MORPHOLOGICAL AND ADABOOST FACE DETECTORS COMPARISON

K. Kucharski, W. Skarbek

Institute of Radioelectronics
Warsaw University of Technology
K.Z.Kucharski@elka.pw.edu.pl

G.N. Stamou, N. Nikolaidis and I. Pitas

Department of Informatics
Aristotle University of Thessaloniki
Box 451, 54124 Thessaloniki, GREECE
Email: $\{$*gstamou,nikolaid,pitas*$\}$*@aiia.csd.auth.gr*

ABSTRACT

In this paper two novel face detection methods based on different approaches are presented. The Morphological Elastic Graph Matching face detector uses some expert knowledge about a layout of the face and mutual relations between its most parts. It allows for accurate detection of frontal faces with some tolerance regrading view, in-plane rotation etc. but it is relatively slow. The high speed of the detector is a crucial parameter in many face detection applications, e.g. the face recognition. To this end the Central Two-step AdaBoost detector is presented which defines a cascade of the extended weak classifiers that accelerates the detection speed to a high degree.

1. INTRODUCTION

Face detection methods can be divided into two main categories, namely knowledge-based methods and appearance-based methods [1]. First group utilizes to certain degree the human perception of how the face should look like, what means that knowledge about standard layout of the face and mutual relations between its most important parts, as perceived by the human, is always somehow embedded into the algorithms. Second group, on the other hand, tries not to assume any prior knowledge about the appearance of the face but rather aims at extracting its most important features directly from a representative training set of faces.

In this work two novel approaches belonging to both categories are discussed. The Morphological Elastic Graph Matching face detector, which belongs to the knowledge-based methods category, is described in section 2. Two Step Central AdaBoost detector, a member of appearance-based face detection techniques, that adopts the AdaBoost methodology, is presented in section 3. In a concluding section 4 both algorithm are evaluated using challenging database CMU+MIT.

2. FACE DETECTION BASED ON MORPHOLOGICAL ELASTIC GRAPH MATCHING

Elastic Graph Matching (EGM) [2] has previously been used in object recognition [2], face authentication - verification [3] gesture recognition and tracking [4].

The basic representation for faces using the Morphological Elastic Graph Matching (MEGM) algorithm is a labelled graph. The graph vertices are associated with feature vectors. The latter are calculated based on techniques originating from scale-space image analysis, which are applied to the image points that correspond to the graph vertices. Additionally, the graph edges encode information about the relative position of the graph vertices. Feature vectors can be built in different ways. The multiscale morphological dilation-erosion of the image by a scaled circular structuring function [3] has been employed to form such vectors, both because it reduces the computational complexity and provides good facial feature representations, since dilations-erosions deal with local extrema in the image.

Let $V = \{(i,j)\}$, $i = 1, ..., M$, $j = 1, ..., N$ be the set of vertices of an $M \times N$ graph. The outputs of multiscale dilation-erosion for the integer scale parameter of the structuring function $\sigma = -\sigma_m, ..., \sigma_m$ form the feature vector $\mathbf{j}$, also called "jet":

$$\mathbf{j}(\mathbf{x}_{i,j}) = ((f \star g_{\sigma_m})(\mathbf{x}_{i,j}), ..., (f \star g_1)(\mathbf{x}_{i,j}), f(\mathbf{x}_{i,j}), \\ (f \star g_{-1})(\mathbf{x}_{i,j}), ..., (f \star g_{-\sigma_m})(\mathbf{x}_{i,j})) \tag{1}$$

where $\mathbf{x}_{i,j} = (x_{i,j}, y_{i,j})$ denotes the coordinates $x_{i,j}, y_{i,j}$ of vertex (i,j) in the image and $(f \star g_\sigma)(x)$ denotes the multiscale dilation-erosion of the image $f(\mathbf{x})$ by $g(\mathbf{x})$. $\sigma = \{-9, ..., 9\}$ both for computational complexity and accuracy reasons.

The MEGM algorithm detects a face by performing translations and deformations of a reference face graph aiming at minimizing a cost function. The cost function is based on both the norm of the difference between the feature vectors that correspond to the same graph vertex in the reference and the test images and the geometrical distortion between the reference and the candidate graph configurations. Let the superscripts r and t denote the reference and test images respectively and $N((i,j))$ denote the four-connected neighborhood of vertex (i,j). The goal of the detection algorithm is to find the set $\{\mathbf{x}^t_{i,j}\}$ of graph vertex coordinates $\mathbf{x}^t_{i,j}$ in the test image that minimizes the cost function:

$$C(\{\mathbf{x}^t_{i,j}\}) = \sum_{(i,j)\in V} \left\{ C_v(\mathbf{j}(\mathbf{x}^t_{i,j}), \mathbf{j}(\mathbf{x}^r_{i,j})) + \lambda \sum_{(k,l)\in N((i,j))} C_e((i,j),(k,l)) \right\} \quad (2)$$

where $C_v(\mathbf{j}(\mathbf{x}^t_{i,j}), \mathbf{j}(\mathbf{x}^r_{i,j}))$ denotes a similarity measure between the feature vectors at graph vertex (i,j) in the reference and the test image and $C_e((i,j),(k,l))$ is a term penalizing the deformations of the graph, defined by:

$$C_e((i,j),(k,l)) \triangleq \|(\mathbf{x}^t_{i,j} - \mathbf{x}^r_{i,j}) - (\mathbf{x}^t_{k,l} - \mathbf{x}^r_{k,l})\|, \quad (k,l) \in N((i,j)) \quad (3)$$

Normalized correlation has been used as the similarity measure in (2):

$$C_v(\mathbf{j}(\mathbf{x}^t_{i,j}), \mathbf{j}(\mathbf{x}^r_{i,j})) \triangleq \frac{\mathbf{j}(\mathbf{x}^t_{i,j}) \cdot \mathbf{j}(\mathbf{x}^r_{i,j})}{\|\mathbf{j}(\mathbf{x}^t_{i,j})\| \; \|\mathbf{j}(\mathbf{x}^r_{i,j})\|}, \quad (4)$$

between the feature vectors at graph vertex (i,j). $(\cdot)$ denotes the inner product. The optimization of (2) is performed in a simulated annealing framework with additional penalties imposed by the graph deformations [3]. Since $C_e((i,j),(k,l))$ penalizes only the graph deformations and not the translations of the entire graph, the vector of graph vertex coordinates $\mathbf{x}^t_{i,j}$ in the test image can be the result of a translation $\mathbf{s}$ of the entire graph (prior to any deformations) and a bounded local perturbation $\boldsymbol{\delta}_{i,j}$ of the graph vertex coordinates in the reference image:

$$\mathbf{x}^t_{i,j} = \mathbf{x}^r_{i,j} + \mathbf{s} + \boldsymbol{\delta}_{i,j}, \quad \|\boldsymbol{\delta}_{i,j} \leq \boldsymbol{\delta}_{max}\| \quad (5)$$

The first stage of the proposed method involved the creation of the reference face image and consequently the reference face graph. For this purpose, a facial database was used to calculate the mean face. The images of the database were normalized and aligned multiple times, with respect to different features of the face, so as to get the best possible accuracy. The reference face graph was then created by selecting 64 (8×8) points around the eyes, eyebrows, nose, mouth and face contour and calculating the feature vectors (1). This process was repeated for different sizes (starting from 16×16 pixels).

The next stage was to train the proposed face detection system. For each of the resulting reference face graphs in the previous stage, we applied the MEGM algorithm between the "reference" face graph and all the images of a face database other than the one used to create the reference face graph. The values of the similarity measure introduced earlier (4) were recorded. The same process was repeated for a database of non-face samples created from web images. Based on the recorded values, we calculated the thresholds for the similarity measure that best separate face from non-face samples. These can be used to accept or reject a candidate region as a face.

During the test stage, the MEGM algorithm was applied between the reference face graph and each test image, using a constrained search, since an exhaustive search would be too slow. Merging of the results was performed both in spatial terms (i.e. adjacent detections) and in scale terms (detection of the same test image region in different sizes). In both cases, the best result was retained.

A first inspection of the results revealed an increased rate of false detections. To eliminate this, a statistical study on a number of geometrical properties of the human face contour was performed and thresholds for these properties were derived. These were used as a verification step to eliminate false alarms.

3. CENTRAL TWO-STEP ADABOOST DETECTOR

The Central Two-step AdaBoost detector connects two cascade-like detector structures, namely a cascade of the extended weak classifiers and the AdaBoost cascade [5] that is a serial connection of strong classifiers being sets of weak classifiers selected in course of the AdaBoost training scheme [6].

The structure of the central two-step classifier (Figure 2) introduces two novelties in comparison to the AdaBoost detector. It replaces first part of the AdaBoost cascade with the cascade of extended weak classifiers and it uses central part of faces as the positives in the training set of L labelled image windows $O_1, \ldots, O_L$ instead of holistic facial images.

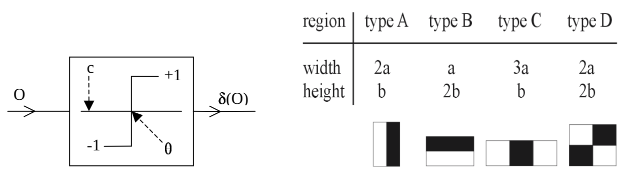

Fig. 1. Diagram of the weak classifier used in the classic AdaBoost approach (left) and basic region types associated with it (right).

The idea of the extended weak classifier is based on an observation that with help of basic region types showed in

Figure 1 only contrasts at neighbouring areas are actually calculated. Therefore, the search space of the discriminative contrast features is restricted to the special sub-set. An extended weak classifier is a weak classifier associated with a region R consisting of an arbitrary number of rectangular sub-regions R^+ and R^- that may occur at any locations within the image window.

The second observation takes into consideration the speed of the cascade-based detector which depends to a very large extent on how many classifiers the first stages of the detector are composed of. The false rejection rate and false acceptance rate, that are both actual quality measures of the classic algorithm, do not come up until the strong classifier level. This double evaluation can possibly produce certain amount of inertia and thus delay the cascade algorithm convergence which is guaranteed by the AdaBoost properties anyway [7]. Hence, another key idea behind the extended weak classifier is to move its evaluation in terms of false acceptance rate and false rejection rate lower, on the level of a weak classifier, to be able to reach the stop condition of a single cascade's stage for the smaller number of its weak components.

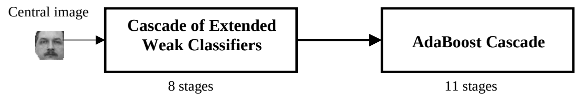

Fig. 2. A diagram of the central two-step classifier.

The main steps of the extended weak classifier algorithm are as follows:

Algorithm *Extended weak classifier*

Input:
– training image windows: $O_1, \ldots, O_L$
– Labels of the training image windows: $y_1, \ldots, y_L$
– Algorithm's parameters: fr, fa_{thr}, eps

Output:
– The best extended weak classifier: Ω_{opt}
– Associated optimal threshold: θ_{opt}
– Minimum false acceptance rate: fa_{opt}

Method: – Execute steps $1-6$

1. $fa_{opt} \leftarrow 1$; $fa_{prev} \leftarrow 0$; $\Omega_{opt} \leftarrow \emptyset$;

While $(fa_{opt} < fa_{thr})$:

For all available regions R:

For $j = 1, \ldots, L$:

2. if $\Omega_{opt} \neq \emptyset$, than $c_j \leftarrow$*calcContrast*(O_j, Ω_{opt});
   otherwise $c_j \leftarrow 0$;
3. $c_j \leftarrow c_j + \sum_{(x,y)\in R} O_j(x, y)$;
4. $(\theta^*, fa^*) \leftarrow$*findBestThr*$(fr, c, y)$;
5. if $fa^* < fa_{opt}$, than
   $\theta_{opt} \leftarrow \theta^*;\ \Omega_{opt} \leftarrow \{\Omega_{opt}, \omega\};\ fa_{opt} \leftarrow fa^*$;

if $\frac{fa_{prev} - fa_{opt}}{fa_{prev}} < eps$, than break;

$fa_{prev} \leftarrow fa_{opt}$:

6. return $(\Omega_{opt}, \theta_{opt}, fa_{opt})$;

Every region in the above algorithm is described by the compound parameter $\Omega = \{(x_1, y_1, a_1, b_1, \text{sign}_1), \ldots, (x_M, y_M, a_M, b_M, \text{sign}_M)\}$, where M is a number of its sub-regions and sign identifies every sub-region as R^+ or R^-. The algorithm is sub-optimal in that sense it builds the output extended weak classifier incrementally, therefore in current iteration uses a complete extended weak classifier from the previous iteration and scans all possible sub-regions to find the best one to add to the present structure. The algorithm is controlled by three parameters, a maximum acceptable false rejection rate fr, a false acceptance rate fa_{thr} to be achieved and a threshold eps that stops the algorithm if adding further sub-regions does not improve the trained classifier performance by a satisfactory margin.

The extended weak classifier that satisfies the constraints implied by parameters fr and fa_{thr} is, in fact, the stage of a cascade in its own right. The output structure of that kind, learned using the training set of central faces, is illustrated in Figure 3. The maximum acceptable false rejection rate and maximum acceptable false acceptance rate were set to 0.001 and 0.5, analogously to the AdaBoost cascade training process, performed separately. Comparison of weak classifiers numbers at first six stages of those two cases (2 vs. 8, 2 vs. 11, 3 vs. 18, 3 vs. 19, 3 vs. 26, 3 vs. 38, explicitly shows that a few times less computations is necessary to perform the first part of detection with the cascade of extended weak classifiers. Despite the training stuck at the eighth stage with 17 sub-regions reaching false acceptance rate 0.6 it succeeded to perform the task because its total false acceptance rate was already below 0.01.

Therefore, the cascade of the weak classifiers ensured fast detection and the standard AdaBoost cascade was used to push the overall false acceptance down to ensure accuracy of the detection. The resulting AdaBoost cascade part consisted of 11 stages and 232 weak classifiers, with subsequent

Fig. 3. Regions associated with the extended weak classifiers at all eight stages of a cascade of the extended weak classifiers.

stages having respectively $10, 13, 14, 18, 19, 22, 24, 26, 29, 28,$ 29 weak classifiers. The whole central two-step detector reached false acceptance level about $6 \cdot 10^{-6}$.

4. EXPERIMENTAL RESULTS

The proposed algorithms have been both tested on the CMU-MIT frontal subset that consists of 130 gray-level real-world images with 507 manually labelled faces and it is a very challenging set of images due to broad spectrum of different types of complex backgrounds, fairly large variability both of a face size and a number of faces in a single image, and finally different quality of photos. Accuracy of both detectors was evaluated in terms of true detection rate td and a false acceptance rate fa, i.e. false alarms. The overall comparison of the described algorithms is shown in Table 1 while exemplary images are presented in Figure 4. The variable numbers regarding MEGM detector refer to different parameters of the algorithm that can be tuned to achieve the best compromise between speed and accuracy. In case of MEGM detector processing time depends on the required accuracy and is between 3 and 6 seconds, while Two-step Central AdaBoost detector processes a CIF frame on Pentium IV 3.2 GHz in about 250 ms.

Algorithm	td	fa
MEGM	75-87 %	5-10 %
Two-step Central AdaBoost	78.5 %	23

Table 1. Results of the proposed face detection methods on the CMU-MIT face database.

5. ACKNOWLEDGEMENT

The work presented was developed within VISNET, a European Network of Excellence (http://www.visnet-noe.org), funded under the European Commission IST FP6 programme.

6. REFERENCES

[1] M.-H. Yang, D. J. Kriegman, and N. Ahuja, “Detecting faces in images: A survey,” *IEEE Trans. PAMI*, vol. 24, no. 1, pp. 34–58, 2002.

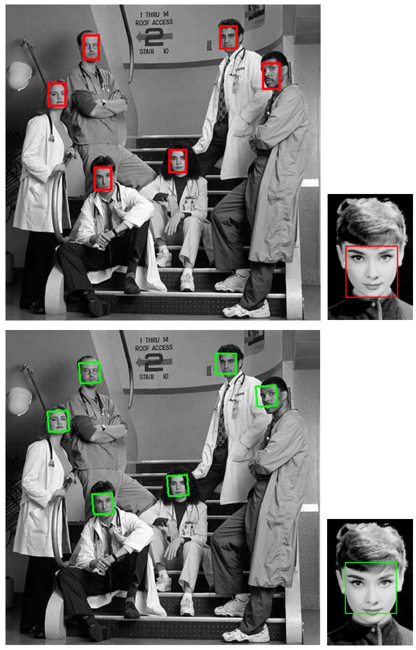

Fig. 4. Exemplary detection results on MIT-CMU database; MEGM detector (upper row), Two-step Central AdaBoost detector (lower row)

[2] M. Lades, J.C. Vorbrüggen, J. Buhmann, J. Lange, C. v.d. Malsburg, R.P Würtz, and W. Konen, “Distortion invariant object recognition in the dynamic link architecture,” *IEEE Transactions on Computers*, vol. 42, no. 3, pp. 300–311, March 1993.

[3] C. Cotropoulos, A. Tefas, and I. Pitas, “Frontal face authentication using morphological elastic graph matching,” *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 555–560, April 2000.

[4] G. Stamou, N. Nikolaidis, and I. Pitas, “Object tracking based on morphological elastic graph matching,” in *IEEE International Conference on Image Processing (ICIP 2005))*, Genova, Italy, September 2005.

[5] K. Kucharski, “Optimization of face detection with adaboost cascade algorithm,” *Proc. Workshop on Image Analysis for Multimedia Interactive Services WIAMIS*, 2005.

[6] P. Viola and M. Jones, “Robust real-time object detection,” *2nd Int’l Workshop on Stat. and Comp. Th. of Vis. –Modeling, Learning, Computing and Sampling*, 2001.

[7] W.Skarbek and K. Kucharski, “Image object localisation by adaboost classifier,” in *ICIAR 2004 Proceedings, Part I*, A. Campilho and M. Kamel, Eds., 2004, LNCS, pp. 511–518.