Available online at www.sciencedirect.com
SCIENCE DIRECT®
Real-Time Imaging 10 (2004) 339–350
www.elsevier.com/locate/rti

A wavelet subspace method for real-time face tracking

Rogerio S. Feris[a], Volker Krueger[b], Roberto M. Cesar Jr.[c,*]

[a]Department of Computer Science, University of California, Santa Barbara, CA 93106, USA
[b]Department of Computer Science, Aalborg University Esbjerg, Niels Bohrs Vej 8, 6700 Esbjerg, Denmark
[c]Department of Computer Science, University of São Paulo, Rua do Matão 1010, 05508-900 São Paulo-SP, Brazil

Available online 17 September 2004

Abstract

In this article, we present a new method for visual face tracking that is carried out in a wavelet subspace. Initially, a wavelet representation for the face template is created, which spans a low-dimensional subspace of the image space. The video sequence frames, where the face is tracked, are then orthogonally projected into this subspace. This can be done efficiently through a small number of applications of the wavelet filters. All further computations are performed in the low-dimensional wavelet subspace, allowing real-time processing. An effective performance assessment is carried out to show robustness with respect to facial expression and affine deformations, as well as the efficiency of our method, which allows real-time face tracking.

1. Introduction

This paper addresses the issue of affine real-time face tracking. Real-time (RT, 25/30 Hz) is a major requisite for many human–computer-interface (HCI) and surveillance applications, as well as for tele-conferencing and tele-teaching tasks. For gesture, gaze, and pose estimation applications, tracking has to be not only fast, but also precise. A variety of tracking approaches already exists [1–4] and the literature is briefly reviewed in Section 2. In this work, we will use Gabor wavelet networks (GWN) [5,6] in order to represent the face to be tracked. It is worth saying that we have already discussed a GWN-based real-time tracking method in [7]. The method discussed in the present paper is related to that approach and poses a considerable enhancement in terms of efficiency, while exploiting the sparseness of a wavelet representation.[1]

GWN combine the RBF networks' and the Gabor wavelets' advantages. Objects are represented through a linear combination of Gabor wavelets where the parameters of each of the Gabor functions (such as orientation, position and scale) are optimized to reflect the particular local image structure. The use of Gabor wavelet networks has several advantages, namely

1. By their very nature, Gabor wavelet networks are invariant to some degree to affine deformations and homogeneous changes in image brightness;
2. Gabor filters are good feature detectors [9,10] and the optimized parameters of each Gabor wavelet reflect the underlying image structure;
3. The Gabor wavelet weights are linearly related to the Gabor filter responses and thus also reflect the underlying local image structure;
4. The precision of the representation can be varied to any desired degree ranging from a coarse to an almost photo-realistic representation by simply varying the number of used wavelets. Depending on the available computer power and the necessary tracking precision, the number of wavelets can be dynamically varied.

The above properties allow us to define a subspace of the image space being given by the selectively chosen Gabor wavelets. Property 3 allows an easy projection of any image into the subspace, while property 2 assures

*Corresponding author.
E-mail addresses: rferis@cs.ucsb.edu (R.S. Feris), vok@cs.aue.auc.dk (V. Krueger), cesar@vision.ime.usp.br (R.M. Cesar Jr.).

[1]This paper presents an extended and improved version of the conference paper [8].

1077-2014/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.rti.2004.06.002

the great sparseness of the wavelet representation. These aspects underly the approach discussed in this paper.

This paper is organized as follows. Some related work is discussed in Section 2 while some wavelet preliminaries are presented in Section 3. We will review the GWN framework in Section 4. In Section 5, we will introduce our subspace tracking approach which is shown to be this paper's main contribution as an improvement of the original method. Effective performance assessment results corroborating our approach are described in Section 6.1, while Section 6.2 discusses the main advantages and drawbacks of our method with respect to previous ones. The paper is concluded in Section 7.

2. Related work

Many human face tracking approaches have been proposed over the past years. The originality of each approach, in general, relies on the choice of the face model or representation used for tracking. Examples include holistic representations, ranging from coarse models (e.g. color-based [11]) to templates and parameterized models [12,3], face representations based on contours [13,14], and also representations based on a sparse collection of facial features [15,16]. We briefly discuss some representative methods in this section. 3D face models (e.g., [1]) are not covered, since our work is focused on planar face tracking.

Holistic face representations that combine simple visual cues such as color and motion [11] have been widely used for face tracking. Representing a human face as a moving skin-color blob allows real-time tracking, also robust to out-of-plane face rotations. On the other hand, a coarse model may be confused with other objects of similar color and does not provide precise face location in each video sequence frame.

The work proposed by Hager and Belhumeur [3] describes an efficient framework for planar object tracking. Using a discrete, region-based object representation, tracking is posed as determining parameters that describe the object motion along the sequence. Illumination variation, a major problem in holistic approaches, is handled by lower-dimensional linear subspaces. They also use techniques from robust statistics to treat partial occlusion. The system fails under significant facial expression variation or out-of plane rotations.

Active appearance models (AAMs) were proposed by Cootes et al. [12] as a robust statistical technique for face tracking. In this approach, an appearance face model, encoding both texture and shape information, is built from training data. Tracking is achieved by fitting the learned model along the sequence, through an analysis by synthesis approach. Different facial expressions and illumination changes are well approximated, as long as they are represented in the learning data. Recently, view-based AAMs [17] were proposed to handle out-of-plane face rotations. The optimization process for model fitting is relatively fast, but still does not run in real-time.

Contour-based face models are used to track the elliptical contour of the head or the contour of specific facial features. The idea of active contours, or snakes [14], is to use a curve that is deformed and attracted to an image contour, in response to a force system. The active shape model approach (ASM) [13] is similar in spirit to snakes. The main advantage is that the curve is deformed according to the allowed object deformations, statistically learned from training data. Tracking is posed as determining the shape parameters along the sequence to better fit the image evidence, subject to the shape constraints represented by the model. Both approaches work well on images where the contours are well defined, but may fail in cluttered scenes. Moreover, a good curve initialization must be provided.

Another well-established route toward efficient face tracking is to represent a face based only on a sparse collection of features. Such methods tend to be insensitive to global changes in the intensity and composition of the incident illumination. Maurer and Malsburg [16] use Gabor filters as visual features for face tracking. A set of face points is initially chosen a priori, and for each selected point, a feature vector called jet is computed through the application of Gabor wavelets, with different scales and orientations. Jets are then tracked individually with subpixel accuracy, under large out-of-plane face rotations. However, the method is computationally expensive and does not run in real-time.

Mckenna and Gong [18] combined facial feature tracking based on Gabor jets with an ASM to impose global constraints and improve tracking. They also showed the usefulness of Gabor filtering in head pose estimation [19]. Their face tracking and pose estimation system runs in real-time, but requires specialized hardware for the Gabor wavelet transform computation. Our approach uses a new wavelet representation for the face image that is even sparser than the Gabor jet representation. Our face model is based on wavelets whose parameters and weights are optimized to approximate the original face image. A related face model was proposed by Howell and Buxton [20], which combines Gabor wavelets with RBF networks. But this work considers the Gabor filtering as preprocessing to provide input to the RBF network, while in our method the wavelets are part of the network.

Finally, it is important to mention that the face models discussed above could be part of higher-level tracking frameworks [21,22]. The condensation framework [21] allows probabilistic tracking through the

propagation of the conditional object state distribution over time, based on a dynamic and measurement model. Zhou et al. [23] used condensation for simultaneous tracking and recognition of human faces. More recently, Wang et al. [24] showed that condensation tracking is further improved by learning the intrinsic object structure in embedded space. In addition to condensation, the incremental focus of attention (IFA) [22] framework could be used as a layer-based representation for fast tracking recovery.

3. Wavelet preliminaries

The wavelet transform is a mathematical tool that has been developed and applied in very different fields such as signal and image processing, astronomy, medicine and finances, to name but a few [25–28]. This section discusses some of the main concepts related to the 2D wavelet transform [26].

The gray-level image can be seen as a function $f(x,y)$, $(x,y) \in \mathbb{R}^2$, where $\mathbb{R}^2 = \mathbb{R} \times \mathbb{R}$ denotes the real plane. Formally, the image $f(x,y)$ is represented as a square integrable (i.e. finite energy) function defined over $\mathbb{R}^2$, which is denoted as $f \in \mathbb{L}^2(\mathbb{R}^2)$ [25]. We also denote 2D points in the real plane as bold letters for simplicity's sake, e.g. $\mathbf{x} = (x,y)$. Therefore, $f(\mathbf{x})$ represents a gray-level image. The continuous wavelet transform of a 2D signal $f(\mathbf{x})$ can be defined as

$$W_\psi(\mathbf{b},\theta,s) = C_\psi^{-1/2}\,\frac{1}{s}\int \psi^*(s^{-1}r_{-\theta}(\mathbf{x}-\mathbf{b}))f(\mathbf{x})\,\mathrm{d}^2\mathbf{x} \qquad (1)$$

where C, ψ, $\mathbf{b}$, θ, and s stand for the normalizing constant, the mother wavelet, the translation vector, the rotation angle and the dilation (i.e. scale) parameter, respectively. ψ^* denotes the complex conjugate of the mother wavelet ψ, which is a function that is well-localized in the space and in the frequency domains and presents zero mean (i.e. null DC component) [25]. The resulting transform is often complex, depending on the chosen wavelet. There are many alternatives for visualizing and interpreting the obtained wavelet transform, and the reader is referred to [29] as a good reference on this subject.

One of the most popular mother wavelets is the Gabor wavelet [8], also called Morlet [25], which can be efficiently tuned to specific frequencies [30,25]. This property allows the application of this wavelet to detect specific image structures of interest whilst leaving undesirable artifacts such as noise out. Furthermore, the Gabor wavelet is known to be directional, in the sense of being effective in selecting orientations. The 2D Gabor wavelet is defined as

$$\psi_M(\mathbf{x}) = \exp(\mathrm{j}\mathbf{k}_0\mathbf{x})\,\exp\left(-\tfrac{1}{2}\|A\mathbf{x}\|^2\right) \qquad (2)$$

where $j = \sqrt{-1}$ and

$$A = \begin{bmatrix} \varepsilon^{-1/2} & 0 \\ 0 & 1 \end{bmatrix}, \quad \varepsilon \geqslant 1 \qquad (3)$$

is an array that defines the anisotropy of the filter, i.e. its elongation in a given direction [25]. In other words, the larger ε is, the more elongated is the mother wavelet along the x-axis with respect to the y-axis. By varying the rotation parameter θ of the wavelet transform (Eq. (1)), it is possible to orient the elongated wavelet in any desirable direction.

The Gabor wavelet is a complex exponential multiplying a 2D Gaussian, where $\mathbf{k}_0$ is a vector that defines the frequency of the complex exponential in the x and in the y directions, i.e.

$$\mathbf{k}_0 = \begin{bmatrix} k_x \\ k_y \end{bmatrix}. \qquad (4)$$

The Gabor wavelet is shown in Fig. 1(a) as a surface and 1(b) as an image. Because this wavelet is represented on a complex number plane, only the real part is shown.

The role played by the wavelet parameters is fundamental, both the transform parameters (i.e. $\mathbf{b}$, θ, and s, see Eq. (1)) and the Gabor wavelet parameters. The size of the mother wavelet is changed by varying the scale parameter s. The translation parameter $\mathbf{b}$ is responsible for shifting the mother wavelet throughout the image and is therefore important in detecting and representing specific structures that may appear anywhere in the image. By varying the orientation parameter θ, an anisotropic mother wavelet selects structures in different orientations. As far as the Gabor wavelets parameters are concerned, when the frequency parameter k_y is increased, the number of wavelet oscillations in the y direction is also increased. On the other hand, changing the elongation parameter may lead to wavelets from a wavelet with no elongation to a much more elongated one in a given direction. It is clear from these considerations that the parameters may be configured in different ways in order to make the wavelet suitable for analyzing image structures of different shapes, which is explored by the face representation approach explained below.

The existence of so many parameters that vary continuously lead to a transform flexible enough to represent the image information in a suitable way. Furthermore, the fact that the wavelet transform of Eq. (1) has an inverse means that no information is lost. Nevertheless, that wavelet transform is seldom calculated and used by varying all parameters and it can obviously not be visualized in such high dimension parameter space. Instead, some approaches attempt to keep some parameters fixed. For instance, the scale and angle parameters (s and θ, respectively) may be kept fixed for some a priori defined values $s = s_0$ and $\theta = \theta_0$,

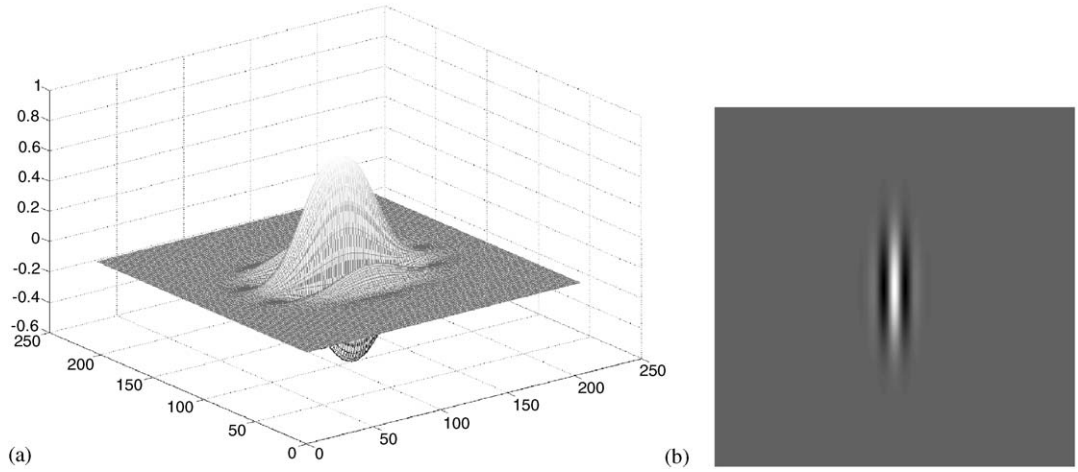

Fig. 1. Gabor (or Morlet) wavelet shown as a surface in (a) and as an image in (b).

with the resulting wavelet transform being represented as a 2D function defined in Eq. (5) (the so-called *position representation* [25])

$$W_\psi(\mathbf{b}) = C_\psi^{-1/2}\frac{1}{s_0}\int \psi^*(s_0^{-1}r_{-\theta_0}(\mathbf{x}-\mathbf{b}))f(\mathbf{x})\,\mathrm{d}^2\mathbf{x}. \tag{5}$$

The approach followed in this paper explores a different idea. Instead of keeping some of the parameters fixed, an optimization algorithm is applied in order to find a good parameter configuration to represent a face, as explained in the next section.

4. Gabor wavelet networks

The basic idea of the wavelet networks was first stated in [31], and the use of Gabor functions is inspired by the fact that they are recognized to be good feature detectors [9,10].

The GWN is explored by taking the imaginary part of the original Gabor wavelet (which is complex-valued). Furthermore, the complex exponential frequency is taken as 1 and the elongation parameter is controlled by taking two different scale parameters associated to the x- and y-axes, i.e. s_x and s_y. To define a GWN, we start out, generally speaking, by taking a family of N odd Gabor wavelet functions $\boldsymbol{\Psi} = \{\psi_{\mathbf{n}_1}, \ldots, \psi_{\mathbf{n}_N}\}$ of the form

$$\begin{aligned}\psi_{\mathbf{n}}(x,y) = {} & \exp\left(-\tfrac{1}{2}[s_x((x-c_x)\cos\theta-(y-c_y)\sin\theta)]^2\right.\\ & \left.+[s_y((x-c_x)\sin\theta+(y-c_y)\cos\theta)]^2\right)\\ & \cdot\sin(s_x((x-c_x)\cos\theta-(y-c_y)\sin\theta)),\end{aligned} \tag{6}$$

with $\mathbf{n} = (c_x, c_y, \theta, s_x, s_y)^{\mathrm{T}}$. Here, c_x, c_y denote the translation of the Gabor wavelet, s_x, s_y denote the dilation, and θ denotes the orientation. The choice of N is arbitrary and related to the degree of desired representation precision of the network. In order to find the GWN for a function $f \in \mathbb{L}^2(\mathbb{R}^2)$ (f dc-free, w.l.o.g.) the energy functional

$$E = \min_{\mathbf{n}_i, w_i \text{ for all } i}\left\|f - \sum_i w_i\psi_{\mathbf{n}_i}\right\|_2^2 \tag{7}$$

is essentially minimized with respect to the weights w_i and the wavelet parameter vector $\mathbf{n}_i$. The wavelet parameter vectors $\mathbf{n}_i$ are optimized under a side condition of a Lagrange multiplier $\phi(\mathbf{n}_1, \ldots, \mathbf{n}_N)$. The function ϕ assures that the $\mathbf{n}_i$ do not diverge but stay within a range as specified in [32].

The two vectors

$$\boldsymbol{\Psi} = (\psi_{\mathbf{n}_1}, \ldots, \psi_{\mathbf{n}_N})^{\mathrm{T}}$$

and

$$\mathbf{w} = (w_1, \ldots, w_N)^{\mathrm{T}}$$

define then the *Gabor wavelet network* $(\boldsymbol{\Psi}, \mathbf{w})$ for the function f.

In other words, considering a discrete image[2] I, a Gabor wavelet network is defined through an N-dimensional vector of weights w_i and an N-dimensional vector of Gabor wavelets $\psi_{\mathbf{n}_i}$, where the weights w_i and the parameter vectors $\mathbf{n}_i$ are chosen such that the weighted sum of Gabor wavelets $\psi_{\mathbf{n}_i}$ approximates the discrete gray value image I optimally.

From the optimal wavelets $\boldsymbol{\Psi}$ and weights $\mathbf{w}$ of the Gabor wavelet network, the function f can be (closely)

[2]We use the notation $f, g, \ldots$ to generally refer to continuous functions, while we use $I, J, \ldots$ when we explicitly refer to discrete gray value images.

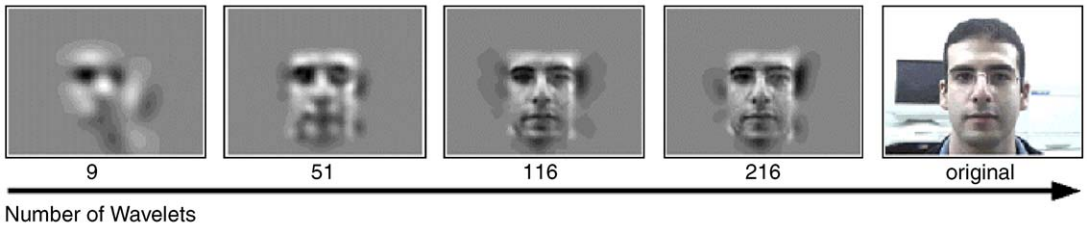

Fig. 2. Face images reconstructed with different number of wavelets.

reconstructed by a linear combination of the weighted wavelets

$$\hat{f} = \sum_{i=1}^{N} w_i \psi_{\mathbf{n}_i} = \boldsymbol{\Psi}^{\mathrm{T}} \mathbf{w}.$$

Clearly, the quality of the image representation and reconstruction depends on the number N of wavelets and can be varied to reach almost any desired precision.

An example of reconstruction can be seen in Fig. 2. In this example, a family of 216 wavelets has been distributed over the inner face region of the rightmost image I by minimizing Eq. (7). Different reconstructions $\hat{I}$ obtained by the application of Eq. (8) for different values of N are shown.

4.1. Direct calculation of weights

The weights w_i of a GWN are directly related to the local filter responses of the Gabor filters $\psi_{\mathbf{n}_i}$. Gabor wavelet functions are not orthogonal, thus implying that, for a given family $\boldsymbol{\Psi}$ of Gabor wavelets, it is not possible to calculate a weight w_i by a simple projection of the image onto the Gabor wavelet $\psi_{\mathbf{n}_i}$. In fact, a family of dual wavelets $\tilde{\boldsymbol{\Psi}} = \{\tilde{\psi}_{\mathbf{n}_1} \ldots \tilde{\psi}_{\mathbf{n}_N}\}$ has to be considered. The wavelet $\tilde{\psi}_{\mathbf{n}_j}$ is the dual wavelet of $\psi_{\mathbf{n}_i}$ iff $\langle \psi_{\mathbf{n}_i}, \tilde{\psi}_{\mathbf{n}_j} \rangle = \delta_{i,j}$. With $\tilde{\boldsymbol{\Psi}} = (\tilde{\psi}_{\mathbf{n}_1}, \ldots, \tilde{\psi}_{\mathbf{n}_N})^{\mathrm{T}}$, we can write $[\langle \boldsymbol{\Psi}, \tilde{\boldsymbol{\Psi}} \rangle] = \mathbb{1}$. In other words, given $g \in \mathbb{L}^2(\mathbb{R}^2)$ and a GWN $\boldsymbol{\Psi} = \{\psi_{\mathbf{n}_1}, \ldots, \psi_{\mathbf{n}_N}\}$, the optimal weights for g are given by $w_i = \langle g, \tilde{\psi}_{\mathbf{n}_i} \rangle$. It can be shown that $\tilde{\psi}_{\mathbf{n}_i} = \sum_j (\boldsymbol{\Psi}^{-1})_{i,j} \psi_{\mathbf{n}_j}$, where $\boldsymbol{\Psi}_{i,j} = \langle \psi_{\mathbf{n}_i}, \psi_{\mathbf{n}_j} \rangle$ [6].

The above equations allow us to define the operator

$$\mathscr{T}_{\boldsymbol{\Psi}} : \mathbb{L}^2(\mathbb{R}^2) \longmapsto \langle (\psi_{\mathbf{n}_1}, \ldots, \psi_{\mathbf{n}_N}) \rangle \tag{8}$$

as follows: given a set $\boldsymbol{\Psi}$ of optimal wavelets of a GWN, the operator $\mathscr{T}_{\boldsymbol{\Psi}}$ represents an orthogonal projection of a function g onto the closed linear span of $\boldsymbol{\Psi}$ (see Eq. (8) and Fig. 3), i.e.

$$\hat{g} = \mathscr{T}_{\boldsymbol{\Psi}}(g) = g\tilde{\boldsymbol{\Psi}}\boldsymbol{\Psi} = \sum_{i=1}^{N} w_i \psi_{\mathbf{n}_i} \quad \text{with } \mathbf{w} = g\tilde{\boldsymbol{\Psi}}. \tag{9}$$

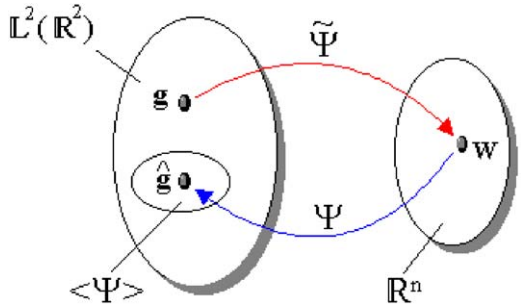

Fig. 3. A function $g \in \mathbb{L}^2(\mathbb{R}^2)$ is mapped by the linear mapping $\tilde{\boldsymbol{\Psi}}$ into the vector $\mathbf{w} \in \mathbb{R}^N$. The mapping of $\mathbf{w}$ into $\mathbb{L}^2(\mathbb{R}^2)$ is achieved with the linear mapping $\boldsymbol{\Psi}$. Both mappings constitute an orthogonal projection of a function $g \in \mathbb{L}^2(\mathbb{R}^2)$ into the subspace $\langle \boldsymbol{\Psi} \rangle \subset \mathbb{L}^2(\mathbb{R}^2)$.

5. Face tracking in wavelet subspace

In this section, we will present our efficient approach for real-time face tracking in wavelet subspace. For this, we first consider a GWN $(\boldsymbol{\Psi}, \mathbf{v})$ that is optimized for a certain face image. As previously mentioned, $\langle \boldsymbol{\Psi} \rangle$ defines a low-dimensional wavelet subspace, while $\mathbf{v}$ is the optimal weight vector obtained by an orthogonal projection of the facial image onto the closed linear span of $\boldsymbol{\Psi}$. We will call the components of $\mathbf{v}$ *reference weights*.

Since the affine deformation is a native degree of freedom of a GWN, we exploit this property to perform face tracking. Basically, the set of basis functions $\{\psi_{\mathbf{n}_1}, \ldots, \psi_{\mathbf{n}_N}\}$ that span the wavelet subspace $\langle \boldsymbol{\Psi} \rangle$ is affinely distorted, until the set of weights, obtained by the orthogonal projection of the current frame into this subspace, is nearest to the reference weight vector $\mathbf{v}$. In other words, we deform the wavelet network to match the face image in each frame, and this can be carried out efficiently in the low-dimensional wavelet subspace. An affine deformation of a wavelet network (and, respectively, the wavelet subspace $\langle \boldsymbol{\Psi} \rangle$) is carried out by considering the entire wavelet network as a single wavelet [6]

$$\boldsymbol{\Psi}_{\mathbf{n}}(\mathbf{x}) = \sum_i w_i \psi_{\mathbf{n}_i}(\mathbf{SR}(\mathbf{x} - \mathbf{c})), \tag{10}$$

where the vector $\mathbf{n} = (c_x, c_y, \theta, s_x, s_y, s_{xy})$[3] defines the dilation matrix $\mathbf{S}$, the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{c}$, respectively.

[3] To include the shear in the parameter set of the wavelets, see [6].

The orthogonal projection of a face image I into the subspace $\langle \Psi \rangle$, given by the set of weights $\mathbf{w}$, is computed very fast through a small number of wavelet filtrations. Recall from Section 4.1 that $w_i = \langle I, \tilde{\psi}_{\mathbf{n}_i} \rangle$, where $\tilde{\psi}_{\mathbf{n}_i} = \sum_j (\Psi^{-1})_{i,j} \psi_{\mathbf{n}_j}$. This corresponds to the following equation:

$$w_i = \sum_j (\Psi^{-1})_{i,j} \langle I, \psi_{\mathbf{n}_j} \rangle. \tag{11}$$

Thus, the optimal weights w_i are derived from a linear combination of wavelet filtrations, where the coefficients are given by the inverse of matrix $\Psi_{i,j} = \langle \psi_{\mathbf{n}_i}, \psi_{\mathbf{n}_j} \rangle$. It can be shown that the matrix $\Psi_{i,j}$ is, except for a scalar factor, invariant with respect to affine transformations of the wavelet network. It can therefore be computed offline and beforehand. Hence, the weights w_i are computed efficiently with Eq. (11) through a local application $\langle I, \psi_{\mathbf{n}_i} \rangle$ for each of the N Gabor wavelets $\psi_{\mathbf{n}_i}$.

Let $\mathbf{n} = (c_x, c_y, \theta, s_x, s_y, s_{xy})$ be an affine parameter vector which describes a network parameterization. Our objective is to gradually change these parameters until the orthogonal projection of the current frame J into the wavelet subspace $\langle \Psi \rangle$, given by the set of weights $\mathbf{w}$, is nearest the reference weights $\mathbf{v}$. In other words, we must optimize the parameters $\mathbf{n}$ with respect to the energy functional

$$E = \min_{\mathbf{n}} \|\mathbf{v} - \mathbf{w}\|_{\Psi} \tag{12}$$

with

$$w_i = \sum_j \frac{1}{s_x \cdot s_y} (\Psi^{-1})_{i,j} \langle J, \psi_{\mathbf{n}_j}(\mathbf{SR}(\mathbf{x} - \mathbf{c})) \rangle, \tag{13}$$

where $\mathbf{S}$ is the dilation matrix, $\mathbf{R}$ is the rotation matrix and $\mathbf{c}$ is the translation vector, all defined through the vector $(c_x, c_y, \theta, s_x, s_y, s_{xy})$. During tracking, this optimization is done for each successive frame. As there is not much difference between adjacent frames, the optimization is fast. To minimize the energy functional (12), the Levenberg–Marquardt algorithm was used.

In Eq. (12) we used the notation $\|\mathbf{v} - \mathbf{w}\|_{\Psi}$ to refer to a distance between vectors $\mathbf{v}$ and $\mathbf{w}$ of the subspace $\langle \Psi \rangle$. However, it is not yet clear how this distance measure should be calculated. One could use the Euclidean distance between two vectors, as done in [33]. However, that distance measure misses any interpretation in this context and all wavelets are equally treated, even though they might be of different scales.

Therefore, we propose a different distance measure, which is based on the Euclidean distance between vectors of the image space. We thus define the difference $\|\mathbf{v} - \mathbf{w}\|_{\Psi}$ as the Euclidean distance between the two respective images f and g

$$\begin{aligned} \|\mathbf{v} - \mathbf{w}\|_{\Psi} &= \|\mathscr{T}_{\Psi}(f) - \mathscr{T}_{\Psi}(g)\|_2 \\ &= \left\| \sum_{i=1}^{N} v_i \psi_{\mathbf{n}_i} - \sum_{j=1}^{N} w_j \psi_{\mathbf{n}_j} \right\|_2. \end{aligned} \tag{14}$$

Various transformations lead to

$$\begin{aligned} \|\mathbf{v} - \mathbf{w}\|_{\Psi} &= \left[\sum_{i,j} (v_i - w_i)(v_j - w_j) \langle \psi_{\mathbf{n}_i}, \psi_{\mathbf{n}_j} \rangle \right]^{1/2} \\ &= (\mathbf{v} - \mathbf{w})^t \Psi_{i,j} (\mathbf{v} - \mathbf{w}). \end{aligned} \tag{15}$$

The matrix of pairwise scalar products $\Psi_{i,j} = \langle \psi_{\mathbf{n}_i}, \psi_{\mathbf{n}_j} \rangle$ is the same as the one in Section 4.1 and Eq. (11). Note that if the wavelets $\{\psi_{\mathbf{n}_i}\}$ were orthogonal, then Ψ would be the unity matrix and Eq. (15) would describe the same distance measure as proposed in [33].

The method presented here is related to the one described in [7], which also uses GWN for face representation. However, that technique performs tracking through a typical pattern matching in the image-space, using the wavelet representation as template. Compared to that method, our approach poses a considerable enhancement in terms of efficiency, as it provides a great data reduction, considering that tracking is done in a low-dimensional wavelet subspace. Indeed, a similar pattern matching is carried out (see Eqs. (12) and (14)), but this is done based only on wavelet weights, sparing out the computationally demanding template reconstruction and pixel-wise sum-of-squared-difference computation required in the image-based tracking.

6. Experiments

In this section, we describe our experimental results and then discuss the advantages and drawbacks of our technique with respect to other face tracking methods.

6.1. Results

The proposed approach for affine face tracking was successfully tested on various image sequences (see http://www.vision.ime.usp.br/~cesar/journals/rti04-tracking/ for a representative set of test sequences). The tracking videos in this directory are: *demo.mpg*, which shows a challenging sequence with variations in expression, pose and lighting; *demo2.avi*, which demonstrates tracking under fast motion and background distractions; *failure.avi*, a typical failure case situation; *person11.avi* and *person12.avi*, which shows two examples of a database of 40 sequences that we collected for facial analysis using tracking information; *trackscale.avi*, which demonstrates tracking under large scale changes; *optimized.avi* and *non-optimized.avi*, a

comparison between tracking with adaptive and non-adaptive filters. All test sequences showed a person in motion, more or less frontal to the camera, so that the facial features are always visible. Note that we show the position of specific facial features in the tracking videos. This is just to show the precision of our results; it does not mean that the features were tracked individually.

Our experiments demonstrate the ability of our method to track a face as it undergoes changes in pose and expression. Although we have represented the face as a rigid object undergoing limited motion, we realize that different face expressions and small depth variations exhibited by facial features are enough to be well-approximated by the affine wavelet model. Furthermore, since Gabor wavelets are DC free, our approach also showed robustness with respect to homogeneous changes in image brightness.

Concerning tracking precision, we have evaluated the proposed method with respect to the number of wavelets used in the representation. As the first step, we recorded a video sequence that shows a person undergoing different poses and facial expressions (*demo.mpg*). On the face of that person, a GWN with 116 Gabor wavelets was optimized and used to estimate the "ground truth" affine parameters in each frame. A human operator adjusted the frames where facial feature points were not precisely located, for the ground truth estimation. Fig. 4 illustrates some frames of our subspace tracking results, using this wavelet representation. Note that the tracking method is robust to variations in rotation, scale, and strong expression changes. Fig. 5 shows tracking frames under large scale changes (see *trackscale.avi* for this test sequence).

Now, we want to analyze the decrease of precision when using a smaller number of wavelets in the representation. For that, from the large GWN above, GWNs that contained only the largest 51, 22 and 9 wavelets, sorted according to decreasing normalized weights, were used for tracking. The graphs in Fig. 6 depict the estimation of the face parameters x-position, y-position and angle θ as well as the ground-truth in each frame.

In the graphs, we can see that the tracking precision increases with an increasing number of applied wavelets. On the other hand, the tracking speed is improved when the number of wavelets is small. Clearly, the number of applied wavelets is task dependent and can be dynamically changed. Within a specific application (e.g., face tracking), we could evaluate the computational power and adjust *automatically* the number of wavelets to keep a specific frame rate. For other applications (e.g. facial expression synthesis), we would require more wavelets for better reconstruction, but this is a task-dependent selection.

Fig. 4. Sample frames of our wavelet subspace tracking. Note that the tracking method is robust to facial expressions variations as well as affine deformations of the face image.

Fig. 5. Tracking frames under large-scale variation.

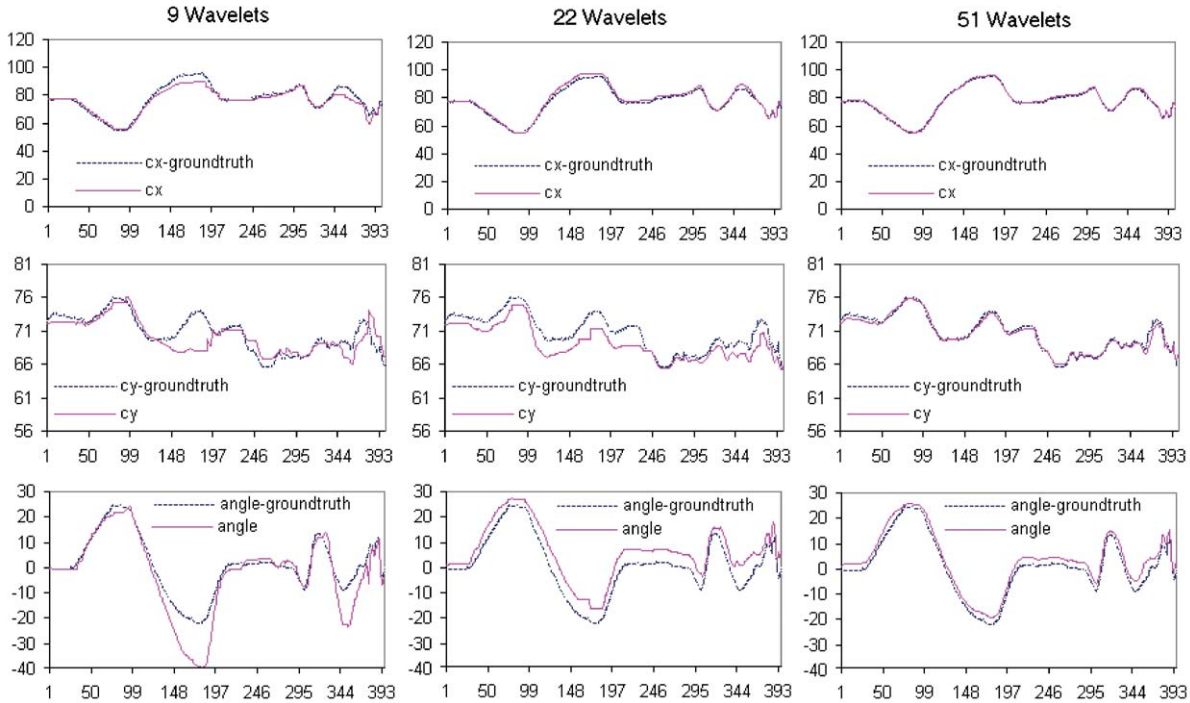

Fig. 6. Estimation of the face parameters x-position, y-position and angle θ in each frame, using GWNS with 9, 22 and 51 wavelets. The ground-truth is depicted to illustrate the decrease of precision when considering few wavelets.

The resolution of the image frames showed in Fig. 4 is 160×120 pixels and the size of the inner face region in which the GWN was optimized is 50×65 pixels. Using only 9 wavelets, the computing time for each Levenberg–Marquardt cycle was 15 ms on a 1 GHz Linux-Athlon. Higher performance is achieved for smaller face regions, or fewer parameters (e.g. just translation and rotation). It is worth saying that the number of cycles for each frame depends on the distance of the initial parameters from the local minimum, which is directly related to object speed. Increasing the number of wavelets in the representation leads to a more precise but slower tracking. For instance, using 51 wavelets, a computing time of 85 ms per cycle was required in each frame.

We compared tracking efficiency with the work we proposed in [7]. This work also makes use of a Gabor wavelet network representation, but tracking is carried out in image domain, requiring a computationally expensive pixelwise operation in each frame. In contrast, our technique is efficiently carried out in wavelet subspace. Comparing with the image-based method, we observed in our experiments a speedup of a factor two for each single Levenberg–Marquardt cycle, but with a slight increase in the number of needed cycles. Fig. 7 illustrates this comparison.

In our experiments, we verified that, by considering a GWN of 40×40 pixels, the optimization correctly converges in each frame if the inter-frame motion is approximately 10 pixels in x and y position, 10 degrees in orientation and 20% in scale. This allows tracking under fast object motion, as shown in the *demo2.avi* video sequence.

An interesting question to be considered is whether tracking with non-adaptive wavelets would achieve similar results as tracking based on wavelet networks.

To answer this question, we compared tracking in both cases, using a small number of wavelets. More specifically, we considered an example where the filters are initially displaced in a 2×2 and 3×3 grid in the inner face region, with random scales and orientations. If we keep such wavelet parameters fixed, and make only the weights adaptive, the image reconstruction is poor and tracking fails, as shown in Fig. 8 (upper row). For testing tracking based on wavelet networks, we used the same set of wavelets for the network initialization. After making both wavelet parameters and weights adaptive, the representation encodes more information and tracking is carried out successfully (Fig. 8, bottom row). The video sequences *optimized.avi* and *non-optimized.avi* illustrate this comparison. We also refer to the work of one of the authors of this paper, which demonstrates the advantages of using adaptive wavelets for pose estimation [5]. This could be explored in view-based tracking for handling out-of-plane face rotations.

6.2. Discussion

Gabor filters as measurement models have been successfully employed in state-of-the-art face tracking systems (see e.g., Eyematic/Neven Vision product—http://www.nevenvision.com/product-ffT.html). When compared to other methods, our technique would inherit the same pros and cons of traditional Gabor filter tracking methods [16,34], which considers wavelets with fixed parameters. However, for a small number of filters (desired for efficient tracking), our approach offers superior performance (as shown in Fig. 8), due to the fact that adaptive wavelets provide a richer compact face representation. Moreover, our method is carried out efficiently in wavelet subspace, thus allowing real-time processing.

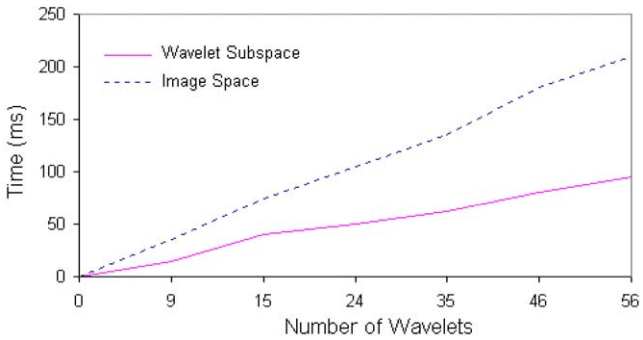

Fig. 7. Comparison of tracking carried out in image space and wavelet subspace.

Why not use other subspace methods (such as PCA [35]) or LDA [36]) for face tracking, rather than wavelet networks? Most subspace methods only work in very specific scenarios. PCA, for instance, generally only works if the data is distributed normally in the feature space. For most applications e.g., for faces in real-world scenarios, the data distribution is not known and PCA has already shown limited success. In this context, support vector machines [37] become very attractive. They are closely related to RBF networks and exploit the fact that the basis functions are positive semi-definite. Indeed, a publication on the theory of networks, very closely related to our work [38], studies the optimization of the weights and parameters of radial basis functions under the Tikhonov stabilizer. Though not stated explicitly, we use a similar method in order to find the weights and parameters for each of the network wavelets. Wavelet networks and SVMs in this respect have a great similarity, however with the major difference that an aquivilent of the Mercer's theorem for odd functions is missing, which would allow to decouple the odd Gabor wavelets into two orthogonal components (as done with the SVM-kernels). Given the good experiences with Gabor wavelet networks, a deeper understanding of this subspace method is wishful. Presently, there is interesting ongoing research to investigate and understand that problem [39].

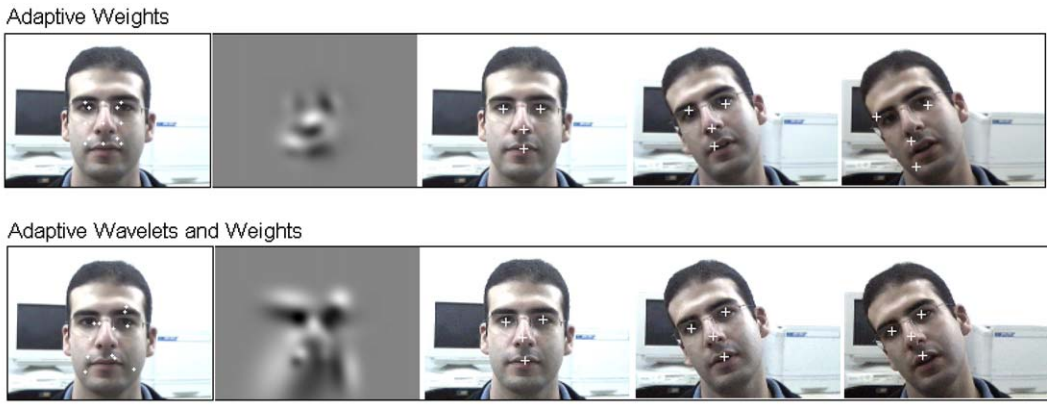

Fig. 8. Tracking comparison using adaptive and non-adaptive filters. Note that making both wavelets and weights adaptive allows better image reconstruction and tracking with a small number of filters.

Even though we do not consider an explicit deformation parameter (such as in deformable models like AAM [12] and ASM [13]), our sparse representation with spatially localized wavelets allows great robustness with respect to large expression changes. We refer to the video sequence *demo.mpg*, which shows our tracking result under large facial deformations—a typical case where traditional correlation-based methods would fail. Moreover, since we have a feature-based wavelet representation, partial occlusions and outliers have a local effect in our model. Thus, a robust norm, such as Geman McClure [40] could be used to discard outliers. This compares favorably with other holistic representations such as PCA, where the error due to e.g. partial occlusions is spread over all the image.

As already mentioned, Gabor filters are DC-free, thus achieving a certain amount of localized normalization for illumination. Fig. 9 illustrates two frames of the test video sequence showed in Fig. 4 with respective histograms, showing the lighting variation. We note, however, that our method fails under strong illumination changes. In such cases, we suggest the use of explicit models for illumination compensation, such as those based on spherical harmonics [41].

In this paper, we have not handled the issue of out of plane face rotations. Fig. 10 shows a typical failure example, where the user turns his head away from the camera. On the other hand, the works of McKenna and Gong [19] and Maurer and Malsburg [16] have proven the usefulness of Gabor wavelets for coping with head pose variations. Gabor filtered pose manifolds in fact offer more linearity than intensity-based pose manifolds [42]. We intend to exploit this information in a view-based scheme for achieving tracking under large pose variations.

It is well known that Gabor filters are good feature detectors [9,10]. They have been widely used for feature localization [33], but reported tracking methods are computationally expensive. With a small set of optimally selected filters, we are able to achieve real-time tracking, while keeping good accuracy.

GWNs invite the closest comparison with the well-known Gabor jet approach, which was also used for face tracking [16]. The advantage of GWNs is that they offer a sparser representation of image data. This is because the wavelet parameters are selectively chosen from the continuous space, in contrast with the Gabor jet approach, which is based on the discrete wavelet transform. For example, considering just 52 wavelets, GWNs provide a good representation for a face image (see Fig. 2), whereas the Gabor jet approach would require many more wavelets to get a comparable representation. In our search, we have not seen a face tracking approach based on Gabor filters with parameters and weights optimally tuned. This allows *efficient* tracking with a small number of wavelets, which would not be possible if the filters are not made adaptive.

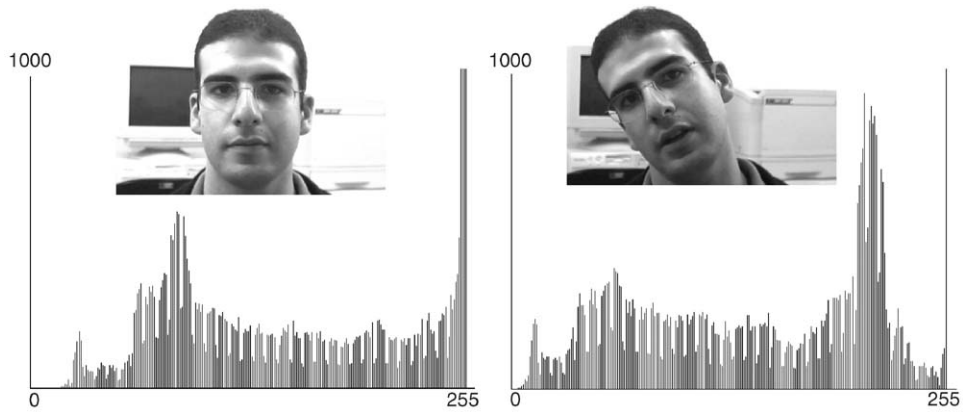

Fig. 9. Two frames of the test video sequence showed in Fig. 4 with respective histograms.

Fig. 10. A typical failure situation, where the user turns his head away from the camera.

Finally, it is also worth mentioning the work of Hager and Belhumeur [3], which uses an affine model to track the face. This work has the advantage of being even faster than GWN tracking. On the other hand, the GWN approach avoids the discrete data interpolation required in the former, since, in this case, we deal with a continuous wavelet representation.

7. Conclusions

We have presented a novel method for real-time face tracking which is carried out in wavelet subspace. Our method uses a face representation based on Gabor wavelet networks (GWNs), which allows us to perform efficient tracking under affine face deformations and homogeneous changes in image brightness. This method has been used by a graph-based facial feature segmentation algorithm described in [43], leading to successful results. Our experiments also show invariance to significant expression changes.

As future work, we plan to address the problems of out of plane face rotations and significant illumination changes, using a view-based scheme and an explicit model for illumination compensation.

Our method could also be part of more general tracking frameworks. For instance, the *incremental focus of attention* framework [22] could be used for fast failure recovery, having our wavelet subspace tracker as the feature tracking layer. Another example is the Condensation framework [21], where our wavelet features could be used for an effective measurement model. In this case, Condensation tracking would replace our Levenberg–Marquardt optimization, possibly reducing problems with local minima.

Acknowledgements

This work was partially funded under the Grants FAPESP (99/01487-1), FAPESP (99/12765-2) and CNPq (300722/98-2). The valuable comments made by the reviewers helped us improve the paper in different ways.

References

[1] Ahlberg J. Using the active appearance algorithm for face and facial feature tracking. In: ICCV'01 workshop on recognition, analysis and tracking of faces and gestures in real-time systems. Vancouver, BC, Canada; 2001. p. 68–72.

[2] Comaniciu D, Ramesh V, Meer P, Real-time tracking of non-rigid objects using mean shift. In: IEEE conference on computer vision and pattern recognition, vol. 2, Hilton Head Island, SC, 2000. p. 142–49.

[3] Hager G, Belhumeur P. Efficient region tracking with parametric models of geometry and illumination. IEEE Transactions on Pattern Analysis and Machine Intelligence 1998;20(10):1025–39.

[4] Ji Q, Yang X. Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. Real-Time Imaging 2002;8(5): 357–77.

[5] Krüger V, Sommer G. Efficient head pose estimation with gabor wavelet networks. Image and Vision Computing 2002;20:665–72.

[6] Krueger V, Sommer G. Gabor wavelet networks for object representation. Journal of the Optical Society of America 2002;19(6):1112–9.

[7] Krueger V, Happe A, Sommer G, Affine real-time face tracking using gabor wavelet networks, in: Proceedings of the International conference on pattern recognition—ICPR-2000, vol. 1, 2000. p. 127–30.

[8] Feris RS, Krueger V, Cesar Jr RM. Efficient real-time face tracking in wavelet subspace. In: Proceedings of the second international workshop on recognition, analysis and tracking of faces and gestures in real-time systems, 8th IEEE international conference on computer vision (RATFG-RTS-ICCV 2001) Vancouver, Canada; 2001. p. 113–8.

[9] Manjunath BS, Chellappa R. A unified approach to boundary perception: edges, textures, and illusory contours. IEEE Transactions on Neural Networks 1993;4(1):96–107.

[10] Mehrotra R, Namuduri K, Ranganathan R. Gabor filter-based edge detection. Pattern Recognition 1992;52(12):1479–94.

[11] Raja Y, McKenna S, Gong S. Tracking and segmenting people in varying lighting conditions using color. In: International conference on automatic face and gesture recognition. Nara, Japan; 1998. p. 228–33.

[12] Cootes T, Edwards G, Taylor C. Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence 2001;23(6):681–5.

[13] Cootes T, Cooper D, Taylor C, Graham J. Active shape models—their training and application. Computer Vision and Image Understanding 1995;61(1):38–59.

[14] Kass M, Witkin A, Terzopoulos D. Snakes: active contour models. In: First international conference on computer vision; 1987. p. 259–68.

[15] Colmenarez A, Frey B, Huang T. Detection and tracking of faces and facial features. In: International Conference on Image Processing; 1999. p. 657–61.

[16] Maurer T, von der Malsburg C. Tracking and learning graphs and pose on image sequences of faces. In: International conference on automatic face and gesture recognition; 1996. p. 176–81.

[17] Cootes T, Wheeler G, Walker K, Taylor C. View-based active appearance models. Image and Vision Computing 2002;20:657–64.

[18] McKenna S, Gong S, Würtz R, Tanner J, Banin D. Tracking facial feature points with Gabor wavelets and shape models. In: International conference on audio and video-based biometric person authentication. Crans-Montana, Switzerland; 1997. p. 35–42.

[19] McKenna S, Gong S. Real-time face pose estimation. Real-Time Imaging 1998;4.

[20] Howell J, Buxton H. Towards unconstrained face recognition from image sequences. In: International conference on automatic face and gesture recognition; 1996. p. 224–9.

[21] Isard M, Blake A. Condensation—conditional density propagation for visual tracking. International Journal of Computer Vision 1998;29(1):5–28.

[22] Toyama K, Hager G. Incremental focus of attention for robust vision-based tracking. International Journal of Computer Vision 1999;35(1):45–63.

[23] Zhou S, Krueger V, Chellappa R. Face recognition from video: a condensation approach. In: International conference on automatic face and gesture recognition. Washington DC, USA; 2002. p. 224–9.

[24] Wang Q, Xu G, Ai H. Learning object intrinsic structure for robust visual tracking. In: International conference on computer vision and pattern recognition. Wisconsin, USA; 2003. p. 227–33.
[25] Antoine J-P, Carette P, Murenzi R, Piette B. Image analysis with two-dimensional continuous wavelet transform. Signal Processing 1993;31:241–72.
[26] Arnéodo A, Decoster N, Roux SG. A wavelet-based method for multifractal image analysis. i. methodology and test applications on isotropic and anisotropic random rough surfaces. The European Physical Journal B 2000;15:567–600.
[27] Daugman JG. Complete discrete 2-d Gabor transforms by neural networks for image analysis and compression. IEEE Transactions on Acoustics, Speech, and Signal Processing 1988;36(7):1169–79.
[28] Mallat S, Hwang WL. Singularity detection and processing with wavelets. IEEE Transactions on Information Theory 1992;38(2):617–43.
[29] Grossmann A, Kronland-Martinet R, Morlet J. Reading and understanding continuous wavelet transform. In: Combes J-M, Grossmann A, Tchamitchian P editors. Wavelets, time–frequency methods and phase-space (Proceedings of Marseille 1987). Berlin: Springer; 1989.
[30] Antoine JP, Barache D, Cesar Jr RM, Costa LF. Shape characterization with the wavelet transform. Signal Processing 1997;62(3):265–90.
[31] Zhang Q, Benveniste A. Wavelet networks. IEEE Transactions on Neural Networks 1992;3:889–98.
[32] Lee TS. Image representation using 2D Gabor wavelets. IEEE Transactions on Pattern Analysis and Machine Intelligence 1996;18:959–71.
[33] Wiskott L, Fellous J, Krueger N, Malsburg C. Face recognition by elastic bunch graph matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 1997;19:775–9.
[34] Steffens J, Elagin E, Neven H. Tracking and segmenting people in varying lighting conditions using color. In: International conference on automatic face and gesture recognition. Nara, Japan; 1998. p. 516–21.
[35] Turk M, Pentland A. Face recognition using eigenfaces. In: International conference on computer vision and pattern recognition. Maui, HI, USA; 1991. p. 586–91.
[36] Belhumeur P, Hespanha J, Kriegman D. Eigenfaces versus fisherfaces: recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence 1997;19(7):711–20.
[37] Williams O, Blake A, Cipolla R. A sparse probabilistic learning algorithm for real-time tracking. In: International conference on computer vision. Nice, France; 2003.
[38] Poggio T, Girosi F, A theory of networks for approximation and learning, A.I. Memo 1140, Artificial Intelligence Lab, MIT, 1989.
[39] Zhu Y, Comaniciu D, Ramesh V, Schwartz S. Parametric representations for nonlinear modeling of visual data. In: International conference on computer vision and pattern recognition, Hawai, USA; 2001. p. 553–60.
[40] Black M, Anandan P. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. Computer Vision and Image Understanding 1996;63(1):75–104.
[41] Sato I, Okabe T, Sato Y, Ikeuchi K. Appearance sampling for obtaining a set of basis images for variable illumination. In: International conference on computer vision. Nice, France; 2003. p. 800–7.
[42] Gong S, McKenna S, Psarrou A. Dynamic vision: from images to face recognition. London, UK: Imperial College Press; 2000.
[43] Colliot O, Tuzikov A, Cesar Jr RM, Bloch I. Approximate reflectional symmetries of fuzzy objects with an application in model-based object recognition. Fuzzy Sets and Systems 147(1): 141–163, 2004.