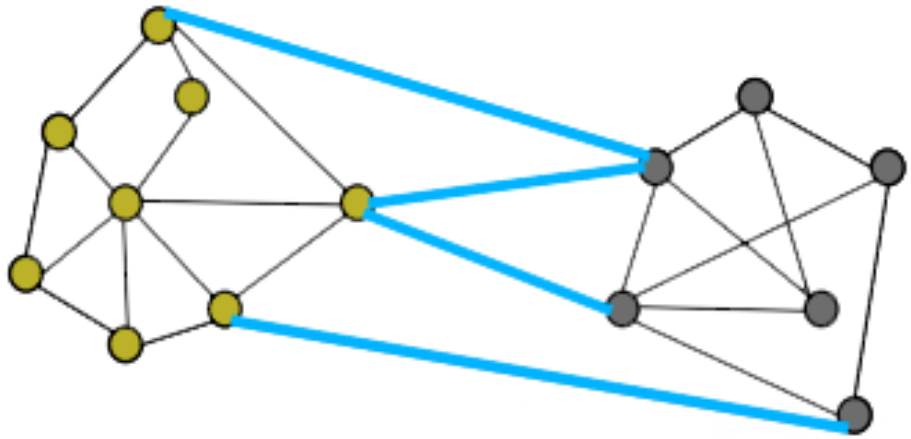
Сегментация

Идея:
строим взвешенный граф **G = { V, E }**.
пикселы изображения соответствуют вершинам **V**,
а геометрическая, яркостная и текстурная близость между двумя пикселами ***i, j*** соответствует весу ребра ***S_i,j***
("*S*" от "similarity" - близость пикселов)

http://www.cis.upenn.edu/~jshi/GraphTutorial/

Сегментация

Тогда задачу сегментации на две области можно сформулировать как задачу разбиения множества вершин графа на две части. Такое разбиение называется **разрезом**.

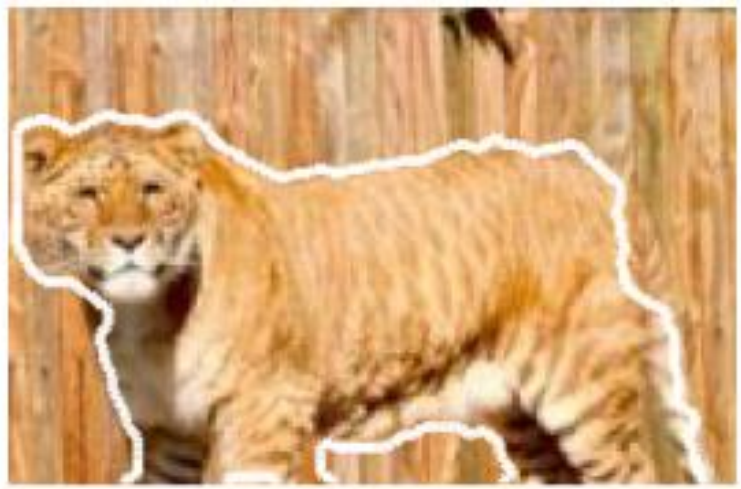


$$cut(A, \bar{A}) = \sum_{i \in A, j \in \bar{A}} S_{i,j}$$ - величина разреза.

Минимальный разрез (т.е. разрез с минимальной величиной) - объявим решением задачи сегментации.

Сегментация

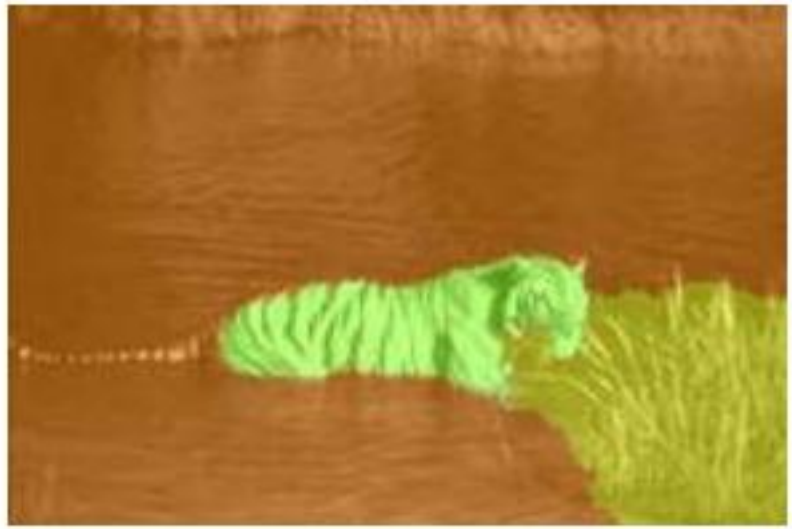
Дополнив метод минимального разреза ручной первоначальной разметкой областей, можно получить весьма хорошие результаты (правда, уже не полностью автоматические):



James Malcolm, Yogesh Rathi, Allen Tannenbaum
A Graph Cut Approach to Image Segmentation in Tensor Space
http://jgmalcolm.com/pubs/malcolm_tc.pdf

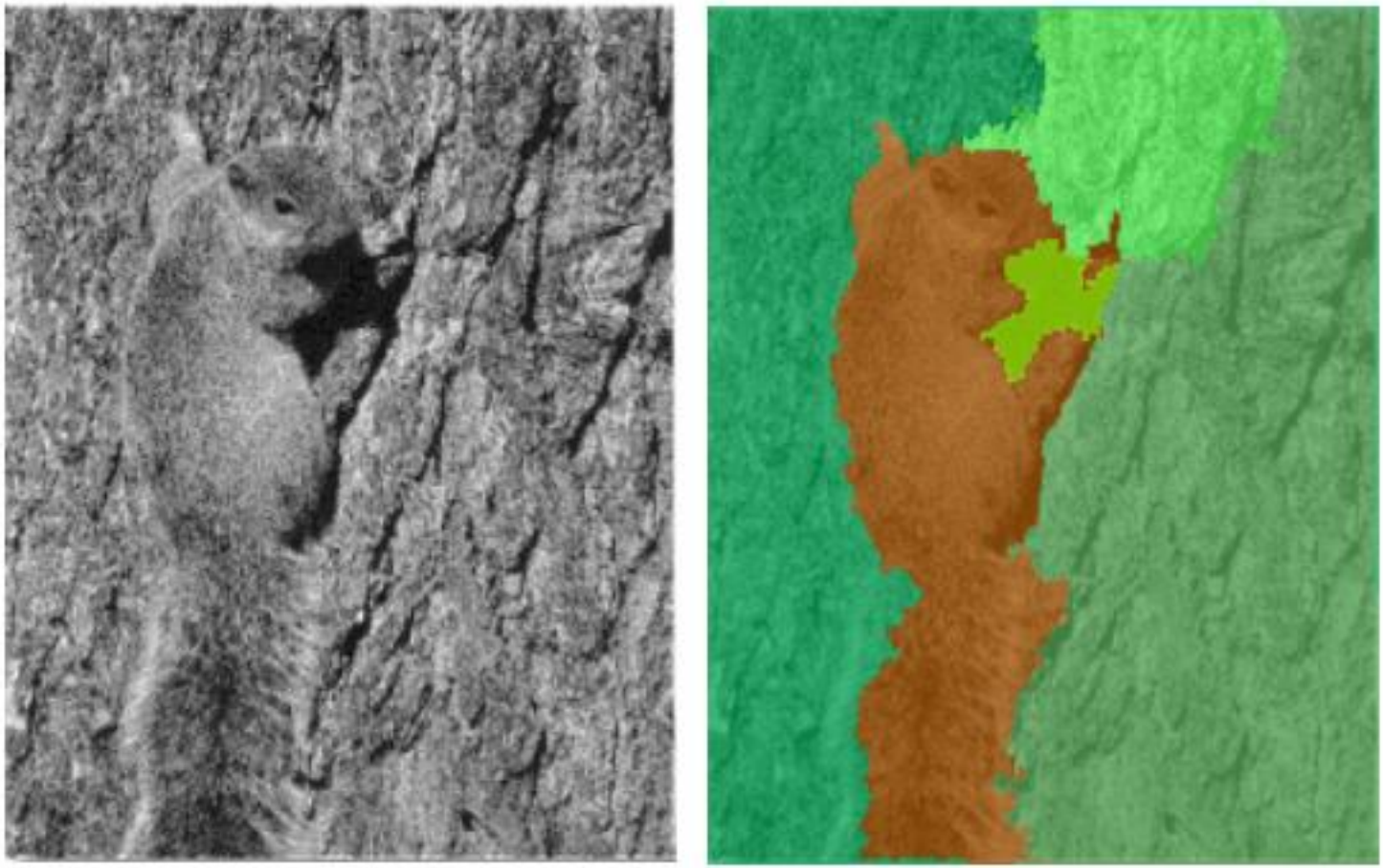
Сегментация

Добавив идею многомасштабного анализа (полностью автоматический результат):



Eitan Sharon, Segmentation by Weighted Aggregation, CVPR’04
http://www.cis.upenn.edu/~jshi/GraphTutorial/Tutorial-ImageSegmentationGraph-cut4-Sharon.pdf

Сегментация



Eitan Sharon, Segmentation by Weighted Aggregation, CVPR’04
http://www.cis.upenn.edu/~jshi/GraphTutorial/Tutorial-ImageSegmentationGraph-cut4-Sharon.pdf

Сегментация

Eitan Sharon, Segmentation by Weighted Aggregation, CVPR’04
http://www.cis.upenn.edu/~jshi/GraphTutorial/Tutorial-ImageSegmentationGraph-cut4-Sharon.pdf

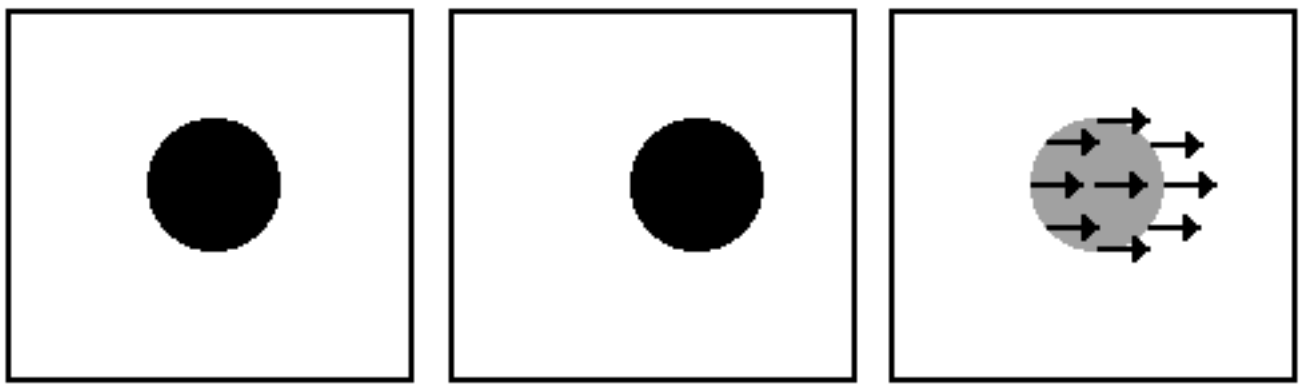
Оптический поток

Оптический поток (optical flow) - это векторное поле видимого движения объектов, поверхностей и ребер в визуальной сцене, вызванное относительным движением между наблюдателем и сценой.

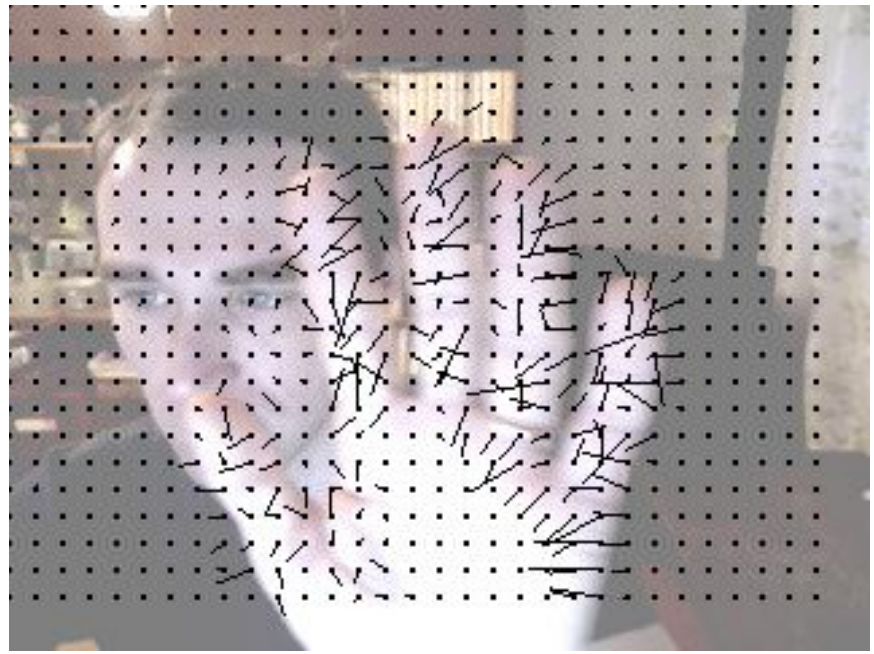
Оптический поток

Обычно рассматривается оптический поток, возникающий при рассмотрении двух кадров видео.

Для каждого пиксела (x, y) оптический поток представляет собой вектор $(f(x, y), g(x, y))$, характеризующий сдвиг:



Оптический поток



http://www.ultimategraphics.co.jp/jp/images/stories/ultimate/BCC/optical.jpg

Применения оптического потока

1. Для определения направления, в котором движутся объекты в кадре. Видео

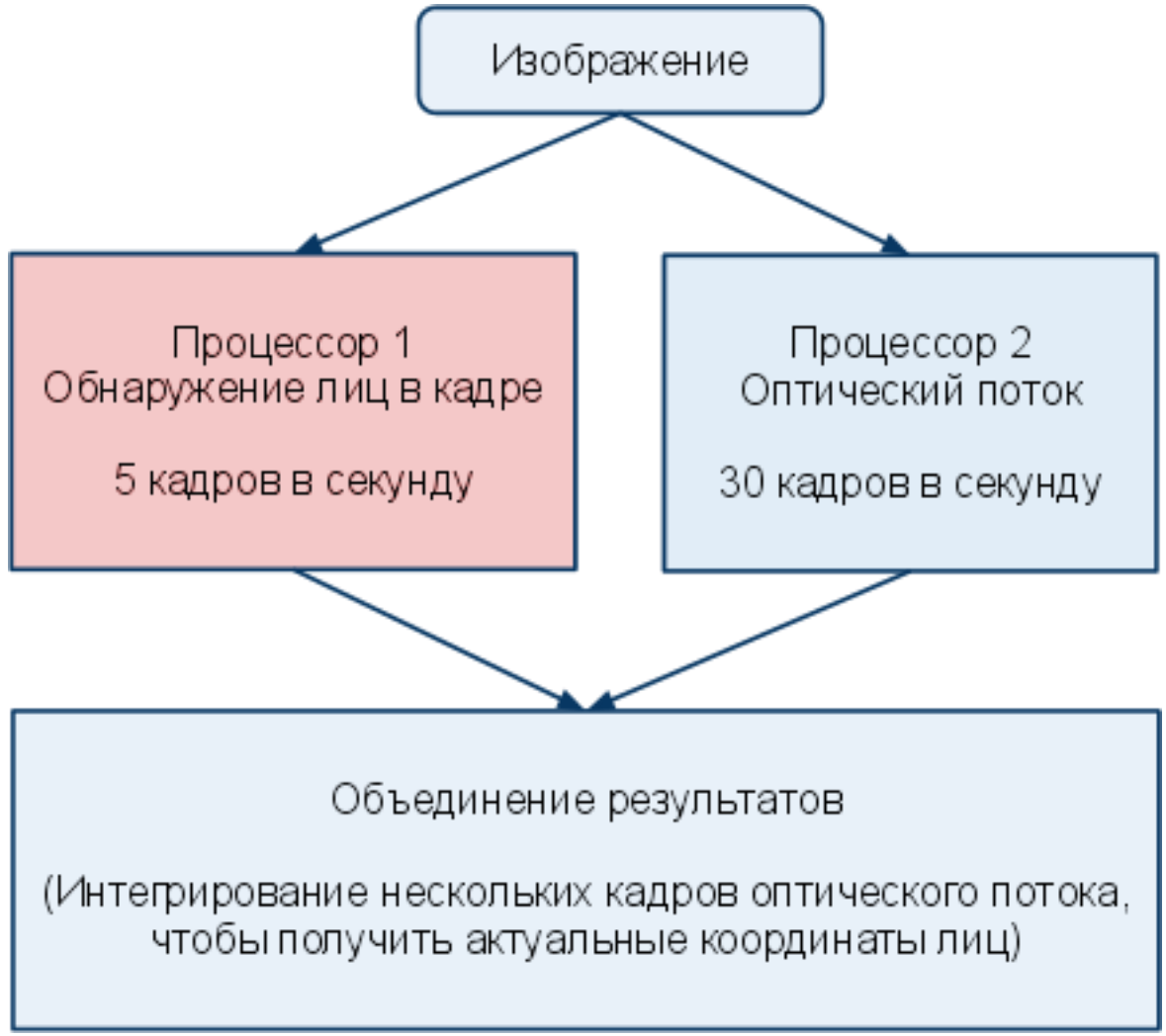
2. Для сегментации движущихся областей в кадре для последующего анализа.

3. Для восстановления формы трехмерного объекта, возле которого движется камера.

4. Как вспомогательный способ повышения устойчивости алгоритмов обнаружения объектов, если они находятся не в каждом кадре.
Например, для задач поиска лиц, маркеров и т.п.

Применения оптического потока

Повышение устойчивости распознавания лиц.



Видео — Зеленые окружности - результат Процессора 1, фиолетовые прямоугольники - Объединение результатов.

Методы вычисления оптического потока

(I) Блочные ("наивные" методы)

Для каждой точки ищется сдвиг, минимизирующий разность в локальном окне.

(II) **Дифференциальные** (наиболее используемые)

Оценка производных по x, y, t.

1. **Lucas-Kanade** - очень быстрый.
2. **Farneback** - достаточно качественный, но работает медленней
3. **CLG** - качественный, но пока не реализован в OpenCV
4. Пирамидальный Lucas-Kanade, вычисляющийся только на "точках интереса"
5. Horn-Schunk - не очень устойчивый к шумам

(III) На основе дискретной оптимизации (ресурсоемкие)

Решение строится с помощью методов min-cut, max-flow, линейного программирования или belief propagation.

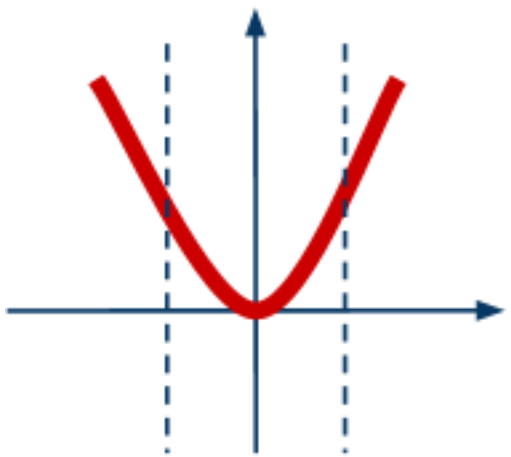
Методы вычисления оптического потока

Сегодня в OpenCV реализовано несколько алгоритмов, лучший из них - Farneback.

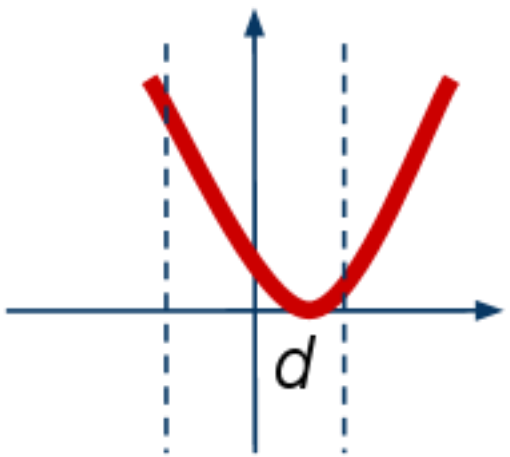
(Gunnar Farneb□ack "Two-Frame Motion Estimation Based on Polynomial Expansion", Proceedings of the 13th Scandinavian Conference on Image Analysis Pages 363-370 June-July, 2003)

Методы вычисления оптического потока

Идея заключается в аппроксимации квадратичной функцией яркостей пикселов в окрестности некоторого пиксела на обоих кадрах. Используя коэффициенты полиномов, можно вычислить сдвиг - который объявляется значением оптического потока в данном пикселе.



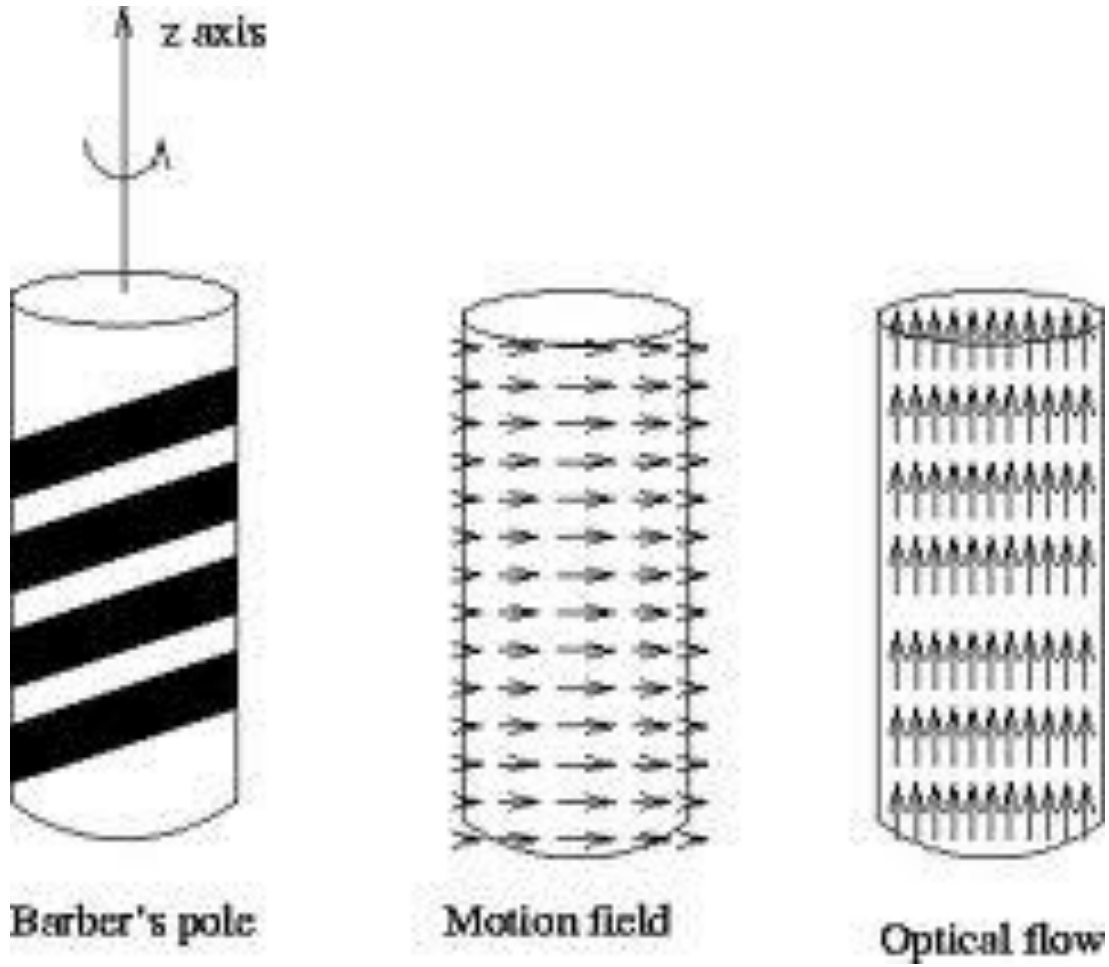
$y = x^2$



$y = x^2 + bx + c = (x - d)^2$

Проблемы оптического потока

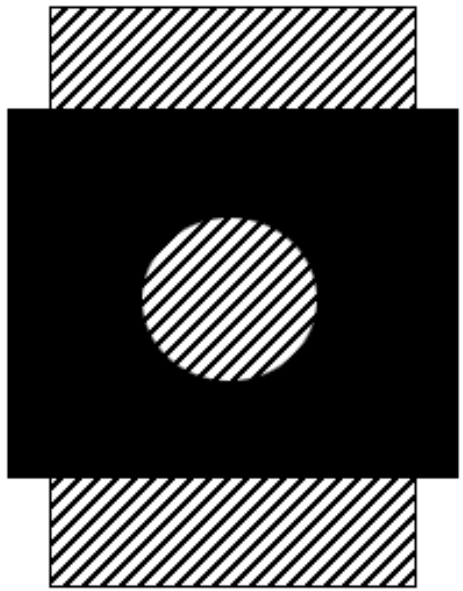
Оптический поток и фактическое поле движения могут не совпадать, и даже быть перпендикулярными:



Проблемы оптического потока

Проблема апертуры - неоднозначность определения движения, вызванная рассмотрением движения лишь локально (не анализируя края объекта).

Проявляется особенно сильно
- в малотекстурированных сценах
- в комбинаторных сценах типа полосок и шахматных досок



http://pages.slc.edu/~ebj/sight_mind/motion/Nakayama/aperture_problem.GIF

Домашнее задание 1 из 2

Для получения зачета по этим лекциям "автоматом"

Построить картинку с объектами, состоящими из черных полос и клеток на белом фоне, затем вторую картинку, где эти объекты сдвинуты на расстояние, превышающее ширину полос. Затем рассчитать и вывести на экран получившийся оптический поток.

Результаты присылайте на **perevalovds@gmail.com**

6. Возможности и ограничения в **сложных** задачах компьютерного зрения

- Поиск лиц - алгоритм Виолы-Джонса
- Поиск пешеходов - алгоритм HOG

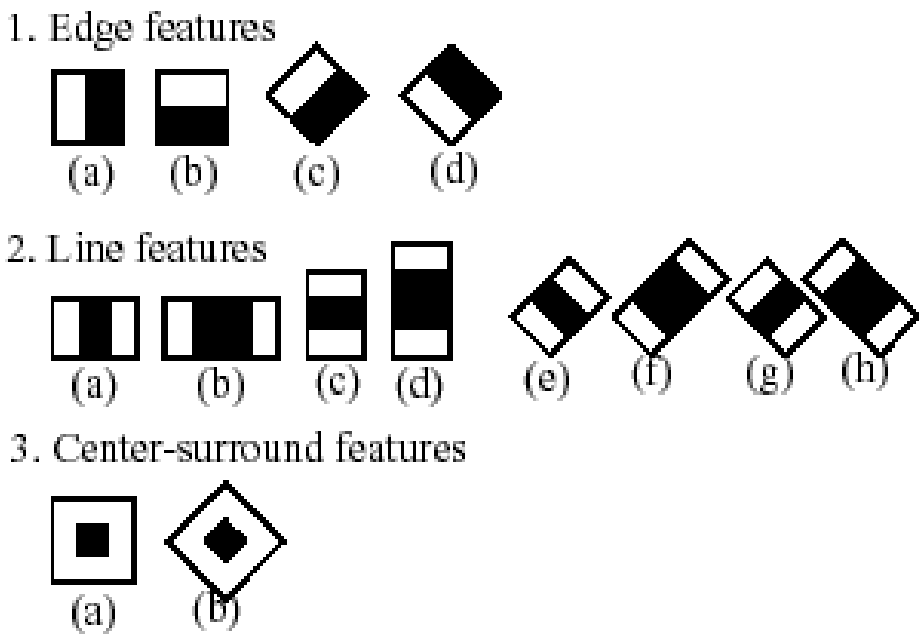
Перейти к оглавлению

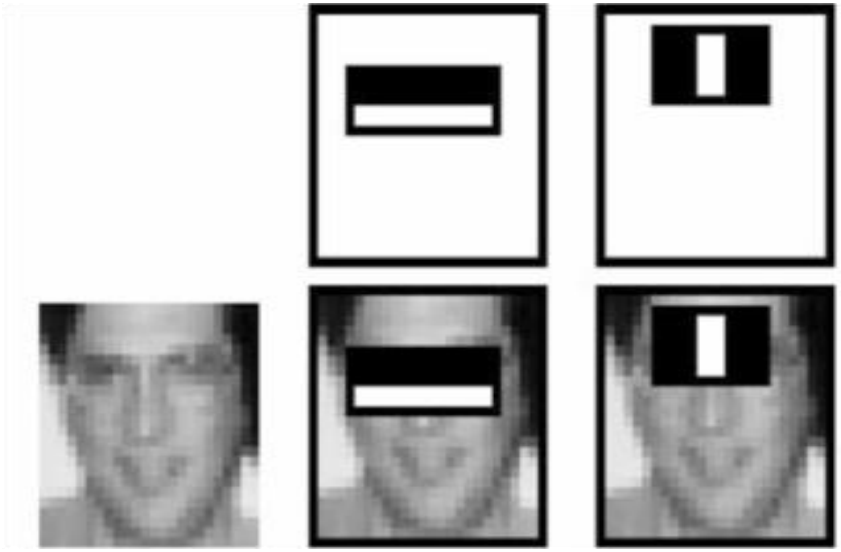
Поиск лиц - алгоритм Виолы-Джонса

Алгоритм Виолы-Джонса является сегодня базовым для поиска фронтальных лиц в кадре.

Поиск лиц - алгоритм Виолы-Джонса

Для работы используется набор "признаков типа Хаара",





http://www710.univ-lyon1.fr/~bouakaz/OpenCV-0.9.5/docs/ref/pics/haarfeatures.png
http://alliance.seas.upenn.edu/~cis520/wiki/uploads/Lectures/viola_jones_first2_small.png

Поиск лиц - алгоритм Виолы-Джонса

На стадии обучения из избыточного набора признаков методом **бустинга** строится набор классификаторов.

Поиск лиц - алгоритм Виолы-Джонса

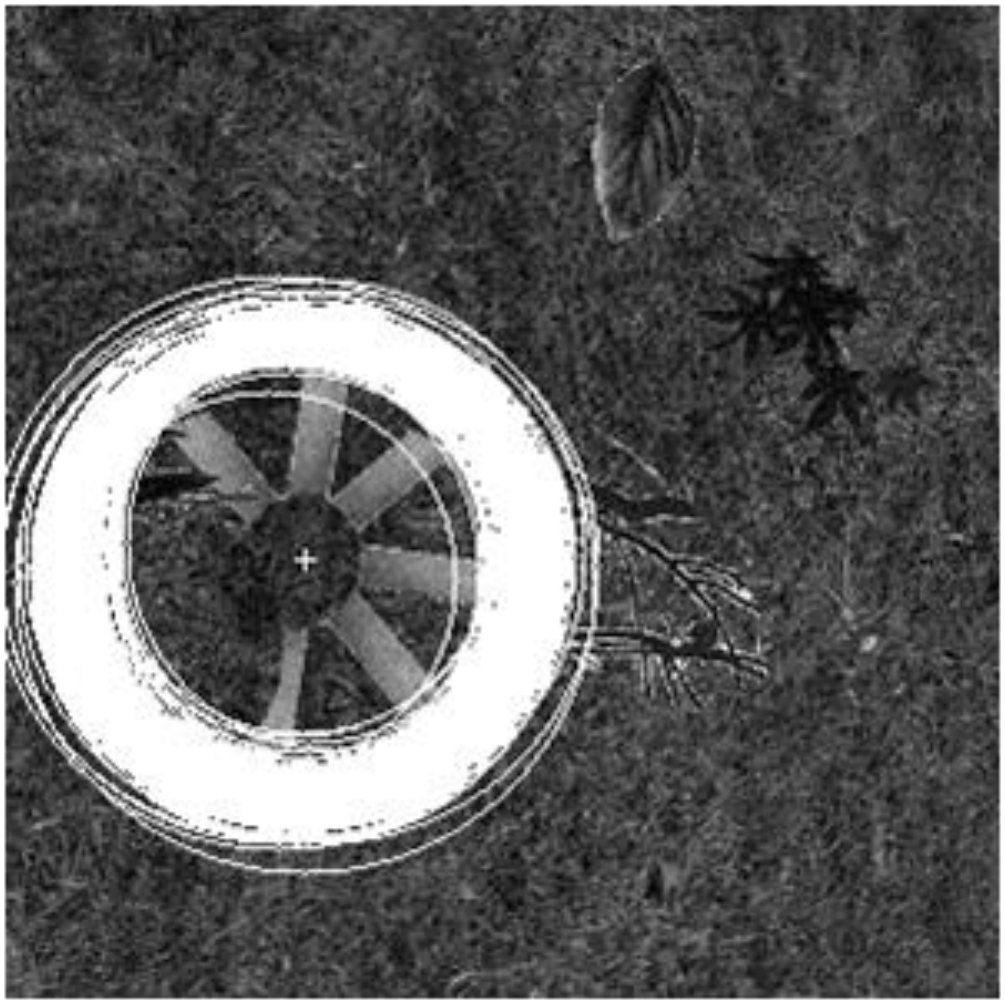
- работает хорошо на фронтальных лицах.

Проблемы
- для лиц в профиль не работает из-за причесок.
- для всего человека или верхней части - мне не удалось добиться распознавания.

Это связано, по-видимому, с тем, что алгоритм способен хорошо обучиться распознаванию внутренних контуров практически неизменного объекта. А с внешними изменяющимися контурами он работает не очень хорошо.

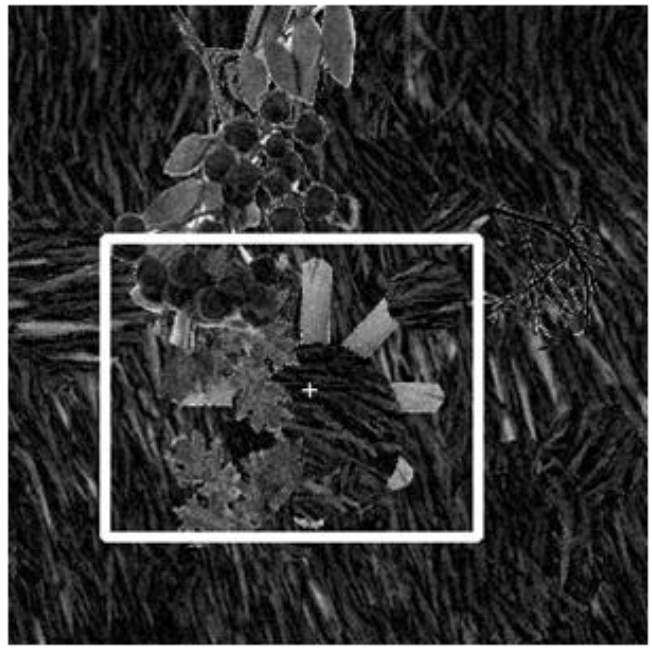
Поиск лиц - алгоритм Виолы-Джонса

Применение для поиска загороженных объектов в траве.



Поиск лиц - алгоритм Виолы-Джонса

Применение для поиска загороженных объектов в траве.



•частота пропуска: **0.158** *(158 среди 1000 положительных примеров) ;*

• частота ложной тревоги: **0.049** *(40 среди 1000 положительных примеров, 58 среди 1000 отрицательных примеров);*

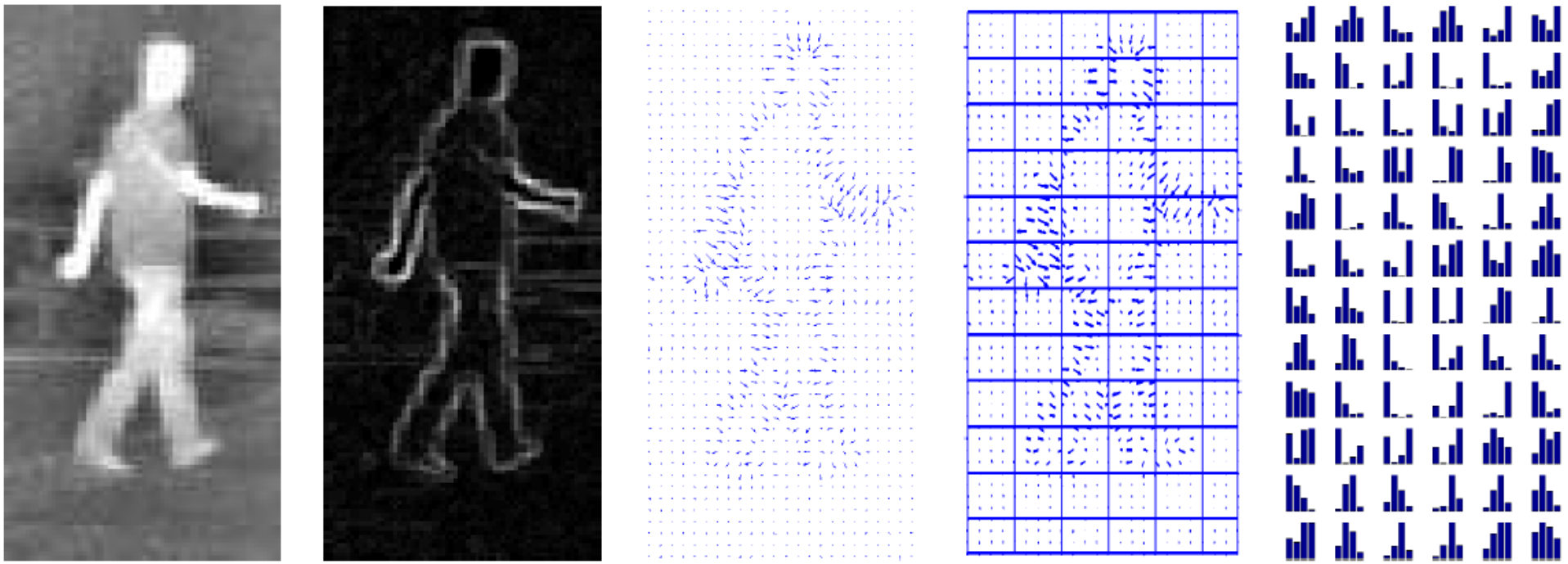
• таким образом, частота правильного обнаружения составила **84.2%**, частота ложной тревоги **4.9%**.

Поиск пешеходов - алгоритм HOG

HOG = Histogram Of Gradients,
гистограмма градиентов.

Поиск пешеходов - алгоритм HOG

Принцип работы: изображение разбивается на области, в которых считается направление градиента. Эти направления аккумулируются в гистограммы. Полученный вектор используется для распознавания образов (метод SVM).



http://ericbenhaim.free.fr/images/hog_process.png

Поиск пешеходов - алгоритм HOG

Алгоритм достаточно надежно может распознавать автомобили, мотоциклы, велосипеды.

Демо:

Видео http://www.youtube.com/watch?v=BbL2wWy8KUM

Проблемы:

Судя по демо-видео, алгоритм не очень хорошо работает с людьми в юбках, или он был обучен на распознавание людей под другим ракурсом.

7. Новые применения компьютерного зрения

- Интерактивные мультимедиа системы
- 3D-иллюзия
- Мэппинг
- Динамический мэппинг

Перейти к оглавлению

Интерактивные мультимедиа системы

Funky Forest (2007, T.Watson)

Видео http://zanyparade.com/v8/projects.php?id=12

Интерактивные мультимедиа системы

Body paint

Видео http://www.youtube.com/watch?v=3T5uhe3KU6s

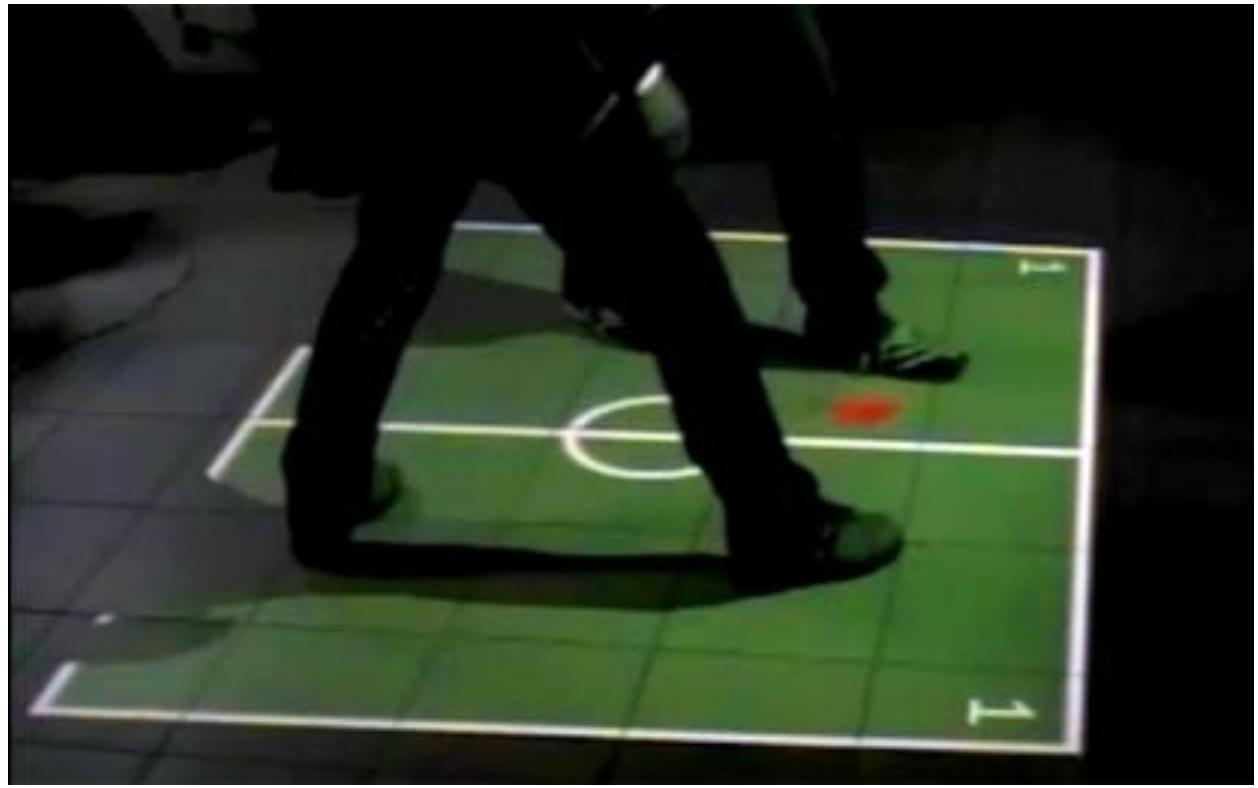
Интерактивные мультимедиа системы

Проекция на руки зрителей

Yoko Ishii and Hiroshi Homura, It’s fire, you can touch it, 2007.

Интерактивные мультимедиа системы

Напольные игры



Видео с чемпионата по напольному пинг-понгу, база отдыха Хрустальная-2011

Чемпионат проводился в рамках конференции "Современные проблемы математики" - 42-я Всероссийская молодежная школа-конференция, база отдыха Хрустальная, 2 февраля 2011 г.

www.playbat.net

Видео

3д-иллюзия

Это иллюзия восприятия объемного тела на (плоской) поверхности, которая достигается путем точной имитации геометрии и светотени для данного тела из той точки, где стоит зритель.

http://justinmaier.com/wp-content/uploads/2006/05/ATT5082002.jpg

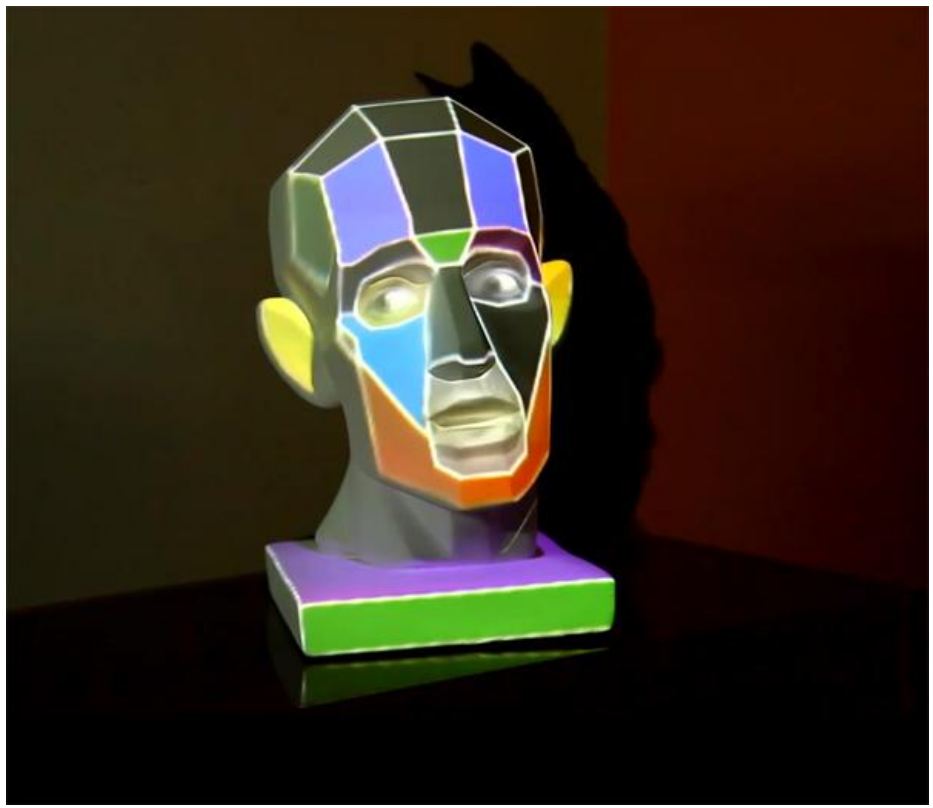
3д-иллюзия

Создание иллюзии 3д с помощью трекинга головы и глаз. Сейчас встраивается в портативные игровые приставки, имеющие камеры.

Видео http://www.youtube.com/watch?v=o5tlIIOXMs4

Мэппинг

Мэппинг (projection mapping) – осуществление проекции видео не на специальные экраны, а на другие объекты для их “оживления”.



Видео http://www.youtube.com/watch?v=BLNqZ1Nbo7Q

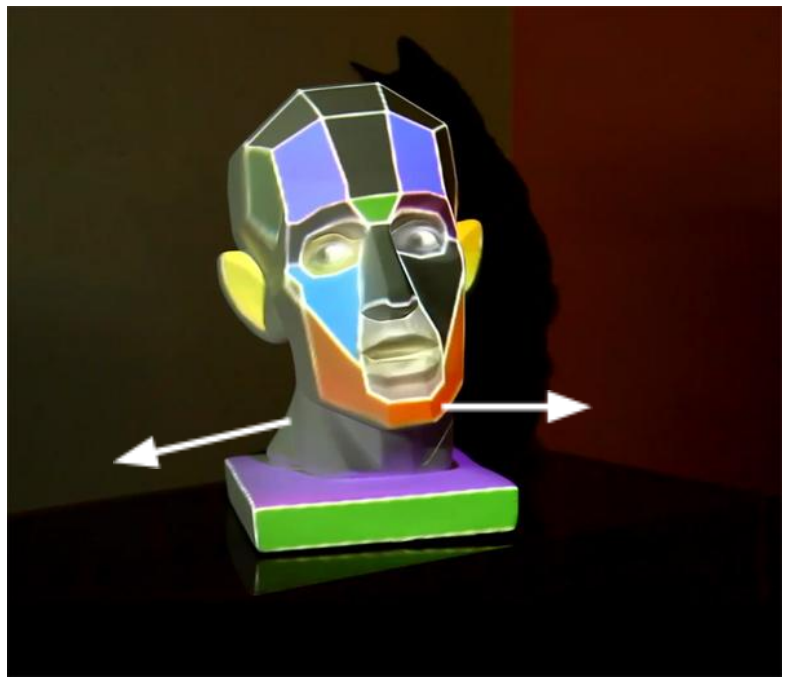
Мэппинг

Сегодня популярен мэппинг на здания (Architectural Projection Mapping).

Видео http://www.youtube.com/watch?v=BGXcfvWhdDQ

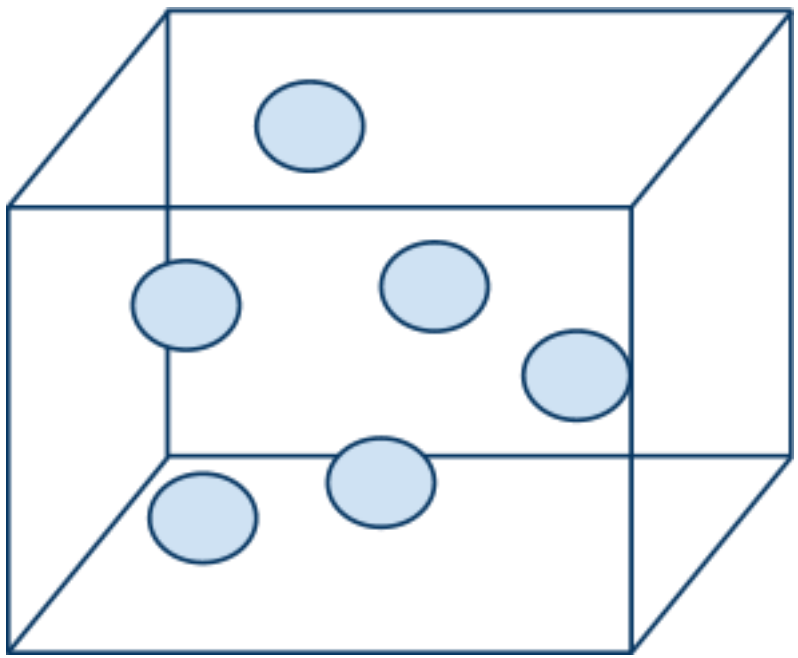
Динамический мэппинг

Идея: использовать методы трекинга и методы Markerless AR для синхронизации движущегося объекта и изображения с проектора.



Динамический мэппинг

Более радикально: отслеживать движение нескольких объектов (скачущие шары, падающие листы бумаги), и осуществлять проекцию на них.



Домашнее задание 2 из 2

Для получения зачета по этим лекциям "автоматом"

Сделать трекинг (обнаружение положения) падающего мячика, который затем отскакивает и скачет.

Заснять видео, на котором виден мяч и поверх показано положение мяча, найденного компьютером. Видео выложить на **youtube**, ссылку выслать на **perevalovds@gmail.com**.

За пределами зрения и графики: Физические вычисления

Водный музыкальный инструмент,
звук запускается волнами

Aleatoric water musical instrument

Видео http://www.youtube.com/watch?v=CZ_KijiwQHE

Заключение

- Литература
- Дружественные лекции и семинары
- Партнеры
- Контакты

Перейти к оглавлению

Литература

Компьютерное зрение

1. *Гонсалес Р., Вудс Р.* Цифровая Обработка Изображений.
2. *Л. Шапиро, Дж. Стокман* Компьютерное зрение.

OpenCV

1. Документация OpenCV
C++: http://opencv.willowgarage.com/documentation/cpp/index.html

2. *G. Bradski, A. Kaehler* Learning OpenCV: Computer Vision with the OpenCV Library
- к сожалению, для версии OpenCV для C, а не C++.

3. Мои лекции по OpenCV для C++ (матмех, осень 2010)
http://uralvision.blogspot.com/2010/12/opencv-2010.html

Дружественные лекции и семинары


Разработка интерактивных систем на OpenFrameworks

Содержание курса

Для кого этот спецкурс

Контакты

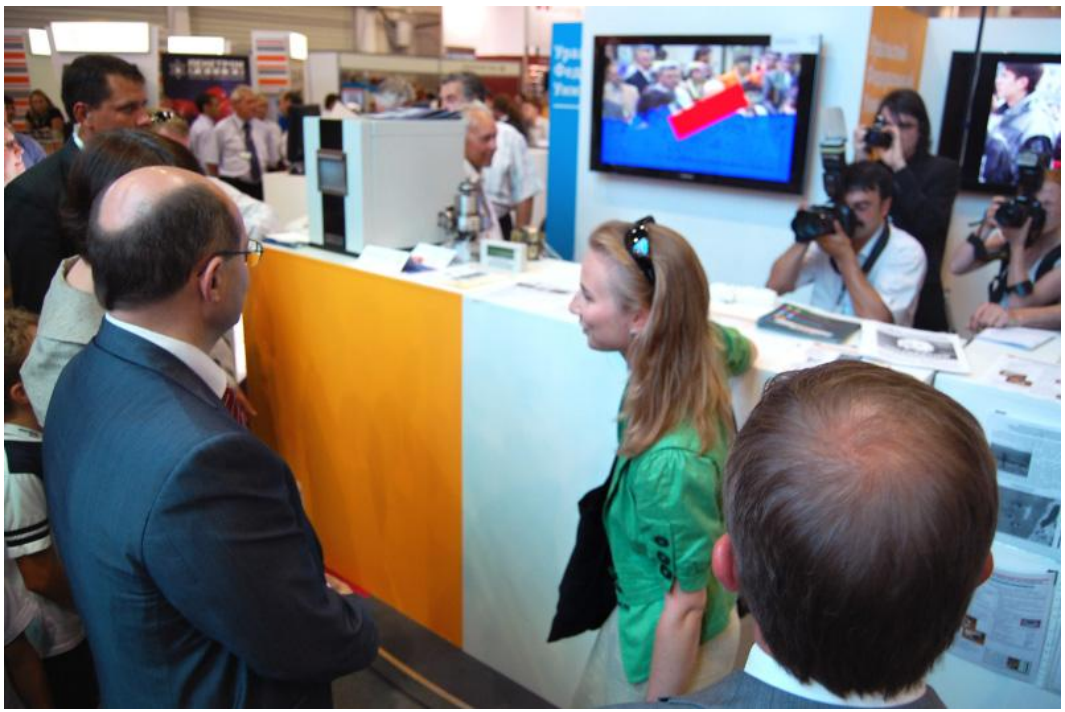
Спецкурс по **openFrameworks** с элементами **OpenCV**, матмех УрГУ, весенний семестр 2011.

Программа спецкурса и время занятий будет на сайте **www.uralvision.blogspot.com**

Дружественные лекции и семинары

УрГУ - Факультет искусствоведения и культурологи
Екатеринбургский ф-л Государственного Центра современного искусства
УрО РАН

Программа "Искусство, Наука, Технологии"



art.usu.ru/index.php/confer/229
www.art.usu.ru, www.ncca.ru

Дружественные лекции и семинары

Программа "Искусство, Наука, Технологии"
2-го марта, 18:30
Тело как интерфейс: Wearable Computing

ПРИГЛАШЕНИЕ К УЧАСТИЮ с 10-15 минутным сообщением и без него.
Просьба сообщить о желании и теме выступлений Ксении Федоровой, ksenfedorova@gmail.com
Семинар будет проходить в медиатеке УрГУ (Ленина 51, 4 этаж, 413а)
Подробная информация - http://www.scribd.com/doc/49078187/Body-As-Interface

Дружественные лекции и семинары

Программа "Искусство, Наука, Технологии"

21-22 апреля

2-й Международный семинар

"Теории и практики медиаискусства"

Председатель орг.комитета Ксения Федорова, куратор ЕФ ГЦСИ, ksenfedorova@gmail.com

Информация о семинаре будет опубликована на ncca.ru, art.usu.ru, usu.ru и www.uralvision.blogspot.com

http://www.ago.net/assets/images/assets/past_exhibitions/2007/steinkamp.jpg

Партнеры

www.playbat.net (интерактивные системы)

ООО Бизнес-Фрейм (консалтинг комп. зр.) www.bframe.ru

ООО 5-е измерение (интерактивные системы)

gorodaonline.com (сеть информационно-деловых порталов)

Анимационная студия "Муль-ОН" www.mult-on.ru

Анимационная студия "Аниматех" www.anima-tech.ru

Контакты

Перевалов Денис Сергеевич
perevalovds@gmail.com

Эта лекция опубликована на
www.uralvision.blogspot.com