

# Facial Feature Detection Using A Hierarchical Wavelet Face Database

Rogério Schmidt Feris

Jim Gemmell

Kentaro Toyama

*Microsoft Research*

Volker Krüger

*University of Maryland*

January 9, 2002

Technical Report

**MSR-TR-2002-05**

Microsoft Research

Microsoft Corporation

One Microsoft Way  
Redmond, WA 98052

# Facial Feature Detection Using A Hierarchical Wavelet Face Database<sup>1</sup>

Rogério Schmidt Feris, Jim Gemmell, Kentaro Toyama

*Microsoft Research*

Volker Krüger

*University of Maryland*

## **Abstract**

WaveBase is a system for detecting features in a face image. It has a database of faces, each with a two-level hierarchical wavelet network. When a new face image is presented to the system for face detection, WaveBase searches its database for the “best face” – the face whose first level wavelet network most closely matches the new face. It also determines an affine transformation to describe any difference in the orientation of the faces. By applying the affine transformation to the position of the features in the best face, approximate feature positions in the new face are found. Second level wavelet networks for each feature are then placed at these approximate positions, and allowed to move slightly to minimize their difference from the new face. This facilitates adjustments in addition to the affine transformation to account for slight differences in the geometry of the best head and the new head. The final position of the wavelet network is WaveBase’s estimate of the feature positions. Experiments demonstrate the benefit of our hierarchical approach. Results compare favorably with existing techniques for feature localization.

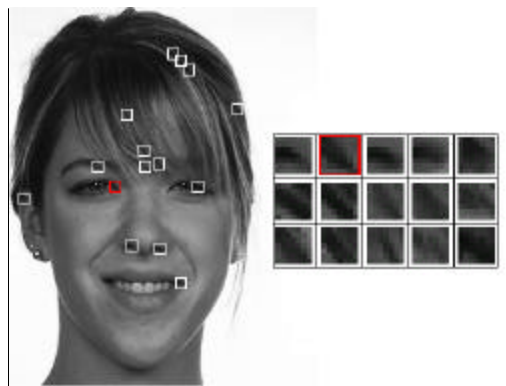
---

<sup>1</sup> A reduced version of this paper was submitted to the 5th International Conference on Automatic Face and Gesture Recognition.

# 1 Introduction

Automated initialization of feature location is a requirement of many tracking algorithms that take advantage of temporal continuity of the target. In this paper, we describe an approach to automatic initialization using hierarchical wavelet networks. Our application is facial feature localization for the purpose of initializing facial feature tracking, but the approach is applicable to other target types.

Tracking algorithms that are based on tracking sets of compact visual features, such as edge corners or small image patches, are especially difficult to initialize because each feature in itself is rarely unique – brute-force raster-scan searches of such small features will result in many possible candidates, of which only a small handful may be desirable matches (Figure 1). This suggests that features with larger support should be used, but features with larger support are also likely to be less precise in their localization, as image features far away from the feature in question bias localization. For example, many frontal face detectors [15, 16, 17] could trivially be converted to frontal eye detectors, by assuming that eyes are located at certain relative coordinates with respect to a detected face, and in fact, some face detectors overlay markers on the eyes, as evidence of a detected face [15, 16]. At a given resolution, whole faces contain more information than the eyes alone, and so the larger support of the face provides greater constraints in the search for eyes. On the other hand, the larger support also means that eye localization is imprecise because the face-eye relationship varies from image to image. Variations in facial geometry alone make it impossible to pinpoint pupils or eye corners using such a technique.



**Figure 1 Candidates for an eye corner from a face image**

We present a system, WaveBase, which solves this problem via a hierarchical search using Gabor wavelet networks (GWNs, [9]). This approach allows effective object representation using a constellation of 2D Gabor wavelets that are specifically chosen to reflect the object properties. For application to facial feature detection, we construct a training database of face images and their 2-level GWN representations. The first level GWN, representing the entire face, is used to find a face in the database that is similar to the target, and to determine an affine transformation to describe any difference in the orientation of the faces. The second level GWNs, representing each feature, are initialized in positions according to the affine transformation from the first level GWN. They are then allowed to move slightly to minimize their difference from the new face. This facilitates adjustments to account for slight differences in the geometry of the database face and the target. The final position of the child-wavelet networks is the estimate of the feature positions.

WaveBase was developed as part of the GazeMaster project, which attempts to use a combination of vision and graphics technology to attain eye contact in videoconferencing. GazeMaster uses a feature-tracker to perform pose estimation. The feature-tracker follows the corners of the eyes, the nostrils, and the mouth corners. However, initialization of the feature tracker is manual and awkward. It requires that the feature positions be known in the first frame of the video, and then it can track the features in the subsequent frames.

In order to achieve this initialization automatically, we imagined two steps: a per-user initialization that may take up to a few minutes, and a per-session initialization that takes less than one second. For the per-

user initialization, a new user would press a key while looking into the camera. After confirming that they have taken a good head-on image, WaveBase will attempt locate the features in the image. Once this is done, a GWN representation of the user's head and each feature is created and saved. The per-session initialization again requires the user to look into the camera and press a key. Facial feature detection will then be performed just as it is in the per-user step, but in this case the best head is already known (it is the saved representation of the user's head) eliminating the time required to search the database of faces.<sup>2</sup>

The remainder of the paper will use examples in which the features consist of eye corners, nostrils and mouth corners, due to WaveBase's relationship to GazeMaster. However, WaveBase does not know that it is finding these particular features, or that it is working with faces, so the technique is applicable to feature detection in general.

The remainder of the paper is organized as follows. In Section 2, we explain Gabor wavelet networks, which form the basis for our approach, and introduce hierarchies of GWNs, as well. Section 3 discusses the algorithmic details of our feature-localization system and shows results on a hand-annotated database of faces and facial features. Finally, Section 4 reviews related work.

## 2 Wavelet Networks

A wavelet network consists of a set of wavelets and associated weights, where its geometrical configuration is defined with respect to a single coordinate system. It can be further transformed by a parameterized family of continuous geometric transformations. Wavelet Networks [20] have recently been adapted for image representation [9] and successfully applied to face tracking, recognition, and pose estimation [1, 9]. Here, we apply them to the problem of feature localization.

### 2.1 Basics

The constituents of a wavelet network are single wavelets and their associated coefficients. We will consider the odd-Gabor function as mother wavelet. It is well known that Gabor filters are recognized as good feature detectors and provide the best trade-off between spatial and frequency resolution [11]. Considering the 2D image case, each single odd Gabor wavelet can be expressed as follows:

$$\mathbf{y}_{\mathbf{n}_i}(\mathbf{x}) = \exp \left[ -\frac{1}{2} (\mathbf{S}_i (\mathbf{x} - \boldsymbol{\mu}_i))^T (\mathbf{S}_i (\mathbf{x} - \boldsymbol{\mu}_i)) \right] \cdot \sin \left[ (\mathbf{S}_i (\mathbf{x} - \boldsymbol{\mu}_i)) \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right], \quad (1)$$

where  $\mathbf{x}$  represents image coordinates and  $\mathbf{n}_i = (s^x, s^y, \mathbf{q}, \mathbf{m}^x, \mathbf{m}^y)$  are parameters which compose the terms  $\mathbf{S}_i = \begin{pmatrix} s_i^x \cos \mathbf{q}_i & -s_i^y \sin \mathbf{q}_i \\ s_i^x \sin \mathbf{q}_i & s_i^y \cos \mathbf{q}_i \end{pmatrix}$ , and  $\boldsymbol{\mu}_i = \begin{pmatrix} \mathbf{m}_i^x \\ \mathbf{m}_i^y \end{pmatrix}$  that allow scaling, orientation, and translation. The parameters are defined with respect to a coordinate system that is held fixed for all wavelets that a single wavelet representation comprises.

A Gabor wavelet network for a given image consists in a set of  $n$  such wavelets  $\{\mathbf{y}_{\mathbf{n}_i}\}$  and a set of associated weights  $\{w_i\}$ , specifically chosen so that the GWN representation:

$$\hat{\mathbf{y}}(\mathbf{x}) = \sum_{i=1}^n w_i \mathbf{y}_{\mathbf{n}_i}(\mathbf{x}) \quad (2)$$

best approximates the target image.

---

<sup>2</sup> Significant changes in the user's appearance, e.g. due to growing or cutting a beard, or a change in environment that leads to very different lighting conditions may require the per-user initialization to be repeated.

## 2.2 Compression as Learning

Assuming we have a single training image,  $\mathbf{I}^t$ , that is truncated to the region that the target object occupies, we learn GWN representation parameters as follows:

1. Randomly drop  $n$  wavelets of assorted position, scale, and orientation, within the bounds of the target object.
2. Perform gradient descent (e.g., via Levenberg-Marquardt optimization [13]) over the set of parameters  $\{w_i, \mathbf{n}_i\}$ , to minimize the difference between the GWN representation and the training image:

$$\arg \min \left\| \mathbf{I}^t - \sum_{i=1}^n w_i \mathbf{y}_{\mathbf{n}_i}(\mathbf{x}) \right\|^2. \quad (3)$$

3. Save the geometric parameters,  $\mathbf{n}_i$ , and the weights,  $w_i$ , for all  $n$  wavelets. Let  $\mathbf{v} = [w_1 \ w_2 \ \dots \ w_n]^T$  denote the concatenated vector of weights.

Step 2 minimizes the difference between the GWN representation of the training image and the training image itself. A reasonable choice of  $n$  results in a representation that is an effective encoding of the training image. One advantage of the GWN approach is that one can tradeoff computational effort with representational accuracy, by increasing or decreasing  $n$  (see Figure 2).

We note here that if the parameters for a wavelet,  $\mathbf{y}_{\mathbf{n}_i}(\mathbf{x})$ , are fixed, then its coefficient,  $w_i$ , on an image,  $\mathbf{I}$ , can be computed directly from the image by taking the inner product of the wavelet's dual,  $\tilde{\mathbf{y}}_{\mathbf{n}_i}(\mathbf{x})$ , with  $\mathbf{I}$  where  $\langle \mathbf{y}_{\mathbf{n}_i}(\mathbf{x}), \tilde{\mathbf{y}}_{\mathbf{n}_j}(\mathbf{x}) \rangle = \mathbf{d}_{i,j}$ , for  $i, j$  with respect to the GWN representation,  $\mathbf{Y}$  (see [1, 9] for more details).

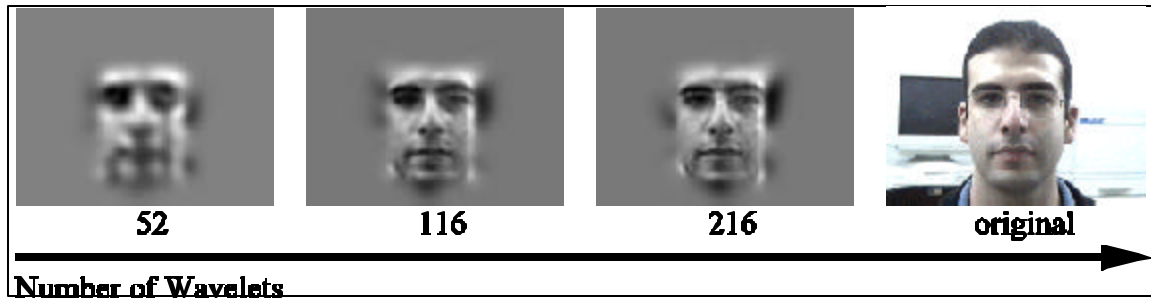


Figure 2 Facial reconstruction with 52, 116 and 216 wavelets compared with the original image.

## 2.3 Localization

GWNs may be further transformed by a bijective geometric transformation,  $\mathbf{T}_{\mathbf{a}}$ , parametrized by  $\mathbf{a}$ , such that the GWN representation  $\mathbf{Y}(\mathbf{x})$  is mapped to  $\mathbf{Y}(\mathbf{T}_{\mathbf{a}}^{-1}(\mathbf{x}))$ . Localization of an object represented by  $\mathbf{Y}$  can then be seen as finding the optimal parameters,  $\mathbf{a}$ , of  $\mathbf{T}$  that allow  $\mathbf{Y}(\mathbf{T}_{\mathbf{a}}^{-1}(\mathbf{x}))$  to best reconstruct a portion of the image. Given a hypothesized set of parameters,  $\mathbf{a}$ , one way to determine whether it performs a good reconstruction is to compute  $\mathbf{Y}(\mathbf{T}_{\mathbf{a}}^{-1}(\mathbf{x}))$  and then compute the  $L_2$ -norm between it and the image (within  $\mathbf{Y}$ 's support region).

If the transformation  $\mathbf{T}$  is linear, the transformation can be “pushed back” to the individual wavelets,  $\mathbf{y}_{\mathbf{n}_i}(\mathbf{x})$  that make up the GWN representation. In this case, we do not have to laboriously reconstruct images to compute the  $L_2$ -norm. Instead, given a hypothesized set of parameters,  $\mathbf{a}$ , we can now transform the constituent wavelets accordingly, compute  $\mathbf{w}$ , their weights on the image,  $\mathbf{I}$ , and directly compute  $L_2$ -norm as follows:

$$\|\mathbf{I} - \mathbf{T}_{\mathbf{a}}^{-1}(\mathbf{x})\|^2 = \|\mathbf{v} - \mathbf{w}\|_{\Psi}^2 = \sum_{i,j} (v_i - w_i)(v_j - w_j) \langle \mathbf{y}_{n_i}, \mathbf{y}_{n_j} \rangle. \quad (4)$$

where  $v_i = \langle \mathbf{I}(\mathbf{x}), \mathbf{Y}_{n_i}(\mathbf{T}_{\mathbf{a}}^{-1}(\mathbf{x})) \rangle$ .

The terms  $\langle \mathbf{Y}_{n_i}, \mathbf{Y}_{n_j} \rangle$  are independent of  $\mathbf{a}$  up to a scalar factor, thus further facilitating on-line computations.

## 2.4 Hierarchical Wavelet Networks

Hierarchical wavelet networks are best envisioned as a tree of wavelet networks. Each node of the tree represents a single wavelet network together with its coordinate system. Each child node is associated with a fixed local coordinate system that is positioned, scaled, and oriented with respect to its parent. Child nodes represent wavelet networks in themselves. Relationships between the wavelet parameters in a parent node and a child node are not fixed a priori. That is, this hierarchical structure only imposes direct constraints on the relative positioning of coordinate systems between nodes, not on the wavelets themselves.

Structured in this way, wavelet networks occurring higher (toward the root) in the tree constrain their child-node wavelet networks in such a way as to avoid significant geometric deviations while offering enough flexibility that local distortions can still be modeled.

## 3 Implementation

WaveBase was developed to provide initialization for GazeMaster's 3D facial pose tracker. The tracking system (described in [2, 3]) uses nine tracked features on a subject's face – inner and outer corners of both eyes, three points on the nose, and two mouth corners. Each feature is tracked by a combination of low-resolution, sum-of-absolute-differences template matching and iterative sub-pixel tracking of small image patches [7, 10]. Both feature-tracking algorithms require accurate initial localization of the nine features, per subject, in order to track. Previously, these points were initialized manually for each subject; by implementing the algorithms described above, we were able to automate this process for a range of subjects. In the remaining sequences, facial features will refer to eight of these features (not including the nose tip – this is estimated as the midpoint between nostrils, because local image information is insufficient for accurate localization).

### 3.1 Training Database

Our training database includes the following for each face:

- the original image,
- a bounding box for each facial feature,
- a bounding box for the whole face,
- a GWN representation of the region inside the face bounding box, and
- a GWN representation of the region inside each facial feature bounding box.

Faces are well-represented with a GWN of 52 wavelets, as shown in Figure 2 (Cf. the Gabor jet approach, which would require many more wavelets). Each facial feature is represented by a GWN comprising nine wavelets. Figure 3 shows a face image from the database, with the level-one GWN representation of the face, and the level-two GWN representation of the features.

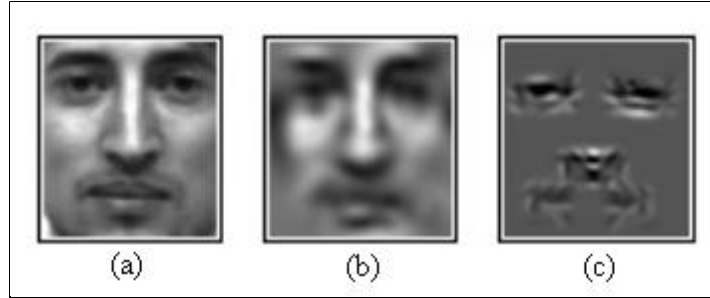


Figure 3 Training database: (a) face image (b) GWN representation of face (c) GWN of features

### 3.2 Level One: Face Matching

The first step in feature localization we call face matching. The task is to find the “best match” face from our database of faces. In order to rate one face as best, we need an algorithm that gives a face in the database a score as a match for the new face. As we shall see, this score doesn’t necessarily indicate how similar the faces appear, but should be a good predictor of whether it is a good face for approximating feature locations in the new face.

Assume we are given a face image together with the approximate location of the face. The approximate face location would typically come from face detection [15, 16, 17]). For our experiments, we knew the face location and simply used this.

We use the first level of the GWN hierarchy and a nearest-neighbor algorithm for face matching. For each candidate face, we begin by determining an affine transformation of the level-one GWN that registers the candidate with the target image, as explained in Section 2.3. Including an affine transformation allows similar faces that have some difference in head pose (rotation) to be discovered. Levenberg-Marquardt optimization is used to find the best affine parameters.

Once we have found this transformation, we can score the difference between the transformed wavelet representation and the new face. A good score should indicate that the transformed feature locations from the database face are close to the feature locations in the new face. We have found that pixel-wise difference is not the best score for predicting feature locations. It seems that even when features are near each other, the pixel-wise difference may be large. For example, suppose the eye corners are very close, but are slightly out of alignment. The eyebrows may then not align, and where there is eyebrow in one face there will be skin in the other, and vice versa, which will yield a large pixel-wise difference. To score the database face, we allow the scale values of each wavelet in the transformed wavelet representation to be adjusted until the pixel-wise difference is minimized. We then set the score to the sum of the differences of the optimized scales from the original scales. This residual score is given in equation (4). Our experiments have shown that good residual scores do imply the transformed database face feature locations being near the new face feature locations.

Note that after level one, we can generate reasonable hypotheses for feature positions already, simply by applying the affine transformation to the relative positions of the features with respect to the whole face, as marked in our database. The success rate of these first-level hypotheses is given in Table 1. In the next subsection, we show how these estimates are further refined by level-two analysis.

### 3.3 Level Two: Feature Localization

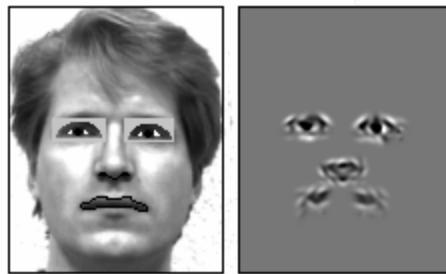
Level One gives us an initial starting point for finer search. The refinement process is identical in the abstract to how we computed the affine transformation in Level One. The details are slightly different:

We do not allow arbitrary affine transformations for facial features, because local features tend to have far fewer image constraints. A problem akin to the “aperture effect” comes into play, and this is aggravated by searching over too many degrees of freedom. We already have the affine transformation of the face, which will include an overall rotation value. We assume that any additional difference between the faces is composed of position changes only. While this may not be strictly true, it prevents the optimization from finding false positive matches in other similar features at different orientations or scales. For each feature,

we search within a limited window for a position that minimizes the difference in wavelet subspace between a candidate level-two feature GWN, and the target image. The location with the minimum value is deemed to be the location of the feature. Note that the location for each feature as output by WaveBase is a bounding box (i.e. the bounding box of the feature wavelet network that we have positioned).

When fine-tuning the feature locations, it is not clear that the features of the best-match face are in fact the best features to consider. It may be that some other face, which is overall more dissimilar, has features that are more similar. Or, it may be that some processed (e.g. edge enhanced) or hand-drawn features may work better in practice (Figure 4 shows an edge-enhanced face that we included in our database). Therefore, WaveBase allows candidate feature GWNs may be drawn from any of the faces in our database,<sup>3</sup> not just the GWNs that are associated with the best-match face from Level One. We select the most similar feature, as measured by a residual score exactly as in level one. This gives even a relatively small database the power to match a considerable segment of the population, by mixing and matching features from different faces.

Clearly, the success of WaveBase depends on it containing at least one face in its database that is sufficiently similar to the new face to allow discovery of an affine transformation for the face, and hence an approximation of feature positions. We do not know how many faces would be required to ensure that any other face (or some high percentage of other faces) in existence can be sufficiently matched. Perhaps it is the case that only some small number of representative faces are required. At this early stage of our research, we are not concerned with minimizing the size of the database. Our focus at present is achieving accuracy and showing a proof of concept. Space optimizations are left for future studies.



**Figure 4 – Edge-enhanced face, and the GWN representation of its features.**

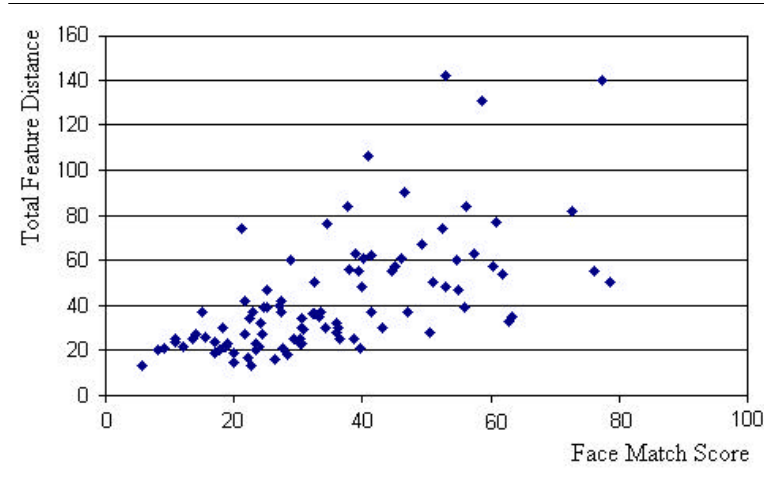
### **3.4 Results**

Experimental validation of our approach was obtained by constructing a database of 100 faces, drawn from the Yale and FERET Face Databases [4, 12]. To test, we performed a series of leave-one-out experiments on each of the 100 faces. In each experiment we consider one face and apply feature localization using the remaining database of 99 faces. For each set of automated feature localizations, we compare with the hand-marked locations of each feature.

---

<sup>3</sup> Including any processed or hand-drawn heads that have been added to the database.



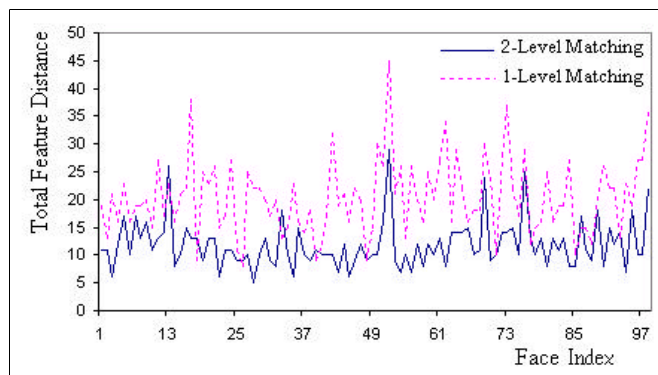


**Figure 5 – First-level matching: Sum of feature position differences versus face match score for one face.**

Figure 5 plots the sum of feature position differences versus face score for a single face, with all other faces in the database scored against it. This figure demonstrates that a good score always corresponds to a small position difference. To show that there is considerable advantage to additional layers in the hierarchy, we compare feature localization results using only one level to using both levels.

Feature	1-level detect rate	2-level detect rate
Left eye outside corner	0.81	0.95
Left eye inside corner	0.90	0.94
Right eye inside corner	0.93	0.94
Right eye outside corner	0.78	0.96
Left nostril	0.86	0.95
Right nostril	0.88	0.94
Left lip corner	0.65	0.87
Right lip corner	0.65	0.88

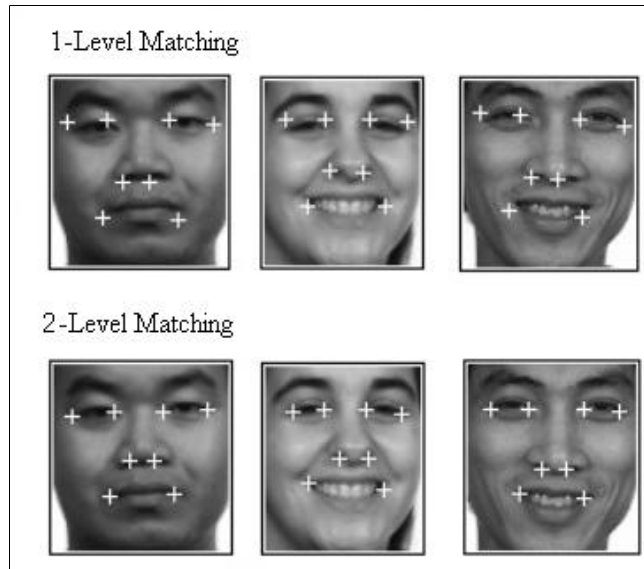
**Table 1. Feature localization accuracy for 1- and 2-level hierarchies. A feature was counted as accurately detected if it was localized to within 3 pixels of the point marked by hand.**



**Figure 6 – Sum of feature position differences for each face, plotted for 1-level and 2-level matching.**

Table 1 compares feature localization rates for both 1 and 2-level systems. An “accurate” localization is characterized as one in which the feature was localized to within 3 pixels (L2-distance) of the hand-marked position. Note that features are localized consistently more accurately for all features with two levels rather than one. Figure 6 shows this same trend broken down differently. The solid line indicates the total SAD in feature position between 2-level localization and hand-annotation; the dashed line is for 1-level localization. Except in two or three rare instances, the 2-level localization is far superior.

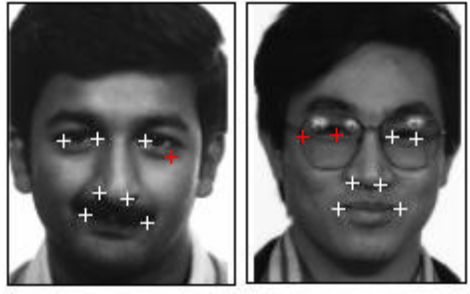
Finally, we offer some examples out of the 100 experiments for visual examination. Figure 7 shows a clear improvement in feature localization with two levels. Note that just about every feature is accurately localized by two-level matching. Figure 8 and Figure 9 illustrate further cases of accurate and inaccurate detection cases using the two-level hierarchy. Figure 9 shows examples of some rare failure cases. Among failures, these examples are typical – eyebrows or shadows under the eyes are sometimes mistaken for the eyes themselves, and specular reflection from glasses can obfuscate eye corners. See the appendix for the full results.



**Figure 7 – Feature detection results show improved accuracy from using hierarchical localization.**



**Figure 8 – Feature detection results showing accurate detection.**



**Figure 9 – Feature detection results showing inaccurate detection.**

## 4 Related Work

Other facial feature detection approaches exist. One approach detects feature points using hand-crafted geometric models of features [19]. The goal of this work, however, is in detection of faces by looking for groups of facial features, so feature localization accuracy is low. Other work trains separate face and facial feature detectors, where features are trained for maximum discriminability from among a training set [5]. This work is presented without quantitative measures of feature localization. Steerable filters and geometrical models have also been used to find facial features with high accuracy[8]. A coarse-to-fine image pyramid is employed to localize the features, but the technique requires high-resolution imagery in which sub-features such as the whites of the eye are clearly visible as such. Color segmentation can also be used to estimate approximate feature locations [6]. These estimates, reported to have a precision of up to 2 pixels, can be further refined via grayscale templates to sub-pixel accuracy. For each individual and each face feature nine 20 x 20 pixel templates are given, but no generalization to unknown faces is discussed. Finally, neural networks have been used to detect eyes and eye corners [14]. Results approach 96% correctly detected eye corners while allowing a variance of two pixels, but these results are for eyes only, which are less deformable than mouths.

Lastly, GWNs invite the closest comparison with the well-known Gabor jet representations of facial features [18]. The advantage of GWNs is that they offer a sparser representation of image data: Where jets can require up to 40 complex Gabor filters to approximate the local image structure around a single feature point, GWNs can make do with nine, as in our implementation. This is a direct consequence of allowing wavelets in a GWN to roam continuously in their parameter space during training. Edge features, which are building blocks of more complex features, are thus efficiently captured at various scales by GWNs.

## 5 Conclusion

We have presented a hierarchical wavelet network approach to feature detection. Our method takes a coarse-to-fine approach to localize small features, using cascading sets of GWN features.

We tested our results on the task of facial feature localization, using one- and two-level hierarchies. For the one-level implementation, GWNs are trained for the whole face; for two levels, the second-level GWNs are trained for each of eight facial features. Experiments show that the two-level system outperforms the one-level system easily, verifying the usefulness of a hierarchy of GWNs for feature localization. Results compare favorably with other algorithms on this task.

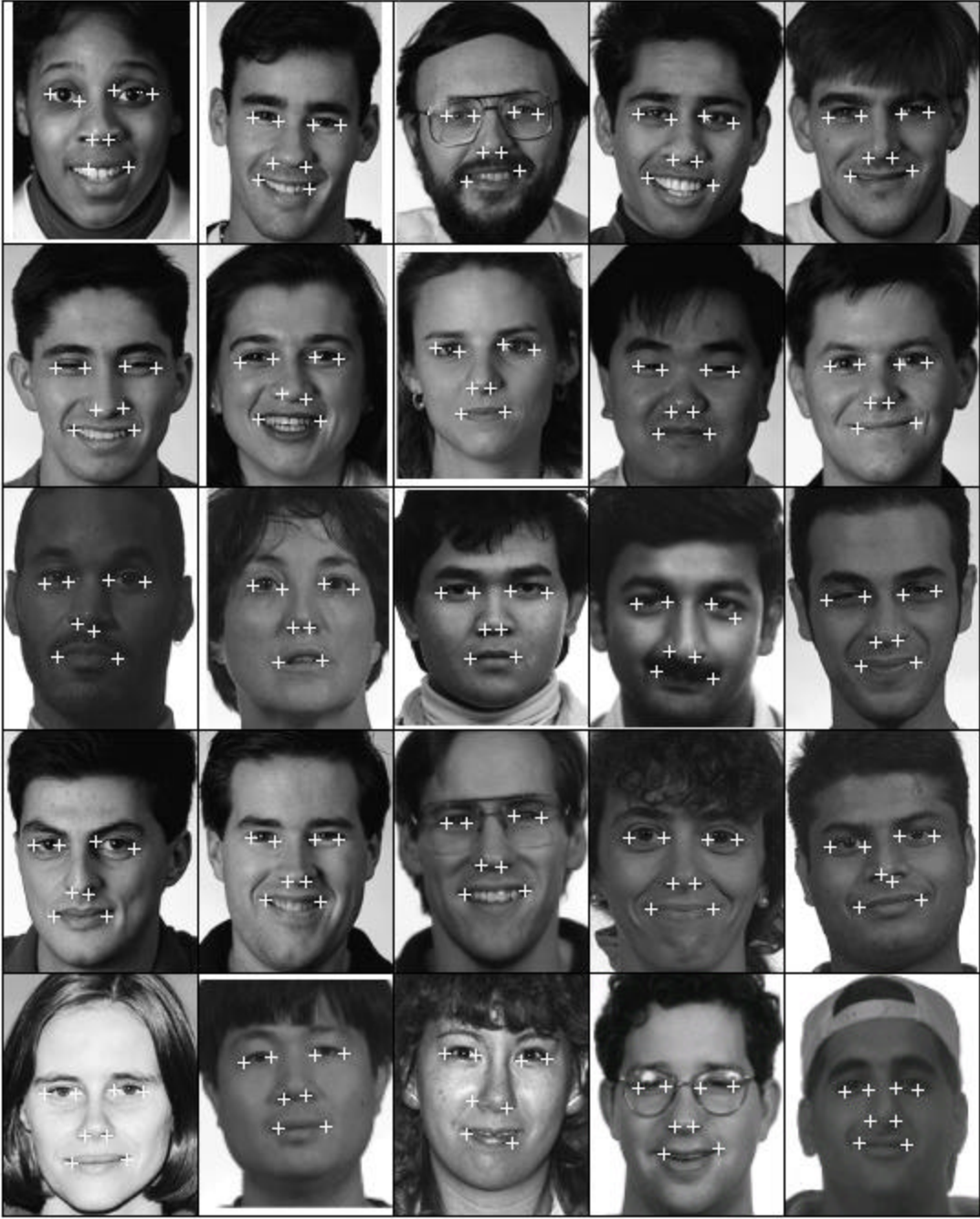
Some remaining issues include the following: How can we determine the minimum number of wavelets required for each GWN? Can a subset of wavelets in a given network be sufficient for good matching at a particular level? Finally, how can we minimize the number of GWNs necessary at each level to capture the broad range of the set of real targets? We hope to examine these questions as future work.

## 6 References

- [1] Rogerio Feris, Volker Krüger, and R. Cesar Jr, Efficient Real-Time Face Tracking in Wavelet Subspace, *Proceedings of the Int. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, Vancouver, BC, Canada, 2001, in conjunction with ICCV'01.

- [2] Gemmell, Jim, Zitnick, C. Lawrence, Kang, Thomas, Toyama, Kentaro, and Seitz, Steven, Gaze-awareness for Videoconferencing: A Software Approach, *IEEE Multimedia*, 7(4):26-35, Oct-Dec 2000.
- [3] Kentaro Toyama and G. Hager, Incremental Focus of Attention for Robust Vision-Based Tracking, *International Journal of Computer Vision*, 35(1):45-63, 1999.
- [4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Patt. Anal. and Mach. Intel.*, 19(7):711–720, 1997. Special Issue on Face Recognition.
- [5] A. Colmenarez, B. Frey, and T. Huang. Detection and tracking of faces and facial features. 1999.
- [6] H. Graf, E. Casotto, and T. Ezzat. Face analysis for synthesis of photorealistic talking heads. In *Proc. Int'l Conf. on Autom. Face and Gesture Recog.*, pages 189–194, Grenoble, France, March, 28-30, 2000.
- [7] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10):1025–1039, October 1998.
- [8] R. Herpers and et al. Edge and keypoint detection in facial regions. In *Killington, VT, Oct. 14-16*, pages 212–217, 1996.
- [9] V. Krüger. Gabor wavelet networks for object representation. Technical Report CS-TR-4245, University of Maryland, CFAR, May 2001.
- [10] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Int'l Joint Conf. on AI*, pages 674–679, 1981.
- [11] B. Manjunath and R. Chellappa. A unified approach to boundary perception: edges, textures, and illusory contours. *IEEE Trans. Neural Networks*, 4(1):96–107, 1993.
- [12] P. Phillips, H. Moon, S. Rizvi, and P. Rauss. The feret evaluation. In H. W. et al., editor, *Face Recognition: From Theory to Applications*, pages 244–261, 1998.
- [13] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes, The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK, 1986.
- [14] M. Reinders, R. Koch, and J. Gerbrands. Locating facial features in image sequences using neural networks. 1997.
- [15] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. Patt. Anal. and Mach. Intel.*, 20:23–38, 1998.
- [16] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *Proc. Computer Vision and Patt. Recog.*, pages 749–751, Hilton Head Island, SC, June 13-15, 2000.
- [17] P. Viola and M. Jones. Robust real-time face detection. In *ICCV01*, page II: 747, 2001.
- [18] L. Wiskott, J. M. Fellous, N. Krüger, and C. v. d. Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. Patt. Anal. and Mach. Intel.*, 19:775–779, 1997.
- [19] K. Yow and R. Cipolla. Feature based human face detection. *Image and Vision Computing*, 15:713–735, 1997.
- [20] Q. Zhang and A. Benveniste. Wavelet networks. *IEEE Trans. Neural Networks*, 3:889–898, 1992.

**APPENDIX: Results for all faces.<sup>4</sup>**



---

<sup>4</sup> Although we included an edge-enhanced face in the database (Figure 4) we did not perform feature detection on it

