Eye-blink Detection Using Gradient Orientations

Tomáš DRUTAROVSKÝ*
Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xdrutarovsky@is.stuba.sk

Abstract. State-of-the-art algorithms offer a real-time human face and eye detection which can be used to detect eye-blink. In this paper we present a feature descriptor. Gradient orientations and magnitudes are used to build the feature descriptor. We use the difference between samples of open and closed eyes. Gradient descriptors are further used to train the SVM. SVM can predict the open or closed eye state from the given input data. Due to the knowledge of the state of the eye (open or closed), eye-blink frequency and duration can be computed. These parameters are used to establish the level of sleepiness what can be used to prevent the driver from incoming microsleep.

1 Introduction

Today’s technologies offer us the opportunity to capture video of a driver in the car. Information about eye openness can contribute to the estimation of blink frequency and duration. These two values are considered to be the main signs of sleepiness [10]. Further analysis of these parameters can contribute in microsleep prevention.

Webcams and smartphones are available and provide a real-time image acquiring with sufficient frame resolution. World of computer vision knows algorithms for human’s face and eye detection and tracking what can be used for real-time eye sample processing. This leads to ideas of blink detection and eye state determination. Several works are devoted to the distinction between open and closed eye, but all of them have gaps in the robustness of the solution. That is because of different environment conditions, illuminations changes and various eye appearances.

Our goal is to construct a gradient orientation descriptor, which could describe open and closed eyes differently so it could be used to train the Support Vector Machine (SVM) [2]. We propose a method of gradient information calculation from the frame and processing them so the gradient features are more distinguishable. Knowing the difference between open and closed eye provides several advantages. We can use the information to detect eye-blinks or percentage of eye closure.

* Bachelor study programme in field: Informatics
  Supervisor: Ing. Andrej Fogelton, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava
2 Related work

The automatic drowsy driver monitoring and accident prevention system was presented in [3]. The system analyzes pupils using Horizontal Symmetry Calculation (HSC). The system receives input colored frames from a video camera and measures the eye-blink duration of a driver constantly. After face detection the neural network-based detector is used for the precise eye pupils position. The head rotation angle is calculated using the vertical position of both pupils. If eye detection in the next frame fails, the angle helps to determine the right face and eye position. Detected pupils are analyzed using HSC to determine whether the eyes are Open or Closed. HSC uses the fact that folded closed eye samples have more difference pixels than open eye samples. Mentioned algorithm was tested on ZJU database and it achieved 94.8% accuracy for eye-blink detection. This method is dependent on samples’ quality and precise eye alignment and cropping.

Work presented in [7] solves the problem of blink detection using the frame pixel difference. Authors estimate blink when the pixel intensity difference of consecutive frames is higher than the preset threshold. Method also uses thresholds to consider a non-uniform change of the illumination for each eye in consecutive frames and also an exclusion of voluntary blinks from the further analysis. Similar method was proposed in [6] where authors localize eye positions according to the significant change of the intensity in the consecutive frames. After successful eye localization, the system learns how an open eye appears so it can be used in the next phase of blink detection. Eye closure is estimated due to correlation score between current eye template and learned template examples. Mentioned algorithms require video samples with no bigger face moves and uses relatively many thresholds.

In the paper [5] authors proposed a vision-based drowsiness detector which can detect a driver in a realistic driving simulator. This detector is based on the infrared stereo camera. The detector constantly tracks eyes and estimates the percentage of closure also known as PERCLOS. For the best estimation, PERCLOS values were compared to results of several psychological experiments. First, face and eyes are detected using Viola – Jones detector [12] and detection failures are then corrected using Kalman filter. Subsequently, the sequence of filters is used to improve the frame quality so the PERCLOS can be estimated more accurately. PERCLOS is calculated from the ratio between the iris height in the frame and the nominal value which is assigned during a ten-second calibration at the start of tracking. This detector reaches high recall (90.68%) and low false positive rates using their own database consisting of 25 hours of driving. However, the system uses an infrared stereo camera what makes it an expensive solution with high hardware requirements.

Another method of eye state determination is measurement of pixel amount in eye regions. Authors in [4] presented a method of eye-blink detection using intensity horizontal projection (IHP). The method uses the fact that iris has lower IHP value than other regions around the eye. This means that closing of the eyelid causes noticeable changes in the histogram of IVP values. On the other hand, algorithm presented in [8] measures eye-blink using intensity vertical projection (IVP). After applying the median filter, the IVP of eye regions without eyebrows is measured. It is considered that the ratio of maximal IVP to minimal IVP for the open eye is higher than for the closed eye. Open eye has also higher maximal IVP value than closed eye. Although these methods seems obvious, they work accurately only for specific eye types. Therefore we can not consider them robust and reliable enough.

3 Gradient Orientation Descriptor

One of the most significant characteristics for the human discrimination of eye states is an ability to recognize shape. Shape is often described by gradient orientations. We want to use that fact and build a feature descriptor which could help the computer to discriminate different eye states using the gradient orientations. We propose the gradient calculation and orientation sorting for each eye sample. We use weight map to adjust weights of values which are added to the orientation bins.
3.1 Gradient calculation and sorting

First step of descriptor construction is the gradient calculation from the image. In this paper we do not discuss face and eye detection and tracking. We assume that the input eye sample contains nothing but eye aligned in the center of the sample. In our work we used Viola – Jones detector to obtain eye rectangles. After that, we align rectangles according to the pupils using the gradient pupil locator. In the case of the closed eye sample, we assume that eyelashes or eyelids' link is located in the center of the sample.

Our method uses Sobel operator for the edge detection, because it is partly rotation invariant. Sobel gives better results in computing diagonal edges than the edge detector which uses $[-1, 0, 1]$ kernel and image preprocessed with Gaussian filter.

Input image is decomposed into derivatives – horizontal image gradients $dx$ and vertical image gradients $dy$. Using these two images we can compute gradient magnitude $m$ for each pixel as maximum of absolute values from both images (Equation 1). We can also compute gradient orientation $\alpha$ using the tangent function (Equation 2). This approach was inspired by the work presented in [1].

$$m_{x,y} = \max(abs(dx_{x,y}), abs(dy_{x,y})) \quad (1)$$

$$\alpha_{x,y} = \tan(dx_{x,y}, dy_{x,y}) \quad (2)$$

Another important operation is constructing of the weight map. This map is used to control the weight of the image pixels so the pixels in the center of the image have higher weights than pixels at the image border. Each point in the eye sample has own weight $w_{x,y} = e^{n_{x,y}}$. In our algorithm, $n_{x,y}$ is linear interpolation between 0 for border pixels and 12 for center pixels, according to the pixels position in the weight map. Exponent values were chosen due to the empirical test results. This approach is inspired by the FREAK descriptor [11] which used knowledge about human’s retina and vision sharpness.

After these operations, we have two essential values for each image pixel so we can sort gradient orientations into 360 bins. Each gradient orientation is dispersed from $-10$ to $10$ degrees and classified into bins to avoid errors in orientation calculation. Appropriate bin is increased with value from weighted map. However, we consider only gradients with magnitude higher than 10 (we consider magnitude values from 0 to 255) to avoid non-significant gradients.

3.2 Orientations function

Following step of the gradient descriptor construction is the processing of the 360-binned orientation function. Input orientation function has a lot of local minima and maxima, therefore we smooth the function. Each bin is smoothed using average value from the four closest bins. We smooth the function until it has 4 extremes exactly – two minima and two maxima. As we consider aligned eye samples, these extremes are often around 0, 90, 180, 270 degrees. We use the fact that open eyes (Figure 1) have maximum with highest value around 90 degrees, but closed eyes have higher maximum around 270 degrees bin (Figure 2). That is because an open eye has more gradients oriented vertically down near the eye center. Moreover, open eye sample has bigger disperse of the maximum peak what is caused by higher amount of horizontal orientations around the iris.

Smoothed function is then aligned to the first local maximum. Rotation invariance can be guaranteed by shifting bin values according to the rotation angle. This angle can be computed as the angle between line connecting both irises and horizontal line.

Orientations function with 4 extremes is used to construct the final gradient descriptor. Our proposed descriptor consists of 8 numbers – four pairs. Each pair represents one extreme or peak and consists of two numbers $rate_{x,y}$ and $distance_{x,y}$. $rate_{x,y}$ represents the rate of the peak height among other peaks. This means that summed size of all four peaks gives 1. $distance_{x,y}$ represents
the number of bins located between considered peak and the closest right peak. Descriptor $\hat{O}$ represents descriptor of averaged vector from 200 open eye samples (Equation 3).

\[
\hat{O} = [0.59, 0.30, 0.10, 0.18, 0.24, 0.22, 0.07, 0.30]
\]  

(3)

4 Evaluation

Both types of eye states (open and closed) are used to train the SVM classifier. In this paper we construct the SVM from 200 open eye and 200 closed eye samples. Trained SVM is used for further state prediction. Testing was performed on other 200 open and 200 closed eyes and results are summarized in the Table 1.

Evaluation of the descriptors’ accuracy was performed on our own eye dataset, which consists of the six individuals sitting in front of the camera. Eye samples were acquired using Viola – Jones eye detector and gradient pupil locator at the average resolution of $24 \times 24$ pixels.

Our method based on the gradient orientations and image weighting achieved accuracy of 87.8%. We have compared our final gradient descriptor to two other methods.

First of the compared methods is the gradient descriptor without weighting. Absence of weighting supports the fact that the strong links like eyelashes or eyebrows can cause loss of accuracy. This eye features have strong gradients which are not ignored as in the weighting method. Therefore the method achieved accuracy of 71.0%.

Another compared method is SIFT descriptor [9]. Our implementation of the descriptor has one point of interest located in the center of the sample with interest area stretched on the whole sample.
Table 1. Results of all tested methods. TP represents true positive rate (correctly identified closed eye) and TN represents true negative rate (correctly identified open eye).

<table>
<thead>
<tr>
<th>Method</th>
<th>TP</th>
<th>TN</th>
<th>overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>No weighted</td>
<td>67.5%</td>
<td>74.5%</td>
<td>71.0%</td>
</tr>
<tr>
<td>Weighted</td>
<td>93.5%</td>
<td>82.0%</td>
<td>87.8%</td>
</tr>
<tr>
<td>Sift</td>
<td>95.0%</td>
<td>93.5%</td>
<td>94.3%</td>
</tr>
</tbody>
</table>

Final descriptor has 128 dimensions – quantized gradient directions, as usual. This descriptor achieved accuracy of 94.3%.

5 Conclusion

In this paper we proposed the method of constructing feature descriptor based on gradient orientations. Our solution involves gradient orientations calculation and sorting, weighting and descriptor building. Descriptors of open and closed eye samples were used to train the SVM, which is used to predict results. Proposed method method achieves accuracy of 87.8%.

Although we do not achieve the accuracy of SIFT descriptor, our descriptor consists of 8 numbers only which might result in the potential performance increase. Our method depends on the eye alignment what can affect the gradient orientations and subsequently the feature descriptor. We consider the result encouraging and see the potential of the method in the further enhancement.

We want to focus on the location of significant eye regions and use them to adjust the weighting. Moreover, we want to use the improved descriptor to implement eye-blink detector. Knowing the information about the current eye closure can lead to the estimation of the blink duration and blink frequency. Our future aim is to build a detector with fair trade-off and acceptable solution robustness.

References


