Eye-tracking mouse control project

Mikhail Chekanov
Optimization Class Project. MIPT

Introduction

In this work a new approach in eye-tracking mouse control is presented. It combines Viola-Jones face detection algorithm, variance projection function and SIFT keypoints detector.

Viola-Jones face detection algorithm

Algorithm for robust face detection is described in Paul Viola's and Michael J Jones' work[3].

Method extracts various features from an image. Each feature is calculated by adding and subtracting gray-scaled sub-window image pixels' values according to different templates(*Haar-like features*, see Fig. 1)

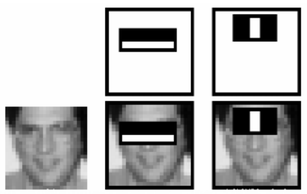

Figure 1: Example of Haar-like features

Importance of each feature is determined with *AdaBoost* algorithm. To accelerate face-recognition process features are divided in a set of groups. In the first group there are few but most reliable features. Each next group consist of a larger number of less important features than previous one. Therefore, sub-window is analyzed with classifier processing the first group's features, if it doesn't passes this classification it is discarded, otherwise it is sent for analysis with second group and so on. If sub-window passes all classifications it is considered to be a face-like region. This approach is called *Cascade of Classifiers*

Viola-Jones algorithm could be used in detection regions of eyes, nose, mouth etc. We use it to determine a small region of eye on image. After that we are looking for specific eye points in this region.

Variance projection function

Guo-Can Feng and Pong Chi Yuen's work [1] describes a method for detecting eye's key points with variance projection function(VPF).
Let $I(x, y)$ is an image. For each row and column we could calculate mean value:

$$m_x(y) = \frac{1}{m}\sum_{x=1}^{m} I(x, y)$$

$$m_y(x) = \frac{1}{n}\sum_{y=1}^{n} I(x, y)$$

Then variance projection functions are defined as:

$$\sigma_x(y) = \frac{1}{m}\sum_{x=1}^{m}(I(x, y) - m_x(y))^2$$

$$\sigma_y(x) = \frac{1}{n}\sum_{y=1}^{n}(I(x, y) - m_y(x))^2$$

The idea of eye's key points' detection with this function is that VPF has a significant increment in specific points of eye (see Fig. 2) Therefore, these points could be detected with maximums of VPF's derivatives: $\frac{\partial\sigma_x}{\partial y}, \frac{\partial\sigma_y}{\partial x}$.

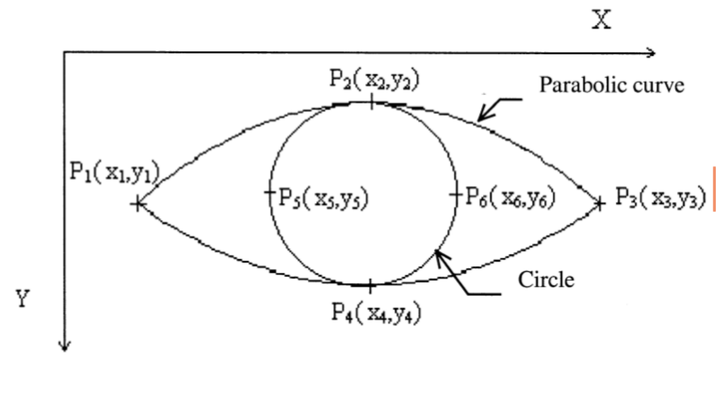

Figure 2: Eye's key points

SIFT keypoints detector

Originally SIFT(Scale Invariant Feature Transform) was described in [2].

SIFT descriptor is used for detecting corners and blobs on image. It looks for extreme of image's Laplacian of Gaussian points (in practice this value is approximated with difference of two Gaussians (DoG) with various sigmas). It also determines the size of blob measuring DoG at different scales(see Fig. 3). Properly configured descriptor is guaranteed to find key point – center of pupil, as pupil is a significant blob in eye region's image. Moreover, pupil is the darkest point of eye's image. Thus we could distinguish pupil's key point from others comparing their pixels' value.

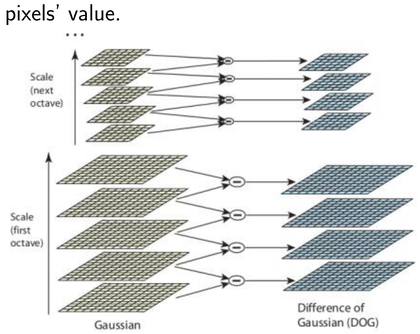

Figure 3: SIFT's calculation of image's DoG

Algorithm

The code is presented here.

```
LEN = const;
x_array = [current x-position of mouse] * LEN;
y_array = [current y-position of mouse] * LEN;
coef_column = [const array];
while True do
    I = input image;
    face = the left of found with Haar Cascade face-like regions in I ;
    leftEye = the left of found with Haar Cascade eye-like regions in face;
    kps = SIFTDetectKeypoints(leftEye);
    threshold = MAX_VALUE;
    res_kp = None ;
    for kp in kps do
        mean = mean value of I1 region of kp;
        if mean < threshold then
            threshold = mean;
            res_kp = kp ;
        end
    end
    Calculate m_x, m_y, σ_x, σ_y;
    Calculate dσ_x, dσ_y;
    left_border = position of the left big enough maximum for |dσ_y|;
    right_border = position of the right big enough maximum for
     |dσ_y|;
    bottom_border = position of the lowest big enough maximum
     for |dσ_x|;
    top_border = position of the highest big enough maximum for
     |dσ_x|;
    new_x = x-position of mouse + x-position of res_kp -(left_border
     + right_border) / 2;
    new_y = y-position of mouse + y-position of res_kp -(top_border
     + bottom_border) / 2;
    append (new_x, x_array) * coef_column to x_array;
    remove x_array[0];
    append (new_y, y_array) * coef_column to y_array;
    remove y_array[0];
    move mouse to (x_array[-1], y_array[-1]);
end
```

Algorithm 1: Mouse-control algorithm

Results

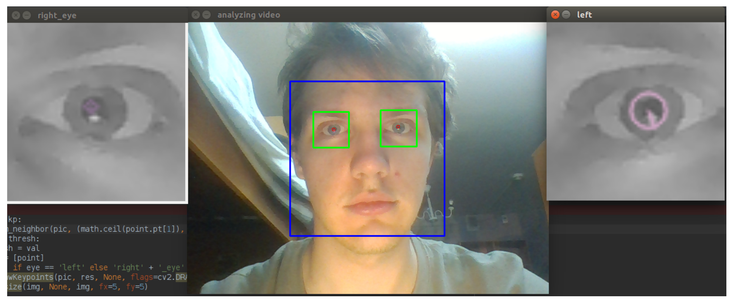

Figure 4: Face, eyes and pupil detection

At Fig. 4 results of detection face-like region, eye-like regions and center of pupil is presented. On gray-scaled enlarged eye images pupil's keypoint detected by SIFT detector is presented. The size of the circle represent determined size of blob.

References

[1] Guo-Can Feng and Pong Chi Yuen. Variance projection function and its application to eye detection for human face recognition. *Pattern Recognition Letters*, 19(9):899–906, 1998.

[2] David G Lowe et al. Object recognition from local scale-invariant features. In *iccv*, volume 99, pages 1150–1157, 1999.

[3] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.