CNS 186 3/19/02

Feature-Based Face Detection Using Gaussian Derivative Filters

California Institute of Technology
Pasadena, CA 91125 – USA
minhtam@caltech.edu
March 2002

Problem formulation 3/5
Accomplishments 37/40
Write up / Presentation 5/5

more background would have been useful

$\boxed{45/50}$

$\boxed{A}$ for project

Christof Koch

Abstract

A feature-based face detection method using second derivative of Gaussian filters is presented. Filter responses are processed to determine regions of interest. Face localization is then performed by matching image features with a model. The global detection scheme is tested with real images with varied conditions. The results are finally presented and the performance of the detection evaluated.

Quite nice algorithm here, Minh Tam, not bad. And it seems to work quite nicely on ~~some~~ many faces. I hope you learned to appreciate your brain – and the tremendous computational job it is performing – much more now. Your choice of particular techniques, e.g. thinning, VOF, HOF etc does appear a bit idiosyncratic. Can you think of a more general method to generate face detector algorithms? Or dog, cat, car or license-plate detectors?

Introduction

In computer vision, face detection still remains a field of research. Lots of attempts have been made in order to automatically detect and recognize human faces in natural scenes. The variety of background, conditions and the non-rigidity of human faces make this computation difficult. So it is a challenging task to find an efficient model which could cope with all these variations.

Several techniques for computational face detection exist nowadays. Some emphasize the speed of computation, others the robustness of the detection. And generally, most of them in order to be efficient required to be performed with respect to some constraints (i.e. faces in frontal view).

Whereas neural network-based and feature matching-based algorithm are well spread ?? techniques, this paper will present a different approach. The detection scheme proposed in this project is a feature-based method using oriented Gaussian derivative filters. The use of these filters is controversial since their response may generally present more false alarms than previous cited techniques. However, the advantage is that they can highlight all the features which have a common direction, by blurring all the edges which have a different orientation from the one of the filters. Moreover, as human faces present variations in shape, expressions and illumination, it is necessary to find an approach that avoid if not all at least one of these problems. The use of Gaussian derivative filters could also be justified since their response are not very sensitive to a change of the face shape, expression or even illumination.

This paper will deal with front-view faces mainly for optimization and tuning purpose. However all the algorithms can be easily extended to cope with varied scenes. By using several filters with appropriate orientation, it will be shown that good performance can be achieved.

A. Method

The method adopted is a coarse-to-fine approach. The algorithm is divided into 2 major parts. The first one is critical since it performs image enhancement so that main face features can be extracted, whereas the second one processes the data coming from that extraction.

The algorithm scheme is the following.

First part:
- Image filtering
- Feature extraction

Second part:
- Vectorization and grouping
- Determining regions of interest
- Face localization

Each step will now be presented in detail.

1. Image filtering

By using Gaussian derivative filters, the challenge is to be able to highlight all the common features present in faces without highlighting the noisy background. Instead of using multi-scale Gaussian derivative filters, only 2 second derivative of a Gaussian shown in figures 1 and 2, were chosen. The reasons are multiple. First, filtering is a long process, so by limiting the number of filters the computation time will be reduced. Second, features that Gaussian derivative filters with large variance can detect, can also be detected by Gaussian derivative filters with small variance. Third, the larger the variance, the more blurred the filter response.

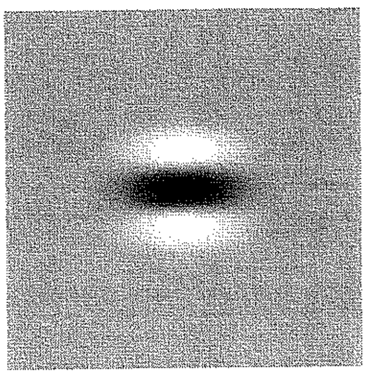

Figure 1 : ***second derivative Gaussian filter oriented horizontally***

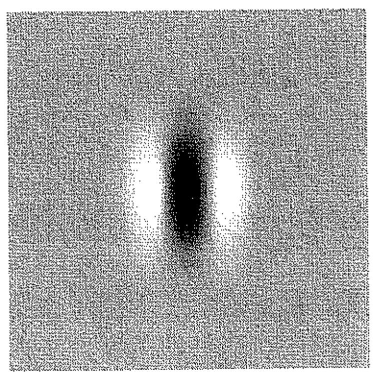

Figure 2 : second derivative Gaussian filter oriented vertically

The horizontal Gaussian derivative filter present a variance of 2 in its long axis whereas has a variance of 0.8 in its short axis. The vertical Gaussian derivative filter is just the same as the horizontal one but is rotated of 90°. Several Gaussian filters with different variance were tested but those with values mentioned above gave better results by producing detailed responses. Figure 3 and 4, 5 show the original image and the corresponding output of the filters, which is evaluated as follows:

Let G be the output of a filter response, Yi the intermediate output and Y the final output.

$Y_1 = G^2$: Computes the energy of the response

$Y_2 = \log(Y_1)$: Effects of strong edges are lessened (in some work, to eliminate the effects of strong edges, a threshold is used instead. By doing so, it is possible to also eliminate face features, therefore a logarithm is applied to the output)

$Y = (Y_2)^2$: Reduces the effect of noise

$Y = (\log G^2))^2 = 4(\log G)^2$

Figure 3 : *original image*

Figure 4 : *image filtered with the horizontal-oriented filter (HOF)*

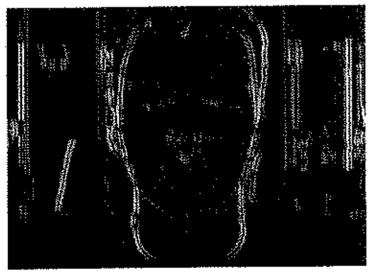

Figure 5 : *image filtered with the vertical-oriented Gaussian (VOF)*

In a general manner, if the orientation of a subject is unknown, several oriented filters have to be used sequentially. For each of them, by using a perpendicular second Gaussian derivative filters, the boundary of the face can be brought to the fore.

2. Feature extraction

This step extracts face features with high energy. The idea is to perform a non-maxima-suppression algorithm. As interesting features can have low energy, it is necessary to perform that suppression locally.

A simple algorithm is used. For each pixel of the preceding outputs, the number of connected pixels which has a higher intensity is counted. If this number is less than a certain threshold, then the pixel is said to be a local maximum (white pixels), else it is switched off (black pixels). After several tests on different images, this threshold was chosen to be 2 connected pixels for better results.

The next step is to apply a skeletonization algorithm on the outputs. The aim is to remove any extra pixels while preserving the properties and features of the images. From that, lines will be obtained and will be processed for the vectorization.

Figure 6 and 7 show the output of the non-maxima-suppression algorithm followed by the skeletonization.

Figure 6 : *features extracted for the HOF*

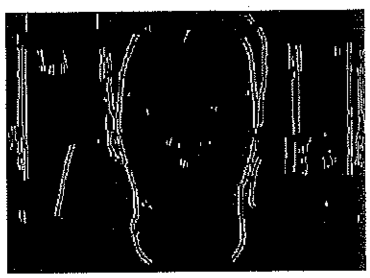

Figure 7 : *features extracted for the VOF*

3. Vectorization and grouping

The vectorization process converts lines into vectors and groups them according to their closeness. Closeness is calculated by sampling 2 lines and comparing relative distances between sample points. If for some sample points, this distance turns out to be below a certain threshold, then these vectors (or lines) or classified in the same group. The distance used is the Euclidean distance. Clusters of vectors are thus formed as shown figure 8 and 9.

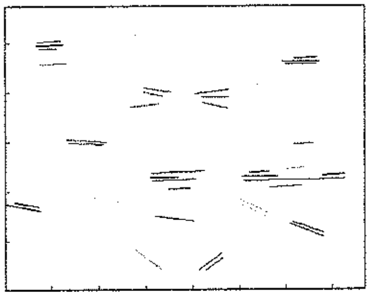

Figure 8 : ***clusters of vectors for the HOF***

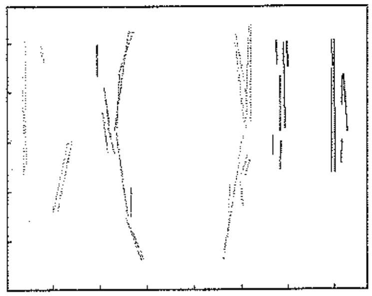

Figure 9 : ***clusters of vectors for the VOF***

Then, pairs of clusters are found by comparing similarities between clusters. A pair is found when the angle formed by their centroid does not exceed 10 degrees. This is a perceptual value, found to be appropriate after several tests on different scenes. Figure 10 et 11 show these pairs.

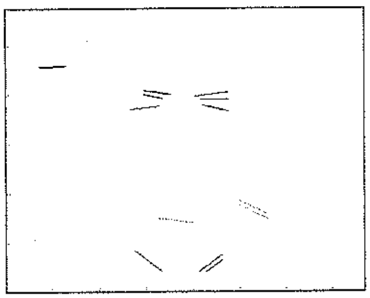

Figure 10 : ***pairs of clusters for the HOF***

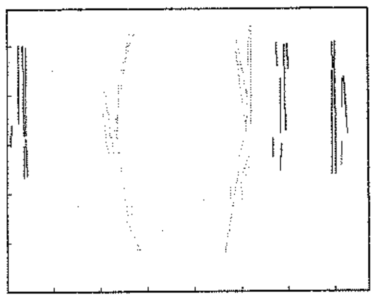

Figure 11 : ***pairs of clusters for the VOF***

Identifying pairs has 2 purposes. First, it is used to localize possible *eyes* (output from HOF). Second, it divides space into smaller regions of interest (output from VOF).

4. Regions of interest

By combining the results obtained so far, specific regions of interest can be determined. As *vertical* pairs (cluster pairs resulting from the VOF) delimit a region in space. An assignment is performed. *Horizontal* pairs (cluster pairs resulting from the HOF) are assigned to a *vertical* pair if and only if they are inside the region delimited by the *vertical* pair. Besides those *horizontal* pairs, single clusters (opposed to a pair) are also assigned the same way provided that they are situated in the same region of space. Then regions where a face may be found are localized.

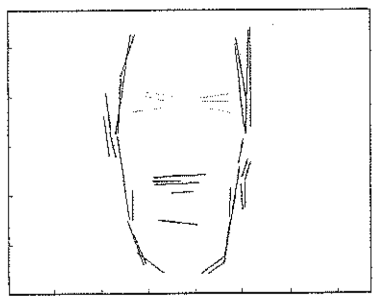

Figure 12 : ***regions of interest***

5. Face localization

The final step of the algorithm is to check the presence of a face in each of the regions of interest found previously. For that purpose a model is built and correspondences are looked for. The model is shown opposite and consists in estimating relative position of main features of a face such as the eyes, the mouth, the boundary of the head from 4 directions. The distance estimation and error tolerance made are perceptual and comes from tests as well. If it is possible to match all the features of the model with some clusters in a region of interest then a face is detected. So the output of the algorithm is not a probability to have a face, but simply is a yes/no answer in so far as it comes from a decision tree (left-right boundary detected -> eye detected -> mouth detected).

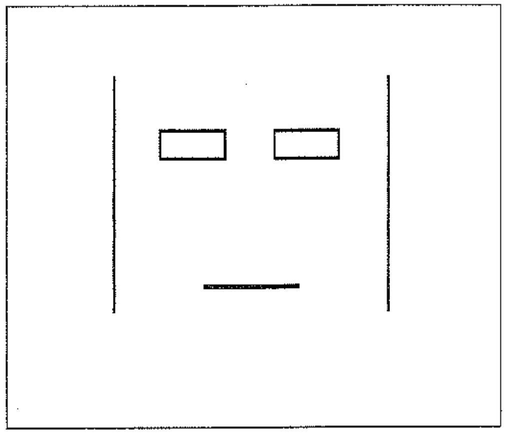

Figure 13 : ***model used for the detection***

B. Results

To test the algorithm, the database from BioID Technology Research was used. The dataset consists of 1521 gray level images with a resolution of 384x286 pixel. Each one shows the frontal view of a face. All parameters were tuned so that they can detect faces corresponding to that size. Tests were run without changing the parameters of the algorithm.

Variation of illumination and face expressions

For these cases, all faces were correctly detected. However the eye position is only an approximation since the algorithm only performs an estimation of the face features.

Variation of human faces

The following faces were correctly detected.

The following faces were not detected.

On the average, the algorithm detected 12 cases correctly out of 15, which represents 80 % correct cases.

Discussion

By using Gaussian derivative filters, the algorithm can cope with problems related to variations in illumination and face expressions, in so far as edges of features are only involved. Even with a noisy background the algorithm succeeds most of the time in correctly detecting the faces.

From the results, it can be seen that the algorithm fails when there are many edges around the face. This is the reason why women are often not detected because of edges present in the hair. The other reason why the algorithm fails is because it has been tuned to detect specific range of faces. If the face features are smaller than expected then the algorithm will fail to detect them because of inappropriate thresholds.

Although all the tested images had the same size, the computation time varied from 1 minute to 3 minutes depending on the complexity of the scene. The more detailed the scene, the longer the computation time, the larger the probability of error.

Nevertheless, by construction, the algorithm is flexible in the sense that it is modular. Each part of it can be improve separately to obtain better results since the steps are sequential.

One of the main advantage of this algorithm is that it does not require any training sequence to perform, whereas one of its main drawback is that it is sensitive to the tuning of the thresholds.

In conclusion, improvements have still to be made however the obtained results are promising.

References

[1] K. C. Yow and R. Cipolla, "Human Face Location", CUED/F-INFENG/TR 239, Oct. 1995.

[2] S. Ravela and A. R. Hanson, "On Multi-Scale Differential Features for Face Recognition", Dept. of Computer Science, University of Massachusetts, Amherst.

[3] O. Jesorsky, K. J. Kirchberg and R. W. Frischholz, "Robust Face Detection Using the Hausdorff Distance", Third International Conference on Audio- and Video-based Biometric Person Authentication, Springer, Lecture Notes in Computer Science, LNCS-2091, pp. 90-95, Halmstad, Sweden, s6-8 June 2001