Affine Real-Time Face Tracking using a Wavelet Network

Volker Krüger, Alexander Happe and Gerald Sommer
Computer Science Institute, Christian-Albrechts University Kiel
Preußerstr. 1-9, 24105 Kiel, Germany
Tel: ++49-431-560496, FAX: ++49-431-560481
email: vok@ks.informatik.uni-kiel.de

Abstract

In this work we will present a method for visual face tracking that is based on a wavelet representation of a face template. The wavelet representation allows arbitrary affine variations of the facial image, it allows to generalize from an individual face template to a rather general face template and it allows to adapt the computational needs of the tracking algorithm to the computational resources available. The method presented runs in real-time (25 Hz) on a Linux Pentium 450 MHz and was tested on several common sequences including the salesman-sequence.

1 Introduction

In this paper we will address the issue of tracking faces. Tracking faces in real-time (RT) (25 Hz) is of major importance for many applications including Human-Computer-Interfaces (HCI), surveillance applications, teleconferencing or teleteaching. Also applications such as gesture- and gaze detection are often stated applications that heavily depend on precise tracking algorithms. Yet, the issue of tracking faces is far from being solved satisfactorily. Many tracking systems use color as a clue for tracking but they track imprecisely and they are not capable of distinguishing between faces and non-faces [9; 1; 16]. Other tracking systems use a previously given template (gray value, active contour, etc), while allowing affine variations of the facial image. These systems track precisely as shown in the excellent work of [6], but the templates are either of individual persons or are computationally expensive [7; 5] so that tracking is not in RT. Other systems use a separate neural network or eigenfaces for verifying for the face during tracking. However, these systems are not invariant with respect to affine variations of the facial image [13]. In [12] a system is presented that is able to track faces independently of face orientation and gesture. The systems uses the wavelet jet bunch-graph approach [18] but tracks with less than 1 fps.

In this work we will present an approach for RT face tracking that allows arbitrary affine deformations of the facial image in order to compensate for different poses. For this we use a wavelet representation for a given face template. In the following context the wavelets are not used in the manner as introduced by Mallat or Deaubechies [2; 11] where, in the discrete case of image representation, homogeneous sampling schemes are considered. Neither will we use a bunch-graph like approach that represents only arbitrary local image features [18]. The major contribution of this work will be the introduction of a "third way" of using wavelets. It is a similar approach to the one in [19]: A wavelet representation (*wavelet network*) of a face template will be found by optimizing the weights and the degrees of freedom of each single wavelet (2D dilation, 2D translation, orientation) with respect to the pixel wise difference to a given face template. With this, the face template is represented by a linear combination of these specially optimized wavelets. The wavelets are optimized within the continuous phase space and are therefore not limited to the homogeneous discrete grid of the discrete phase space. Furthermore they are optimized with respect to the image contents and do not represent just arbitrary local image features. This representation has several advantages:

1. A face template (or image template, in general) can be represented by a very small set of weighted wavelets.
2. The degrees of freedom of each single wavelet cover the affine deformations without a shear factor and so do the degrees of freedom of their linear combination (*superwavelet*).
3. It is straight forward to introduce a shear factor

to allow the linear combination of the wavelets to deform affinely.

4. The computer power needed for tracking will depend upon the number of wavelets to evaluate. This number can be chosen dynamically upon the computer power available.

5. The number of wavelets used decides on the amount of detailed template information contained in the wavelet representation. Using less wavelets the representation becomes more general which allows to track also arbitrary faces.

By exploiting these advantages, we establish real-time tracking of arbitrary faces by optimizing the affine parameters of the entire wavelet representation at each image frame while being able to dynamically adapt the number of used wavelets and with this the computer resources needed.

In section 2 we will give a short introduction to wavelets. In section 2.1 we explain how we find a wavelet representation for a given face template and in section 3, the tracking algorithm is presented. In section 4 we test the tracker on the well known sequences "salesman" and conclude with final remarks in section 5.

2 Introduction to wavelets

In this section we will give a short introduction to the wavelet transform and review very shortly the two common wavelet approaches for image representation taken to date. A function $\psi \in \mathbb{L}^2(\mathbb{R}^2)$ that satisfies

$$0 < C_\psi = 4\pi^2 \int_{\mathbb{R}} \int_{\mathbb{R}} \frac{\|\hat{\psi}(\omega_x, \omega_y)\|^2}{\|(\omega_x, \omega_y)\|} d\omega_x \, d\omega_y < \infty$$

is called a *wavelet*. Here, $\hat{\psi}$ is the Fourier transform of ψ. For any function $f \in \mathbb{L}^2(\mathbb{R}^2)$ the continuous wavelet transform is given by

$$\begin{aligned} (Lf)(c_x, c_y, s_x, s_y, \theta) &= \int_{\mathbb{R}^2} f(\mathbf{x}) \psi(\mathbf{SR}(\mathbf{x} - \mathbf{c})) d\mathbf{x} \\ &= < f, \psi_{c_x, c_y, \theta, s_x, s_y} > \\ &= < f, \psi_n > , \end{aligned} \tag{1}$$

with the rotation matrix $\mathbf{R}$, the dilation matrix $\mathbf{S}$ and the translation vector $\mathbf{c}$:

$$\begin{aligned} \mathbf{R} &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \\ \mathbf{S} &= \mathrm{diag}(s_x, s_y) \\ \mathbf{c} &= (c_x, c_y)^T \ . \end{aligned}$$

θ denotes the rotation angle of the wavelet $\psi(\mathbf{x})$, s_x, s_y the scaling in x and y direction and c_x, c_y the translations in x and y direction. In this sense, the wavelets $\psi_\mathbf{n}$ are dilated, rotated and translated versions of the original wavelet ψ, called *mother wavelet*. The five dimensional parameter vector $\mathbf{n}$ is given by these parameters:

$$\mathbf{n} = (c_x, c_y, \theta, s_x, s_y) \ .$$

The function f can always be reconstructed by integration over all wavelet parameters:

$$\begin{aligned} f &= C_\psi^{-1} \int (Lf)_\mathbf{n} \psi_\mathbf{n} \, d\mathbf{n} \\ &= C_\psi^{-1} \int < f, \psi_\mathbf{n} > \psi_\mathbf{n} \, d\mathbf{n} \end{aligned}$$

This integration is done over the entire continuous phase space.

A natural way to define the discrete wavelet transform is to discretizise the phase space and to assign discrete values to the wavelet parameters as follows: $s_x = (s_{x_0})^m, s_y = (s_{y_0})^m, c_x = n s_0 (s_{x_0})^m, c_y = k s_0 (s_{y_0})^m, \theta = \theta_l = l\theta_0$, with $m, n, k, l \in \mathbf{Z}$. With this the discrete wavelet transform is given by

$$(L^d_{mnkl} f) = < f, \psi_{mnkl} > \ . \tag{2}$$

Equation (2) can be interpreted as an abstract representation of f by its wavelet coefficients. To represent f uniquely (if possible at all), a huge number of wavelet coefficients are generally needed. How good f is represented by its coefficients $L^d_{mnkl} f$ and how many are needed depends on the chosen wavelet and on the values s_{x_0}, s_{y_0}, s_0 and θ_0. This type of representation is also applied within the bunch graph approach [18], where, however, only a few prominent feature points are represented by their wavelet coefficients. Of course, a reconstruction of the image is in this case not possible.

An image f can also be represented (in the sense of an approximation) by a linear combination of weighted wavelets

$$f = \sum_{mnkl} w_{mnkl} \psi_{mnkl} \ . \tag{3}$$

In the sense of equation (3), the function f is approximated by a linear combination of weighted wavelets where wavelets and weights are found by optimization (see below).

In some special cases (orthogonal cases), $w_{mnkl} = L^d_{m,n,k,l} f$, however generally, $w_{mnkl} \neq L^d_{m,n,k,l} f$.

2.1 The third way

Gabor functions are well known to provide the best possible tradeoff between spatial resolution and frequency resolution and are therefore an important tool in image analysis. Furthermore, the use of Gabor filters in image analysis is biologically motivated as they model the response of the receptive fields of the orientation-selective simple cells in the human visual cortex [3; 8]. In fact, it has been suggested [4; 14] that the receptive field response of simple cells can be described by the particular family of 2D Gabor wavelets.

It is important to know how well an image I can be represented by using Gabor wavelets in case of a discrete phase space. This topic was studied by Lee [10] and it came out that the dilation step widths, the translation step width and the angular step θ_0 have to be extremely small which leads to a highly redundant image representation. Thus, one way to represent an image is to use such a general and highly redundant Gabor wavelet decomposition (it should be mentioned that this decomposition allows to represent *any* $f \in \mathbb{L}^2(\mathbb{R}^2)$). However, a huge number of wavelets coefficients are needed. Another possibility is to use a Gabor jet representation [18]. Even though this is obviously a very lossy image representation with roughly 800 wavelet coefficients, it seems to work very well in the context of face recognition.

Inspired by [19], we want to propose here, as a major contribution of this work, a third way by representing an image using a *wavelet network*. This representation has the advantage of being even sparser than the Gabor jet representation. Yet it encodes nearly the entire image information and allows a good reconstruction.

Generally speaking, we start out by selecting a family of N (not necessarily Gabor) wavelets $\Psi = \{\psi_{\mathbf{n}_1} \dots \psi_{\mathbf{n}_N}\}$. The parameters $\mathbf{n}_i$ (dilation, translation and orientation) may be chosen arbitrarily. In order to find a representation for a function $f \in \mathbb{L}^2(\mathbb{R}^2)$ (f DC-free, without limitation of generality) we minimize the energy function

$$E = \min_{\mathbf{n}_i, w_i \text{ for all } i} \|f - \sum_i w_i \psi_{\mathbf{n}_i}\|_2^2 \qquad (4)$$

with respect to the weights w_i and the wavelet parameters $\mathbf{n}_i$.

Definition: Let $\psi_{\mathbf{n}_i}$, $i = 1 \dots N$ wavelets. The two vectors

$$\begin{aligned} \Psi &= (\psi_{\mathbf{n}_1}, \dots, \psi_{\mathbf{n}_N})^T \text{ and} \\ \mathbf{w} &= (w_1, \dots, w_N) \end{aligned}$$

define a *wavelet network*.

Figure 1. The left image shows the original face image, the right image shows its reconstruction using formula (5) with an optimal wavelet net Ψ of just $N = 52$ odd Gabor wavelets.

It may be mentioned that it was proposed before [2; 10; 4] to use an energy function (4) in order to find the optimal weights w_i for non-orthogonal wavelets $\psi_{\mathbf{n}_i}$. We enhance this approach to also find optimal parameters of each wavelet $\psi_{\mathbf{n}_i}$. The parameters $\mathbf{n}_i$ are chosen from continuous phase space. This is exactly the great advantage over the discrete approach: While in the case of a discrete phase space image structure has to be interpolated, wavelets can in the continuous case be selectively chosen to reflect precisely the local image structure.

Using the optimal wavelets Ψ and weights $\mathbf{w}$ the original image can be (closely) reconstructed by a linear combination of weighted wavelets:

$$\begin{aligned} \tilde{f} &= \sum_{i=0}^{N} w_i \psi_{\mathbf{n}_i} \qquad (5) \\ &= \Psi \mathbf{w}^T \ . \end{aligned}$$

Of course the quality of the reconstruction depends on the number N of used wavelets.

In this work, we use the Gabor function as mother wavelets ψ and restrict our considerations to the odd part (imaginary part) only:

$$\begin{aligned} \psi_{\mathbf{n}}(x,y) &= \Big(-\frac{1}{2}\Big[s_x\left((x-c_x)\cos\theta - (y-c_y)\sin\theta\right)\Big]^2 \\ &+ \Big[s_y\left((x-c_x)\sin\theta + (y-c_y)\cos\theta\right)\Big]^2\Big) \\ &\times \sin\Big(s_x\left((x-c_x)\cos\theta - (y-c_y)\sin\theta\right)\Big) \qquad (6) \end{aligned}$$

with $\mathbf{n} = (c_x, c_y, \theta, s_x, s_y)$. The left image fig. (1) shows the original image I, the right image $\tilde{I}_{4,6}$ its reconstruction (5) using an optimal wavelet net Ψ of just $N = 52$ odd Gabor wavelets.

The choice of the mother wavelet ψ is arbitrary. We use the odd Gabor function as mother wavelet throughout this paper because we found that this function leads to the best reconstruction results. Furthermore, since the odd Gabor wavelets are DC-free, they are invariant to homogeneous illumination changes.

In order to minimize formula (4) and to find an optimal wavelet net Ψ, $\mathbf{w}$ for an image I, we use the Levenberg-Marquard gradient descent method [15]. Such a method finds a local minimum, a careful selection of initial parameters is therefore important. Our initialization and optimization scheme is similar to the Laplacian pyramid scheme: First we position equidistantly 4×4 coarse wavelets within the prominent image region (in our case the inner face region) (fig. 2, top left). These 16 wavelets define the first pyramid layer. They are then optimized with respect to the energy function (4). The optimization result,$\tilde{I}_4$, is shown in fig. 2, top right. In a second step we calculate the difference image between the original image and its reconstruction, $I - \tilde{I}_4$, which is then, in turn, approximated by 6×6 finer wavelets (2nd row, left). These wavelets define the 2nd pyramid layer. The result is shown in the 2nd row, right. Adding the two right images together leads to image $\tilde{I}_{4,6}$, shown in the 3rd row. To the left, the positions of the first layer wavelets are sketched. The original image I is for comparison shown in fig. 1. This proceeding may be continued for further pyramid layers. It should be mentioned that at each indicated wavelet position in fig. 2, just one single wavelet is located. Their initial orientations are random and their initial scales are constant in each layer and chosen with respect to the distance to the neighboring wavelet. A coarse-to-fine strategy for optimization is intuitive because the energy function (4) can be better minimized using first coarse and then fine wavelets.

3 Affine Face Tracking in Image Sequences

We have given in the preceding section an extensive introduction to wavelet networks. This is needed to understand the basic principles of the wavelet representation. Now, all tools are in place and we can continue merely straight forward by describing in the next subsection 3.1 how a wavelet net, taken as a face template, can be repositioned (translated, scaled and rotated) on a new face such that the wavelets will be positioned in the new image on the same facial features as in the old one. In subsection 3.2 we will extend this approach to allow a repeated repositioning of the template net in a sequence of images.

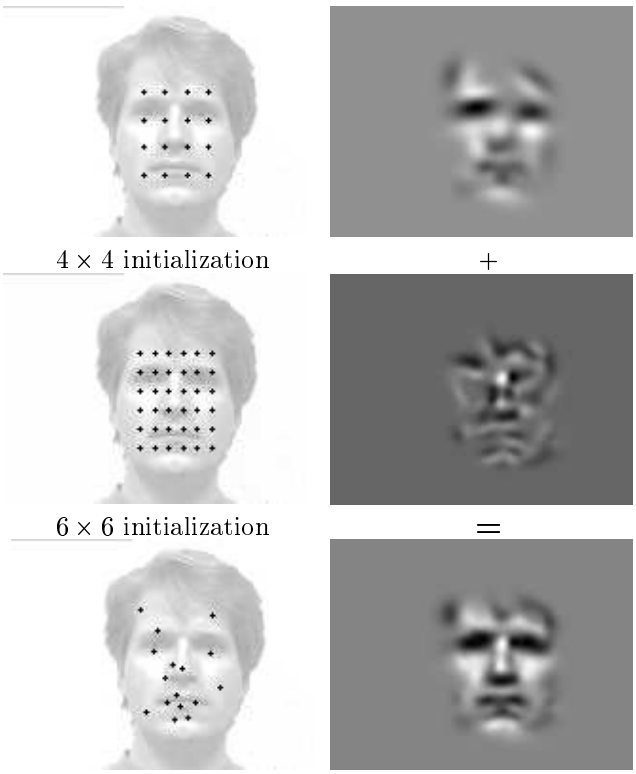

Figure 2. These images demonstrate the idea of the Laplace pyramid-like initialization and optimization scheme. The wavelet net is first initialized with the wavelets sketched in the top left image. The optimization results in the net shown in the top right image. The difference between that image and the original image is then approximated by the wavelets that are initialized according to the center left image. The optimization result is shown in the center right image. The bottom right image shows finally the sum between the top right image and the center right one. The bottom left image shows the positions of the wavelets of the top image.

3.1 Repositioning Wavelet Networks in Single Images

In this subsection we will demonstrate how an entire wavelet net can be translated, scaled and rotated so that its wavelets are finally positioned on the same facial features as in the original image. An example for this can be see in fig. 2, where in the bottom left image the original positions of the wavelets are marked and in fig. 4, where in new images the wavelet positions of the *repositioned* wavelet network are marked.

Positioning a wavelet net is established by using a

superwavelet [17].

Definition: Let $\Psi = (\psi_{\mathbf{n}_1}, \ldots, \psi_{\mathbf{n}_N})$, $\mathbf{w} = (w_1, \ldots, w_N)$ be a wavelet net. A *superwavelet* $\Psi_{\mathbf{n}}$ is defined to be a linear combination of the wavelets $\psi_{\mathbf{n}_i}$ such that

$$\Psi_{\mathbf{n}}(\mathbf{x}) = \sum_i w_i \psi_{\mathbf{n}_i}(\mathbf{SR}(\mathbf{x} - \mathbf{c})) , \quad (7)$$

where the parameters of vector $\mathbf{n}$ of superwavelet Ψ define the dilation matrix $\mathbf{S}$, the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{c}$.

A superwavelet $\Psi_{\mathbf{n}}$ is again a wavelet [17] that has the wavelet parameters dilation, translation and rotation. Therefore, we can handle it in the same way as we handled each single wavelet in the previous section, and we may optimize its parameters $\mathbf{n}$ with respect to some energy function E:

$$E = \min_{\mathbf{n}} \|I - \Psi_{\mathbf{n}}\|_2^2 \quad (8)$$

For optimization, we can elegantly use the same optimization procedure as in the preceding section. In several experiments we have found that the initialization that has to be supplied to the gradient decent method may be within the range of approximately ± 10 px in position, $\pm 20\%$ in scale and $\pm 10°$ in orientation (see below for further comments). An example of the optimization process can be seen in fig. 3: Shown are the initial values of $\mathbf{n}$, the values after 2 and 4 optimization cycles and the final values after 8 cycles, each marked with the white square. The square refers to the image region where the wavelets were initially homogeneously distributed as shown in fig. 2. Its center position marks the center position of the corresponding superwavlet. The superwavelet used in fig. 3 is $\tilde{I}_4$ of fig. 2 i.e. it is derived from the person in fig. 1. A further example can be seen in fig. 4. The left images have to be compared to the bottom left image in fig. 2: It can be seen that the wavelets are repositioned correctly to the same corresponding facial features. The images to the right of fig. 4 show the reconstructions using the repositioned wavelet nets.

The image distortions of a planar object that is viewed under orthographic projection is described by six parameters: translation c_x, c_y, rotation θ and dilation s_x, s_y and s_{xy}. The degrees of freedom of a wavelet only allow translation, dilation and rotation. However, it is straight forward to include also shearing and thus allow any affine deformation of $\Psi_{\mathbf{n}}$. For this, we enhance the parameter vector $\mathbf{n}$ to a six dimensional vector

$$\mathbf{n} = (c_x, c_y, \theta, s_x, s_y, s_{xy})$$

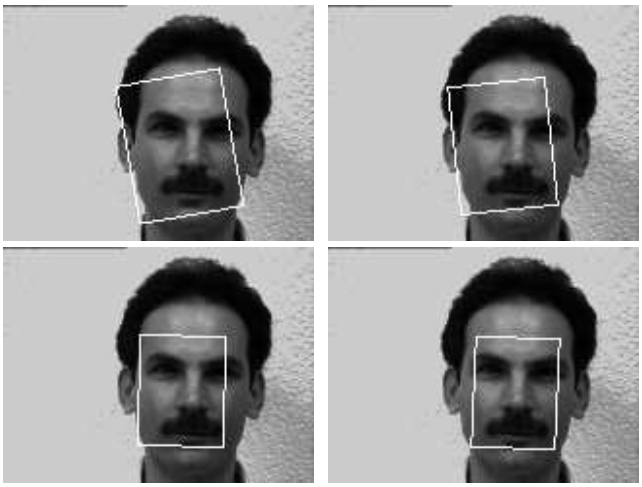

Figure 3. The images show the 1st, the 2th, the 4th and the 8th (final) step of the gradient descent method optimizing the parameters of a superwavelet. The top left image shows the initial values with 10 px. off from the true position, rotated by $10°$ and scaled by 20%. The bottom right image shows the final result. As superwavelet, $\tilde{I}_4$ of figure 1 was used.

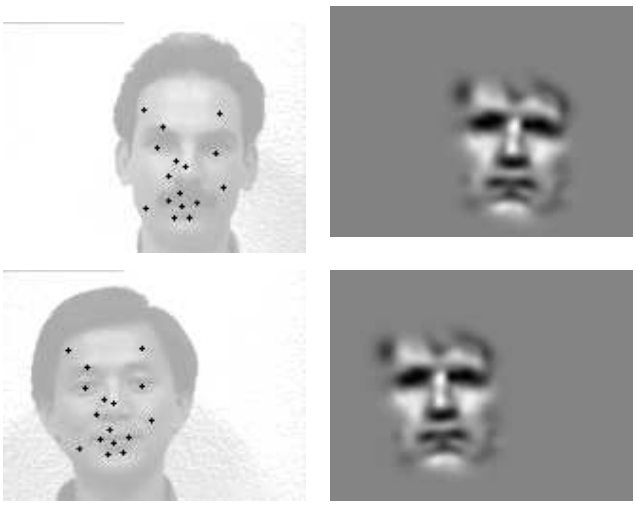

Figure 4. The images show the positions of each of the 16 wavelets after repositioning the wavelet net (left) and the corresponding reconstruction(right). The reconstructed faces show the same orientation, position and size as the ones they were repositioned on.

By rewriting the scaling matrix $\mathbf{S}$,

$$\mathbf{S} = \begin{pmatrix} s_x & s_{xy} \\ 0 & s_y \end{pmatrix} ,$$

we become able to deform the superwavelet $\Psi_{\mathbf{n}}$ affinely.

The repositioning of the superwavelet can be understood as warping where the original face, represented by the wavelet net Ψ, is warped onto the new face. This idea is shown in fig. 5.

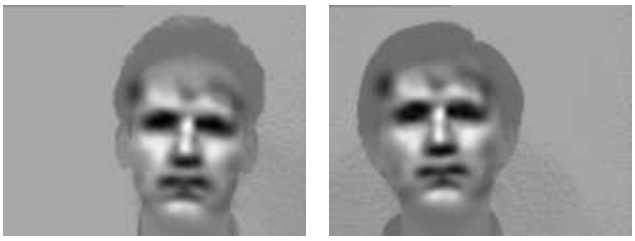

Figure 5. The two images show the wavelet net $\tilde{I}_{4,6}$, repositioned onto the two test images of fig. 4. This demonstrates that the repositioning process can be understood as warping the superwavelet onto the new test faces.

3.2 Affine Tracking using Wavelet Networks

The technique of repositioning a superwavelet with respect to the energy function (8), as it was explained in the preceding subsection, can also be applied to image sequences. For this, (8) may be rewritten to

$$E \quad = \quad \min_{\mathbf{n}_t} \|I_t - \Psi_{\mathbf{n}_t}\|_2^2 \ . \qquad (9)$$

so that for the frame I_t at time step t the superwavelet $\Psi_{\mathbf{n}_t}$ is optimized with respect to the energy function (9). To derive $\mathbf{n}_{t+1}$ for a successive frame I_{t+1} at time $t+1$, $\mathbf{n}_t$ from the preceding frame is used as initial value. These initial values were in our experiments good enough that the optimization always converged quickly (see section 4).

Initial values $\mathbf{n}_0$ for the very first frame I_0 can be derived from the color blob information of the face as experiments have shown. A color blob is given by its mean value and its standard deviation. The mean value gives a clue about the position and a first clue about the scale and the orientation can be calculated from the standard deviation matrix. For the test sequence of fig. 6, we have chosen $\mathbf{n}_0$ by hand because the sequences is a gray scale sequence.

The number of wavelets that make up the superwavelet can be adapted: The maximum number is given by the number N of wavelets in the wavelet network. However, we are free to use less wavelets. Each wavelet of the superwavelet has to be evaluated during the repositioning process, so that using less wavelets results in a respective speedup. To choose the best wavelets $\psi_{\mathbf{n}_i}$, the absolute value of their weights, $\|w_i\|$, can be used as a measure of importance.

Techniques for affine motion prediction have not yet been incorporated into the tracker. Such techniques should result in a significant speedup.

4 Experiments

Using the superwavelet $\tilde{I}_4$, we have found in several experiments that the initialization of $\mathbf{n}_0$ may vary from the correct parameters by approx. ± 10 px in x and y direction, by approx. 20% in scale and by approx. $\pm 10°$ in rotation (see fig. 3). Of course, these are only approximate values since they depend on the number of wavelets used, on the template face and on the scale of the used wavelets. In our case, 10 px. correspond to $\approx 1/6$ of the width of the white box in fig. 3, marking the inner face region.

We have further tested the positioning procedure on the Yale face database. This database consists of 15 different individuals, showing eights different facial expressions, the faces are approximately all of the same size. The wavelet net $\tilde{I}_4$ can be considered as a rather general face template. Using this wavelet net, the positioning procedure converged correctly on 13 individuals (independent of expression) by just giving the approximate image center as initial values. This shows two things:

1. it shows, that the wavelet net template is not fixed to one individual and that it is sufficiently general.
2. it shows, that the reposition algorithm is quite stable with respect to its initial values.

For face tracking, using color blob information as initial values for $\mathbf{n}$ seems to be precise enough. We have tested the face tracker on several sequences, including the salesman sequence. Example frames can be seen in fig. 6, the entire mpeg-sequence (64k) may be downloaded from

`http://www.ks.informatik.uni-kiel.de/~vok/research/wavelet_tracker/salesman.mpg.`

Figure 7 shows for each frame the square error in pixels between the "ground truth" center position of the faces and the estimated center position of the superwavelet (which is the center of its white box). The ground truth was estimated by hand, an estimation error of $\pm$ one pixel in x- and y- direction is possible. For tracking, only 9 (!) wavelets were used. This, however, limits the tracking precision which can be noticed in fig. 6 and 7.

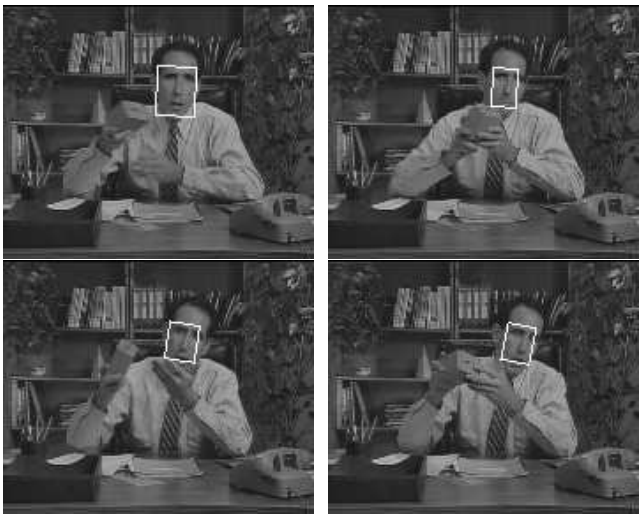

Figure 6. The images shows from left to right frame 11, frame 50, frame 120 and frame 137.

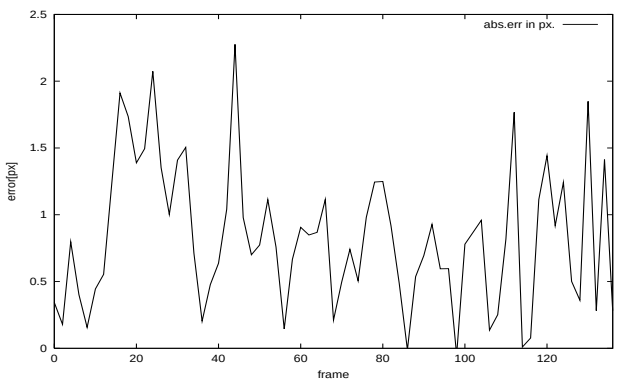

Figure 7. The figure shows the Euclidean distance between the estimated position of the superwavelet and the "ground truth".

During tracking, since we track with 25 Hz, successive frames are sufficiently similar so that the gradient descent method never needed more than two cycles for each time step until reaching the minimum. Using 9 wavelets the computing time consumed for each frame was less than 35 ms on a 450 MHz Linux Pentium. Using more wavelets instead, the tracking becomes more precise but slower. Furthermore, when the system tracks slower, the initial values for the positioning procedure are less precise because of the increased temporal distance between successive image. Using all 16 wavelets of the first pyramid layer, the tracker consequently reaches only 10 Hz. However, this loss of speed probably would have been less drastic if a motion prediction had been used.

5 Conclusion

In this paper we presented a novel approach for real-time face tracking. The major contributions of this work are

1. the introduction of the notion *wavelet network*. The wavelet network is used to approximate a template image while using a relatively small set of wavelets.

2. the introduction of the notion *superwavelet*, that allows us to handle an entire wavelet network as a single wavelet and to deform the wavelet net affinely.

Even though we are by no means the first to use wavelet networks and superwavelets, we are still the first to apply them consequently in practice and to study and exploit their properties and advantages. As far as we know, both terms have so far only been of theoretical interest.

3. The development of a tracking approach that exploits the advantages of the superwavelet which has several advantages:

   - The flexible representation of the template using a wavelet network allows a tracking that is not fixed to one particular individual.
   - The tracker is able to cope with affine deformations of the tracked object.
   - The tracker uses only moderate hardware resources.
   - The resources needed for tracking can be dynamically adapted to the resources available. This is particularly useful when the tracker is to be used in conjunction with other programs.

In several experiments the tracking and the repositioning algorithm was successfully evaluated and compared to the ground truth. It will be future work to incorporate this tracking technique into our active camera mount [9] and to study and evaluate the properties and qualities of our wavelet representation. Furthermore it is future work to use wavelet networks for face- and gesture recognition.

Acknowledgment

The images used are derived from the Yale Face Database. We thank K. Daniilidis from UPenn, U. Mahlmeister from Philips (Hamburg) and N. Krüger for valuable discussions.

References

[1] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *IEEE Conf. Computer Vision and Pattern Recognition, CVPR*, pages 232–237, 1998.

[2] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Trans. Informat. Theory*, 36, 1990.

[3] J. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized two-dimensional visual cortical filters. *Journal Opt. Soc. Am.*, 2(7):1160–1168, 1985.

[4] J. Daugman. Complete discrete 2D Gabor transform by neural networks for image analysis and compression. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 36(7):1169–1179, 1988.

[5] F. d.l. Torre, S. Gong, and S. McKenna. View-based adaptive affine tracking. In *Proc. Fifth European Conference on Computer Vision*, volume 1, pages 828–824, Freiburg, Germany, June 1-5, 1998.

[6] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

[7] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 1998.

[8] J. Jones and L. Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58(6):1233–1258, 1987.

[9] V. Krüger, R. Herpers, K. Daniilidis, and G. Sommer. Teleconferencing using an attentive camera system. In *Int. Conf. on Audio- and Video-based Biometric Person Authentication*, pages 142–147, 1999.

[10] T. S. Lee. Image representation using 2D Gabor wavelets. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(10):959–971, 1996.

[11] S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[12] T. Maurer and C. v.d. Malsburg. Tracking and learning graphs on image sequences of faces. In *Int. Conf. on Automatic Face- and Gesture-Recognition*, pages 176–181, Killington, Vermont, USA, Oct. 14-16, 1996.

[13] S. McKenna and S. Gong. Recognizing moving faces. In H. Wechsler et.al., editor, *Face Recognition – From Theory to Applications*, NATO ASI Series, pages 578–588. Springer, 1998.

[14] M. Porat and Y. Zeevi. The generalized Garbor scheme of image representation in biological and machine vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(4):452–468, July 1988.

[15] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes, The Art of Scientific Computing.* Cambridge University Press, Cambridge, UK, 1986.

[16] Y. Raja, J. McKenna, and S. Gong. Tracking and segmenting people in varying lighting conditions using color. In *Int. Conf. on Automatic Face- and Gesture-Recognition*, pages 228–233, Nara, Japan, April 14-16, 1998.

[17] H. Szu, B. Telfer, and S. Kadambe. Neural network adaptive wavelets for signal representation and classification. *Optical Engineering*, 31(9):1907–1961, 1992.

[18] L. Wiskott, J. M. Fellous, N. Krüger, and C. v. d. Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):775–779, July 1997.

[19] Q. Zhang and A. Benviste. Wavelet networks. *IEEE Trans. on Neural Networks*, 3(6):889–898, Nov. 1992.