REAL-TIME THREE-STAGE EYE FEATURE EXTRACTION

Chien-Nan Chou (周建男), Wei-Ting Peng (彭維廷), Yi-Ping Hung (洪一平)

Dept. of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan
Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan
E-mail: r96922021@ntu.edu.tw

ABSTRACT

The diversification of the development in human-computer interaction provides users more and more interactive experience which is totally different than usual. As one of the sense organs that serve as a proxy for human attention and intention, we can give users better interaction closer to their need by observing the change of the eyes. This paper proposes a low-cost system, using a non-infrared off-the-shelf webcam, which can extract and track the eye features on the face. For getting a real-time and robust result, we propose a three-stage method based on the low variety in the geometry of faces so that these eye features just need to be searched in the small eye rectangle derived from that. Also, our system works without IR lights which are easily affected by environmental ambient light, and it ensures getting high accuracy outdoors as well as indoors and being easy to set up in real life.

1. INTRODUCTION

The eyes are a sense organ which plays an important role in our daily life. There is lots of information coming from the eyes as a proxy for human attention and intention. By looking at the eyes, we get some rich cues, such as how you feel or what you see, which could be valuable for use in psychophysics and human interaction. As one of the most salient features at human face, automatic eye feature extraction has a wide range of usage in the computer science area. Especially in human-computer interaction (HCI), the use of eye feature extraction has significant potential to enhance the quality of interface. For example, by tracking the eye corners on the face we can get the precise position of the head of the user and use such the information to adapt some aspect of the display, or we can estimate the direction of the eye gaze by extracting the center of the iris to substitute for the traditional mouse input.

Nowadays, there have been many commercial eye tracking systems and literatures which frequently rely on the infrared (IR) light for a robust environment and an accurate result. However, the use of IR light introduces some drawbacks [1, 2]: first, most people are not able to get a successful calibration result due to false reflections from glasses, interference with the ambient light, or occlusion by eyelids or eyelashes. It would get worse in some situations, such as the users wearing glasses or the ambient light at an outdoor environment. Second, users may feel uncomfortable and their eyes tend to dry out as the result of long-time direct exposure to the IR light even if users’ eyes have the invisibility of the spectrum of infrared light. Third, the use of IR source and IR camera increases the cost such that the price of current IR-based commercial systems is expensive. For the purpose of creating a stable, comfortable, and low-cost eye tracking system, the use of non-IR off-the-shelf products as elements is the best choice.

In this paper, we propose an approach based on human visual characteristics, using the geometry of faces, which can extract and track those eye features – the two eye corners and the iris circle with properties of scale, translation, and small range rotation invariance and real-time, accurate, and robust result. We just need to use an off-the-shelf low-cost webcam without IR-light to obtain the frontal face of the user. Compared with the existing IR-based eye tracking methods, the proposed method is simple to setup and can work both indoor and outdoor. Most of all, it does decrease the cost of building such a system in both software and hardware.

The rest of the paper is organized as follows. In chapter 2, we provide a detailed review on the related works of eye feature extracting approaches. Chapter 3 describes our method in detail to extract eye features including the eye corners and the iris circle. Then, we test our method on the BioID database and compare the measure results with state of the art methods in chapter 4, followed by the conclusions in chapter 5.

2. RELATED WORKS

Eye localization/tracking techniques have been developed for decades, and there are three broad categories of these methodologies according to the distinct difference of hardware use or measurement [3]:

(1) Electro-OculoGraphy (EOG), which relies on
recording the electric potential differences of the skin surrounding the ocular cavity; (2) scleral contact lens/search coil, which involves attaching a mechanical or optical reference object mounted on a contact lens for measuring the sensitive movement of the eye; (3) Photo-OculoGraphy (POG) or Video-OculoGraphy (VOG), which relies on the measurement of visible features of the eye by using image processing techniques, e.g., the pupil, the limbus, or a corneal reflection of a closely directed light source. Considering the accuracy, the first two techniques are good choices. Unfortunately, the primary obstacle of using these techniques is either too intrusive or too expensive to be applied extensively in daily life. VOG or POG is the least invasive technique which would be easily accepted for routine use. They can be subdivided into two modalities, active infrared-illumination approaches and passive appearance-based approaches, based on the use of active light illumination or not.

3. EYE TRACKING SYSTEM

Our eye tracking system is based on a video-based non-intrusive approach. The proposed system does not require any dedicated and expensive hardware. The only hardware is a low-cost off-the-shelf webcam while using such a technique. It outputs the information of the eye features such as eye corners or iris centers, which can be applied into many applications. In this chapter we will introduce the architecture of our feature extraction system and describe the principle of each component.

3.1 System Architecture

In our system, the input image or video frame for the proposed method contains a user’s face with the eyes not occluded so as to extract them. The user is free to move around the field of the camera view. In order to quickly and accurately extract eye features, the proposed system can be divided into three stages: face detection, eye detection, and feature extraction. The fine eye region is detected at the second stage after the face region is marked in face detection, and then eye features, two eye corners and an iris circle, are extracted from this region at the final stage. The searching space at each stage is bound to the previous stage due to the step-by-step property of our system, and therefore both the false alarm and the processing time can be reduced at the same time. Fig. 1 shows the overview of our system architecture. The details of each component will be described in the following sections.

3.2 Face and Eye Detection

Face detection is the first step of the system framework and determine the later two stages to run or not. It is an indispensible step to eliminate the interference from the background while allowing the user to move free. Current face detection techniques have been developed for a long time and can recognize faces efficiently and accurately. In our work, the face position is estimated by using the boosted cascade face detector proposed by Viola and Jones [4].

Fig. 2: The searching region for the eye detection.

Applying human visual property in the recognition of faces, people can identify a human face from a very far distance, even though the details are vague. That means the symmetric characteristic is enough to be recognized. Human face is made up of eyes, nose, mouth, chin, etc. There are differences in shape, size and structure of those organs, but they are always on the right places [5, 6]. The eyes are beneath the eyebrows while they are almost symmetrical to the nose. With the above face detection result, we demarcate the possible regions for searching the fine eye region. It avoids the false eye detection on the nose holes or mouth. Fig. 2 shows the searching region for the eye detection, which is based on the facial geometry. Similarly, we adopt the boosted cascade eye detector proposed by Castrillón et al. [7]. The fine eye location is estimated quickly and accurately due to the boundary of the searching region.

3.3 Eye Feature Extraction

Fig. 3: Eye feature extraction.
Eye feature extraction is the most important step in our system. It processes the fine eye region derived from the earlier eye detection and extracts the eye features. According to the difference in the classification of the features, this section can be divided into two subsections: eye corner location and iris circle location. Fig. 3 shows the details of the two subsections in the feature extraction. Eye corner location is a composite of detection and tracking. While detection step locates the current eye corners, tracking step takes into account the temporal coherence contained in a video stream in order to build a more robust detector. Similarly, iris circle location is composed of two steps, initialization and refinement. The iris center is roughly estimated in the first step, and then the iris circle is refined to locate the fine iris center and radius. For more details, we will describe the two subsections in feature extraction respectively.

3.3.1 Eye Corner Location
The eye corner is the anchor point which determines the location and orientation of the eye or the face. The shape and orientation of the eye corner is stable across different people and facial expressions. For the purpose of utilizing the pattern of the eye corner, Zhu and Yang [21] designed a simple preset pair of filters to detect eye corners. If the eye region is always darker than the skin region around the eye corner, the method based on the intensity pattern can successfully find the corner. In fact, they do not consider the existence of the white sclera so that a false detection occurs as the white sclera is obvious. Similarly, Sirohey and Rosenfeld [8] designed the wedge filters based on the color information around the corners. They consider that the color distribution of the sclera region can be distinguished from the flesh tone in the face. The wedge filters are made up of three wedge-shaped parts as shown in Fig. 4. The largest wedge looks for the flesh tone and the smallest wedge looks for the sclera tone. There is a “don’t care” region between the sclera wedge and the flesh wedge allowing for any possible squinting of the eye. A corner, either right or left, is detected if the average value of the pixels in each wedge satisfies its comparative color tone. However, the color distributions of the flesh tone and the sclera tone are overlapped and easily influenced by the light condition. This searching result will be doubtful.

![Fig. 4: The wedge filter.](Image)

In this paper, we propose an eye corner template using Gabor wavelets, which utilizes the special pattern of the eye corner on the Gabor feature space. The Gabor wavelet proposed by Lades et al. [9] is a powerful tool in image feature extraction and can be defined as:

\[
\psi_{\mu,\nu}(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left( -\frac{x^2}{2\sigma^2} \right) \exp \left( \frac{x}{\sigma} \right)
\]

(1)

where \(\mu\) and \(\nu\) define the orientation and the scale of the Gabor kernel, and the wave vector \(k_{\mu,\nu}\) is defined as:

\[
k_{\mu,\nu} = k_0 e^{i \phi} \quad \text{with} \quad k_0 = \frac{\mu}{\sigma} \quad \text{and} \quad \phi = \frac{\pi \nu}{8}
\]

(2)

where \(k_{\text{max}}\) is the maximum frequency and \(f\) is the spacing factor between kernels in the frequency domain.

![Fig. 5: The Gabor feature space of an eye image.](Image)

At an appropriate scale and orientation, filtering with a single Gabor wavelet gives a prominent response to the corresponding pattern of intensity variations in the image. Ville et al. [10] proposed a simple Gabor feature space for invariant object recognition, which uses Gabor wavelets with different coefficients to describe a feature vector. Fig. 5 shows the Gabor feature space of an eye image at 5 different scales and 8 orientations. The eye corner, the intersection of the two eyelid curves and the end point of eyelid curves, is one of the most salient features. We found that the structure is highlighted in the Gabor feature space at the first two rows. Encouraged by these results, we utilize the Gabor feature space to design a template for detecting the eye corner. The Gabor feature space \(G\) of an eye image at two scales and eight orientations can be defined as:

\[
G = \begin{pmatrix}
g_{1,1} & g_{1,2} & \cdots & g_{1,8} \\
g_{2,1} & g_{2,2} & \cdots & g_{2,8}
\end{pmatrix}
\]

(3)

where \(g_{i,j}\) means the response image obtained from convolving the eye image with the Gabor wavelet at the \(i\)th scale and the \(j\)th orientation. For illumination invariance, \(G\) can be normalized as:

\[
G' = \frac{G}{\sqrt{\sum_{i,j} |g_{i,j}|^2}} = \begin{pmatrix}
g'_{1,1} & g'_{1,2} & \cdots & g'_{1,8} \\
g'_{2,1} & g'_{2,2} & \cdots & g'_{2,8}
\end{pmatrix}
\]

(4)

and then let \(f\) be the mean image of these response images as:

\[
f = \frac{1}{n} \sum_{i,j} g'_{i,j}
\]

(5)

where \(n = 16, i = 1,2\) and \(j = 1,2,\ldots,8\). Fig. 6 shows some Gabor representations \(f\) in different eye images. We found that the eye corner structure is still obvious when eye blinked or half closed, so we wish to design an eye corner template based on this special corner.
structure like Zhu’s method [21] or Sirohey’s method [8]. However, instead of manually designing a heuristic template, we learn the structure by selecting some corner images in the Gabor representation and then train the reasonable template.

Fig. 6: Gabor representation of eye images.

Fig. 7: Samples of inner eye corner images.

To construct an inner eye corner template, we select 40 inner eye corner images $t$ cut from certain Gabor representations $f$ such as Fig. 6. The size of these images is 11x11, and the center of each image is corresponding to the manually located eye corner. Some inner eye corner images are shown in Fig. 7. The final inner eye corner template $T_R$ is constructed by calculating the average of $t$.

$$T_R = \frac{1}{N} \sum_{i=1}^{N} t_i \quad (6)$$

where $N = 40$. Note that the number $N$ is based on the training data set. In our experiment, 40 images are sufficient to obtain the template with the obvious corner structure. $T_R$ is shown in Fig. 8(a). Similarly, the outer eye corner template $T_T$ is constructed by the same way and shown in Fig. 8(b).

Fig. 8: The eye corner templates. (a) The inner corner (b) The outer corner

In order to detect the eye corner in an eye region, we first obtain the Gabor representation of the eye region using the same coefficients as the Gabor feature space $G$ and being normalized by Equation (4) and (5), and then a template matching method is used to find the location with the best response. Unlike a traditional matching approach using the squared difference or the cross correlation to measure the similarity between the template and the corresponding subimage in the search window, we use the normalized correlation coefficient method proposed by Betke et al. [11], which is invariant to changes in image intensity. This measure can be defined as:

$$r(u,v) = \frac{\sum_{i,j} [F(x+i,y+j)-T_{u,v}] F(x,y)-T}{\sum_{i,j} [F(x+i,y+j)-T] \sum_{i,j} [F(x,y)-T]} \quad (7)$$

where $F$ is the image, $T$ is the template, $T$ is the mean value of the template, and $T_{u,v}$ is the mean value of $F$ in the corresponding region under the template. Note that because the image and the template are normalized to unit length, the intensity of the image will not influence the measure of the similarity. In order to reduce the computational cost, the matching operation could be calculated on part of the eye region due to the structure of an eye. Fig. 9 shows the eye corner detection results.

Fig. 9: The eye corner detection. (a) The eye image. (b) The Gabor representation of (a). (c) The result of template matching.

We found that our detection method can successfully locate eye corners in images. However, applying it to video images thirty times per second will necessarily result in unstable locations. In order to build a more robust detector, a tracking method is used to take into account the temporal coherence contained in a video stream. In our system, the eye corners detected in the current frame are tracked by a modified version of the Lucas-Kanade tracking algorithm [12] in the next frame. We assume that the intensity values around the eye corner do not change but merely shift from one position to another. Consider an intensity template $I_i(x)$ over an $n \times n$ region $R$ in the reference image at time $t$. We wish to find the translation $d$ of the region in the following frame $I_i(x+d)$ at time $t$ by minimizing a cost function $E$ defined as:

$$E = \sum_{x \in R} [I_i(x+d)-I_i(x)]^2 \quad (8)$$

and the minimization for finding the translation $d$ can be calculated in iterations:

$$d_{i+1} = d_i + \left[ \sum_{x \in R} \frac{\partial E}{\partial x} \right] \left[ \sum_{x \in R} \frac{\partial E}{\partial x} \right]^{-1} \quad (9)$$

where $d_0$, the initial estimate, can be taken as zero if only small displacements are involved.

Consecutive frames of a video stream may contain large motions such as sudden head movements, which are possible to lose tracking. In order to track large motions without losing sub-pixel accuracy, a pyramid method with reduced resolution is used. Each image is decomposed into 4 levels from the original finest
resolution image to the coarsest resolution image. In our implementation, a 10x10 search window is used for all levels. Rapid and large displacement can be tracked robustly while maintaining sensitivity to sub-pixel facial motion.

### 3.3.2 Iris Circle Location

The Iris is a membrane in the eye, responsible for controlling the amount of light reaching the retina. In this section, the Iris circle means the boundary between the iris and the white sclera. Despite the difference in the color of the iris, the boundary is still obvious because the iris is always darker than the surrounding sclera. By utilizing the property above, the iris circle can be located in two steps, the initialization and refinement. They are described in detail as follows.

From the above method, the iris center is approximately estimated, but it may not be the precise location due to the cover with the eyelids. In order to refine the iris center and estimate the iris radius, we further find the circular shape of the iris. First, the edge map of the eye image eliminating the highlight is obtained by using the Canny edge detection. For finding the sample points of the iris boundary in the edge map, the algorithm begins at the approximate iris center as a starting point, and then the intersections are obtained along rays extending radially away from the starting point. According to the structure of an eye [23], we realize the diameter of the iris is always smaller than the length between the two corners, so the length of each ray are limited to half the length between the two corners. For each ray we detect at most three sample points before halting. An example set of sample points is shown in Fig. 11(a). We restrict the direction of the rays because the iris is likely to be occluded by the eyelids and eyelashes. The range of angles is an adjustable to accommodate different users, but is initially taken to include -45° to 45° and 135° to 225°. One ray per 5 degrees is traced resulting at most 108 candidate sample points. However, in a real situation there are large outliers due to eye blinks. In order to eliminate these outliers, an upper eyelid point is obtained by tracing a vertical ray from the starting point and finding an intersection, and then those sample points above the two links between the upper eyelid point and the two eye corners respectively are excluded (Fig. 11(b)).

The candidate sample points may still contain outliers. A circle is fitted to the candidate sample points using the Random Sample Consensus (RANSAC) paradigm proposed by Fischler and Bolles [13]. RANSAC is an effective model estimation method as there is a large but unknown percentage of outliers in samples points. Unlike a least-squares fitting approach, it reduces the influence on the accuracy caused by these outliers. We introduce two restrictions on the RANSAC fitting process to increase the robustness of the inlier selection process. First, only candidate circles that the starting point is included within the covered areas are considered. Second, based on the structure of an eye [23], the ratio of the iris diameter to the length of the two eye corners is about 1:3, so only candidate circles with a reasonable ratio are considered. The inliers and outliers are shown as green and red crosses respectively in Fig. 11(c). The final circle fit is shown in Fig. 11(d). Figure 3.12 shows the results on different subjects. Eye features are successfully extracted from the eye of each user.
3.3.3 Blink Detection

While the eyes are blinked, the iris center is occluded by the upper eyelid. In this situation, iris circle location is useless and meaningless, so the blink detection is used to estimate what the status of the eyes is. In addition, it is also considered an important message in HCI and psychology. For example, Chau and Betke [14] present a HCI system designed for people with severe disabilities. People that are severely paralyzed or afflicted with diseases such as ALS (Lou Gehrig’s disease) or multiple sclerosis are unable to move or control any parts of their bodies except for their eyes. The system detects the user’s eye blinks and uses them to provide input to the computer in the form of a mouse click.

To detect the user’s eye blinks in our current system, a fast but inexpensive way is to reuse the obtained feature information. As stated before, we consider the eyes blinked when the iris center is occluded by the upper eyelid. However, one blink means an action that a user opens his/her eyes after closing. Whether the iris center is occluded or not is just used to determine the status of the eye at each frame. We can define the measure as follows.

Let \( a(i) \) represent the status of the eye at the \( i \) frame.

\[
\alpha(i) = \begin{cases} 
\text{Open} & \text{if } H_i \geq H_s, \\
\text{Closed} & \text{otherwise}, 
\end{cases}
\]  
(11)

where \( H_i \) and \( H_s \) are the values of the height from the upper boundary of the eye region to the iris center and the upper eyelid point respectively. Fig. 13 shows the two status of the eye. As the eye changes from an open status to a closed status, we ensure that the blink occurs. The duration of the blink is counted from the closed moment to the reopened moment.

4.1 Accuracy of Iris Center Location

The BioID database consists of 1521 grayscale images of 23 different subjects with a resolution of 384x286 pixels. These facial images are taken during several sessions at different places, i.e. this dataset features uncontrolled illumination and background variations. In addition, the positions of the subjects change both in scale and pose. In some instances the eyes are closed, or turned away from the camera. In many samples the subjects wear glasses where the eyes are hidden by the frame of spectacles or strong highlights on the glasses. Due to these conditions, the BioID database is usually considered a more challenging dataset. The ground truth of the left and right iris centers is provided with the dataset.

To validate the accuracy of our iris center location method, the normalized error, indicating the error obtained by the worse eye estimation, is used to measure the error rate between the estimated iris center locations and the ground truth. This measure was proposed by Jesorsky et al. [16] and is defined as:

\[
e = \frac{\max(d_{\text{left}}, d_{\text{right}})}{\omega},
\]  
(12)

where \( d_{\text{left}} \) and \( d_{\text{right}} \) are the Euclidean distance between the located eyes and the relative ones in the ground truth, and \( \omega \) is the Euclidean distance between the eyes in the ground truth. Since the distance of the two inner eye corners is roughly equal to the eye length, \( \epsilon \leq 0.25 \) (a quarter of the interocular distance) roughly corresponds to the distance between the iris center and the eye corners, \( \epsilon \leq 0.1 \) corresponds to the range of the iris, and \( \epsilon \leq 0.05 \) corresponds to the range of the pupil. Besides the maximum (worse eye) normalized error, we also measure the minimum (best eye) normalized error so as to give upper and lower bounds to the accuracy, and an average between the best and worse estimation.

4. EVALUATION

So far we realize our system how to extract eye feature step by step, which is started from a face to eyes, and then from two eye corners to an iris center. The location of the iris center would have obvious variations when any preceding step gets worse result. In order to validate our system performance, we choose to measure the error of iris center location. In this section we test the proposed method on the BioID [15] face database and compare the measure results with state of the art methods in the literature which use the same database and the same accuracy measure. Furthermore, we describe the processing time, the criterion that our method is efficient enough to be applied in real-time applications.

![Fig. 12: The results on different subjects.](image)

![Fig. 13: The two status of an eye. (a) Open. (b) Closed.](image)
Table 1: Accuracy results.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (e ≤ 0.05)</th>
<th>Accuracy (e ≤ 0.10)</th>
<th>Accuracy (e ≤ 0.25)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td>80.36%</td>
<td>97.05%</td>
<td>99.65%</td>
</tr>
<tr>
<td>Valenti [24]</td>
<td>84.10%</td>
<td>90.85%</td>
<td>98.49%</td>
</tr>
<tr>
<td>Türkkan [25]</td>
<td>19.00%</td>
<td>73.68%</td>
<td>99.46%</td>
</tr>
<tr>
<td>Asteriadis [17]</td>
<td>74.00%</td>
<td>81.70%</td>
<td>97.40%</td>
</tr>
<tr>
<td>Bai [26]</td>
<td>37.00%</td>
<td>64.00%</td>
<td>96.00%</td>
</tr>
<tr>
<td>Campadelli [18]</td>
<td>62.00%</td>
<td>85.20%</td>
<td>96.10%</td>
</tr>
<tr>
<td>Hamouz [19]</td>
<td>59.00%</td>
<td>77.00%</td>
<td>93.00%</td>
</tr>
<tr>
<td>Cristinacce [20]</td>
<td>56.00%</td>
<td><strong>96.00%</strong></td>
<td>98.00%</td>
</tr>
<tr>
<td>Jesorsky [16]</td>
<td>40.00%</td>
<td>79.00%</td>
<td>91.80%</td>
</tr>
</tbody>
</table>

The graph in Fig. 14 shows the accuracy of our method for different e. It is clear that our method get nearly optimal results at the normalized error of 0.25 and 0.1 due to the step-by-step property, which means the estimated iris center is almost located within not only the region between the two corners but also the range of the iris. For further validating the accuracy, we compare our results with state of the art methods in the literature which use the BioID database and the same error measure. Table 1 shows the results compared with other methods for a normalized error smaller than 0.05, 0.1, and 0.25 respectively. It can be seen that, for a normalized error smaller than 0.25, our method achieved superior accuracy with respect to the other methods. Similarly, for iris location (e ≤ 0.1), our method also excel these methods in accuracy. In the case of more accurate eye center location (e ≤ 0.05), our method still exceeds the others in accuracy, except for the one proposed by Valenti and Gevers [24]. This can be justified by the fact that they train a classifier to find the possible one from all candidate iris centers. The result is easily influenced by the training set which is not clearly described in the paper. If the training sets and the test set are obtained from the same data set, of course the accuracy is superior. When using the basic approach not including the classification, their method only has a 77.15% accuracy rate for e ≤ 0.05 which is lower than our method.

4.2 System Efficiency

In many applications, eye feature extraction is usually used as a preprocessing step, so the processing time of our proposed method needs to be as fast as possible. Since our system is based on the step-by-step and coarse-to-fine property, it allows for a real-time implementation. On a 2.0GHz Intel Core 2 Duo, using a single core implementation, the system was able to process about 32.26 frames per second on a 640x480 image, which takes about 16ms per frame for face and eye detection and 16ms per frame for eye feature extraction, and therefore the final frame rate is only limited by the webcam’s frame rate (which is usually about 30fps).

5. CONCLUSIONS

In this paper, we presented a proposed system to infer eye features, including two eye corners and an iris circle, using the geometry of faces for largely decreasing the searching region and the characteristic of eyes for accurately locating the eye features. In the eye corner part, a Gabor-based template is designed to detect the initial position, and the Lucas-Kanade tracking algorithm is used for referring the temporal coherence to get a more robust and sub-pixel result. In the iris part, some pre-processes such as opening operator and Gaussian-shape filter are utilized to initialize the iris center, and then RANSAC circle fitting is used to refine the iris center and radius. The step-by-step framework of our method actually yields low computational cost and eliminates the false detection caused by the background, eyebrows, nose holes, mouth, etc.
An extensive evaluation was performed to validate the accuracy of the proposed approach. The comparison with the state of the art approved that our method is able to achieve high accuracy. Furthermore, our system can process about 32.23 frames per second on a 640x480 video stream, showing that it can be applied in any real-time application.

REFERENCES


