

# Информатика. OpenCV

Знакомство с библиотекой  
алгоритмов компьютерного зрения  
и обработки изображений

# Материалы

- В интернете их очень-очень много:  
<http://robocraft.ru/page/opencv/>
- Запрос в гугле «learning opencv pdf» 😊

# Hello world!

```
#include <opencv/cv.hpp>
#include <iostream>

using namespace cv;

int main()
{
    Mat image;
    image = imread("donov.jpg", CV_LOAD_IMAGE_COLOR); // Read the file

    if (!image.data) ..... // Check for invalid input
    {
        std::cout << "Could not open or find the image" << std::endl;
        return -1;
    }

    namedWindow("Display window", WINDOW_AUTOSIZE); // Create a window for display.
    imshow("Display window", image); ..... // Show our image inside it.

    waitKey(0); ..... // Wait for a keystroke in the window
    return 0;
}
```

# Градиент изображения

```
--Mat blurred;  
--GaussianBlur(image, blurred, Size(15, 15), 0, 0);  
  
--Mat grayScale;  
--cvtColor(blurred, grayScale, COLOR_BGR2GRAY);  
  
--Mat dX, dY;  
--Sobel(grayScale, dX, CV_32F, 1, 0);  
--Sobel(grayScale, dY, CV_32F, 0, 1);  
  
--Mat mag, ang;  
--cartToPolar(dX, dY, mag, ang);  
  
--auto it = std::max_element(mag.begin<float>(), mag.end<float>());  
--auto max_elem = mag.at<float>(it.pos());  
--mag /= max_elem;
```

# Захват видео

```
#include <opencv/cv.hpp>
#include <opencv/highgui.h>

using namespace cv;

int main()
{
    VideoCapture cap(0); // open the default camera
    if (!cap.isOpened()) // check if we succeeded
        return -1;
    namedWindow("frame", CV_WINDOW_AUTOSIZE);
    while (true)
    {
        Mat frame;
        cap >> frame; // get a new frame from camera
        imshow("frame", frame);
        if (waitKey(30) == 27) break;
    }
    // the camera will be deinitialized automatically in VideoCapture destructor
    return 0;
}
```

# Упражнение 1

Сделайте так, чтобы

- на видео выделялись контура с помощью `cv::Canny`
- измените цветовую палитру на `hsv`

# Ищем лицо на фото

```
CascadeClassifier face_cascade;

void detectAndDisplay(Mat frame)
{
    std::vector<Rect> faces;
    Mat frame_gray;
    cvtColor(frame, frame_gray, COLOR_BGR2GRAY);
    equalizeHist(frame_gray, frame_gray);

    // Detect faces
    face_cascade.detectMultiScale(frame_gray, faces, 1.1, 2, 0 | CASCADE_SCALE_IMAGE, Size(30, 30));

    for (const auto& f : faces) // Iterate through all current elements (detected faces)
    {
        Point pt1(f.x, f.y);
        Point pt2(f.x + f.height, f.y + f.width);
        rectangle(frame, pt1, pt2, Scalar(0, 255, 0), 2, 8, 0);
    }

    namedWindow("Display window", WINDOW_AUTOSIZE); // Create a window for display.
    imshow("Display window", frame); // Show our image inside it.
    waitKey(0);
}

if (!face_cascade.load("haarcascades/haarcascade_frontalface_alt.xml")) {
    printf("--(!)Error loading\n");
    return (-1);
}

Mat image = imread("donov.jpg", CV_LOAD_IMAGE_COLOR); // Read the file
detectAndDisplay(image);
```

# Упражнение 2

- Нужно научиться детектировать лица на видео

<http://alereimondo.no-ip.org/OpenCV/34>



# Что можно сделать

- Есть аналогичные каскады Хаара для кисти руки. Можно написать какую-нибудь игру, которой управлять без клавиатуры...