Face detection based on generic local descriptors and spatial constraints

Veronika Vogelhuber, Cordelia Schmid

Face Detection based on Generic Local Descriptors and Spatial Constraints

Veronika Vogelhuber Cordelia Schmid
INRIA Rhône-Alpes & GRAVIR-CNRS, 655 av. de l'Europe, 38330 Montbonnot, FRANCE

Abstract

In this paper we present an algorithm for face detection that is based on generic local descriptors (e.g. eyes). A generic descriptor captures the distribution of individual descriptors over a set of samples (training images). This distribution is assumed to be a Gaussian mixture model and is learnt using the minimum description length principle (MDL). A descriptor of an unknown image may then be classified as one of the generic local descriptors. Robustness is achieved by using spatial constraints between locations of descriptors. Experiments show very promising results.

1 Introduction

Face detection is a well known problem in computer vision with many potential applications, such as structuring image databases or surveillance. For recognition of a particular person the detection of a face in an image is the first necessary step before comparing the face to a given face gallery.

In this paper we derive a method for learning a face representation which is based on generic local descriptors. These descriptors characterize eyes, nose and corners of the lips. Each generic descriptor is represented by a Gaussian mixture model which is computed from individual descriptors of sample images. The estimation of the Gaussian mixture is realized by an Expectation-Maximization (EM) algorithm combined with a Minimum-Description-Length (MDL) algorithm. The MDL algorithm allows to select the number of Gaussians. Moreover, spatial constraints are used and their variability is described by a Gaussian model.

This face representation is used to detect faces in an unknown image. We first search for instances of generic local descriptors. We then verify the spatial constraints between these descriptors (for example between eyes and mouth has to be a nose).

1.1 Previous Work

Different approaches for detecting faces exist. These approaches differ in the face representation as well as in the learning algorithm. Most approaches use a global representation of the face, that is they learn the distribution of the face patch [5, 6, 9]. The face patch is of high dimensionality and the choice of the learning algorithm is therefore important. Moghaddam and Pentland [5] for example deal with the problem of high dimensionality by first applying an eigenspace decomposition to the set of face patches. They then use a Gaussian mixture model to learn the distribution of the most significant eigenvectors. However, the number of Gaussians used is not determined automatically. Sung and Poggio [9] learn the distribution of the high dimensional feature vectors by means of a few view-based face and non-face model clusters. The necessary number of cluster is determined manually. Osuna et al. [6] use a support vector machine to learn the distinction between face and non-face patches. No a priori information such as the number of cluster is necessary.

Global face representations present several disadvantages. Firstly, they are not robust to occlusions. They are also not robust to deformations as the face patch is rigid. Furthermore, a global representation requires learning in the presence of high dimensional data which is a difficult problem. It is also more difficult to make a global representation invariant to image transformations. More recent approaches therefore use a local face representation [2, 7, 10]. Burl et al. [2] detect facial features by correlation and use spatial constraints between these features for verification. Their face representation is not learnt but selected manually. This is the main different with our approach where facial features as well as spatial constraints are learnt from a set of samples. Wu et al. [10] build two fuzzy models to describe skin color and hair color, respectively. These models are used to extract regions in an "unknown" image. These regions are then compared with a head-shaped model using a fuzzy pattern-matching method. Rikert et al. [7] describe a face image by a set of local feature vectors (Gabor filters). Feature vectors of the training images are clustered and the most discriminant clusters are used to describe the face class. They don't use any spatial constraints for verification.

1.2 Overview of the paper

Section 2 explains how to learn a face representation. In section 3 we explain how to detect faces in an "unknown" image. Experimental results for detecting faces are presented in section 4. In section 5 we discuss the potential extensions of this work.

2 Learning a face representation

A face is represented by a set of generic local descriptors and spatial relations between these descriptors. Figure 1 illustrates our face representation. The generic descriptors characterize left and right eye, nose as well as left and right corner of the lips.

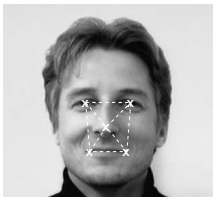

Figure 1. Our face representation is based on generic local descriptors and spatial constraints.

2.1 Local generic descriptors

Given a training set of face images we select for each image the same characteristic locations. One of our characteristic locations is for example the middle of the right eye (cf. figure 1). At each of these locations we compute a local descriptor, that is a vector of local image descriptions. We have used Gaussian derivatives to describe the image locally. The set of descriptors for a characteristic location is used to calculate the distribution of a statistical variable which represents a generic local descriptor and captures its different aspects. Figure 2 illustrates this idea for the right eye.

This distribution is estimated using a Gaussian mixture model. Such a model is appropriate, as there exist a lot of distinctions between the eyes of different persons (for example an Asiatic eye and a European one). A Gaussian mixture model allows to combine very different descriptors by building different sub-classes. Learning of the sub-classes is not supervised and is described in the next section.

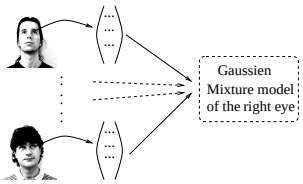

Figure 2. Learning a generic local descriptor.

2.2 Estimation of the Gaussian mixture model

The Gaussian mixture model for a generic descriptor D_i is defined by $D_i = \{D_{ij}\}_{j=1..K_i}$ where K_i is the number of Gaussians of the ith descriptor. Each D_{ij} is defined by $(w_{ij}, \mu_{ij}, \Sigma_{ij})$ where w_{ij} is a weighting component, μ_i the mean value and Σ_{ij} the covariance matrix. The number of Gaussians K_i and the Gaussian model M_i are iteratively determined using a Minimum-Description-Length (MDL) algorithm. The idea is to try different numbers of Gaussians and different Gaussian models. At each iteration, that is for a Gaussian model M_i and a number K_i of Gaussians, the parameters $w_{ij}, \mu_{ij}, \Sigma_{ij}$ are estimated using the Expectation-Maximization (EM) algorithm. The selection of the optimal model is realized by punishing the more complex models and by taking into account the goodness of fit. The log-likelihood measures this goodness of fit. The function to be minimized is

$$h(D_{ij}) = -L(D_{ij}) + \frac{1}{2} card(D_{ij}) \ln(n)$$

where $L(D_{ij})$ is the log-likelihood measure, $card(D_{ij})$ the number of free parameters and n is the number of descriptors used to estimate D_{ij}. Details on the MDL algorithm are given in [1]. In this paper the number of Gaussians K varies from 1 to 10 and we have used 5 different Gaussian models. These models differ in the covariance matrix :

$M_1 : \Sigma_{ij} = \sigma_{ij}^2 I$. I is the identity matrix.

$M_2 : \Sigma_{ij} = \sigma_{ij}^2 C$. C is a diagonal matrix with $\det(C) = 1$.

$M_3 : \Sigma_{ij} = \sigma_{ij}^2 C_{ij}$. C_{ij} are diagonal matrices with $\det(C_{ij}) = 1$.

$M_4 : \Sigma_{ij} = \sigma_{ij}^2 C$. C is a symmetric matrix with $\det(C) = 1$.

M_5 : without any restrictions, the so called full covariance matrix. It is the most complex model.

EM algorithm The EM algorithm is used for finding maximum likelihood parameter estimates when there is missing or incomplete data. In our case, the missing data is the Gaussian cluster to which the descriptor belongs and the incomplete data is the labeling of each descriptor to its Gaussian. We estimate values to fill in for the incomplete data (the "E Step"), compute the maximum likelihood parameter estimates using this data (the "M Step"), and repeat until the log-likelihood increases by less than 1 % from one iteration to the next (if this does not happen within 500 iterations, a stop is forced). A detailed description of the EM algorithm is given in [3].

2.3 Spatial configurations

Spatial configurations are used as constraints to increase the detection performance. It is well known that such constraints are very powerful [8]. They are represented by angles and length ratios between the locations of the generic local descriptors. We represent angles and length ratios by Gaussian variables with mean and standard deviation. This allows to capture variability due to morphological differences.

3 Detecting faces

The face representation learnt in section 2 is used to detect faces in unknown images. Figure 3 illustrates the different steps of our face detection algorithm (for correspondence color - generic local descriptor see figure 4). The first step is to compute a descriptor for each location of the "unknown" image and to decide which is the most likely generic descriptor. Such a classification will not reject any descriptor. We have therefore added a "refusing point step"

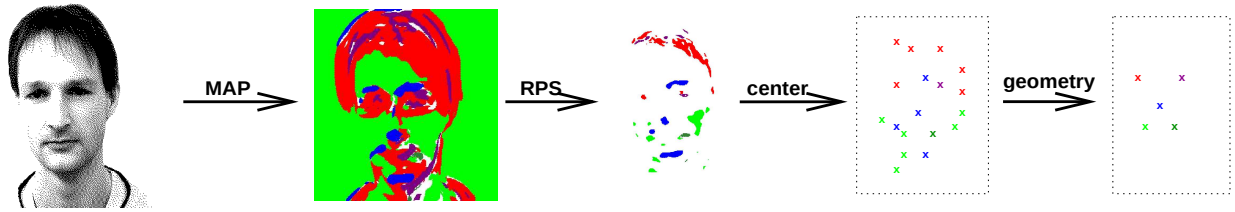

Figure 3. Illustration of our face detection algorithm.

(RPS) which rejects points which are too far from the distribution. For the remaining points we then compute connected components and the centers of these components. The final step is the verification of spatial constraints between the center points. Only configurations which verify the spatial constraints are kept. Note that in figure 3 only one configuration verifies the spatial constraints.

Figure 4. Correspondence between colors and generic local descriptors.

3.1 Classification of a descriptor

A descriptor X of an "unknown" image is classified as one of the generic local descriptors by the maximum a posteriori principle (MAP).
The probability of a d-dimensional descriptor X for a generic descriptor D_i is :

$$p(X|D_i) = \sum_{j=1}^{K_i} w_{ij}\, p(X|(\mu_{ij}, \Sigma_{ij}))$$

where K_i is the number of Gaussian, w_{ij} the weighting factors of each Gaussian and $p(X|(\mu_{ij}, \Sigma_{ij})) = \frac{1}{(2\pi)^{d/2}|\Sigma_{ij}|^{1/2}} \exp^{-\frac{1}{2}(X-\mu_{ij})^t \Sigma_{ij}^{-1}(X-\mu_{ij})}$.
The probability of a generic descriptor D_i given X is

$$p(D_i|X) = \frac{p(X|D_i)\, p(D_i)}{p(X)}$$

$p(D_i)$ are assumed to be equal. The most likely generic local descriptor is

$$\hat{D} = \arg\max_{D_i} p(D_i|X)$$

3.2 Refusing point step (RPS)

The refusing point step (RPS) rejects points which are too far from the distribution. This is measured by the Mahalanobis distance between the descriptor X and the most likely generic descriptor $\hat{D}$. As one generic descriptor D_i may consist of several components D_{ij}, the distance is defined as the sum of the distances between the descriptor X and components D_{ij}:

$$d_M(X, D_i) = \sum_j \sqrt{(\mathbf{X} - \mu_{\mathbf{ij}})^{\mathbf{t}} \mathbf{\Sigma_{ij}^{-1}} (\mathbf{X} - \mu_{\mathbf{ij}})}$$

For each generic local descriptor we determine a distance threshold. This threshold is learnt from the training set by calculating the average of the distances and adding a scaled factor of the variance.

3.3 Computation of connected components

As illustrated in figure 3 to each generic descriptor corresponds an region of the "unknown" image. Each region is extracted by a connected component algorithm and is then represent by its center. Moreover, very small regions are rejected.

3.4 Verification of spatial constraints

The spatial configuration of the centers of the connected components is compared to the learnt spatial constraints to reject or accept an "unknown" image as containing a face. If a face is detected, constraints allow to localize the face position. The verification of the constraints is based on statistical analysis. To allow for missing detection or occlusion we do not require all spatial constraints to be verified.

4 Experiment results

4.1 Implementation details

For our experiments we have used five generic local descriptors. They are computed at the following locations : right and left eye, nose and right and left corner of the lips. The spatial relations between these descriptors are used for verification. Image locations in the set of training images are selected manually.

The individual local descriptors used here are a set of Gaussian derivatives named "local jet" by Koenderink and van Doors [4]. Derivatives are computed up to third order and for 3 different scales : $\sigma \in \{3, 5, 7\}$.

4.2 Experimental setup

The Achermann face database (University of Bern, Switzerland) used for our experiments contains 300 images of 30 different persons (cf. figure 5). We have used 150 images of 15 different persons to estimate our face representation. The remaining 150 images are used as test images.

4.3 Results of the learning step

The distribution of our generic local descriptors is described by Gaussian mixture models. We have used on average 4 Gaussians and the preferred Gaussian model was M_4.

Figure 5. Images of the Achermann face database.

4.4 Results for face detection

Our algorithm correctly detects faces in all of the 150 test images of the Achermann face database. Figure 3 and figure 6 show two examples of our detection algorithm. Note that the refusing point step is crucial to reduce the complexity of the geometric verification.

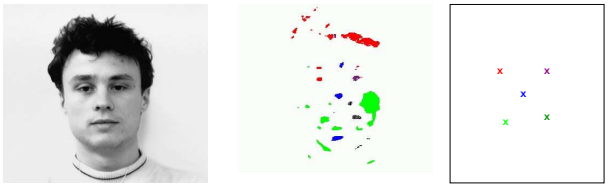

Figure 6. Result of our face detection algorithm. The image on the left is the test image. The middle image displays the detected generic descriptors after the RPS step. On the right are shown the points which verify the spatial constraints.

We have also tested our algorithm on face images not present in the database as well as on non-face images. The result for a face image is shown in figure 7. Note that the eyes have been detected correctly, but left and right eye have been inverted. This is due to the visual similarity of left and right eye. We have included such an inversion in our spatial constraints. Tests for non-faces images have been conducted on the Columbia database. This database contains 1440 images of 20 3D objects. No face has been detected for these images. An example is presented in figure 8.

5 Conclusion

We have derived and implemented a face model based on generic local descriptors and spatial constraints between these descriptors. Each of the generic descriptors is represented by a Gaussian mixture model. This allows to capture the variability of individual descriptors over a set of samples (training images). Results are significantly improved using a Refusing Point Step which allows to reject points far from the probabilistic variable. Spatial constraints allow further verification and make our method robust.

A straightforward extension is to include others generic descriptors, that is other characteristic face locations. This will increase the robustness of our method. The use of color

Figure 7. Result of our face detection algorithm. The image on the left is the test image. The middle image displays the detected generic descriptors after the RPS step. On the right are shown the points which verify the spatial constraints.

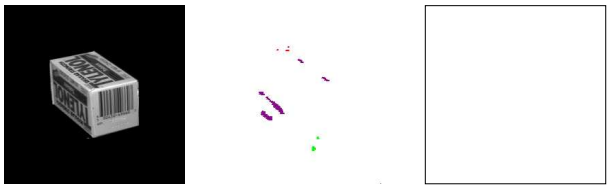

Figure 8. Result for a non-face image. The image on the left is the test image. The middle image displays the detected generic descriptors after the RPS step. On the right are shown the points which verify the spatial constraints. No face is detected.

is also a promising extension. Furthermore, we could extend the algorithm to other types of objects (profile faces, animals etc.).

References

[1] C. Biernacki. *Choix de Modèles en Classification*. PhD thesis, Université de technologie de Compiègne, 1997.

[2] M. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *ECCV*, 1998.

[3] G. Celeux and G. Govaert. Gaussian Parsimonious Models. *Pattern Recognition*, 28(5):781–783, 1995.

[4] J. Koenderink and A. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[5] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *PAMI*, 19(7):696–710, 1997.

[6] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *CVPR*, 1997.

[7] T. Rikert, M. Jones, and P. Viola. A cluster-based statistical model for object detection. In *ICCV*, 1999.

[8] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *PAMI*, 19(5):530–534, 1997.

[9] K. Sung and T. Poggio. Example-based learning for view-based human face detection. *PAMI*, 20(1):39–51, 1998.

[10] H. Wu, Q.Chen, and M. Yachida. Face detection from color images using a fuzzy pattern matching method. *PAMI*, 21(6):557–563, 1999.