Component Fusion for Face Detection in the Presence of Heteroscedastic Noise

Binglong Xie[1,3], Dorin Comaniciu[1], Visvanathan Ramesh[1],
Markus Simon[2], and Terrance Boult[3]

[1] Siemens Corporate Research
755 College Road East, Princeton, NJ 08540, USA
{binglong.xie, dorin.comaniciu, visvanathan.ramesh}@scr.siemens.com
[2] Information and Communication Mobile, Siemens AG
Haidenauplatz 1, 81617 Muenchen, Germany
markus.simon@siemens.com
[3] Computer Science and Engineering Department, Lehigh University
19 Memorial Drive West, Bethlehem, PA 18015, USA
tboult@cse.lehigh.edu

Abstract. Face detection using components has been proved to produce superior results due to its robustness to occlusions and pose and illumination changes. A first level of processing is devoted to the detection of individual components, while a second level deals with the fusion of the component detectors. However, the fusion methods investigated up to now neglect the uncertainties that characterize the component locations. We show that this uncertainty carries important information that, when exploited, leads to increased face localization accuracy. We discuss and compare possible solutions taking into account geometrical constraints. The efficiency and usefulness of the techniques are tested with both synthetic and real world examples.

1 Introduction

It is known that component-based face detection can yield better performance than global approaches when pose and illumination variations and occlusions are considered[9, 5, 6, 11]. While pose and illumination significantly change the global face appearance, since the components are smaller than the whole face, they are less prone to these changes. The component detectors can accurately locate the face components as well. This information should be used to register and normalize the face to a "standard" one, which is appropriate for face recognition. Also, component-based methods can be used to build a detector that can handle partial occlusions [6, 11]. Component-based methods have been also successfully used in other areas, such as people detection [8].

In [5], Heisele et al present a component-based face detector with a two-level hierarchy of Support Vector Machine (SVM) classifiers [2]. The face components are detected independently with the trained SVMs at the first level, and at the second level, a single SVM checks if the geometric locations of the components comply with a face. However, only the largest responses from the component detectors are used when checking the validity of the geometry. Also, SVMs are slow and it should be very challenging to employ them in real-time systems.

In [10], Viola and Jones employ 4 types of rectangular features and use AdaBoosting [4] to automatically build the strong classifier from feature-based weak classifiers. They compute the integral image (similar to the summed area table in [3]) to accelerate the computation of features. Their paper reports a high detection rate, a low false detection rate and the boosted face detector works in real-time.

This paper introduces a new framework for component fusion in the context of the face detection task. Fusion relies on modeling the noise as heteroscedastic and is constrained by a geometric face model. To achieve real-time performance, we employ AdaBoosting when training component detectors. However, our framework is open to various types of component detectors, e.g., SVMs.

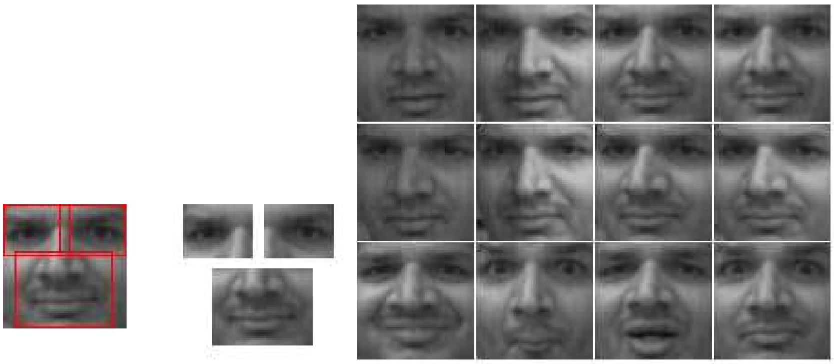

Fig. 1. Left: the components of a face. The left eye component and right eye component are 36 by 28 pixels. The lower face component is 52 by 40 pixels. **Right:** Face examples. The first row and the second row are frontal and turning left faces respectively with 4 different illumination settings.The third row shows faces with different expressions.

2 Component Detectors

In our work, we use 3 components for a face. All the faces are aligned to a 64 by 64 pixel image. We then use three rectangles to cut 3 components, left eye, right eye and lower face, as shown in Figure 1 (left).

Our face database has 1862 faces. The images were taken with 5 poses (frontal, turning left, turning right, tilting up, and tilting down) and 4 illumination conditions(dark overall, lighting from left, lighting from right, and bright overall). There are also some faces with different expressions. Figure 1 (right) shows some examples from the database. We collected more than 6000 pictures as negative examples for detector training.

The AdaBoosting theory states that by adding weak classifiers one can obtain better strong classifiers. However in practice this might not be true, since the weak classifiers are often correlated. To deal with this issue, we use a modified AdaBoosting method that trains the component detectors such that the trained strong classifier is verified to be empirically better at each boosting step.

3 Component-Based Face Model

Suppose we have a probabilistic face model, where each component position has some uncertainty. With the uncertainties, the face model is flexible to describe a variety of possible faces. Assuming Gaussian distributions, in the face model we have a set of 2D points with means $\mathbf{m}_i$, and covariance matrices C_i, $i = 1, 2, ..., N$, where N is the number of components. The face model is a constraint that the components should comply with the geometrical configurations, e.g., the components should not be too far away (see Figure 2 (left)).

The face model is trained from known face examples. We know the exact locations of the components in each training face example, so we can estimate the mean and covariance matrix of each component from these locations.

4 Component Fusion

4.1 Problem Formulation

After the component detectors are trained, we scan the input image to get the component confidence maps, $A_i(\mathbf{x})$, $i = 1, 2, ..., N$, where $\mathbf{x}$ is the location in image, and N is the number of components. We assume confidence map $A_i(\mathbf{x})$ is normalized across all the components.

With the face model $\{\mathbf{m}_i, C_i\}_{i=1,2,...,N}$, the overall face likelihood is:

$$L = \prod_{i=1}^{N} \left[A_i(\mathbf{x}_i) \frac{1}{|2\pi C_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \left(\mathbf{x}'_i - \mathbf{m}_i\right)^T C_i^{-1} \left(\mathbf{x}'_i - \mathbf{m}_i\right) \right) \right] \tag{1}$$

where $\{\mathbf{x}'_i\}$ are rigidly transformed from $\{\mathbf{x}_i\}$ into the face model space, subject to rotation, translation and scaling.

Note the simple maxima of individual component detector responses are not necessarily best choices for component locations under face model constraints. Our goal is to find the best component localization $\{\mathbf{x}_i\}$ with maximal L. We can do exhaustive search with all $A_i(\mathbf{x})$ but that is too expensive.

Since the shape of $A_i(\mathbf{x})$ is often smooth and Gaussian-like, we use a Gaussian shape to approximate it. In other words, the underlying noise model is assumed heteroscedastic, i.e., the noise is both anisotropic and inhomogeneous. We can identify the local maximum as $s_i = A_i(\boldsymbol{\mu}_i)$, where $\boldsymbol{\mu}_i$ is the location of maximum and considered the center of the Gaussian shape. Matei [7] gives a non-parametric method to estimate the "covariance" matrix Q_i in a area B around $\boldsymbol{\mu}_i$:

$$Q_i = \frac{\sum_{\mathbf{x}\in B}\left[A_i(\mathbf{x})\,(\mathbf{x}-\boldsymbol{\mu}_i)\,(\mathbf{x}-\boldsymbol{\mu}_i)^T\right]}{\sum_{\mathbf{x}\in B} A_i(\mathbf{x})} \tag{2}$$

Then the confidence map can be rewritten:

$$A_i(\mathbf{x}) = s_i \frac{1}{|2\pi Q_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\,(\mathbf{x}_i-\boldsymbol{\mu}_i)^T\, Q_i^{-1}\,(\mathbf{x}_i-\boldsymbol{\mu}_i)\right) \tag{3}$$

Therefore,

$$\ln L = \sum_{i=1}^{N}\left[\ln s_i - \frac{1}{2}\ln\left(|2\pi Q_i|\,|2\pi C_i|\right)\right] - \frac{1}{2}d^2 \tag{4}$$

where

$$d^2 = \sum_{i=1}^{N}|\mathbf{x}_i'-\mathbf{m}_i|^2_{C_i} + \sum_{i=1}^{N}|\mathbf{x}_i-\boldsymbol{\mu}_i|^2_{Q_i} \tag{5}$$

$$= \sum_{i=1}^{N}(\mathbf{x}_i'-\mathbf{m}_i)^T C_i^{-1}(\mathbf{x}_i'-\mathbf{m}_i) + \sum_{i=1}^{N}(\mathbf{x}_i-\boldsymbol{\mu}_i)^T Q_i^{-1}(\mathbf{x}_i-\boldsymbol{\mu}_i) \tag{6}$$

In order to maximize L one should minimize d^2. When d^2 is computed for an observation, L or $\ln L$ can be thresholded to make a detection or rejection decision.

4.2 Least Square Fitting

For the beginning, let us simplify the problem so that we only have fixed-point face model $\{\mathbf{m}_i\}$ and fixed-point observations $\{\mathbf{x}_i\}$, for example, taking the means of the face model and maxima of the confidence maps.

Suppose we find the scaling factor s, the rotation R and translation $\mathbf{x}_0$, so that an observation point $\mathbf{x}$ can be mapped to a point $\mathbf{x}'$ in model space.

$$\mathbf{x}' = sR(\mathbf{x}-\mathbf{x}_0) \tag{7}$$

where, the rotation matrix R is a function of θ:

$$R = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \tag{8}$$

Our goal is to minimize the sum of squared error d^2 by choosing the right s, R and $\mathbf{x}_0$:

$$d^2 = \sum_{i=1}^{N}|\mathbf{x}_i'-\mathbf{m}_i|^2 \tag{9}$$

By taking the partial derivatives of Equation (9) with respect to θ, s and $\mathbf{x}_0$, and setting them to zeros (denoting $\mathbf{m}_i = (m_i, n_i)^T$ and $\mathbf{x}_i = (x_i, y_i)^T$), we get the solution:

$$\theta = \arctan \frac{\sum m_i \sum y_i - \sum n_i \sum x_i - N \sum (m_i y_i - n_i x_i)}{\sum m_i \sum x_i + \sum n_i \sum y_i - N \sum (m_i x_i + n_i y_i)} \tag{10}$$

$$s = \frac{N \sum \left(\mathbf{m}_i^T R \mathbf{x}_i\right) - \left(\sum \mathbf{m}_i^T\right) R \left(\sum \mathbf{x}_i\right)}{N \sum \left(\mathbf{x}_i^T \mathbf{x}_i\right) - \left(\sum \mathbf{x}_i^T\right)\left(\sum \mathbf{x}_i\right)} \tag{11}$$

$$\mathbf{x}_0 = \frac{1}{N} \sum \mathbf{x}_i - \frac{1}{sN} R^T \sum \mathbf{m}_i \tag{12}$$

Using the above solution, we can evaluate Equation (9) to get the least square error. A smaller d^2 suggests a larger similarity between the observation and model geometrical configurations. This simple method does not take the individual component confidences into consideration, nor the heteroscedastic model of the noise.

4.3 Fitting Points to a Probabilistic Model

Within this section assume that we have a probabilistic model of 2D points $\{\mathbf{m}_i, C_i\}_{i=1,2,\ldots,N}$. We want to match the observed points $\mathbf{x}_i$ to the model. This case has been analyzed by Cootes and Taylor[1], and here is the summary.

An observation point $\mathbf{x}$ can be mapped to a point $\mathbf{x}'$ in model space:

$$\mathbf{x}' = R\mathbf{x} + \mathbf{t} \tag{13}$$

where, $\mathbf{t} = (t_x, t_y)^T$ and the scaling and rotation matrix R is

$$R = \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \tag{14}$$

Let us denote $\mathbf{a} = (a, b)^T$, and the goal is to find the best $\mathbf{a}$ and $\mathbf{t}$ to minimize the Mahalanobis distance:

$$d^2 = \sum_{i=1}^{N} |\mathbf{x}_i' - \mathbf{m}_i|_{C_i}^2 = \sum_{i=1}^{N} (R\mathbf{x}_i + \mathbf{t} - \mathbf{m}_i)^T C_i^{-1} (R\mathbf{x}_i + \mathbf{t} - \mathbf{m}_i) \tag{15}$$

Taking the partial derivatives of Equation (15) with respect to $\mathbf{a}$ and $\mathbf{t}$, and setting them to zeros, we get the solution:

$$\begin{pmatrix} \mathbf{a} \\ \mathbf{t} \end{pmatrix} = \begin{pmatrix} \sum C_i^{-1} Y_i, & \sum C_i^{-1} \\ \sum Y_i^T C_i^{-1} Y_i, & \sum Y_i^T C_i^{-1} \end{pmatrix}^{-1} \begin{pmatrix} \sum C_i^{-1} \mathbf{m}_i \\ \sum Y_i^T C_i^{-1} \mathbf{m}_i \end{pmatrix} \tag{16}$$

where, $Y_i = (\mathbf{x}_i, J\mathbf{x}_i)$ and

$$J = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \tag{17}$$

4.4 Matching Probabilistic Observations to a Probabilistic Model

With the model $\{\mathbf{m}_i, C_i\}$ and observation $\{\boldsymbol{\mu}_i, Q_i\}$, $i = 1, 2, ..., N$, we want to find the best choices of component locations $\mathbf{x}_i$, and the associated transformation $\mathbf{a}$ and $\mathbf{t}$ to minimize the combined Mahalanobis distance d^2 in Equation (6), where $\mathbf{x}'_i$ is a function of $\mathbf{x}_i$, $\mathbf{a}$ and $\mathbf{t}$ according to Equation (13). Unfortunately, it is hard to find the close form solution to this problem, because the partial derivatives are not linear with respect to $\mathbf{x}_i$, $\mathbf{a}$ and $\mathbf{t}$.

We can use two strategies to solve this optimization problem. One employs numerical optimization methods, such as Levenberg-Marquardt or Newton iterative optimization, which require iterations before convergence.

The other approximates the solution. Notice in Equation (6)

there are two terms. The first term is the Mahalanobis distance in the model space, and the second term is the Mahalanobis distance in the observation space. If we pick $\boldsymbol{\mu}_i$ as the solution for $\mathbf{x}_i$ (this is the first approximation of the solution, though very rough), and use Section 4.3 to match $\boldsymbol{\mu}_i$ to the probabilistic model $\{\mathbf{m}_i, C_i\}_{i=1,2,...,N}$, we end up a biased minimization d^2_{obs} of Equation (6) where the second term is zero. On the other hand, if we pick $\mathbf{m}_i$ as the matched points $\mathbf{x}'_i$ in the model space, and use Section 4.3 to match $\mathbf{x}'_i$ back to the observation $\{\boldsymbol{\mu}_i, Q_i\}_{i=1,2,...,N}$ (denote that the choices in the observation space are $\mathbf{x}''_i$), we end up another biased minimization d^2_{mod} of Equation (6) where the first term is zero. The real minimization must be a tradeoff between these two biased ones. The second approximation of the solution we choose is then the equal average:

$$\mathbf{x}_i = \frac{\boldsymbol{\mu}_i + \mathbf{x}''_i}{2} \tag{18}$$

Further more, we can refine the equal average to get the third approximation, the weighted average approximation, by using the Mahalanobis distances in weighting the average:

$$\mathbf{x}_i = \boldsymbol{\mu}_i + \frac{d^2_{obs}}{d^2_{obs} + d^2_{mod}}(\mathbf{x}''_i - \boldsymbol{\mu}_i) \tag{19}$$

The advantage of the approximations is that they are fast. If the solutions are close to the real minimum, the approximations are more favorable for real-time face detection systems.

5 Experiments

5.1 Synthetic Data

In this experiment, we assume a face model where the centers of the left eye, right eye and lower face components are:

$$\mathbf{m}_1 = \begin{pmatrix} 17.5 \\ -13.5 \end{pmatrix}; \mathbf{m}_2 = \begin{pmatrix} 45.5 \\ -13.5 \end{pmatrix}; \mathbf{m}_3 = \begin{pmatrix} 31.5 \\ -43.5 \end{pmatrix} \tag{20}$$

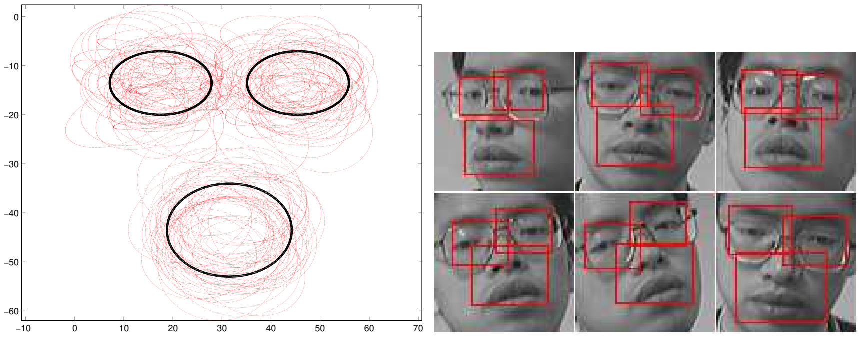

Fig. 2. Left: Face model and 50 examples of the observation distributions. The thick ellipses are the model distribution. The thin ellipses are the randomly generated observation distributions. **Right**: Real world face detection examples from a video with different poses.

and the associated covariance matrices are:

$$C_1 = \begin{pmatrix} 18 & 0 \\ 0 & 7 \end{pmatrix}; C_2 = \begin{pmatrix} 18 & 0 \\ 0 & 7 \end{pmatrix}; C_3 = \begin{pmatrix} 27 & 0 \\ 0 & 15 \end{pmatrix} \tag{21}$$

We randomly generate observation data by adding noise to both the means and covariance matrices of the components in the face model. A 0-mean Gaussian noise with a standard deviation of 4 pixels is added to both x and y directions of the means, and the covariance matrices are also added with a 0-mean Gaussian noise with a standard deviation of 3. The face model and observation examples are shown in Figure 2 (left).

Figure 3 (left) shows the d^2 computed with various approximations. The observation mean approximation has large errors. The equal average and weighted average approximations are very close to the true d^2 obtained by Levenberg-Marquardt optimization. Figure 3 (right) shows the distance error of the best match for each component in average in the observation space. We can see small but noticeable displacement errors for the equal and weighted average methods, compared to Figure 3 (left). This suggests that the when d^2 is close to the minimum, the d^2 surface is quite flat, which is because of the fact that we have relatively large covariances in the face model and observation examples.

5.2 Real World Examples

With AdaBoosting component detectors, our current face detection system runs comfortably at frame rate on a standard laptop with 640 by 480 image size. We tested our techniques with real world examples. Figure 2 (right) shows some examples of handling pose changes. We are currently evaluating the performance of the system on standard face databases.

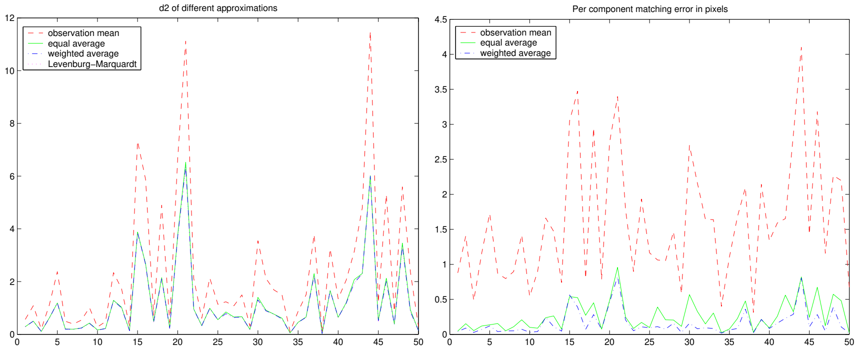

Fig. 3. Left: d^2 from various approximations. **Right**: Both the weighted and average approximations have small localization errors. The x axis is the sample number index in the above graphs.

6 Conclusion

This paper presented a statistical fusion framework for component-based face detection. The framework is tested with component face detectors trained using AdaBoosting, running in real-time. Our work is effective with both synthetic and real world experiments. We do not model the cross-component correlations and this could be part of future work.

References

1. T. Cootes and C. Taylor. Statistical models of appearance for computer vision. Technical report, University of Manchester, September 1999.
2. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
3. F. Crow. Summed-area tables for texture mapping. In *Proceedings of SIGGRAPH*, volume 18, pages 207–212, 1984.
4. Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.
5. B. Heisele, T. Serre, M. Pontil, and T. Poggio. Component-based face detection. In *CVPR01*, pages I:657–662, 2001.
6. T.K. Leung, M. C. Burl, and P. Perona. Finding faces in cluttered scenes using random labeled graph matching. In *Fifth Intl. Conf. on Comp. Vision*, June 1995.
7. Bogdan Matei. Heteroscedastic errors-in-variables models in computer vision. Ph.D. Dissertation, Rutgers, the State University of New Jersey, May 2001.
8. A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Trans. on PAMI*, 23(4), 2001.
9. C. Papageorgiou and T. Poggio. A trainable system for object detection. *IJCV*, 38(1):15–33, June 2000.
10. P. Viola and M. Jones. Robust real-time face detection. In *ICCV01*, page II: 747, 2001.
11. K. Yow and R. Cipolla. Feature-based human face detection. *IVC*, 15(9):713–35, 1997.