Eye tracking by template matching using an automatic codebook generation scheme

1 author:

Marcel J T Reinders
Delft University of Technology

Eye Tracking by Template Matching using an Automatic Codebook Generation Scheme

M.J.T. Reinders

Information Theory Group
Dept. of Electrical Engineering, Delft University of Technology
P.O. Box 5031, 2600 GA Delft, The Netherlands

E-mail: M.J.T.Reinders@et.tudelft.nl

Keywords - Pattern recognition, template matching, incremental learning, feature tracking, eye gesture recognition

Abstract - We present an eye tracking algorithm which is robust against variations in scale, orientation and changes of eye appearances, such as eye blinking. The locations of the eye regions in the different frames are found using template matching. The method is kept invariant for rotations and scale by exploiting temporal information and by using a codebook of eye templates the method is robust against changes in eye region appearances. The entries of the codebook are generated *automatically* during the tracking of the eye regions and eventually represent a distinct set of eye appearances. Classification into different eye gestures, like blinking or different gaze directions, seems possible using these automatically learned patterns.

1 Introduction

Vision-based gesture analysis nowadays receives world wide attention because of the possible revolutionary influence on how man can interact with machines (e.g. [Kjeldsen and Kender, 1996] and [Turk, 1996]). In particular, one is studying the analysis of facial gestures to create a more friendly (inter)face of new multimedia products. Additional interest in automatic facial gesture analysis stems from application areas like image sequence coding, criminal identification, speech aids for the deaf, and surgical medicine [Reinders, 1995].

Vision-based facial gesture analysis implies information extraction from an image sequence concerning the pose and movement of the head as well as the expressions of the face that are related to movements of the facial features such as eyes and mouth. Here, we propose a tracking algorithm which is able to locate the positions of the eye regions in subsequent frames of an image sequence which eventually can be used to find the pose and movements of the head. The proposed scheme automatically learns distinct eye appearances which seem strongly related to eye gestures such as blinking and changing gaze direction. Hence, eye gesture recognition seems to be an additional feature of the proposed tracking method.

Vision-based face recognition - and the strongly related subject of facial analysis - is being studied over the last 20 years [Chellappa et al., 1995]. Although numerous techniques have been proposed throughout the years, traditional template matching - possibly in different identities such as neural networks [Bichsel, 1991] or principal components [Turk and Pentland, 1992] - has been a strong competitor in localizing and recognizing faces or facial features. A study by Brunelli and Poggio [Brunelli and Poggio, 1993] even shows that template matching is superior to geometrical feature-based methods. These results encouraged us to localize eye regions by template matching. Here, we propose a tracking algorithm based on template matching which - after an initialization - automatically learns new eye templates which are stored in a codebook and is robust against changes in eye appearances. Rotation and scale invariance - one of the traditional drawbacks of template matching - is achieved by exploiting temporal information. The proposed method resembles work done by Yau and Duffy [Yau and Duffy, 1989] and Darrell et al. [Darrell et al., 1996] but differs in implementation (e.g. the used similarity measure) and the use of the codebook.

Section 2 discusses template matching. In section 3 the tracking algorithm is introduced and in section 4 results of the proposed scheme are presented. The paper ends with conclusions in Section 5.

2 Template Matching

The basic idea behind template matching is that one has an iconic view (the ‘template’) of the sought feature region that has a high similarity with other images of that feature. Two parameters play an important role within template matching: the iconic view or - as we shall see later - the set of iconic views, and the similarity measure. This section elaborates on different definitions of the similarity measure (Subsection 2.1), different representations of the iconic view (Subsection 2.2), and orientation and scale invariances (Subsection 2.3).

Let the template (iconic view) be denoted by T and the intensity of a specific pixel from this template by $T(u, v)$. Similarly, I denotes the recorded image and $I(i, j)$ the pixel intensity of the pixel at position (i, j). Further, let $S(i, j)$ be the similarity between the template T and the image patch I_T at position (i, j). Having defined a similarity measure the position of the sought feature in the recorded image, denoted by $(\hat{i}, \hat{j})$, is estimated by that position for which the similarity measure shows an extreme value:

$$(\hat{i}, \hat{j}) = arg \max_{i, j} S(i, j) \tag{1}$$

2.1 Similarity Measure

In this section we introduce three different definitions of the similarity measure: (*i*) the normalized cross-correlation, (*ii*) modified cross-correlation, and the (*iii*) mean square error.

Normalized Cross-Correlation

Cross-correlation as similarity measure was already used by Baron [Baron, 1981] for face recognition. To rescale the energy distributions of the template and the image, in order to match their averages and variances, a normalized version has been proposed (see also [Brunelli and Poggio, 1993]). This normalized cross-correlation measure is defined by:

$$S(i, j) = \frac{\langle T \times I_T \rangle - \langle T \rangle \langle I_T \rangle}{\sigma(T)\sigma(I_T)} \tag{2}$$

where, I_T is the patch of image I at location (i, j) that must be matched to T, $\langle\ \rangle$ is the average operator, thus

$$\langle T \rangle = \frac{1}{n} \sum_{u, v} T(u, v) \tag{3}$$

and $\langle\ \times\ \rangle$ represents the pixel-by-pixel product:

$$\langle T \times I_T \rangle = \frac{1}{n} \sum_{u, v} T(u, v) I(i + u, j + v) \tag{4}$$

σ is the standard deviation over the area being matched, e.g.

$$\sigma^2(T) = \frac{1}{n-1}\left(\sum_{u, v} (T(u, v))^2 - \langle T \rangle^2 \right) \tag{5}$$

Modified Cross-Correlation Coefficient

To reduce the influence of small changes in the shape of the feature region Brunelli and Poggio introduced the modified cross-correlation coefficient:

$$S(i, j) = \sum_{u, v} T(u, v) I(i + u + \hat{k}, j + v + \hat{l}) \tag{6}$$

Instead of taking the pixel-by-pixel product of the template and the image patch to be matched (as is the case in Eq. (2)), the intensity value of the template pixel is multiplied with the best matching pixel in a small surrounding window of the corresponding pixel in the image patch, i.e. the pixel with offset $(\hat{k}, \hat{l})$ fulfilling the condition:

$$(\hat{k}, \hat{l}) = arg \min_{k, l} |I(i + u + k, j + v + l) - T(u, v)| \tag{7}$$

This coefficient allows for small local deformations in the computation of the similarity measure. Its normalized form is similar to Eq. (2).

Mean Square Error

It is well known that cross-correlation is computationally expensive. An alternative similarity measure with a lower computational load is the mean square error defined by

$$S(i, j) = \frac{1}{n} \sum_{u, v} (T(u, v) - I(i + u, j + v))^2 \tag{8}$$

Note that in order to apply Eq. (1) this similarity measure should be negated.

2.2 Intensity Normalization

All of the introduced similarity measures are sensitive to illumination changes. Brunelli and Poggio have suggested different ways to preprocess the compared images to get rid of this confounding effect [Brunelli and Poggio, 1993]. In their experiments they compared four different intensity normalizations: (*i*) no preprocessing, i.e. the plain intensities I, (*ii*) normalization by using the ratio of the local value over the average brightness, i.e. $I_T^N = I_T / \langle I_T \rangle$ (similarly for the template image), (*iii*) intensity of the image gradient $I_T^N = D(I_T) = |\partial_x I_T| + |\partial_y I_T|$, and (*iv*) the Laplacian of the intensity $I_T^N = DD(I_T) = (\partial_{xx} I_T + \partial_{yy} I_T)$. They report that the gradient-based normalization gave the best recognition performance. In Section 4 results of experiments comparing representations based on plain intensity and gradient-based normalization are shown.

2.3 Rotation and scale invariance

Clearly, the similarity measures discussed before all suffer from size and orientation differences between the iconic view of the feature and the feature region in the recorded image. To overcome these drawbacks the template can be modified in order to better match with the recorded feature region, i.e.

$$T^m = \boldsymbol{R}\ \boldsymbol{S}\ T \tag{9}$$

where $\boldsymbol{R}$ and $\boldsymbol{S}$ represent rotation and scaling matrices, respectively. Now, to find the position of the feature region in the image, one should find the maximal similarity $S(i, j)$ over all possible rotations and scalings. In our work, however, we used temporal information to predict the scale and orientation of the feature in the image. In Section 3 we elaborate on this topic.

2.4 Region of Interest

To further reduce the computational load one can decide to make use of a region of interest, i.e. the search for maximal $S(i, j)$ (Eq. (1)) is only performed for a part of the image I. In this work we have also used regions of interest which are predicted by the available temporal information and knowledge about the physiognomy.

3 Tracking Eye Regions

Tracking the locations of eye regions throughout an image sequence has become relatively straightforward when using template matching: the iconic view of the sought feature region is matched with every frame of the image sequence. The positions for which the similarity measure peaks - in each of the frames - then identifies the positions of the sought feature region.

In this section we propose our tracking scheme which is an expansion of this relatively simple scheme. In Subsection 3.1 we discuss how to get the iconic view. Subsection 3.2 elaborates on how we achieved rotation and scale invariances as well as our use of the regions of interest.

The shape of the eye regions change drastically during the time lapse of an image sequence, for example the eyes may be blinking. Thus, when only one representative iconic view is used it is very well conceivable - because template matching is not robust against shape changes - that mismatches and consequently error accumulation occurs. In Subsection 3.3 we introduce the concept of a codebook of iconic views. In our method these iconic views are automatically learned simultaneously with the localization of the eye regions.

3.1 Initialization

To identify the eye region in the first frame of the image sequence an ‘initial’ iconic view of an eye should be available. This initial template can be a generic view of the eye region, as averaged over different eye appearances and different persons. In our work, however, we have chosen to create the initial template by manual intervention, i.e. the user should point out the center points of the left and right eye region. Based on the physiognomy the initial templates for the left and right eye are then automatically created.

Suppose the manual assigned left and right eye localizations are denoted by $(\hat{i}_0^{le}, \hat{j}_0^{le})$ and $(\hat{i}_0^{re}, \hat{j}_0^{re})$ respectively (the subscript indicates the zero-th frame number and the superscript indicates either the left or right eye). Then the initial template dimensions are given by

$$\begin{aligned} width(T_0) &= 0.75\ d_{le,re} \\ height(T_0) &= 0.6\ (0.75\ d_{le,re}) \end{aligned} \tag{10}$$

where T_0 denotes the initial template, and $d_{le,re}$ equals the euclidean distance between the left and right eyes, i.e.

$$d_{le,re} = \left\| (\hat{i}_0^{le}, \hat{j}_0^{le}) - (\hat{i}_0^{re}, \hat{j}_0^{re}) \right\| \tag{11}$$

Note that if the eyes are rotated in the initial frame (as shown in Fig. 1) then the image is rotated to align the eyes horizontally before the initial templates are created.

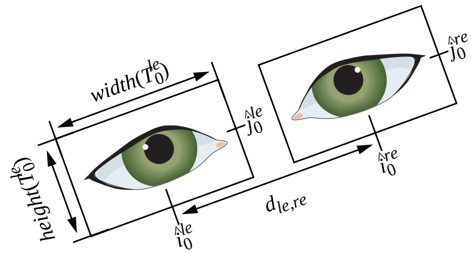


Fig. 1 Definition of initial left and right eye templates based on the manual assigned centers of the left and right eye.

3.2 Rotation and Scaling Invariance and the use of Regions of Interest

The initial template shows a normalized view of the eye region, i.e. the eye region is aligned horizontally and has a fixed scale $(width(T_0), height(T_0))$. To be robust for variations in orientation and scale of the eye regions, as a result of head movements, a rotated and scaled version of the template should be matched. Instead of maximizing over all possible orientations and scalings, we propose to use the estimated eye locations in the previous frame as estimator for the scale and orientation. The change in the scale then becomes

$$s = \frac{\left\| (\hat{i}_{t-1}^{le}, \hat{j}_{t-1}^{le}) - (\hat{i}_{t-1}^{re}, \hat{j}_{t-1}^{re}) \right\|}{\left\| (\hat{i}_0^{le}, \hat{j}_0^{le}) - (\hat{i}_0^{re}, \hat{j}_0^{re}) \right\|} = \frac{d_{le,re}^{t-1}}{d_{le,re}^{0}} \tag{12}$$

and in orientation

$$\alpha = \mathrm{atan}\left(\frac{\hat{j}_{t-1}^{re} - \hat{j}_0^{le}}{\hat{i}_{t-1}^{re} - \hat{i}_0^{le}} \right) \tag{13}$$

The template to be matched can then easily be found by applying Eq. (9).

To avoid a time consuming search over the whole image, we propose to make use of search windows (regions of interest). The use of search windows even increases the robustness of the eye tracker because - if appropriate - the possibility of other similar image patches contained in the search window decreases (in effect excluding, for example, the sometimes very similar image patches in the hair region). The search windows for the left and right eye regions are centered around the position of these regions in the previous frame and their size is again a ratio of the inter eye distance in the previous frame (currently the orientation of the search windows is not adapted).

3.3 Codebook Generation

The above describes a scheme for tracking the left and right eye regions throughout an image sequences. But, as already mentioned, only one iconic view for each eye region is available. This will cause severe problems when tracking the eyes because these regions constantly change in appearance, for example eye blinking or changing gaze direction, and furthermore the illumination is also not constant over the time lapse of an image sequence. Template matching is very sensitive for both these changes. To recuperate from these flaws a set of "*all possible distinct*" iconic views of the eye region - the '*codebook*' - should be available. Then, instead of finding the maximal similarity between *one* template and the image, one should match the image with *all* distinct iconic views in the codebook. The estimated position of the eye region $(\hat{i}, \hat{j})$ then thus becomes:

$$(\hat{i}, \hat{j}) = \underset{i,\, j}{arg} \; \underset{i,\, j,\, c}{max} \; S_c(i, j) \tag{14}$$

where $S_c(i, j)$ indicates the similarity match between template T_c from the codebook against image patch I_{T_c}. Here, we propose a scheme for automatically learning these distinct iconic views of the eye region simultaneously with the localization of the eyes throughout the image sequence.

The basis for this scheme is the fact that the similarity measure indicates how well a template and its best matching image patch resemble. If this resemblance is below a certain threshold q, than this is an indication that the eye region in the present image has a distinct view from the template being matched. At this point, one can decide to insert this 'new' iconic view in the codebook. If the codebook consists of several iconic views, all these views should be macthed against the image before the new feature region is found and before a new view may be inserted in the codebook. To avoid a computational expensive scheme we additionally propose to match only *other* iconic views from the codebook if the similarity measure between the '*current*' template and its best matching image patch is below the previously mentioned threshold q [1].

1. This method presumes that the feature is at least present in the recorded image. To cope with situations were this is not the case one can require that the best similarity measure is at least *above* a second threshold.

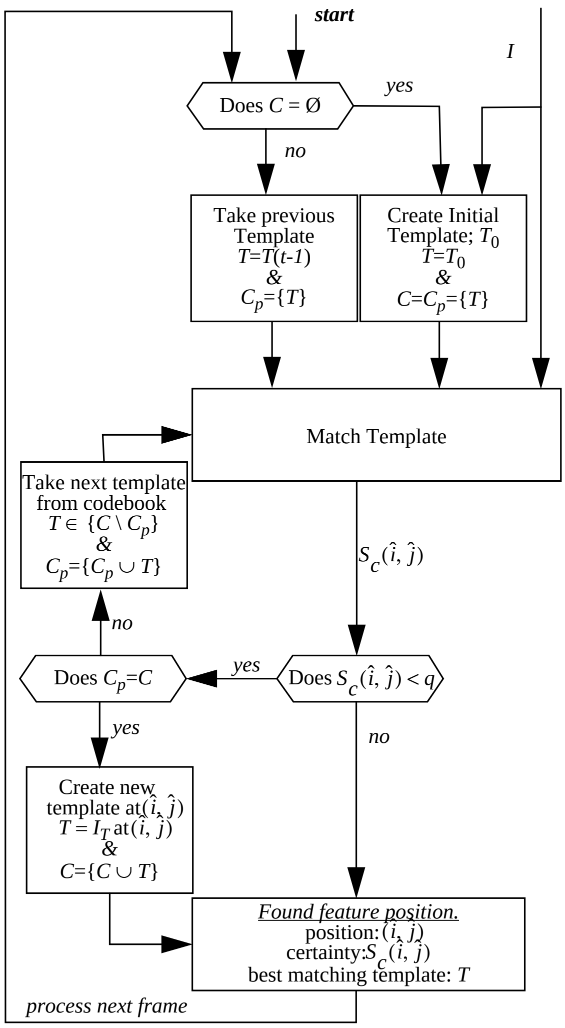


Fig. 2 Proposed tracking scheme for localization and automatic learning of new distinct iconic views.

The complete codebook generation process is shown in Fig. 2. Let C denote the codebook, i.e. the set of iconic views. After initialization (see Subsection II.A), the codebook contains only one template; T_0. Now, suppose that the codebook, after some while, contains three templates $C = \{T_0, T_1, T_2\}$. When processing a new frame one starts by matching the previous best matching template, suppose T_0. The set of already matched iconic views, in this case thus only T_0, is remembered by the set of processed templates: $C_p = \{T_0\}$. Now suppose that the maximal similarity measure for this template $S_0(\hat{i}, \hat{j})$ is above the predefined value q (a high similarity thus). This indicates that the iconic view represented by the template (T_0) and the best matching image patch are very similar and one may conclude that the position of the eye region is found.

In the case that the similarity measure was below q both views are too distinct. Because we didn't process all templates in the codebook at this moment (C_p does not yet equal C) we take the next template in the codebook, for example T_1 (C_p is consequently updated by $C_p = \{C_p \cup T_1\}$). Now, the procedure repeats itself. If, however, template T_1 and T_2 both are processed and the similarity measure is still below the predefined value, there are no views present in the codebook which resemble the feature region in the image. In this case a new template, T_4, is created at the position of the best matching template (see Eq. (14)) and added to the codebook; $C = \{C \cup T_4\}$. To restrict the number of iconic views in the codebook one can choose - if the codebook is full - to replace the *best* matching template, for example T_0; $C = \{\{C \backslash T_0\} \cup T_4\}$ (the latter is not included in Fig. 2).

4 Experiments

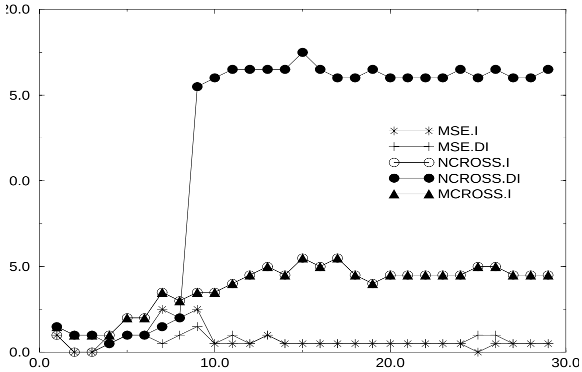
In this section we present the performance of the proposed eye tracker. The first experiment, in Subsection 4.1, shows the generated codebook and localization performance when only a part of the Miss America sequence is processed. Subsection 4.2 presents the results for the whole Miss America and Claire sequence. In Subsection 4.3 the automatically generated iconic views in the codebook for the Miss America sequence are examined.

The performance of the tracking scheme is evaluated by calculating the *block* distance (maximal difference in *x* and *y* direction) between the estimated eye position and the manually determined eye position. If we denote the estimated position by $\hat{\mathbf{x}} = (\hat{i}, \hat{j})$ and the manually assigned point by $\hat{\mathbf{x}}_M = (\hat{i}_M, \hat{j}_M)$ then the frame-by-frame performance indicator is expressed by $d(\hat{\mathbf{x}}, \hat{\mathbf{x}}_M)$.

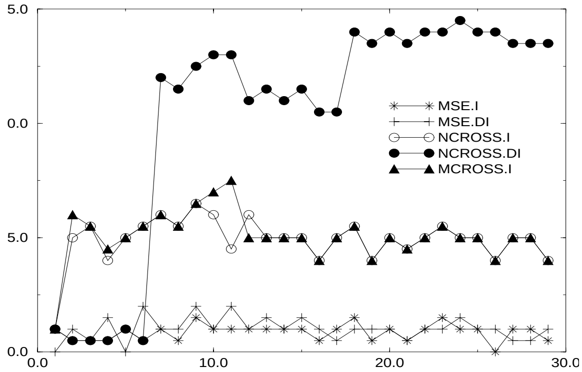
Besides the localization performance, the tracking scheme can also be evaluated by the number of iconic views of the feature region which are present in the codebook at any time instance and the number of these views that are analyzed each frame. Clearly, the number of iconic views in the codebook depend on how many different views of the feature region are observed in the time lapse of the analyzed image sequence. But, both measures can be used to differentiate between the different implementations of the tracking scheme. In the following we denote the codebook size when analyzing frame i by $|C_i|$ and the number of *evaluated* templates by the size of the processed codebook; $|C_{p,i}|$.

4.1 Choosing a Similarity Measure and Intensity Representation

To find out the best combination of similarity measure (Section 2.1) and intensity normalization (Section 2.2) we performed a series of experiments on the first 30 frames of the Miss America sequence. The results of the localization performance are shown in Fig. 3.a and Fig. 3.b for the left and right eye respectively.



(**a**)



(**b**)

Fig. 3 (**a**) shows the tracking performance of the left eye region, $d(\hat{\mathbf{x}}^{le}, \hat{\mathbf{x}}^{le}_M)$, for different settings of similarity measurements and intensity normalizations when analyzing the first 30 frames of the Miss America sequence. (**b**) shows the performance for localizing the right eye;. $d(\hat{\mathbf{x}}^{re}, \hat{\mathbf{x}}^{re}_M)$. Note, MSE.I indicates the use of the Mean Square Error (Eq. (8)) as similarity measure and *no* intensity normalization (see Section 2.2).

From these figures it can be seen that when using the normalized cross-correlation together with gradient-based normalization the proposed tracking scheme is not capable of finding the eye regions. On the one hand, because correlation seems not to have a high localization accuracy (as can be seen from the other figures), and on the other hand because the matching algorithm is distracted by the edginess of the hair region.

Inspecting the other results of the correlation measures (normalized and modified) on the plain intensity, one sees that - although the error is not cumulated - the localization accuracy is not very high. An example is shown in Fig. 4. We have no good explanation for this behavior, but the correlation measure seems distracted by the hair which is in front of the right eye region.

The localization performance when using the mean square error similarity measure is approximately similar when using either the plain intensity or the gradient-based normalization.

Similar results can be seen from Fig. 5 where the size of the codebook is shown as a function of the analyzed frame number. Again there is no clear prefer-

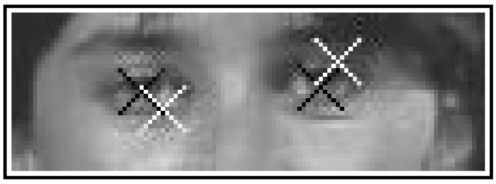


Fig. 4 Example (frame 12 of the Miss America sequence) of the mismatch when using the normalized cross correlation similarity measure and a plain intensity image representation. The manually determined eye locations are indicated by the black crosses and the automatically determined ones by the white crosses.

ence for using the plain intensity or gradient-based normalization.

In Fig. 5 one can additionally see that we have restricted the size of the codebook. In the case of a 'full' codebook and none of the iconic views match the feature region in the image (i.e. $S_c(\hat{i}, \hat{j}) < q \quad \forall c$) the iconic view in the codebook which *best* matches the found feature region is replaced (this view contains the lowest amount of information with respect to the view-based representation).

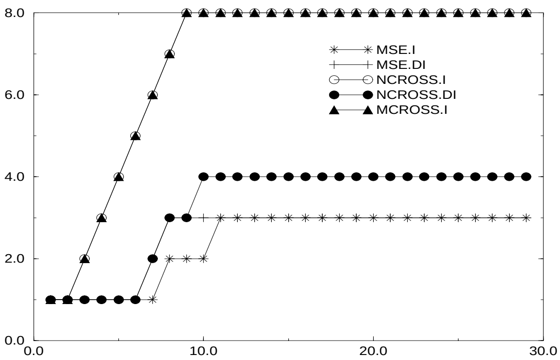


Fig. 5 The (maximal) size of the left and right eye codebooks; $max(|C_i^{le}|, |C_i^{re}|)$, for the different implementations of the tracking scheme. The maximal size of a codebook was set to eight templates.

Based on these experiments we have chosen to use the mean square error as similarity measure and the plain intensity as representation of the feature region. We preferred plain intensity above gradient-based normalization because of the lower computational load.

4.2 Localization Performance

Fig. 6 shows the localization performance when estimating the position of the left and right eye regions in all frames of the Miss America and Claire sequence. From this figure one can see that the localization error in all frames stayed below 4.5 pixels. No error accumulation occurred because of the use of the proposed codebook. Tests using only one template showed that error accumulation easily happened causing large localization errors. Further, one can see from Fig. 6 that in most frames the localization error is approximately

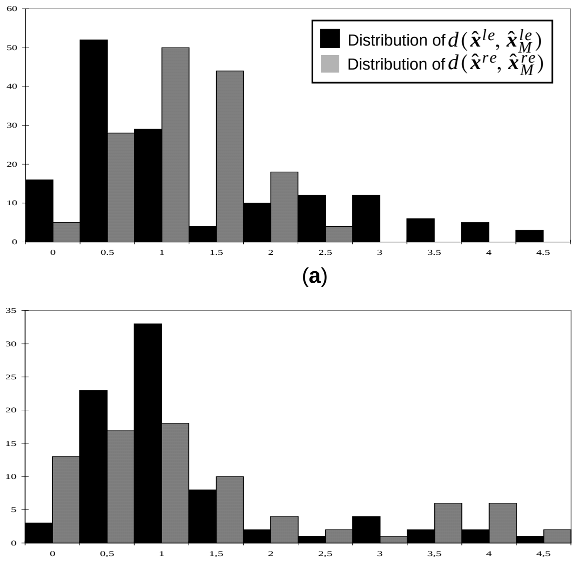


Fig. 6 Distribution of $d(\hat{x}^{le}, \hat{x}_M^{le})$ and $d(\hat{x}^{re}, \hat{x}_M^{re})$ for the tracked left and right eye points; (**a**) when analyzed 150 frames of the Miss America sequence, and (**b**) when analyzed 80 frames of the Claire sequence.

one pixel. In fact, most of the larger localization errors were caused by the difference in the *definition* of the eye region center in the case of manual estimation and automatic estimation. Manually, the center of the eye region was defined by the center of the eye corner points while the automatic scheme tends to find the center of the extreme upper and lower eyelid points. This effect can be seen in Fig. 7 for the left eye region. Further we should point out that, when comparing the performance of different operators similar error distribution are found. The average error of different operators with a reference operator, namely, was measured to be 0.98 pixels.

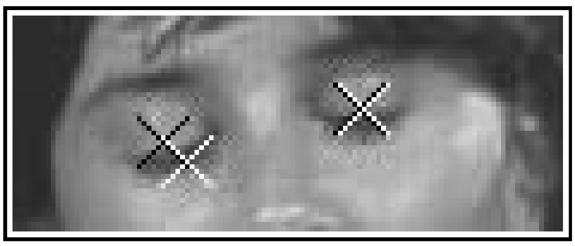


Fig. 7 Frame with largest error for the left eye (4.5 pixels). Note that the manual determined eye centers are defined by the corner points of the eye causing this relatively larger error.

4.3 Codebook generation

Fig. 8 shows the complete - automatically generated - codebook of iconic views of the left and right eye region when *all* frames (150) in the Miss America Sequence are processed. Here one can see that the iconic views in the codebook represent distinct appearances

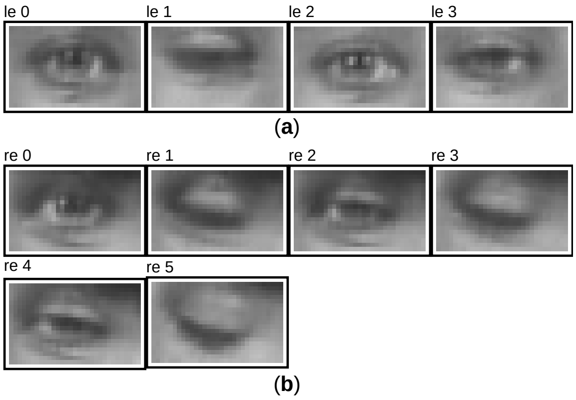


Fig. 8 The left (**a**) and right (**b**) eye codebooks after tracking the eyes in the Miss America sequence over 150 frames.

of the eye region, each showing a different state of eye blinking. Notice that based on this relatively small codebook the performance of the proposed tracking scheme is quite accurate despite *orientation*, *scale* and *shape* variances in the sequence.

It is interesting to examine the identity of the best matching template of the codebook shown in Fig. 8 for each frame of the Miss America sequence. These identities are plotted in Fig. 9. Examination of this plot shows that there are peaks at, for example, the 8th and 47th frame. It turns out that exactly during these frames Miss America has here eyes closed. Something similar happens between frames 116 and 134 where she has here eyes closed for quite some time. Hence, it seems that the iconic views in the codebook are strongly related to the eye gestures and consequently can be used for eye gesture recognition.

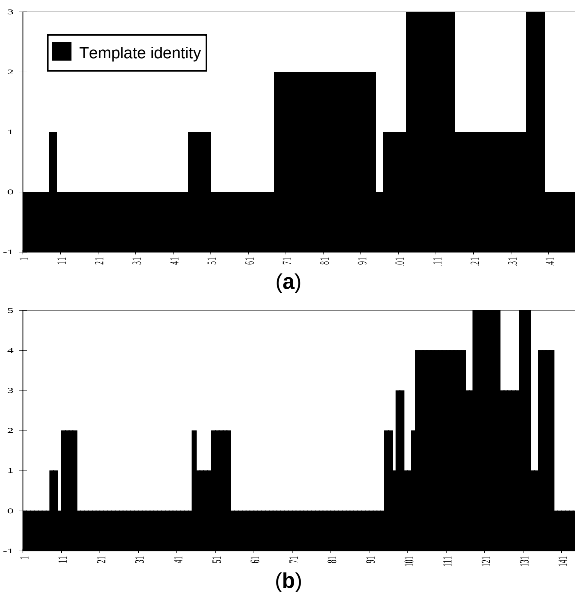


Fig. 9 The best matching template from the codebook shown in Fig. 8 for each frame of the Miss America sequence; (**a**) for the left eye region, (**b**) for the right eye region.

5 Conclusions

In this paper a new tracking scheme is introduced for tracking the location of the left and right eye region in image sequences. The proposed scheme is based on template matching, but is kept invariant for rotation and scale variations by exploiting temporal information. To cope with changing appearances of the sought feature region we proposed to use a codebook of iconic views of that region. The distinct iconic views are learned *automatically* during tracking of the feature region. By a number of experiments we showed that the proposed tracking scheme is quite accurate despite variations in orientation, scale and shape of the feature region.

An additional feature of the proposed method is that the automatically learned iconic views of the eye region correspond to distinct appearances of the eye that are strongly related to eye gestures, like eye blinking. Future research will be directed towards expanding the proposed tracking scheme with eye gesture recognition.

6 References

[1] Baron, R. (1981). Mechanisms of human facial recognition. *International Journal of Man-Machine Studies*, 15:137–178.

[2] Bichsel, M. (1991). *Strategies of Robust Object Recognition for the Automatic Identification of Human Faces*. Ph.D. thesis, ETH Zurich, Zurich.

[3] Brunelli, R. and Poggio, T. (1993). Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052.

[4] Chellappa, R., Wilson, C., and Sirohey, S. (1995). Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, 83(5):704–740.

[5] Darrell, T., Essa, I., and Pentland, A. (1996). Task-specific gesture analysis in real-time using interpolated views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1236–1242.

[6] Kjeldsen, R. and Kender, J. (1996). Toward the use of gesture in traditional user interfaces. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 151–156, Killington, Vermont, USA. IEEE Computer Society Press.

[7] Reinders, M. (1995). *Model Adaptation for Image Coding*. Ph.D. thesis, Delft University of Technology, Department of Electrical Engineering, Information Theory Group, P.O. Box 5031, 2600 GA, Delft, The Netherlands.

[8] Turk, M. (1996). Vision interaction with lifelike characters. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 368–373, Killington, Vermont, USA. IEEE Computer Society Press.

[9] Turk, M. and Pentland, A. (1992). Face recognition using eigenfaces. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–590, Hawai.

[10] Yau, J. and Duffy, N. (1989). A feature tracking method for motion parameter estimation in a model-based coding application. In *Proceedings Third International Conference on Image Processing and its Applications*, pages 531–535, UK.