

Robust Face Detection at Video Frame Rate Based on Edge Orientation Features

Bernhard Fröba and Christian Küblbeck
Department of Applied Electronics
Fraunhofer Institute for Integrated Circuits
Am Weichselgarten 3, D-91058 Erlangen, Germany
Email: {bdf,kue}@iis.fhg.de

Abstract

In this paper we present a novel approach to real-time face detection in arbitrary grey images and video streams using a multi-stage detection pipeline. It is based on an edge orientation matching and a subsequent candidate verification using a SNoW classifier at the final stage. We have developed a simple and efficient method for object modeling and matching using edge orientation information only which we call EOM (Edge Orientation Matching). To make the system more robust we verify face candidates obtained by the EOM with a view based SNoW classifier which operates directly on the grey values. For further speedup the pipelined matcher is combined with a coarse-to-fine grid scan of the image. On average it thus requires less than 10% of all possible image locations to be processed by the detection pipeline. With this method it takes 50 msec on an Athlon 1000MHz PC to analyze an image of spatial size 384×288 . Experimental results on a large database of 18704 grey still images with 19030 detectable frontal faces, including the M2VTS frontal face database, the CMU test set of Rowley and Baluja and the newly released BioID testset, show that the presented approach outperforms methods that use Neural Networks, Bayesian methods and SVMs by far in terms of computational efficiency while showing comparable detection results. The overall recognition performance is above 96%.

1. Introduction

On-line face detection may serve as basic technique for many potential applications such as complex interaction in multi-modal human machine interfaces, face recognition, video coding and virtual reality applications. The fast and reliable detection of faces is a challenging task which provides general insight to real-time computer vision problems. In the past, several approaches to face detection have been made. A widely-used class of methods for find-

ing faces uses various kinds of grey level correlation approaches. Some authors[12, 1, 5, 7] use artificial neural networks trained with normalized face images and non-face images to model the problem domain. Another view-based algorithm was developed by Sung[14]. He also used normalized grey level images to represent a face and non-face class, but he used an alternative classification method based on the combination of a statistical description of the grey value appearance of each class and a subsequent neural network. Osuna[11] uses a support vector machine as classification algorithm.

Many view-based approaches model the appearance of the face in the eigenspace, like Moghaddam and Pentland [10]. The eigenspace is a feature space and is built by singular value decomposition of a large set of training images. Several distance metrics have been proposed for an eigenspace-based face detection algorithm. Moghaddam[10] for example proposes the DIFS (distance in feature space) metric and a probabilistic model is used to model the face class within the feature space. Yang and colleagues[15] propose two methods, one based on self organizing maps and linear discriminant analysis and another one using a mixture of factor analyzers for face detection.

Especially the view-based methods mentioned above are reported to have a good detection performance but the drawback to be computationally rather expensive and at present are not the appropriate choice for real-time applications. Fast methods on the other hand mostly rely on the use of color or motion information. Both may function well under controlled conditions, but may fail in real world applications.

We have designed a pipelined matching algorithm that uses edge orientation at the early stages and grey appearance for verification at the last stage of the pipeline. This approach together with a coarse-to-fine image scan leads to a extremely fast detection system that runs at video-framerate on standard computers. The proposed algorithm is also robust under a wide range of acquisition conditions such as lighting and background variations.

There are several algorithms that use edge orientation information for face processing. Bichsel [2] uses dot products of edge direction maps for the purpose of face recognition. Burl et al. [3] utilize shape statistics of the facial feature (eyes, nose, mouth) configuration learned from a training sample. The feature candidates are detected by what they call Orientation Template Correlation. Maio and Maltoni [8] utilize the hough transform for elliptical face outline detection with a subsequent matching of a manual generated orientation model for verification. The mentioned edge-based methods are not real time for face detection of arbitrary size and the published results are obtained using only small databases.

2. System Overview

Our approach is designed to cope with still images as well as video streams. Figure 2 gives an overview over the still image detector. It consists of two consecutive applied methods EOM (Edge Orientation Matching) and an appearance based method called SNoW. It was first used for face detection by Yang[16] et al.. Both methods are applied at an image location in a pipelined manner. The EOM itself consists of a pipeline of successive matching stages, like displayed in Fig. 2. First the model parts with the highest discrimination power are matched at the early stages. Each stage of the EOM has the ability to reject the current position if the score does not match the score criterion of this stage. Thus most of the examined image locations are ruled out at very early stages with little computation. This is further supported by the edge orientation features as the orientation field of an image is usually populated very sparsely.

An analysis window that passed the EOM-pipeline is verified in the last matching stage by a grey level method. Unlike the previous EOM-pipeline which operates in the orientation domain the last one processes the grey value information directly. It requires expensive normalization of the analysis window. But this last stage is reached only a few times per image so that this does not carry much weight on the whole detection process.

For further speedup the pipelined matcher is combined with a coarse-to-fine grid scan of the image, which on average requires less than 10% of the possible image locations to be processed by the detection pipeline. In the next sections we detail the processing blocks and present the performance figures of the system measured on a large database.

3. Edge Orientation Matching

In this section we derive a formal description of the EOM. First we describe edge orientation fields which are related to gradient fields. The difference is that the angles

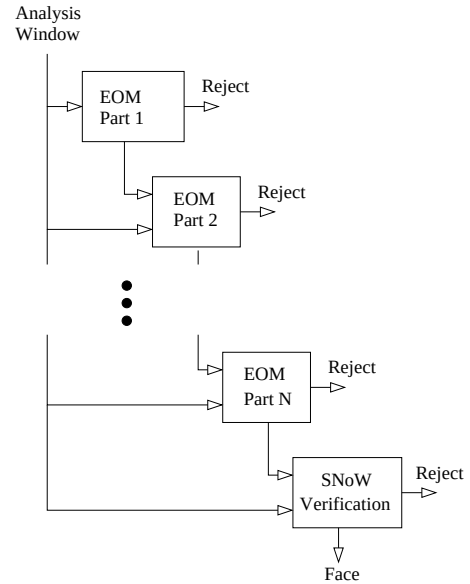


Figure 2. Matching pipeline.

are mapped to $[0 \dots \pi]$ and a threshold on edge strength is used to mask out unreliable edge information. We also introduce an effective distance metric and show how to model an object in the edge orientation domain.

3.1. Edge Orientation Fields

The extraction of edge information (strength and orientation) from a 2-D array of pixels $I(x, y)$ (a grey-scale image) is the basic feature calculation in our detection framework. In this work we use the Sobel method (see for example [4]) for edge processing. It is a gradient-based method which needs to convolve the image $I(x, y)$ with two 3×3 filter masks, K_x for horizontal filtering and K_y for vertical filtering.

$$K_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \text{ and } K_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (1)$$

The convolution of the image with the two filter masks gives two edge strength images $G_x(x, y)$ and $G_y(x, y)$,

$$G_x(x, y) = K_x \star I(x, y), \quad (2)$$

$$G_y(x, y) = K_y \star I(x, y). \quad (3)$$

The absolute value $S(x, y)$, referred to as edge strength and the edge direction information $\Phi(x, y)$ are obtained using:

$$S(x, y) = \sqrt{G_x^2(x, y) + G_y^2(x, y)}, \quad (4)$$

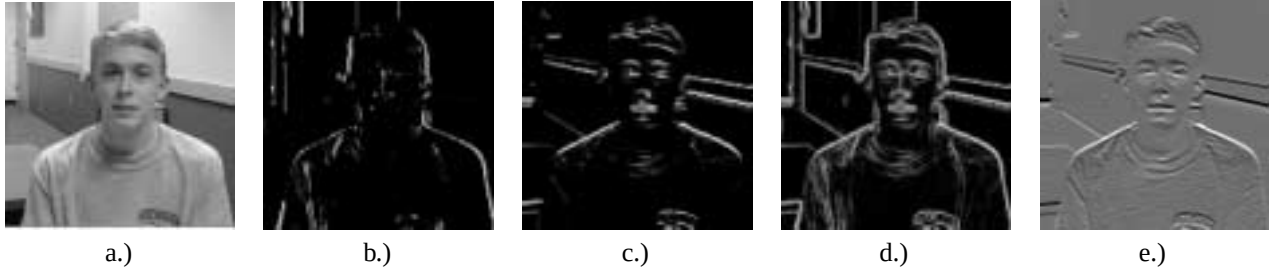


Figure 1. Example of an image (a), the vertical edge image $G_x(x, y)$ from Equ. (2) (b), the horizontal edge image $G_y(x, y)$ from Equ. (3) (c), the edge strength image $S(x, y)$ from Equ. (4) (d) and the edge orientation image $\Phi(x, y)$ from Equ. (5) (e), where the orientations are coded as grey value information

$$\Phi(x, y) = \arctan\left(\frac{G_y(x, y)}{G_x(x, y)}\right) + \frac{\pi}{2}. \quad (5)$$

The edge information on homogenous parts of the image where no grey value changes occur is often noisy and bears no useful information for the detection. To exclude this information we apply a threshold T_s to the edge strength $S(x, y)$ generating an edge strength field $S_T(x, y)$,

$$S_T(x, y) = \begin{cases} S(x, y) & \text{if } S(x, y) > T_s \\ 0 & \text{else} \end{cases} \quad (6)$$

The edge direction as stated in equation (5) takes on values from 0 to 2π . The direction of an edge depends on whether the grey value changes from dark to bright or vice versa. This information is irrelevant for our purposes. Therefore, we map the direction information to a range of value $[0 \dots \pi]$ obtaining a new field

$$\hat{\Phi}(x, y) = \begin{cases} \Phi(x, y) & \text{if } 0 \leq \Phi(x, y) < \pi \\ \Phi(x, y) - \pi & \text{if } \pi \leq \Phi(x, y) < 2\pi \end{cases}, \quad (7)$$

which we call the edge orientation field. The edge orientation information can be rewritten using a complex formula

$$\mathbf{V}(x, y) = S_T(x, y)e^{j\hat{\Phi}(x, y)}, \quad (8)$$

where $\mathbf{V}(x, y)$ is the complex edge orientation vector field and $j^2 = -1$. $S_T(x, y)$ and $\hat{\Phi}(x, y)$ are obtained using equation (6) and (7). The edge orientation vector field can be displayed like shown in figure 3. The elements of $\mathbf{V}$ are referred to as $\mathbf{v}$.

3.2. Orientation Templates

To build a face model, we use a sample of 2000 hand-labeled face images which are different from the test sets described later. The faces are cropped, aligned and scaled to the size 27×32 in the grey level domain. From this

set of normalized face images an average face is computed. We also use vertically mirrored versions of each face in the set for the average face. Finally the edge orientation vector

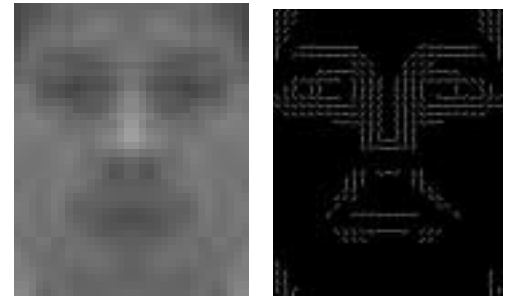


Figure 3. Typical orientation template (right image) $\mathbf{V}_M$ for upright frontal face detection. The orientation information is visualized as vectors without arrows. The left image shows the grey value average face from which the model is constructed.

field $\mathbf{V}_M$ is calculated from the average face.

During training an ordered partitioning of N_M sub-models is searched. The ordering is due to the discrimination power of the sub-models. For finding an optimal partitioning and the associated matching thresholds we also use a nonface-database of 50000 non-face orientation fields of size 27×32 . The optimization process starts with selecting the best feature from the model. The best feature then is the first feature of the most discriminating sub-model. Then the second-best feature is searched and so on. This is done until the best model has F_M / N_M features, where F_M is the number of active feature vectors in the model. The threshold for the sub-model is then chosen to maximize detection on this stage. The whole procedure repeats until the partitioning is

found. Note, that each feature is only used once.

For face detection the model $\mathbf{V}_M$ is shifted over the image, and at each image position (x, y) the similarity between the model and the underlying image patch is calculated. This is done by matching the sub-models successively starting with the most discriminating one. The matching score values from each stage are accumulated and compared against a threshold which is also determined during training. If the threshold criterion is missed at a certain stage the current position is discarded.

3.3. Distance Metrics

In general the orientation matching process can be described as a convolution-like operation

$$c = \sum_i d_l(\mathbf{V}_M^n, \mathbf{V}_I(x, y)), \quad (9)$$

where c is a similarity score between a sub-image at position (x, y) and a the n -th sub-model. The function $d_l()$ calculates the local distance between two single orientation vectors. In the present system the function $d_l()$ is designed to give a low value for a high similarity and a high value for poor similarity. A simple but very efficient distance measure is based on the Sine of the local angular distance. Edge strength is only used to determine whether the distance should be calculated at a certain point. The local distance is written:

$$d_l = \begin{cases} \frac{\sin(|\arg\{\mathbf{v}_i\} - \arg\{\mathbf{v}_m\}|)}{1} & \text{if } |\mathbf{v}_i|, |\mathbf{v}_m| > 0 \\ \text{else} & \end{cases} \quad (10)$$

For computationally efficient execution the distance should be realized as table lookup.

4. Candidate Verification

The face candidates detected by the EOM are passed on to a candidate verification module which constitutes the last stage of the matching pipeline, see Fig. 2. We use an appearance based approach first described in [16]. SNoW is an abbreviation for *Sparse Network of Winnovs* and is a learning network of linear units connecting binary input features with two output nodes. The input features are calculated from the histogram equalized input image patch of size 20×20 . There are 102400 ($= 20 \times 20 \times 256$) binary features, and exactly one of them, namely feature number $256 * (20 * x + y) + I(x, y)$ is set to 1 for each pixel in the image. Since we have 400 pixels in the sub-window, exactly 400 features are set to one and the rest is set to zero. The feature vectors are connected with weights to two output targets, called *face target* and non face target. The decision finally is made from the value of the output node t using a

threshold θ_t . The network or more precisely the weights are trained using the so called Winnov update rule as described in [16] with a database of 12000 faces, produced by slightly scaling and rotating a set of 1000 aligned face patches, and 25000 non faces. The set of non-faces is obtained using a bootstrapping approach similar to that described in [12].

5. Coarse-to-Fine Image Scan

In order to find faces of different sizes we use a multi-resolution image pyramid. The size ratio R_p between two resolution levels is set to $R_p = 1.25$. Each level of the pyramid is searched in a coarse to fine manner. This is done using a sparse grid with a grid step $D_{g^0} = 6$. So we ap-

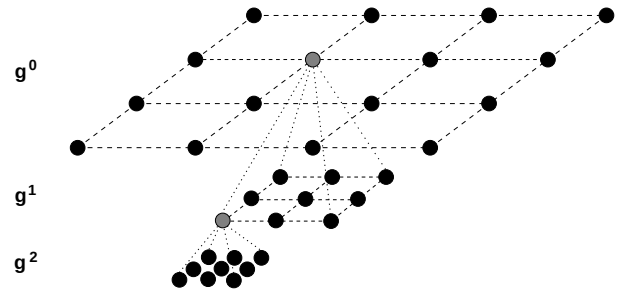


Figure 4. Example of the hierarchical grid search. At first the image is searched at a grid g^0 with resolution $D_{g^0} = 6$. All locations with values below a threshold T_{gs} (light grey tone) are starting points of a local search on grid g^1 with $D_{g^1} = 3$. At the finest grid g^2 a 3×3 neighborhood is evaluated.

proximately evaluate only 2.8% of all possible locations at this coarse level. Each grid point which passes the matching pipeline and has a similarity value below a threshold T_{gs} is a starting point for a local refinement of the search. In the next step we search at the points of a finer structured local grid around the starting point of the coarse grid. The grid step $D_{g^1} = 3$, which means that every 9th point is evaluated. Finally each point of that second step below T_{gs} is evaluated in a 3×3 neighborhood. The grid step of this third level is set to $D_{g^2} = 1$. Figure 4 illustrates the whole procedure. Points with a light grey tone are meant to be below the threshold T_{gs} . The speedup is controlled by the threshold T_{gs} which determines the number of grid points that are starting points for a finer analysis in their neighborhood. The higher T_{gs} is the less likely it will be that we have to perform fine grid searches. On the other hand it is clear that we are about to miss many true face positions then and the detection rate will decrease.

6. Experiments and Results

6.1. Still Images

The image database used for testing the capabilities of the face detector consists of 18704 images, each showing at least one upright face. The whole data set consists of 6 different parts as listed in table 1. The images from the

Name	#Images	#Faces	Comment
Test Set I	10775	10775	complex background
Test Set II	1468	1468	complex, poor quality
Internet	2409	2409	complex background
CMU [12]	179	505	complex, poor quality
BioID [6]	1521	1521	complex background
M2VTS [9]	2352	2352	simple, high quality
Total	18704	19030	

Table 1. Dataset used for the evaluation of our algorithm.

first five sets vary a lot with respect to image quality, lighting conditions, background and the size of the faces displayed. The last set contains the images from the M2VTS [9] database which have an uniform background and the size of the displayed faces is alike. All faces are annotated manually so that the ground truth for the database is known for automatic performance measurement. The CMU [12] and BioID [6] database are publicly available on the Internet and are often used for the benchmarking of face detection algorithms.

We regard a face as detected if the measured face position and size do not deviate from the true values for more than 30%. The face position and size are defined by the position of the eyes and the reference for the relative displacement measurement is the true eye distance. All the results reported in table 2 are obtained using a resolution pyramid with a size factor between two levels of $R_p = 1.25$. The model used for EOM is that displayed in Fig. 3. It consists of $F_M = 720$ orientation vectors. The model is divided into $N_M = 40$ sub-models.

6.2. Comparison with other still image detectors

For benchmarking, we can look at the best detection rates and corresponding number of false detects reported on the CMU database so far. The computation time given in the third place corresponds to an image of size 320×240 and in the fourth place the computer platform is given. Rowley [12] obtained a detection performance of (90.1%, 167,

¹The images of the CMU database where scaled up by 1.25 before detection.

Dataset	Detection	#False	False-Rate
Test Set I	97.76%	169	1/13953306
Test Set II	90.67%	88	1/3650815
Internet	96.39%	48	1/15101254
M2VTS	96.50%	14	1/34324584
BioID	95.67%	64	1/5201107
CMU ¹	87.80%	120	1/1233333
Total	96.63%	503	1/11327348

Table 2. Detection results of the proposed detection system on different databases.

140 sec, SGI O2 175Mhz) using Neuronal Networks for detection. Our system using SNoW only achieves (92.1%, 131, 4 sec, Athlon 1000 Mhz). Schneiderman [13] reports (94.4%, 65, 90sec, Pentium II 450 MHz) for his Bayes approach. The detection rate of the two-stage approach of this paper is a little lower on the CMU dataset (see Tab. 2), but we need only 40 msec to analyze one image of size 320×240 on a Athlon, 1000 Mhz and 50 msec for an image of size 384×288 . Thus even if we take into account that part of the mentioned results are obtained using slower hardware our method is considerably faster.

6.3. Video

For our video experiments we use a video stream of spatial resolution 320×240 which was searched at frame rate of 15 fps on an Athlon 1000MHz computer. This was done by applying the still image face detector, as described in this paper, to each single video frame in real-time. In the video system we apply smoothness constraints on the motion of the detected faces. Thus we can rule out most of the remaining false detects of the static system because false detects usually show up randomly in time and space. The whole video-system will be described elsewhere in near future. To evaluate the performance of the video system we analyze 1/2 hour of German television program. As it was too expensive to annotate all of the 27000 frames manually we only assessed the system qualitatively by overlaying the detections in the video stream. We encountered very little false detects approximately one in every 20 seconds and an estimated detection rate of above 90% on the appearing frontal faces.

7. Summary and Conclusions

In this work we show that edge orientation information is a powerful local feature for object detection. We propose a matching-pipeline which relies solely on edge orientation

features and performs a candidate verification at the last stage of the pipeline. The verification utilizes grey-features and a SNoW Classifier. The combination of a pipelined matching approach and a coarse-to-fine image scan leads to a very fast face detector. This static detection algorithm achieves detection rates of more than 96% on a data base of 18704 of still images having only 503 false detects. The system is able to analyze a video stream of spatial resolution 320×240 at frame rate on an Athlon 1000MHz computer. This means that the static face detector needs less than 40 msec (50 msec for size 384×288) to process a single frame. The introduction of motion constraints further lowers the number of false detects in a video system. We also provide realistic performance figures obtained on a very large database.

Acknowledgments

The work described here was supported by the German Federal Ministry of Education and Research (BMBF) under the project EMBASSI.

References

- [1] S. Ben-Yacoub, B. Fasel, and J. Lttin. Fast face detection using mlp and fft. In *Audio- And Video-Based Biometric Person Authentication*, pages 31–36, Washington, D.C., 1999.
- [2] M. Bichsel. *Strategies of Robust Object Recognition for the Automatic Identification of Human Faces*. PhD thesis, Eidgenössische Technische Hochschule Zurich, Zurich, 1991.
- [3] M. Burl and P. Perona. Recognition of planar object classes. In *Proc. CVPR'96*, 1996.
- [4] K. R. Castleman. *Digital Image Processing*. Prentice Hall, 1996.
- [5] B. Fasel. Fast multi-scale face detection. IDIAP-COM 4, IDIAP, 1998.
- [6] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz. Robust face detection using the hausdorff distance. In *Audio- and Video-based Biometric Person Authentication (AVBPA'2001)*, pages 90–95, Halmstad, Sweden, June 2001.
- [7] S.-H. Lin, S.-Y. Kung, and L.-J. Lin. Face recognition/detection by probabilistic decision based neural network. *IEEE Transactions on Neural Networks, Special Issue on Artificial Neural Networks and Pattern Recognition*, 8(1), January 1997.
- [8] D. Maio and D. Maltoni. Real-time face location on gray-scale static images. *Pattern Recognition*, 33:1525–1539, September 2000.
- [9] K. Messer, J. Matas, J. Kittler, J. Luetttin, and M. G. Xm2vtsdb: The extended m2vts database. In *Second International Conference on Audio- and Video-based Biometric Person Authentication*, pages 71–77, 1999.
- [10] B. Moghaddam and A. Pentland. Maximum likelihood detection of faces and hands. In *International Workshop On Automatic Face- and Gesture-Recognition*, pages 122–128, Zurich, Switzerland, June 26–28 1995.
- [11] E. E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and application. Technical Report 1602, Massachusetts Institute of Technology, March 1997.
- [12] H. A. Rowley. *Neural Network-Based Face Detection*. PhD thesis, Carnegie Mellon University, Pittsburgh, 1999.
- [13] H. Schneiderman. *A Statistical Approach to 3D Object Detection Applied to Faces and Cars*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2000.
- [14] K. K. Sung. *Learning and Example Seletion for Object and Pattern Detection*. PhD thesis, Massachusetts Institute of Technology, January 1996.
- [15] M.-H. Yang, N. Ahuja, and D. Kriegman. Face detection using mixtures of linear subspaces. In *International Conference on Face and Gesture Recognition*, pages 70–76, 2000.
- [16] M.-H. Yang, D. Roth, and N. Ahuja. A snow-based face detector. In *Advances in Neural Information Processing Systems 12 (NIPS 12)*, pages 855–861. MIT Press, 2000.