© *In Proc.* International ECCV 2002 Workshop on Biometric Authentication,
Springer, Lecture Notes in Computer Science, LNCS-2359, pp. 103–111,
Copenhagen, Denmark, June 2002.

Genetic Model Optimization for Hausdorff Distance-Based Face Localization

Klaus J. Kirchberg, Oliver Jesorsky, and Robert W. Frischholz

BioID AG, Germany
{k.kirchberg,o.jesorsky,r.frischholz}@bioid.com,
WWW home page: http://www.bioid.com

Abstract. In our previous work we presented a model-based approach to perform robust, high-speed face localization based on the Hausdorff distance. A crucial step during the design of the system is the choice of an appropriate edge model that fits for a wide range of different human faces. In this paper we present an optimization approach that creates and successively improves such a model by means of genetic algorithms. To speed up the process and to prevent early saturation we use a special bootstrapping method on the sample set. Several initialization functions are tested and compared.

1 Introduction

Face localization is a fundamental step in the process of face recognition. Its aim is to decide whether there is a face in a given image and, in the positive case, to determine the coordinates of the face. The accuracy of the detected face coordinates has a heavy influence on the recognition performance.

In [5] we presented a method for robust frontal face detection based on the Hausdorff distance. This algorithm uses a predefined edge model of the human face to find face candidates in the image. While it is possible to use a simple ellipse as a model, the detection performance can be improved by using a more detailed model. However, it must still represent a wide variety of faces.

In this paper we follow a genetic algorithm approach to generate a face model from scratch and to optimize it based on a fairly large database of sample images. A coding scheme for binary edge models is presented along with the corresponding genetic operators. We describe a bootstrapping optimization framework that speeds up the process by successively adapting the subset of evaluation samples.

Several experiments prove the performance of the system and compare different initialization strategies.

2 Hausdorff Distance-Based Face Detection

This section gives a brief overview of the underlying face detection algorithm. A more detailed description can be found in [5].

The face detection problem can be stated as follows: given an input image, decide whether there is a face in the image or not. If a face was found, return the face coordinates inside the image. In our case these are the coordinates of the left and right eye centers, which is sufficient if the problem is restricted to frontal view faces in images of constant aspect ratio.

For simplicity, we concentrate on the task of finding a single face in an image. The extension to finding multiple faces is straightforward.

We use an edge-based method for finding faces. Therefore we first calculate an edge magnitude image with the Sobel operator. The relevant edge feature points are extracted by a locally adaptive threshold filter to compensate variable illumination. We assume that this procedure will produce a characteristic arrangement of segmentation points in the facial area.

Based on the typical layout that strongly depends on the segmentation steps, we use a face model which itself consists of a set of feature points and can be represented as a binary image.

The feature points of the face model are chosen in a way that the pattern stored in the model is somehow similar to the typically observed patterns in the images' face area.

To detect a face, the model is superimposed over the image at several discrete positions. At each position the similarity between the translated model and the covered part of the image is calculated. A face is considered to be located at the position yielding the highest similarity between model and covered image part. The procedure is illustrated in figure 1. Note that the model can be scaled to allow detecting faces of different sizes.

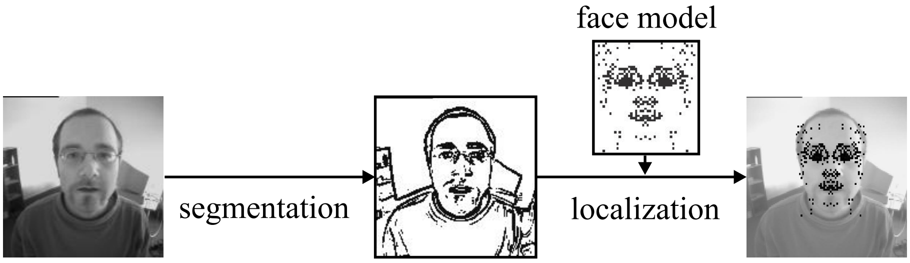


Fig. 1. Face finding procedure

An efficient yet powerful method to calculate the similarity of two binary images is the Hausdorff distance [7], a metric between two point sets. We use a slightly adapted measure, called the (directed) *modified Hausdorff distance* (MHD) [2] to calculate the similarity between the image and the model. Given the two point sets $\mathcal{A}$ and $\mathcal{B}$ and some underlying norm $|| \cdot ||$ on the points, the

MHD is defined as

$$h_{\mathrm{mod}}(\mathcal{A},\mathcal{B}) = \frac{1}{|\mathcal{A}|} \sum_{a\in\mathcal{A}} \min_{b\in\mathcal{B}} \|a-b\| \ . \tag{1}$$

With the two-dimensional point set $\mathcal{A}$ representing the image and $T_p(\mathcal{B})$ representing the translated and scaled model with transformation parameters p, the formula

$$d_{\hat{p}} = \min_{p\in\mathcal{P}} h_{\mathrm{mod}}(\mathcal{A}, T_p(\mathcal{B})) \tag{2}$$

calculates the distance value of the best matching position and scale. The parameters of the corresponding transformation are represented by the parameter set $\hat{p}$.

To make an efficient implementation possible, we use a discrete grid for the model point positions. A model can then be represented by a binary image where white pixels represent the model points. The resolution of this image has to be high enough to be able to represent enough detail but has to be as low as possible to minimize computation time for both the localization procedure and the model optimization process. We used a 45 × 47 model grid which has turned out to be a good trade-off.

One of the major problems of the Hausdorff distance method is the actual creation of the face model. While a simple "hand-drawn" model will be sufficient for the detection of simple objects, a general face model must cover the broad variety of different faces.

3 Genetic Model Optimization

The task of finding a well-suited model for Hausdorff distance based face localization can be formulated as a discrete global optimization problem. An exhaustive search would produce the optimal result (with respect to a given sample set), but due to the exponential complexity it is not computationally feasible. In the broad area of global optimization methods, Genetic Algorithms (GA) form a widely accepted trade-off between global and local search strategy. They were chosen here for they are well-investigated and have proven their applicability in many fields.

Since their invention by Holland [4], Genetic Algorithms have become a standard solution approach for multi-dimensional global optimization problems. We use the algorithm and terminology of the Simple Genetic Algorithm (SGA) described by Goldberg [3].

To formulate our face finding problem as a genetic algorithm, we have to do the genotype coding of the face model, define a fitness function and have to set some more parameters (population size, crossover method etc).

3.1 Genotype Coding

The genotype coding of the face model is done fairly straightforward by a two-dimensional binary genome. We presume that the average face model is symmetric along the vertical axis, which is not exactly true for a single face but sufficient for our purposes. Thus, only the left half of the model is coded in the genome.

3.2 Fitness Function

The fitness function assigns a real-valued number to a given model. This value must reflect the performance of the face localization algorithm with a certain model. During reproduction phase of the GA this value determines an individual's probability to survive and produce offspring. To rate a specific model, it is tested on a set of sample face images. This sample set must be both large enough to be representative and also small enough to allow fast evaluation of the fitness function.

We define the fitness value of a model as the ratio of found faces to the overall number of faces in the set. A face is said to be found if some distance measure between true position and found position is below a certain threshold. We use here the accuracy measure d_{eye} introduced in [5]. Let d_l and d_r denote the distances between the true eye centers $C_l, C_r \in \mathbb{R}^2$ and the expected eye positions, respectively. Then the rating distance is defined as

$$d_{\text{eye}} = \frac{\max(d_l, d_r)}{\|C_l - C_r\|} \tag{3}$$

with the euclidean norm $|| \cdot ||$.

The threshold that defines a face as found was set to $\hat{d}_{\text{eye}} = 0.12$ for the optimization process, which means that we allow a shift of 12% with respect to the distance between the true right and left eye. The true eye positions were marked manually.

We used a set consisting of 1362 face images for evaluation. To speed up evolution, we choose a subset of 80 images. This subset selection is updated every five generations. The process is shown in figure 2.

```
initialize population
do while not converged
   evaluate population on complete set
   build new evaluation set with best model
   run GA for five generations on evaluation set
end do
```

Fig. 2. Optimization process

When a new evaluation set of face images is built, the localization is performed on the whole set of images with the best model of the current population.

For each image we record d_{eye} and sort the images by this distance. The new evaluation set is then compiled from 40 out of the 200 top-ranking images and 40 out of the 200 images with the lowest rating. This makes the GA learn to find new faces while not "forgetting" the others.

3.3 Selection, Crossover, Mutation, Population Size

In the choice of the other genetic operators we mostly follow the suggestions in Goldberg's book [3].

Selection is done by the Roulette wheel scheme, which means each individual's selection probability is directly proportional to its fitness.

As crossover operator we use the natural extension of the one-point crossover. Its function in the two-dimensional case is depicted in figure 3

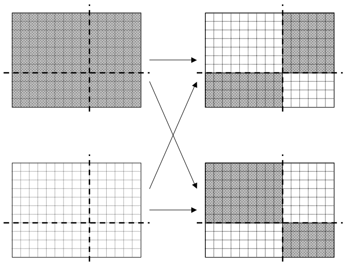


Fig. 3. One-point crossover operator for 2d binary genomes

Mutation is represented by random bit flip with a probability of 0.00025 for each bit.

The population size is constant and is set to 50 individuals.

3.4 Initialization

Another important decision is how to initialize the population. In our experiments we used three different initializations:

- blank model
- average edge model
- hand-drawn model

These initialization options are described in more detail in the next chapter.

4 Experiments

The described approach was first tested on three different setups. They all differed in the method used to initialize the first generation. The purpose of this test was to check the influence of the initialization on the convergence behavior.

In the first method, the population was initialized with random points, each one having a 5% probability of being set. Further genetic parameters were:

crossover rate	0.9
mutation rate	0.00025
population size	50

Some models generated by the GA in this run are shown in figure 4.

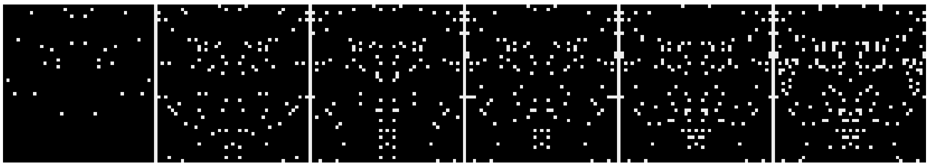


Fig. 4. Models from the randomly initialized GA run

For the second setup, an average edge map was generated from a set of sample images. In the initialization step, the model points were randomly set with a probability proportional to the value of the corresponding point in the average edge map. Some models from this run are shown in figure 5.

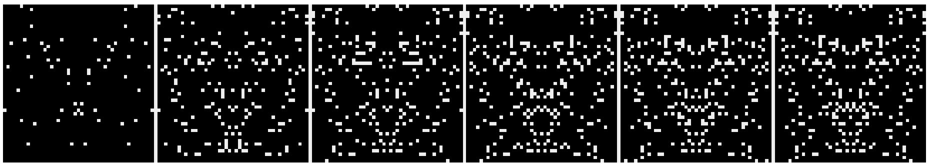


Fig. 5. Models from the average edge map-initialized GA run

The third run was initialized with a hand-drawn face model and 5% of the bits flipped. Figure 6 shows the hand-drawn model and some other models from this run.

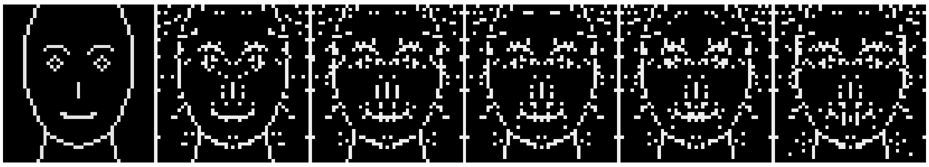
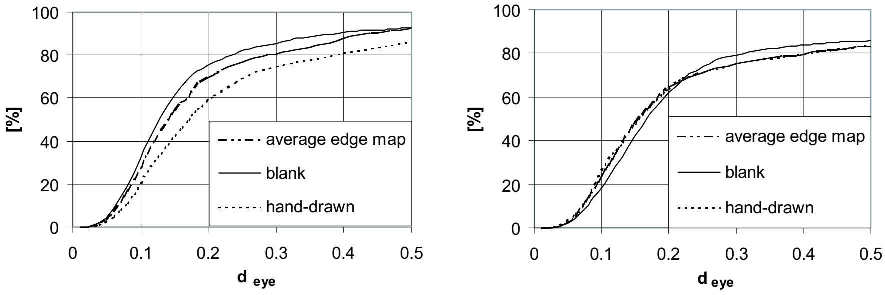


Fig. 6. Models from the hand-drawn initialized GA run

The resulting models were tested on the XM2VTS [6] data set and the BIOID face test set, which is publicly available at [1]. The first mentioned contains 2360, the second 1521 gray level images, each of them showing a single face.

The results for the three models on both sets are summarized in Figure 7. The figure shows the distribution function of the detection results rated using the same method as used by the fitness function described in the previous section (see eq. 3).



(a) (b)

Fig. 7. Distribution function of relative eye distances for the XM2VTS (a) and the BIOID face data set (b) for the best models of the three runs

Regarding a value of $d_{\text{eye}} = 0.25$ more than 80% of the faces are found in the XM2VTS test set. According to the definition of d_{eye} a value of 0.25 equals half the width of an eye. This has shown to be a reasonable threshold for robust face recognition.

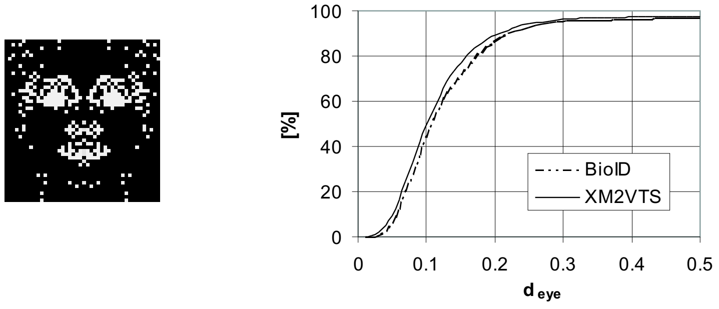
The results on the BIOID test set are a little poorer because this set has been recorded under a larger variety of illumination and face scale and therefore implies a harder problem for face detection systems.

The model gained from blank initialization performs best on both data sets. Therefore we used the blank initialization method to start a second optimization on a larger image database, also using more generations than in the first evaluations.

With the resulting model that is shown in figure 8, together with the belonging distribution functions, the localization performance could be increased up to 92.8% on the BIOID and 94.2% on the XM2VTS dataset (again considering a maximum allowed error of 0.25 relative eye distance).

In comparison, the detection rate for the hand-drawn model itself is 62.3% on the XM2VTS database.

Due to the lack of a common performance measurement for face detection algorithms it is hard to compare different approaches. For example, Smeraldi et al [8] reported a detection rate of 91% for a SVM approach on a subset of 349 images from the M2VTS database. They allowed an absolute tolerance of 3 pixel for eyes and mouth positions.



(a) (b)

Fig. 8. Resulting model (a) and corresponding distance distribution functions for the XM2VTS and the BIOID face set (b).

5 Conclusions

One of the major problems in model-based face detection is the creation of a proper face model. We have presented a genetic algorithm approach for obtaining a binary edge model that allows localization of a wide variety of faces with the Hausdorff search method.

The experiments showed that the GA performs better when starting from scratch than from a hand-drawn model. With this method, the localization performance could be improved to more than 90% compared to roughly 60% for the hand-drawn model.

Genetic Algorithms are a powerful tool that can help in finding an appropriate model for face localization. The presented framework led to a model that performed considerably better than a simple hand-drawn model.

Face localization can be improved by a multi-step detection approach that uses more than one model in different grades of detail. Each of these models can then be optimized separately. This does not only speed up the localization procedure but also produces more exact face coordinates.

References

[1] BioID face database. http://www.bioid.com/research/index.html.

[2] M.P. Dubuisson and A.K. Jain. A modified Hausdorff distance for object matching. In *ICPR94*, pages A:566–568, Jerusalem, Israel, 1994.

[3] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison Wesley, 1989.

[4] John H. Holland. *Adaption in Natural and Artificial Systems.* The University of Michigan Press, Ann Arbor, 1975.

[5] Oliver Jesorsky, Klaus J. Kirchberg, and Robert W. Frischholz. Robust Face Detection Using the Hausdorff Distance. In Josef Bigun and Fabrizio Smeraldi, editors, *Audio- and Video-Based Person Authentication - AVBPA 2001*, volume 2091 of *Lecture Notes in Computer Science*, pages 90–95, Halmstad, Sweden, 2001. Springer.

[6] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTSDB: The extended M2VTS database. In *Second International Conference on Audio and Video-based Biometric Person Authentication*, pages 72–77, March 1999.

[7] W. Rucklidge. *Efficient Visual Recognition Using the Hausdorff Distance*, volume 1173 of *Lecture notes in computer science*. Springer, 1996.

[8] F. Smeraldi, N. Capdevielle, and J. Bigun. Facial features detection by saccadic exploration of the Gabor decomposition and Support Vector Machines. In *Proceedings of the 11th Scandinavian Conference on Image Analysis - SCIA 99, Kangerlussuaq, Greenland*, volume I, pages 39–44, June 1999.