# ZERO-CROSSING BASED IMAGE PROJECTIONS ENCODING FOR EYE LOCALIZATION

*Laura Florea, Corneliu Florea, Ruxandra Vranceanu, Constantin Vertan*

University Politehnica of Bucureşti,
Image Processing and Analysis Laboratory (LAPI)
Splaiul Independenţei, 313, Bucureşti, România

## ABSTRACT

This paper introduces a feature extraction and classification framework that is capable of fast and accurate description of the eye area. The extracted features are based on the combination of integral image projections with TESPAR signal encoding. While image projections were widely used in the field of image analysis, TESPAR was developed for speech recognition and it was only very recently used in conjunction with image signals. Using the unique combinations of the two techniques, we construct features that are easy to compute and provide independency with scale and illumination variation. The computed features are used as inputs into a Multi Layer Perceptron and the capabilities of the resulting framework are proved in an application of eye localization.

***Index Terms—*** Integral Projections, Variance Projections, TESPAR, Eye Localization

## 1. INTRODUCTION

In this paper we introduce and discuss a new type of features that leads to good localization performance, while being intuitive and simple to compute. We exemplify the proposed technique to the problem of eye localization. Even though solutions to this problem do exist (see the review in [1]) the problem of eye localization under various challenges, like coping with the individuality of eyes, occlusion, variability in scale, location, and light conditions, while keeping the computation cost low has still space for improvement.

The framework used for localization relies on feeding the newly introduced feature to a classifier. While the classifier comes in the widely known form of Multi-Layer-Perceptron (MLP), the process of feature extraction innovates by combining two techniques from two different fields of signal and image processing.

Image features create an alterative space to describe the original information. The purpose is to either provide a space where target data is much more distinguishable from the background, either to be more related to the human perception on the specific data. The hereby proposed feature extraction is based on two existing concepts: Integral Projections and TESPAR (Time-Encoded Signal Processing And Recognition) zero crossing based encoding. Hence we will name the resulting features Zero-crossing based Encoded image Projections (ZEP).

The remainder of this paper is organized as follows: Section 2 reviews the concepts related to Integral Projections and describes fast computation methods and Section 3 summarizes the TESPAR technique and includes a short discussion on the properties of the resulted features. The specific way of implementation and the achieved results in the field of eye localization are presented in section 4. The paper ends with a short summary and proposes further continuation paths.

## 2. INTEGRAL IMAGE PROJECTIONS

The integral projections, also named Integral Projection Functions (IPF) or amplitude projections are tools that have been widely used in face analysis. Due to its simplicity, the origins of the technique are somehow vague. It appears as "amplitude projections" [2] in analysis of medical images and as "integral projections" [3] for face recognition. Closely related to the current work, we have to mention [4] and [5] who used integral projection functions and their extensions for eye detection.

For a gray-level image rectangle $F(i,j)$ with $i = i_1 \dots i_2$ and $j = j_1 \dots j_2$, the projection on the horizontal axis is:

$$P_H(j) = \frac{1}{i_2 - i_1} \sum_{i=i_1}^{i_2} F(i,j), \forall j = \overline{j_1, j_2} \qquad (1)$$

and the projection on the vertical axis is:

$$P_V(i) = \frac{1}{j_2 - j_1} \sum_{j=j_1}^{j_2} F(i,j), \forall i = \overline{i_1, i_2} \qquad (2)$$

The projections are perceived as ways to reduce the dimensionality of images from 2D to 1D. The normalization

The work has been co-funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labor, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557 and POSDRU/89/1.5/S/76903.

 

ratio placed before summation is useful in associating the projection to the mean on the specific row or column. Another observation is that given a rectangular region of interest in an image, it can be described up to a certain level of details by its integral projections.

### 2.1. Generalized Projections

Over time, various extensions of the integral projections have been introduced. Feng [4] introduces the Variance Projection Functions (VPF). Considering the image rectangle, $i = [i_1; i_2] \times [j_1; j_2]$, the VPF on the horizontal axis is defined as:

$$V_H(j) = \frac{1}{i_2 - i_1} \sum_{i=i_1}^{i_2} \left(F(i,j) - P_H(j)\right)^2, \forall j = \overline{j_1, j_2} \quad (3)$$

and the projection on the vertical axis is:

$$V_V(i) = \frac{1}{j_2 - j_1} \sum_{j=j_1}^{j_2} \left(F(i,j) - P_V(i)\right)^2, \forall i = \overline{i_1, i_2} \quad (4)$$

The variance projection functions show the variance of the intensity distribution along the specified rows/columns.

While Zhou [5] defines the Generalized Projection Functions as convex combination between IPF and VPF, we consider that true generalization is achieved by extension to higher order statistical moments. The envisaged third order moment is the skewness (and we propose to use the Skewness Projection Function - SPF) while the forth is the kurtosis (and Kurtosis Projection Function - KPF). The reason for this choice lies in the information carried by each of these high order statistics. In terms of image rectangles, the IPF shows the mean value of a specific row/column, the VPF shows if the previous value was obtained as results of a uniform patch of highly varied, the SPF shows if the peak was in the first or second part of the value range, while the KPF shows that distribution if pixels on the specific row/column was flat or peaked. However let us note that for the current application, the projection only up to the variance will be used.

### 2.2. Fast Computation of Projections

Our purpose is to construct features that describe local parts of an image. Hence we must be able to compute in a fast manner the projections associated with a high number of blocks from an image. Taking into account that basically the projection is a sum, we envisaged two alternatives.

The first one is applying the concept of integral image introduced by Viola and Jones [6]. For instance, given the integral image $I(x, y)$, the horizontal IPF on the $i = [i_1; i_2] \times [j_1; j_2]$ image rectangle is found as

$$P_H(j) = \frac{\begin{array}{l}( \quad I(i_2, j_2) + I(i_i - 1, j_1 - 1) - \\ \quad I(i_2, j_1 - 1) - I(i_1 - 1, j_2))\end{array}}{i_2 - i_1} \quad (5)$$

Regarding the higher order projections, we note that they will require integral images computed on higher powers of the initial images. For example the variance is the "mean of the square minus the square of the mean"; therefore it requires the integral image computed on the square of initial values.

The alternate way relies on a version of the same principle. Instead of computing a single integral image, we compute two integral images, both of them oriented: one on horizontal and one on vertical axes. For instance if the image has $M \times N$ pixels the "horizontal integral image" is:

$$I_H(i,j) = \sum_{k=1}^{i} F(k,j), \forall i = \overline{1, M}, \forall j = \overline{1, N} \quad (6)$$

In this case, the horizontal IPF corresponding to the rectangle $i = [i_1; i_2] \times [j_1; j_2]$ is computed as:

$$P_H(j) = \frac{1}{i_2 - i_1} \left(I_H(i_2, j) - I_H(i_1 - 1, j)\right) \quad (7)$$

For higher order projections, the computation requires higher order integral images. We stress that the difference between the two alternatives is that in the first case we need three additions, while in the second case only one addition, but twice the memory. The choice of the specific solution is therefore related to the particularities of the target platform.

## 3. TESPAR ENCODING AND ZEP FEATURE

TESPAR encoding was introduced by King et. al. [7] as a technique for representation and recognition of band limited speech signals. Only recently some applications of TESPAR in the field of image analysis have been reported [8].

The TESPAR encoding is based on the determination of zero-crossings of the target signal. Between two consecutive zero-crossings there is a so–called *epoch*. In the original description an epoch was described by two parameters, while, for our algorithm, we will use three (as shown in figure 1):

- *Duration* - the number of samples of the epoch.
- *Amplitude* - the maximum absolute value of the epoch. It is stored with sign.
- *Shape* - the number of local optima (modes) in the epoch.

Depending on the problem specifics, the parameters of epoch may vary. For instance the amplitude, duration or maximum derivative of each mode may be added.

### 3.1. Zep Feature

Given an image rectangle, we compute the ZEP feature based on the following steps:

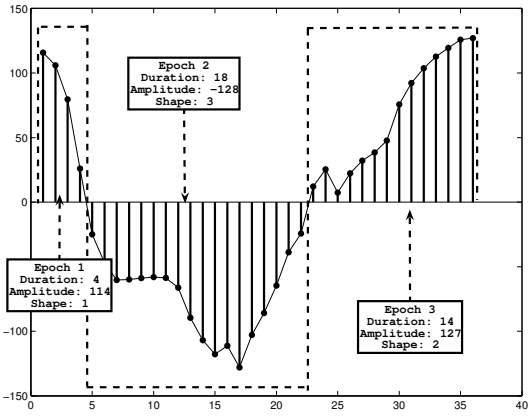


**Fig. 1**. Example of 1D signal (vertical projection of an eye crop) and associated TESPAR encoding. There are three epochs, each with three parameters. The associated code obtained by concatenating epochs would be: $[4, 114, 1,\ \ 18, -128, 3,\ \ 14, 127, 2]$.

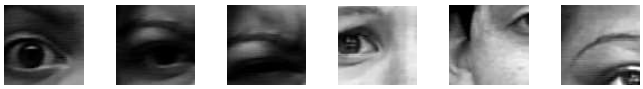

**Fig. 2**. Examples of eyes (the first three) image crops and non-eyes (the right hand three) crops. Please note that eyes expression is not static. Images are taken from [9].

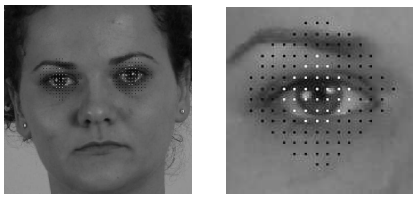

**Fig. 3**. Localization of the center of the squares used as positive examples (white) and negative examples (black). The right hand image is zoomed from the left hand one. The image is taken from the personal database.

1. Compute projection functions. Normalize each projection independently in a symmetrical interval;
2. Arrange projections in a specific order. For instance: $P_H, P_V, V_H, V_V, \ldots$;
3. Encode each projection with TESPAR (as discussed in the previous section and showed in figure 1). Allocate for each projection a maximum number of epochs.
4. Form the final ZEP feature by concatenation of the encoded projections.

While image projections are a simplified representation of images, with each of them carrying specific information, TESPAR encoding simplifies even more the image content. The specific choice of encoding preserves important characteristics of the projections while the simplification allows generalization.

The normalization with respect to the number of elements in the image rectangle in the computation of projections ensures partial scale invariance. To fully achieve the scale invariance property of the ZEP feature we must also normalize the encoded durations to a specific range (e.g. $[0, 255]$). Another important property of the ZEP feature is the independence with respect to uniform variation of illumination. This property is given by the normalization of amplitude in encoding; the shape is by definition invariant to change of illumination. The feature is not invariant to rotation or to non-uniform change in illumination.

## 4. IMPLEMENTATION AND RESULTS

For the problem of eye localization, we restricted the ZEP feature to integral and variance projections. Each projection was encoded with 5 epochs. The projections requiring more epochs were cut, while shorter ones were filled with 0. Therefore the ZEP feature had 60 elements for each considered image rectangle.

Once determined the features, for detection or localization, the extracted data must be feeded into a classifier. We consider that specific combination of epoch parameters describe an object. For instance, the vertical projection of an eye implies small positive epoch (higher amplitude, small duration - between the eyebrow and the eye) followed by large negative epoch (high amplitude large duration - due to the iris and pupil). This implies a specific combination that is characteristic to the Multi Layer Perceptron (MLP).

We have use a two layered feed–forward perceptron trained with back-propagation algorithm. A neuron from the first (hidden) layer should encode the previously described concept, while the second layer should perform the actual classification. In the actual implementation the MLP has 30 neurons on the first layer with sigmoid transfer function and one neuron on the second layer with the same sigmoid transfer function.

For training we used 10000 positive eyes examples and 10000 negative examples. All examples are $71 \times 71$ rectangular image patches (as presented in 2). The positive examples are centered on the eye with a maximal 18 pixels displacement, while negative ones were chosen as close as possible to ground truth eye (as shown in figure 3). Let us note that this specific choice of positive and negative makes the overall classifier less competitive in very high accuracy range, but capable of accommodating eye expressions.

The stringent localization criterion to evaluate the error rate [10] is used for accuracy evaluation. An eye is considered to be correctly determined if the specific error threshold, $\epsilon$, is

smaller than a value. The error is computed as:

$$\epsilon = \frac{\max\{\varepsilon_L, \varepsilon_R\}}{D_{eye}} \quad (8)$$

where $\varepsilon_L$ is the Euclidean distance between the ground truth left eye center and determined left eye center, $\varepsilon_R$ is the corresponding value for the determined right eye, while $D_{eye}$ is the distance between the ground truth eye centers.

Two kinds of tests were performed. A first test involved simple eye identification task, while the second involved eye localization.

In the first case various rectangular crops were encoded with ZEP and feeded to the MLP. The crops were selected from the Cohn-Kanade database [9]. While the database was developed for the study of emotions and contains frontal illuminated portraits, the challenge here is that eyes are in various poses (near-closed, half-open, wide-open).

From the database, we extracted the test set containing 33,335 eyes (in various positions) and 149,166 non-eyes. The system yields, in average a Detection Rate of 91% with a 13.24% False Positives rate. Even though higher performance may be obtained, by adjusting various parameters, we found out that such a system is less accurate in localization.

The second test implied actual identification of eye locations. For this, several steps are required: first we apply a face detector [6]; second we re-sample the estimated face square at $300 \times 300$ pixels and reduced the area of search to rows from 100 to 125 and columns from 75 to 225 (values empirically found on databases). Next, the locations in the given area are scanned (a square with $71 \times 71$ is ZEP encoded) and tested if they are eyes.

The output of the classifier was post-processed. After we found all possible points of eyes (i.e given the MLP value is larger than typical threshold value), we split the image on the left and right side. On each side, only locations where the network has reported values greater than a given percentage (60%) of maximum value of the side patch are kept. The resulting locations will form an image and the geometrical centrum of the largest compact area is considered as eye.

When actual scanning was performed on the Cohn Kanade database, the correct location with precision of $\epsilon < 0.1$, on the entire database was 92.51% and 98.97% with $\epsilon < 0.25$. Examples of results are presented in figure 4.

For a complementary test we envisaged the BioID database [11]. Here 1521 images of 23 different persons have been recorded. Among the set variable environment illumination, facial pose, eye occlusion due to eyeglasses consist as challenges. Visual results are showed in figure 5. The obtained results are shown in table 1.

We have implemented the described solution in C code and it took 25 msec/frame on a Intel i7 processor to localize a pair of eyes. While the code was not optimized and run in single thread, the application was capable of localizing eyes with 40 fps in HD - 720p ($1280 \times 720$).

**Fig. 4**. Cropped face images from Cohn-Kanade database. The ground truth eyes were marked with red (dark grey), while detected eye with green (light grey). Top row shows images where eyes are correctly localized, while bottom shows failure cases.

**Fig. 5**. Cropped face images from BioID database. The ground truth eyes were marked with red (dark grey), while detected eyes with green (light grey). Top row shows images where eyes are correctly localized, while bottom shows failure cases.

For a comparative study we have considered other results reported on the BioID database. Jesorsky et al. [10] proposed a Hausdorff distance based face matching method followed by a MLP eye finder. Niu et al. [12] uses a iteratively bootstrapped boosted cascade of classifiers based on Haar wavelet and Bai [13] changed Reisfelds generalized symmetry transform. Turkan et al. [14] used edge projection to localize the eye area and SVM to precisely determine the position. Asteriadis et al. [16] used the distance to closest edge to describe the eye area. Valenti et. al [15] used isophote properties to gain invariance and hence higher accuracy.

While analyzing the results obtained by the proposed method, we note that only Valenti [15] reported comparable or lower computational time for the Mean Shift (MS) version, which has also lower accuracy at $\epsilon < 0.1$ and $\epsilon < 0.25$. While there are methods more accurate than the hereby mentioned

| **Method** | Acc. $\epsilon < 0.05$ | Acc. $\epsilon < 0.1$ | Acc $\epsilon < 0.25$ |
|---|---|---|---|
| *Proposed* | *57.13* | *88.97* | *98.48* |
| Jesorsky[10] | 40.0 | 79.00 | 91.80 |
| Niu[12] | 75.0* | 93.0 | 98.0* |
| Bai[13] | 37.0* | 64.00 | 96.00 |
| Turkan[14] | 19.0* | 73.68 | 99.46 |
| Val.[15]+MS | 79.56 | 85.27 | 97.45 |
| Val.[15]+SIFT | 84.1 | 90.85 | 98.49 |
| Asteriadis[16] | 74.0* | 81.70 | 97.40 |

**Table 1**. The accuracy (Acc.) of the proposed algorithm and other prior art solutions on the BioID database [11]. *Values estimated from authors graphs.

one, they are more computationally intensive (as reported by authors). Niu requires a huge number of classifiers, Asteriadis needs to search for each pixel where is the closest edge, while Valenti required the SIFT feature descriptor to have high enough accuracy. Therefore, the overall results of our method while being close enough to presented competitors, represent a good compromise between accuracy and complexity.

## 5. CONCLUSIONS

We have shown that TESPAR encoded image projections (named ZEP) are fast and efficient feature detectors. We have studied the achievable performance in the context of eye localization and tested on public and widely used databases.

In the currently described work, we focused on the accuracy improvement and even though we identify the potential for fast computation we did not give the required attention to optimizing the execution time. Another direction is to perform intensive testing to determine the actual potential of the introduced descriptor.

## 6. REFERENCES


[1] D. W. Hansen and Qiang Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 478 – 500, March 2010.

[2] H. C. Becker, W. J. Nettleton, P. H. Meyers, J. W. Sweeney, and C. M. Nice, "Digital computer determination of a medical diagnostic index directly from chest X-ray images," *IEEE Transactions on Biomedical Engineering*, vol. 11, no. 3, pp. 62 – 72, July 1964.

[3] T. Kanade, "Picture processing by computer complex and recognition of human faces," Technical Report, Kyoto University, Department of Information Science, 1973.

[4] G. C. Feng and P. C. Yuen, "Variance projection function and its application to eye detection for human face recognition," *Pattern Recognition Letters*, vol. 19, no. 9, pp. 899 – 906, July 1998.

[5] Z.H. Zhou and X. Geng, "Projection functions for eye detection," *Pattern Recognition*, vol. 37, no. 5, pp. 1049 – 1056, 2004.

[6] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[7] R. A. King and T. C. Phipps, "Shannon, TESPAR and approximation strategies," *Computers & Security*, vol. 18, no. 5, pp. 445 – 453, 1999.

[8] S. Emerich, E. Lupu, and R. Arsinte, "A new approach to iris recognition," in *10th International Symposium on Signals, Circuits and Systems (ISSCS)*, Iaşi, România, July 2011, pp. 1 – 4.

[9] T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00)*, 2000, pp. 46–53.

[10] O. Jesorsky, K. Kirchberg, and R. Frischolz, "Robust face detection using the Hausdorff distance," in *Audio and Video Based Person Authentication*, J. Bigun and F. Smeraldi, Eds. 2000, pp. 90–95, Springer.

[11] "BioID face database," http://www.bioid.com/ downloads/facedb/facedatabase.

[12] Z. Niu, S. Shan, S. Yan, X. Chen, and W. Gao, "2D cascaded adaboost for eye localization," in *Proceedings of the 18th Internationl Conference on Pattern Recognition*, 2006, pp. 1216 – 1219.

[13] L. Bai, L. Shen, and Y.Wang, "A novel eye location algorithm based on radial symmetry transform," in *Proceedings of International Conference on Pattern Recognition*, Hong - Kong, 2006, pp. 511 – 514.

[14] M. Turkan, M. Pardas, and A. E. Cetin, "Edge projections for eye localization," *Optical Engineering*, vol. 47, no. 047007, 2004.

[15] R. Valenti and T. Gevers., "Accurate eye center location and tracking using isophote curvature," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[16] S. Asteriadis, N. Nikolaidis, and I. Pitas, "Facial feature detection using distance vector fields," *Pattern Recognition*, vol. 42, no. 7, pp. 1388 – 1398, 2009.