# Object Tracking Using the Gabor Wavelet Transform and the Golden Section Algorithm

Chao He, Yuan F. Zheng, *Fellow, IEEE*, and Stanley C. Ahalt, *Member, IEEE*

***Abstract*—This paper presents an object tracking method for object-based video processing which uses a two-dimensional (2-D) Gabor wavelet transform (GWT) and a 2-D golden section algorithm. An object in the current frame is modeled by local features from a number of the selected feature points, and the global placement of these feature points. The feature points are stochastically selected based on the energy of their GWT coefficients. Points with higher energy have a higher probability of being selected since they are visually more important. The amplitudes of the GWT coefficients of a feature point are then used as the local feature. This takes advantage of the characteristics of Gabor wavelets which are highly localized in both the time and the frequency domains. The global placement of the feature points is determined by a 2-D mesh whose feature is the area of the triangles formed by the feature points. In this way, a local feature is represented by a GWT coefficient amplitude vector, and a global feature is represented by a triangle area vector. One advantage of the 2-D mesh is that the direction of its triangle area vector is invariant to affine transform. Consequently, the similarity between two local features or two global features can be defined as a function of the angle and the length ratio between two vectors, and the overall similarity between two objects is a weighted sum of the local and global similarities. In order to find the corresponding object in the next frame, the 2-D golden section algorithm is employed, and this can be shown to be the fastest algorithm to find the maximum of a unimodal function. Our results show that the method is robust to object deformation and supports object tracking in noisy video sequences.**

***Index Terms*—Content-based video, feature points, Gabor wavelets, golden section, mesh, object-based video, object tracking.**

## I. Introduction

DIGITAL video processing such as video compression, video indexing, and video manipulation, is changing from frame-based approaches to object-based or content-based approaches [1]–[8]. Instead of treating video as a flow of frames, the video is modeled by a combination of a number of objects spanning different time intervals. These approaches have the advantage of automatically supporting description, indexing, and compression of the video content at the object level. In the newest MPEG standard, MPEG-4 [9], content or media objects are one of the key concepts used to satisfy the needs of authors, service providers, and end users such as far greater reusability and flexibility, higher levels of interaction and more efficient compression. The emerging MPEG-7 [10] will be a standardized description of various types of multimedia content.

Object tracking is an important problem in the field of object-based video processing. When a physical object appears in several consecutive frames, it is necessary to identify its appearances in different frames for purposes of processing. Object tracking attempts to locate, in successive frames, all objects which appear in the current frame. The most straightforward approach to this task is to consider objects as rectangular blocks and use traditional block matching algorithms [11]. However, since objects may have irregular shapes and deformations in different frames, video spatial segmentation and object temporal tracking can be combined [12]–[14]. In [13], a supervised I-frame segmentation and unsupervised P-frame tracking algorithm based on motion estimation is proposed, and in [12], a watershed-based algorithm and a hierarchical block matching motion estimation algorithm are used to segment the first frame of a video sequence, and then temporal tracking is realized by motion projection. A generic point approach in which objects are modeled by five generic points is implemented in [14] to make the tracking process less sensitive to partial occlusion.

In content-based video compression, in order to save bits representing the shape and motion vectors of an object, two-dimensional (2-D) or three-dimensional (3-D) mesh based algorithms have been used [1], [3], [4], [6]. The idea is to represent an object by a set of nodes and connecting line segments which usually form triangles. The tracking information which represents the corresponding relationship of objects among consecutive frames is described by the motion vectors of the nodes. Usually, the motion vectors are computed by traditional motion estimation or optical flow methods. However, in previous work, the motion estimation algorithms are all based on local intensity which can be unstable under challenging conditions such as illumination variation, contrast variation, object zooming, object rotation, and object deformation.

Another problem arises from the use of texture matching. In most cases, objects can be viewed as a combination of pieces of texture, and textures can be used as local features. But intensity based motion estimation methods have difficulties in matching textures because many textures are homogeneous and sensitive to the deformation. Homogeneity causes errors in matching while sensitivity to deformation arising from a small shift may result in a large matching distance.

Gabor functions and wavelets, originally proposed by Gabor [15], have achieved impressive results when used for texture and object recognition. Gabor wavelets are very suitable for

Manuscript received July 20, 1999; revised July 15, 2002. The associate editor coordinating the review of this paper and approving it for publication was Prof. Wayne Wolf.

The authors are with the Department of Electrical Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: hec@ee.eng.ohio-state.edu; zheng@ee.eng.ohio-state.edu; sca@ee.eng.ohio-state.edu).

Digital Object Identifier 10.1109/TMM.2002.806534

representing local features based on a number of useful properties: a) Gabor wavelets are the best localized wavelets, i.e., Gabor wavelets provide the best trade-off between spatial resolution and frequency resolution, b) Gabor wavelets contain a rather large number of parameters, and, c) research in psychology shows that responses of simple cells in the visual cortex can be modeled by the Gabor functions [16]–[18]. Gabor functions have been used for image processing in a number of ways. For example, a family of 2-D Gabor wavelets have been derived for complete image representation using wavelet theory in [19]. In [20] and [21], the GWT coefficients have been used successfully in texture browsing and retrieval. In [22], the magnitude of GWT coefficients was used for 2-D object recognition in a dynamic link architecture approach. The Gabor wavelets are used for local feature representation and a uniform but elastic grid is used for object structure representation. Then, a cost function between the model and the object is defined as a linear combination of the object's local feature difference and global structure deformation. At last, the object recognition is done by simulated annealing based elastic graph matching algorithm which minimizes the cost function. Reference [23] is an advanced version of the dynamic link architecture approach for 3-D object recognition. It utilizes both the magnitude and the phase of the coefficients of the GWT, defines a more sophisticated cost function, and implement a better simulated annealing algorithm.

Although object detection and recognition schemes share some commonality with object tracking, it is typically inefficient to apply the former to object tracking directly. Objects detection algorithms are mainly designed to detect and recognize an object in a more challenging environment and require extensive computation. In object tracking, one has more information to assist in the tracking process. For example, objects in successive frames often obey affine transform constraints and the motion of the object between adjacent frames is limited.

In this paper, we integrate the 2-D Gabor wavelet, dynamic link, and 2-D mesh approaches along with a number of new ideas for object tracking. The amplitudes of the GWT coefficients instead of the local intensity [1], [3], [4] are used to represent local features. A nonuniform 2-D mesh instead of a uniform grid in [22], [23] is used to represent the global structure. In addition, the object is tracked by a 2-D golden section algorithm followed by the elastic graph matching. The technique described here presents three new methods: 1) an innovative feature point selection scheme, 2) an affine-transform invariant mesh representation method, and 3) a fast 2-D golden section search algorithm.

This paper is organized as follows. Section II presents a brief explanation of Gabor functions and wavelet and how they can represent local features. Section III describes our feature point selection scheme. Next, the generation and representation of the mesh is presented in Section IV. Section V explains the similarity measures we define between two mesh representations. Section VI describes the 2-D golden section search algorithm used for object tracking. Experimental results are given in Section VII, and the paper concludes with a summary and a discussion in Section VIII.

## II. Gabor Function, Gabor Wavelets, and Local Features

A 2-D Gabor function, $g(x, y)$, and its Fourier transform $G(u, v)$ are defined as [20], [22], [23]:

$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right]. \tag{1}$$

$$G(u, v) = \exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right]\right\} \tag{2}$$

where $\sigma_x$ and $\sigma_y$ are the standard derivations of $g(x, y)$ along the $x$ and $y$ axes, respectively. Here $\sigma_u = 1/(2\pi\sigma_x)$ and $\sigma_v = 1/(2\pi\sigma_y)$ are the standard derivations of $G(u, v)$ along the $u$ and $v$ axes, respectively. In the spatial domain, the 2-D Gabor function is a product of an elliptical Gaussian and a complex plane wave. In the frequency domain, the Gabor function is just 2-D elliptical Gaussian function shifted along the $u$ axis. The Gabor function is well known for its optimal time-frequency localization. For any $f(t)$ and its Fourier transform, $F(\omega)$, the uncertainty relationship holds [25]:

$$\sigma_t\sigma_\omega \geq \tfrac{1}{2} \tag{3}$$

where $\sigma_t$ and $\sigma_\omega$ are the effective widths of $f(t)$ and $F(\omega)$, respectively. The equality condition is possible only when the signal is a Gabor function, and this makes the Gabor function the best choice for representing local features of a signal.

Gabor wavelets are generated by scaling and rotating the Gabor function:

$$\begin{aligned} g_{mn}(x, y) &= a^{-m}g(x', y'), \\ &\qquad a > 1,\ m = 1:M,\ n = 1:N. \end{aligned} \tag{4}$$

$$\begin{aligned} \begin{bmatrix} x' \\ y' \end{bmatrix} &= a^{-m}\begin{bmatrix} \cos\theta_n & \sin\theta_n \\ -\sin\theta_n & \cos\theta_n \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}, \\ \theta_n &= n\pi/N \end{aligned} \tag{5}$$

where $a$ is the scaling parameter. Gabor wavelets can be understood as a set of Gabor functions with different frequency centers and orientations. The factor of $a^{-m}$ yields a logarithmic frequency sampling. The orientation of the Gabor wavelets is controlled by $\theta$. Since Gabor wavelets are symmetric, we need only specify the value of $\theta$ to realize an evenly sampled space in $[0, \pi]$. In this way the concept of the localization of the Gabor wavelets has been extended to time, frequency and orientation. Indeed, orientation localization is another reason why Gabor wavelets are a very good choice for representing local features.

By convolving an image with Gabor wavelets the Gabor wavelet transform (GWT) of the image $I(x, y)$ can be defined as:

$$\hat{I}(x, y, m, n) = \int I(x', y')g_{mn}(x-x', y-y')\,dx'dy'. \tag{6}$$

Referring to (1), one may observe that the output of the GWT is a complex number. There are a total of $M$ different frequencies and $N$ different orientations, resulting in $M \times N$ coefficients for each image pixel $(x, y)$. The amplitudes of these coefficients can be viewed as the local feature vector of that point [22], [23]:

$$L(x, y) = \left\{abs\left(\hat{I}(x, y, m, n)\right), \ m = 1 : M, n = 1 : N\right\}. \quad (7)$$

This vector represents the local features of point $(x, y)$ and captures the frequencies and orientations which should remain constant during object motion.

Note that using this definition the Gabor function and Gabor wavelets have a nonzero mean, and additionally the GWT has a strong response to the DC component of a signal. In fact, dominant DC components can decrease the effectiveness of the GWT coefficients. So in [20] and [22], $G(0, 0)$ is set to zero to avoid this dominant effect. In our work, $G(0, 0)$ is only scaled since we do not want to totally remove the DC component.

## III. Local Feature Point Selection

After we calculate the GWT of the whole image (frame), we select particular points as the feature points, which represent the object using their local feature vectors. The GWT coefficients of any given point are calculated from the pixels of the local area surrounding the point. Thus the coefficients of adjacent points are calculated from overlapped local areas. As a result, the calculated GWT coefficients are redundant, and it is not necessary to select all points as feature points. For this reason, [22] and [23] have used a uniform grid placed on the object and the nodes of the grid are then selected as the feature points.

Another concern is that different points have different levels of importance in object representation. For example, in a noisy environment, some points with small coefficients may be contaminated by noise, and these points may be ill-suited for use in the object representation. To avoid the selection of these points, an amplitude threshold can be used to exclude them [25]. Furthermore, some points are more important than others and should have a better chance of selection. This can be accomplished by employing a function which depends on the differences between the GWT coefficients of different frequencies [26]. The point with the maximum value of this function in a local neighborhood is selected as a feature point. This approach supports the selection of starting and ending points of line segments as well supporting the selection of any points where there are significant curvatures.

Another commonly used method of object delineation is based on the intensity gradient [3], in which the points with the highest gradient in a local neighborhood are selected as the feature points corresponding to edges of the object. Generally, both the GWT based and intensity gradient based selection methods can be modeled as finding the local maximums of a significance measure function. Only those points with local maximums are selected as feature points $\{(x_f, y_f)\}$.

Unfortunately, while both these schemes generally avoid the selection of ill-suited points, they preclude the selection of those points which, while significant, are not local maximum. Another problem in the local maximum approach is that all local regions are treated equally even though they may make different contributions to the representation of the object. For example, the local maximum of a nonsignificant region may be selected even though the region contributes little or no information to the representation of the object. In a good selection scheme, the distribution of the feature points should be globally balanced. Intuitively, more important regions should be apportioned more feature points than less important regions with similar areas. Reference [4] presents such a method. It relates the size of the local neighborhood with the local intensity gradient. The higher the local intensity gradient, the smaller the local neighborhood. In this way, the more importants regions should have more features points.

We propose a simpler selection scheme to achieve the same goal. We randomly select a point as a feature point with a probability which depends on the significance of the point (significance will be defined later). More significant points are more likely to be selected. As a result, a region with more significant points should have a larger contribution to the representation of the object as compared with other regions simply because there are more feature points in the region. For example, a human face has regions with different levels of significance. The eyes, nose, and mouth have a higher level of significance than other regions. This means that the density of the selected feature points in the regions of eyes, nose, and mouth should be higher than in the other facial regions. However, this does not mean that we totally ignore the latter. We describe this process as follows: First, we define the energy of the GWT coefficients of point $(x, y)$ as:

$$E(x, y) = \sum_{m=1}^{M} \sum_{n=1}^{N} \left|\hat{I}(x, y, m, n)\right|^2. \quad (8)$$

The significance of the point $S(x, y)$ is measured by an increasing function of $E(x, y)$:

$$S(x, y) = fs(E(x, y)). \quad (9)$$

There are two reasons for selecting energy as the measure of the significance. The first reason is that the human visual system can be modeled by the Gabor function [16]–[18]. Higher energy signals represent stronger stimuli to the human visual system. The second reason is that Gabor wavelets can be viewed as edge detectors, which is consistent with the traditional edge-based object representation. Fig. 1 shows that the energy of the GWT coefficients of a face is higher around the significant areas such as eyes, mouth, nose, and the contour of the face.

Secondly, we use the significance function $S(x, y)$ to control the variance of a random variable $R(x, y)$ in order to obtain a random significance function $S_r(x, y)$:

$$S_r(x, y) = S(x, y)R(x, y). \quad (10)$$

Here, a Gaussian random variable $R(x, y) \sim N(0, 1)$ is used whose value is not bounded within an interval. This will give every point a probability of being selected. The random significance $S_r(x, y) \sim N(0, S(x, y)^2)$ also has a Gaussian distribution. A threshold $T$ is set for $S_r(x, y)$, and those points with $S_r(x, y) > T$ will be selected as feature points:

$$\{(x_f, y_f)\} = \{(x, y)|S_r(x, y) > T\}. \quad (11)$$

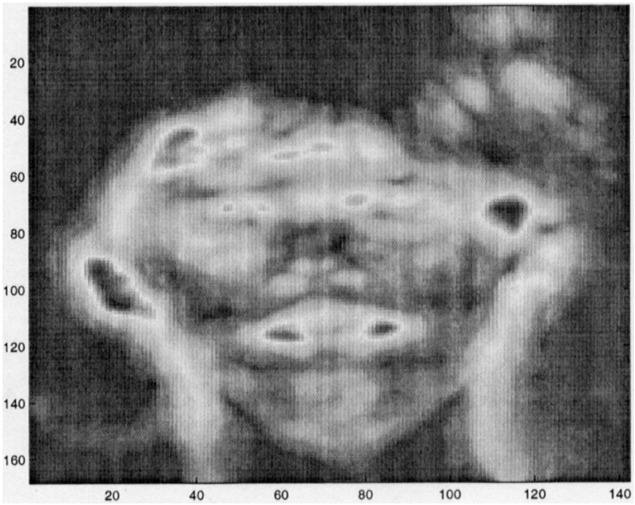


Fig. 1. Energy of the GWT coefficients $E(x, y)$. White color means high energy.

The probability of selecting $(x, y)$ as a feature point is:

$$P((x, y) \in \{(x_f, y_f)\}) = \frac{1}{2} - erf\left(\frac{T}{S(x, y)}\right) \tag{12}$$

where $erf(x)$ is the error function and defined as $(1/\sqrt{2\pi}) \int_0^x e^{-t^2}\, dt$. The points with higher energy are more likely to be selected as $erf(x)$ is an increasing function. All points are evaluated equally. By changing the function $fs(x)$, we can also control the relationship between the energy and the probability. Fig. 2 shows one result of our feature point selection approach. The nodes of the solid mesh are the selected feature points. Most of the feature points are located close to edges, but there are also points located inside the object. Unlike traditional edge-based representations, this is a rich, multi-resolution representation which includes more information from the object. For comparison purposes, the grid based feature point selection as used in [22], [23] is also shown in this figure as nodes of the dashed grid. Although the latter method is applied to the same region and generates the same number of the feature points, it is less efficient than our proposed method. Note that some feature points selected by the grid approach are in the regions which are not important, such as the background or hair. Furthermore, there are no feature points in some important regions such as the eyes and mouth.

Additionally, our statistical selection scheme supports an adaptive number of the feature points. The total number of the feature points can be controlled by adjusting the threshold $T$. One advantage is that the number of feature points changes smoothly with the changing of $T$. It is, of course, possible that two or more selected feature points might be too close, and can be viewed as redundant. To counteract this problem in the selection procedure we compute the distance between the candidate feature points with those points already selected, if one distance is less than a threshold, $d_{\min}$, the candidate point is not selected.

## IV. Global Feature

Next, we need to define the global relationship among these feature points in order to describe the object completely. Since the feature points are not uniformly distributed inside the object, the first step is to connect these points using line segments to form a mesh. Suppose that there are $N_p$ feature points $\{(x_f(n), y_f(n)) | n = 1 : N_p\}$ distributed on the object. First we select a new point $(x_f(i), y_f(i))$ from $i = 1$ to $N_p$ and connect this point to all old points $\{(x_f(j), y_f(j)) | j = 1 : i - 1\}$ with new line segments. Secondly if a new line segment intersects an old line segment, we retain the shorter line segment and remove the longer one. Finally, we organize lines to form triangles. The mesh can also be generated by the other popular algorithms like Delaunay algorithm. We use $N_l$ and $N_t$ to denote the number of the line segments and triangles retained after the procedure is completed, respectively. Fig. 2 shows an example of this mesh generating procedure.

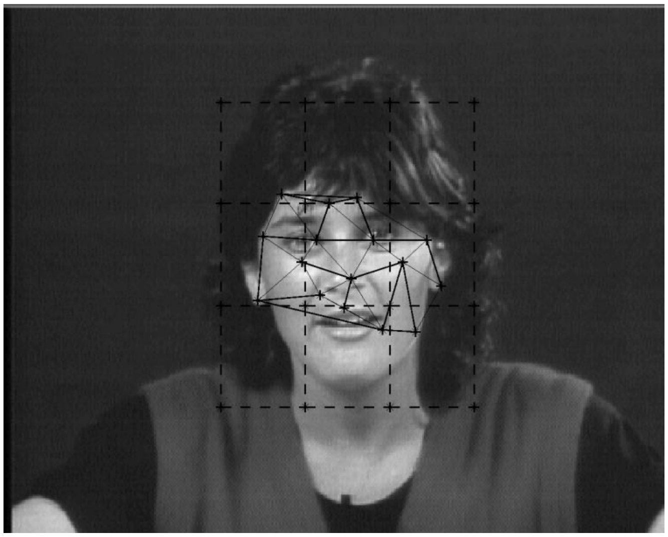


Fig. 2. Feature point selection for a face. The "$*$" points of the mesh are the feature points selected by the technique we elaborate in Section III, as compared to the "$*$" points of the grid which are the feature points selected by a conventional girding technique.

In a video sequence, a physical object may deform in different frames. In [1] and [4], a small region is defined for every node to limit the mesh's deformation, but inside the region there is no constraint. In object recognition, if grid is used to represent the object, deformed grid can be represented by the lengths of the line segments connecting feature points and the angles between line segments [22], [23]. While in most video sequences, the object's deformation can be modeled by an affine transform, length-based or angle-based representations are not invariant under affine transformations. For example, if an uniform grid is scaled along the vertical direction, all vertical line segments are scaled but all horizontal line segments keep their original length. Then if we simply use the length of all the line segments of this grid as a vector to represent the grid's structural information, the resultant vector changes its length and direction after the vertical scaling, which is an example of an affine transform. Similar analysis and results are also valid for mesh representations.

In this paper, we propose an area-based mesh description which is invariant to the affine transform. Our representation of a mesh is very compact and consists of a vector of the areas of all triangles, $G = [a(1), a(2), \ldots, a(N_t)]$, where $a(i)$ is the area of the $i$th triangle.

The affine transform can be modeled by the following formula [27]:

$$\begin{bmatrix} x_a \\ y_a \end{bmatrix} = C \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} g \\ h \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} g \\ h \end{bmatrix} \tag{13}$$

where $(x, y)$ is a point's original coordinate and $(x_a, y_a)$ is its new coordinate after the affine transform. Now suppose an object undergoes an affine transform and the new area vector is $G_a = [a_a(1), a_a(2), \ldots, a_a(N_t)]$, $a_a(i)$ is the area of the $i$th triangle after the affine transform. $[g, h]^T$ represents a shifting and it will not affect the areas of the triangles. Using the Singular Value Decomposition (SVD) [27], the matrix $C$ can be decomposed into three parts:

$$C = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{bmatrix} \begin{bmatrix} \cos\tau & \sin\tau \\ -\sin\tau & \cos\tau \end{bmatrix}. \tag{14}$$

The decomposition shows that the $C$ matrix is actually composed of two rotating and one scaling matrices. Thus, the area of a triangle will only be changed by scaling, while rotating does not affect the area. If the area of a triangle in the current mesh is $a(i)$, the new triangle's area after the affine transform is $a_a(i) = \sqrt{\lambda_1 \lambda_2} a(i)$, and the new area vector would be $G_a = \sqrt{\lambda_1 \lambda_2} G$. Thus, only the length of the area vector changes, and the direction of the vector is invariant to affine transform. This direction-invariant property is guaranteed so long as the motion of the object can be modeled by an affine transform.

## V. Similarity Measure

With the local features and the global feature defined, we can now compute the similarity between the representation of the model (the object in the current frame) and that of an object (which can actually be any arbitrary area) in the new frame. The two representations should have the same structure such as $N_p$, $N_l$,, and $N_t$, and the relationship between points, line segments, and triangles. Since the local features and the global feature represent different features of the object, they should be considered separately when measuring similarity. In the dynamic link architecture methods [22], [23], a cost function is defined as a linear combination of the local features similarity and the grid deformation. Then the matching goal is to minimize the cost function. The linear combination of similarity and deformation is the key idea of elastic graphic matching. On the one hand, it attempts to maximize the local feature similarity; on the other, it attempts to maintain the structure of the object as well as possible. We expand this idea and use a final similarity measure which is the weighted sum of the local and global similarities. The local feature similarity, $S_l$, is the sum of individual local feature similarities between each pair of corresponding feature points. The individual local feature similarity $s_l(i)$ between the local features of corresponding feature points $(x_{fo}(i), y_{fo}(i))$ in the object and $(x_{fm}(i), y_{fm}(i))$ in the model is defined as [22], [23]

$$s_l(i) = \frac{L_O(i) \cdot L_M(i)}{\|L_O(i)\| \times \|L_M(i)\|} \min\left(\frac{\|L_M(i)\|}{\|L_O(i)\|}, \frac{\|L_O(i)\|}{\|L_M(i)\|}\right) \tag{15}$$

where $L_M(i)$ is the local feature vector of $(x_{fm}(i), y_{fm}(i))$ in the model, and $L_O(i)$ is the local feature vector of $(x_{fo}(i), y_{fo}(i))$ in the object. The operator "$\cdot$" is the inner product, $\|\ \|$ is the norm operator used to calculate the length of a vector, and $\min$ returns the smallest component. The final local similarity, $S_l$, is the sum of $s_l(i)$ for all feature point pairs

$$S_l = \sum_{i=1}^{N_p} s_l(i)/N_p. \tag{16}$$

We note that the affine transform does not change the direction of the area vectors. One can therefore define the similarity of the global placement, $S_g$, as a $\cos$ function of the angle between the model's area vector, $G_M = (a_m(1), a_m(2), \ldots, a_n(N_t))$, and the object's area vector, $G_O = (a_o(1), a_o(2), \ldots, a_o(N_t))$:

$$S_g = \cos\left(k \times \arccos\left(\frac{G_M \cdot G_O}{\|G_M\| \times \|G_O\|}\right)\right) \tag{17}$$

where $k$ is a parameter used to balance the sensitivity of the global and local similarities.

The overall similarity between the model and the object is thus defined as:

$$S = \alpha S_l + (1 - \alpha) S_g \tag{18}$$

where $\alpha$ is a weighting parameter which can be selected through experiment. Basically, $\alpha$ controls the trade-off between local and global similarities. Global similarity emphasizes the shape of the object, while the local similarity stresses the local features.

## VI. Searching

The most straightforward method for searching for an object in the next frame is to place the object model to all possible position in the next frame and find the position which manifests the best matching result [11]. It is also used in grid based object recognition [22] and [23]. Applied to the mesh representations of objects, we would thus place the mesh of the model in the current frame to an arbitrary position in the next frame, and then define the position of the nodes of the new mesh in the next frame as a new set of feature points which represents a test object. Then the similarity between these two sets of the feature points can be computed. After testing all possible positions, the test object which has the maximum similarity to the model is selected as the search result. Once the position is found, the object deformation need to be tracked. In [22] and [23], simulated annealing based elastic graphic matching algorithm is applied for global optimization to identify objects. Each node is shifted randomly to find its best matching position. In [3], [4], every feature point is examined independently and no global placement information of features points is used. Consequently the object's structural information may be lost.

Here we propose a 2-D golden section searching algorithm to find the object in the next frame quickly. In a 1-D optimization problem, the golden section algorithm can be used to find a

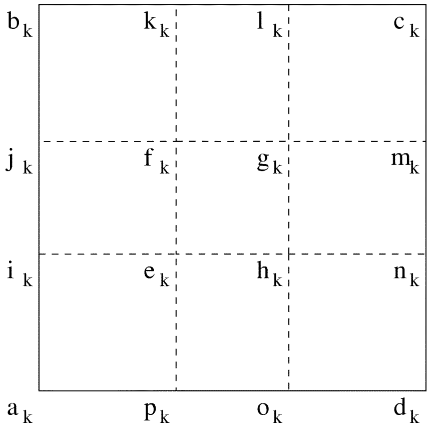


Fig. 3. Two-dimensional golden section algorithm.

global maximum if the function is unimodal [28]. Suppose that a function is defined on $[0, 1]$, and $\overline{x}$ is the unique maximum of $f(x)$ on $[0, 1]$. If $f(x)$ strictly increases for $x \leq \overline{x}$ and strictly decreases for $x \geq \overline{x}$, it is called unimodal. The golden section search is based on Fibonacci numbers which are defined as:

$$\begin{aligned} &F_0 = F_1 = 1, \quad F_2 = F_0 + F_1, \\ &F_3 = F_1 + F_2, \ldots, F_k = F_{k-1} + F_{k-2}. \end{aligned} \tag{19}$$

The golden section algorithm picks the golden-section points $x_{k-1} = F_{k-2}/F_k$ and $x_k = F_{k-1}/F_k$, $k > 2$ to decrease the interval covering $\overline{x}$. If $f(x_{k-1}) \geq f(x_k)$, the new interval will be $[0, x_k]$. Otherwise, the new interval will be $[x_{k-1}, 1]$. This refinement is done recursively until the length of the interval reaches the requirement. As $k \rightarrow \infty$, $(F_{k-1}/F_k) = e \approx 0.618$, so the algorithm is called the golden section algorithm.

Here we expand the acbove golden section algorithm from 1-D to 2-D. Suppose that a function $f(x, y)$ is defined on a rectangle, $a_k b_k c_k d_k$, as shown in Fig. 3, and $(\overline{x}, \overline{y})$ is the unique maximum of $f(x, y)$. If $f(x, y)$ decreases when the distance between $(x, y)$ and $(\overline{x}, \overline{y})$ increases, we can apply the 2-D golden section algorithm to search for the maximum. We first select $j_k$ and $i_k$ as the golden section points of the line segment $a_k b_k$, $k_k$ and $l_k$ as the golden section points of $b_k c_k$, $m_k$ and $n_k$ as the golden section points of $c_k d_k$, and $p_k$ and $o_k$ as the golden section points of $a_k d_k$. We pick the intersection points of $j_k m_k$, $i_k n_k$, $k_k p_k$, $l_k o_k$ as the 2-D golden section points, which are $e_k$, $f_k$, $g_k$ and $h_k$, respectively. If $f(e_k)$ is the largest of the four points, the region covering the global maximum will be refined to a new rectangle $a_k j_k g_k o_k$. New rectangles are generated in a similar way if $b_k$, $c_k$, or $d_k$ is the maximum point.

As we have discussed above, the basic idea for searching is to place the mesh of the model to the new frame and shift it around to find the best matching object. We measure the shifting by $(x, y)$ which is calculated by $(x', y') - (x_c, y_c)$, where $(x_c, y_c)$ is the coordinate of the center of the mesh of the model and $(x', y')$ is the coordinate of the center of the mesh in the new frame. As the motion of the object is often limited between adjacent frames, we can apply the 2-D golden section algorithm by limiting $(x, y)$ within a rectangle and define a function $S_{MO}(x, y)$ on the rectangle. $S_{MO}(x, y)$ is just the similarity between the model whose mesh center is $(x_c, y_c)$ and the test object whose mesh center is $(x', y')$. So the goal is to find the maximum of $S_{MO}(x, y)$. In most cases, if the moving distance is not too large, we assume that the similarity function is unimodal. Fig. 4 shows an example of the similarity function between the first and tenth frames of the Miss America video sequence. From the figure, one can see that the similarity function is close to be unimodal, but there still exist local maximums. These local maximums will affect the accurate tracking of the object. Fortunately, the values of these local maximums are significantly less than the global maximum; therefore, one may possibly avoid the local maximum by continuing the search until a satisfactory maximum of the similarity function is found. This of course will cause more computation that is a trade-off for more robust performance of the proposed method.

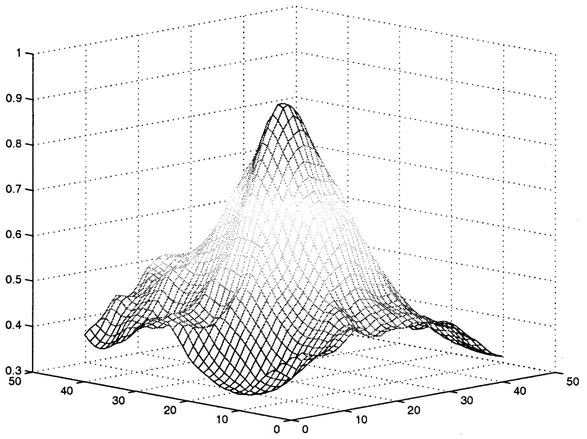


Fig. 4. One similarity function example between the 1st and 10th frames of the Miss America sequence.

The maximum point of $S_{MO}(x, y)$ is the coarse position of the object in the next frame, but it does not reflect the deformation of the object. To refine the position and to track the deformation, we further search for the best matching position for every feature point like [22], [23]. For every feature point, we simply move it around in a local searching region to find its best matching position. These two steps are alternatively applied iteratively until the similarity cannot be further improved, or until all the allocated computation time is consumed. The final result is a mesh on the new frame which reflects not only the position but also the deformation of the object.

In order to decrease the computational cost of the GWT we use a cache scheme during the searching. Consequently, in our search algorithm, we do not need to compute the GWT coefficients of all the points. When the GWT coefficients of a new point are needed, we first look for that point in the cache, and if it is there, we proceed without further computation.

## VII. Experimental Results

Here we apply the proposed object tracking algorithm to several video sequences. The parameters of the Gabor wavelets are selected by the experiments, and for convenience, we express these parameters in the Fourier domain. We choose five frequencies and five orientations. The highest frequency is $\pi/2$ and the scaling ratio is $\sqrt{2}$. The five orientations are $0, \pi/5, 2\pi/5, 3\pi/5$ and $4\pi/5$, respectively. The standard

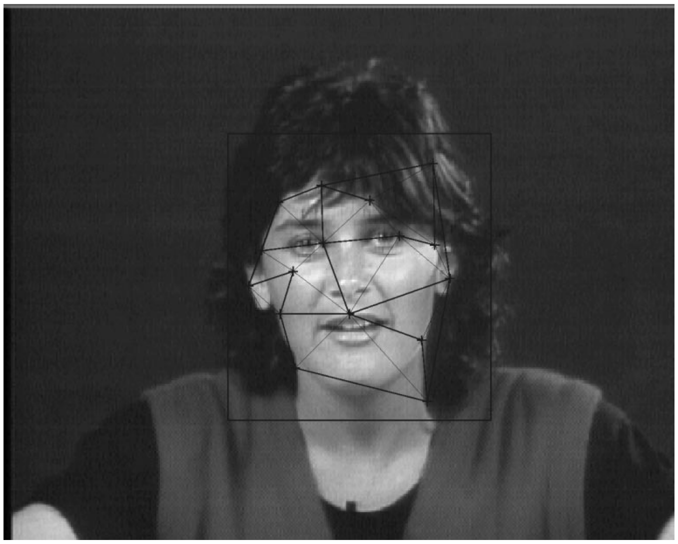


Fig. 5. First frame of the Miss America sequence.

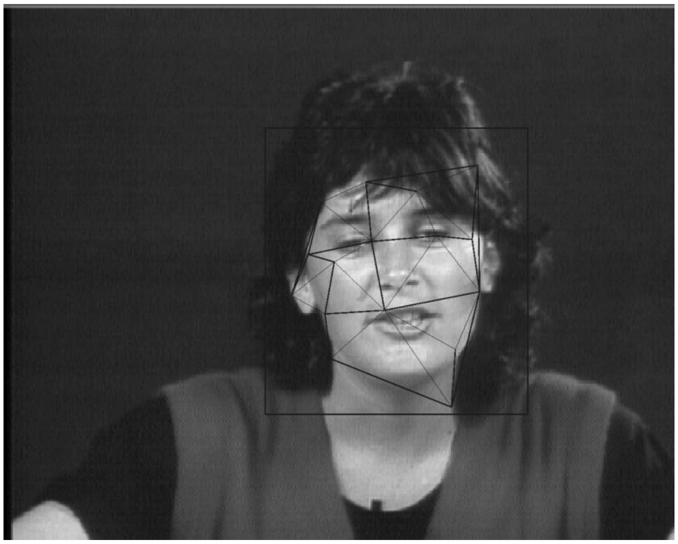


Fig. 6. Tracking result of the 100th frame of the Miss America sequence tracked by the proposed method (mesh) and the method of [11] (block).

derivation, $\sigma_u$, along the long axis of the Gabor wavelet with the highest frequency is $\pi/3$. The scaling ratio of the standard derivation is also $\sqrt{2}$. The ratio between the standard derivation of the long and short axes is 2.

In the first searching step, the size of the searching area in the golden section algorithm depends on the size and the standard derivation of the lowest frequency wavelet in the time domain. We select the size to be $36 \times 36$ pixels. In the second refinement step, the searching rectangle is $10 \times 10$. In the similarity measure, we weigh the global and local feature similarities equal, so $\alpha = 0.5$. $k$ is selected individually based on the deformation of the object. During the tracking, the mesh's geometry and placement are updated frame by frame, but the local feature is not in order to prevent error accumulation and make the tracking robust.

Here we use two video sequences of our experiments. The first is the Miss America sequence which is used to test the ability of the algorithm to track object deformations and shifts. Only the first frame in every five frames is used to force the motion and deformation to be large. In the first frame, a rectangle is interactively selected by the user. Although the user specifies a rectangle to cover the object, we find that after the feature point selection, all the feature points are located inside the object or on the boundary. Fig. 5 shows the first frame of the sequence. Fig. 6 shows the tracking result in the last frame. The solid mesh is the tracking result of our proposed method (note that the large block outside the face is by the approach of [11] which will be mentioned later again). From the result we can see that the face is tracked even with considerable deformation and shift. It can also be noted that the feature point pairs in the first frame and the last frame represent the same physical points, which demonstrates that the GWT coefficients are useful for representing local features. Fig. 7 shows the matching similarity in each step of the proposed tracking method. The experiment also proves that the golden section algorithm is very efficient. In the experiment the first step of the search required only five recursive steps to find the correct position.

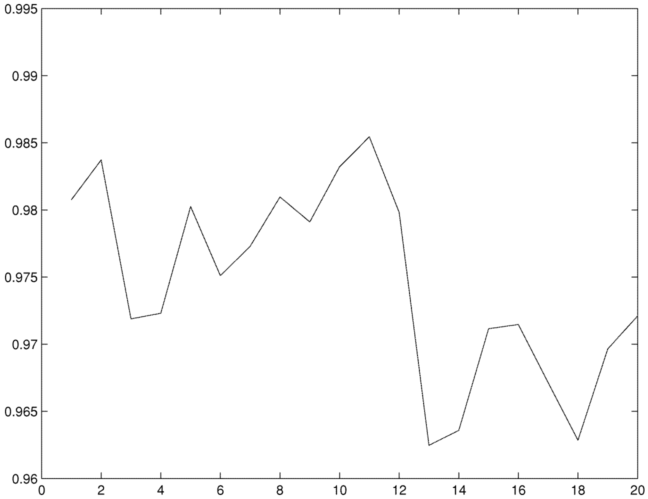


Fig. 7. Matching similarity during the tracking of the Miss America sequence ($x$: Steps of search, $y$: Similarity).

We also compared our method with existing approaches reported in [11] and [1], [4], [6], respectively. Both methods were developed for tracking objects in video. In [11], a simple block matching method is used. The tracking approach is similar to the one used for motion estimation in MPEG, but with a difference in that the search block is much bigger in order to cover the entire object. Matching is achieved by a simple subtraction of the feature block from the testing location, and searching for the minimal difference. One can see that this method does not include any semantic meanings of the feature, and may entirely lose tracking of the object. Even in the best case, it can track the block but does not locate the detailed features of the object. We applied this simple method of [11] to the Miss America sequence and results are shown in Fig. 6. The block is the one which is used by the method of [11] to cover the entire face, while the mesh is by our method. One can see that the block method locates the object, but provides no information about the deformation of the object while our method does both.

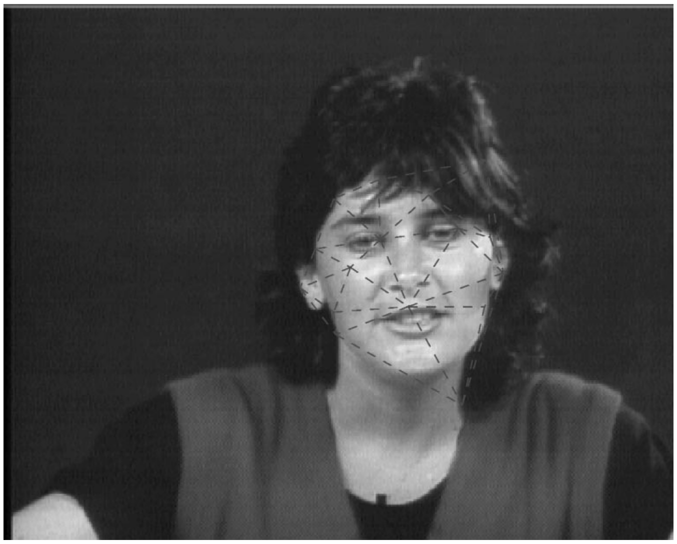


Fig. 8. Tracking result of the 100th frame of the Miss America sequence by independent node-point matching.

In [1], [4], and [6], an object is represented by a hierarchical 2–D or 3–D mesh consisting of Delaunay triangles. The mesh is tracked by applying simple block matching method individually to each node or by optical flow method. We repeat the block matching method. The tracking of the object is conducted by individual nodes and not by the entire set collectively. The mesh in Fig. 8 is the result achieved by applying the block matching method on each mesh node individually. One may observe that one tracked point has very large shift from its correct position. This is caused by the failure of the matching method which uses no global constraints.

Obviously the trade-off of our method is a higher computation cost. This is because of the computation of Gabor wavelet transforms, which is mathematically more complicated. The computation for the block-matching method [11] is the simplest, and is therefore less powerful than the alternatives.

To test the robustness of our approach, we further applied the approach to the sequence titled “Claire.” In this sequence, the deformation of the face is more complex than in the first sequence. Fig. 9 shows the first frame of the sequence. The tracking result is shown in Fig. 10 as solid mesh. Note that even though the object undergoes a significant deformation, our method still tracks the face. Fig. 11 shows the matching similarity at each step of our tracking method. One may notice that the similarity drops significantly at the end. That is because the deformation in the last few frames is very big. In fact the golden section may even fail for certain kind of feature points.

In order to test our tracking method in a noisy environment, we add Gaussian noise to a video sequence before the object tracking is applied. The Miss America sequence is used again. A $10 \times N(0,\, 1)$ Gaussian noise image is added to every frame of the original video. Fig. 12 shows the first frame of the sequence and Fig. 13 show the tracked object in the last frame. Fig. 14 is the matching similarity in every tracking step. The average PSNR of the video is 28 dB.

In order to test our approach under a truly affine transform, we artificially apply an affine transform to the first frame of the

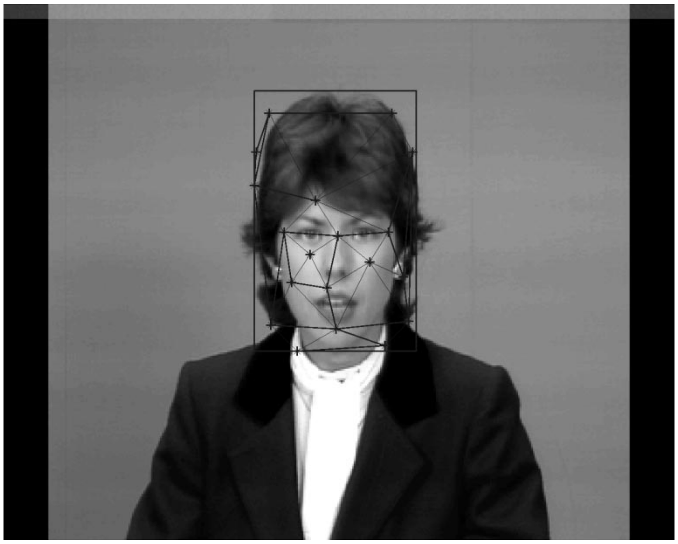


Fig. 9. First frame of the Claire sequence.

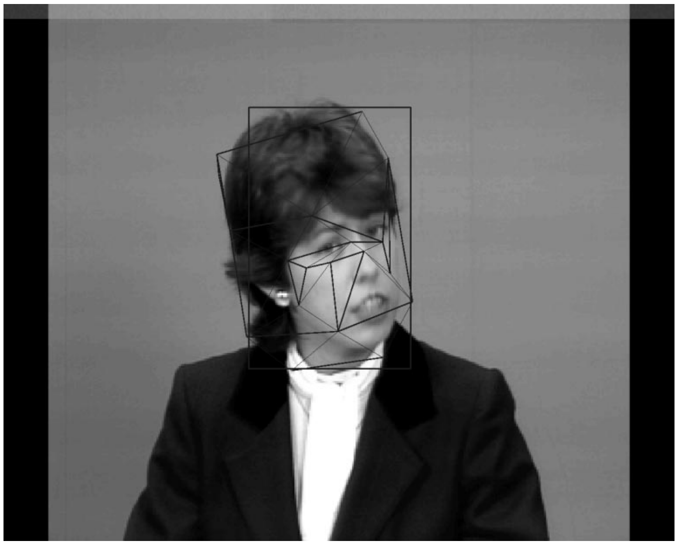


Fig. 10. Tracking result of the 100th frame of the Claire sequence by the proposed method.

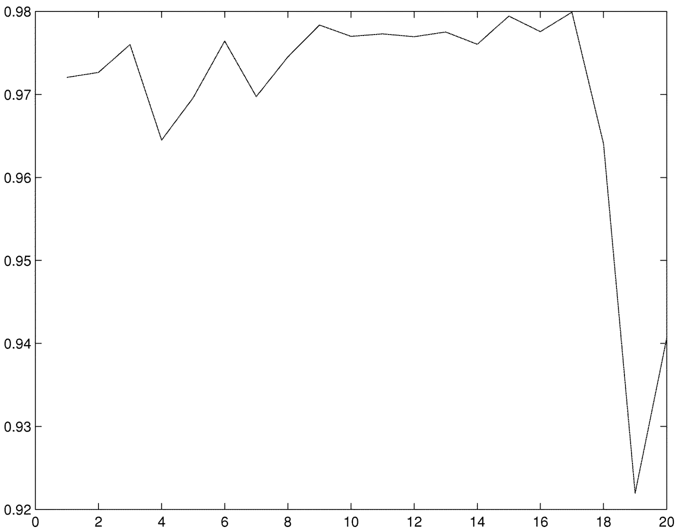


Fig. 11. Matching similarity during the tracking of the Claire sequence ($x$: Steps of search, $y$: Similarity).

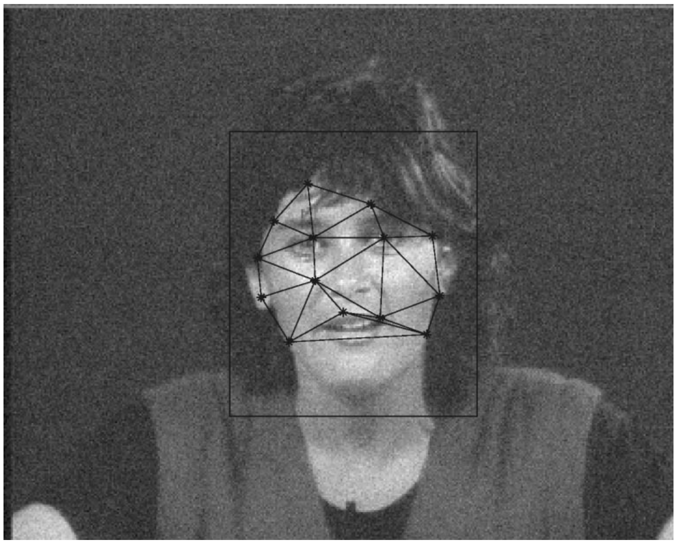


Fig. 12. First frame of the noisy Miss America Sequence.

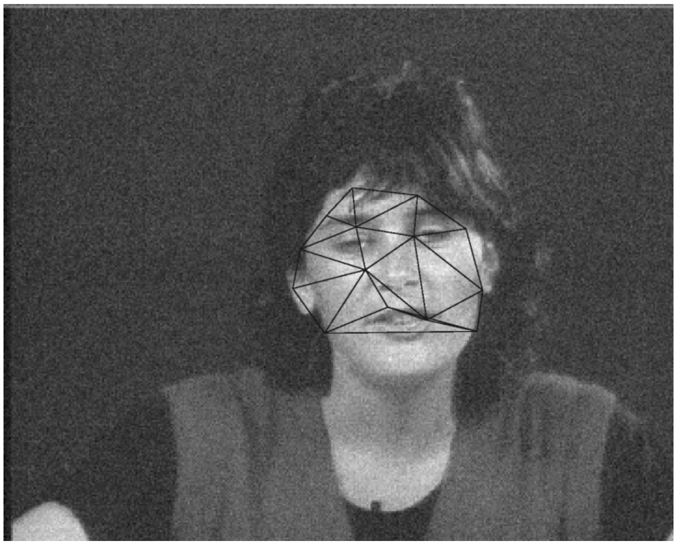


Fig. 13. Tracking result of the 100th frame of the noisy Miss America sequence by the proposed method.

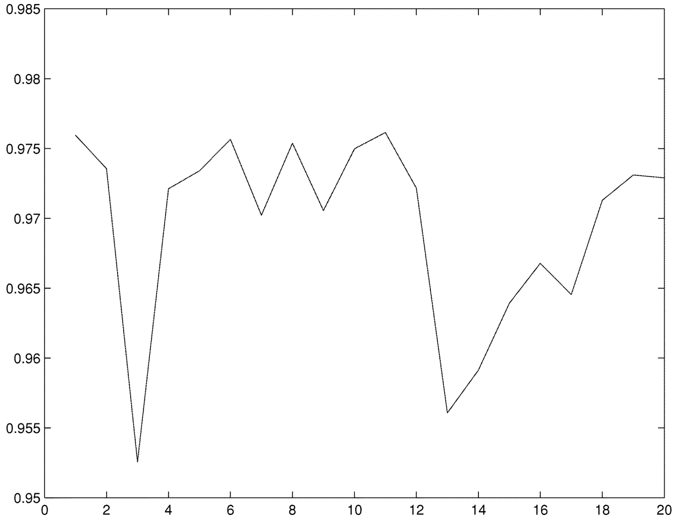


Fig. 14. Matching similarity during the tracking of the noisy Miss America sequence ($x$: Steps of search, $y$: Similarity).

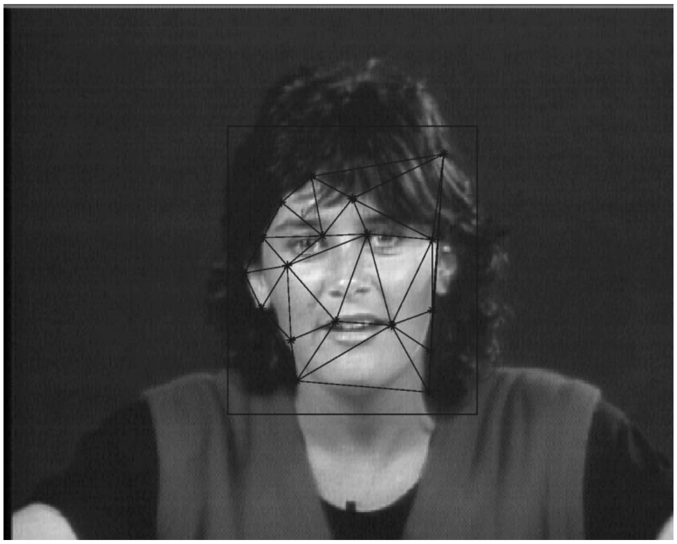


Fig. 15. First frame of Miss America for affine transform.

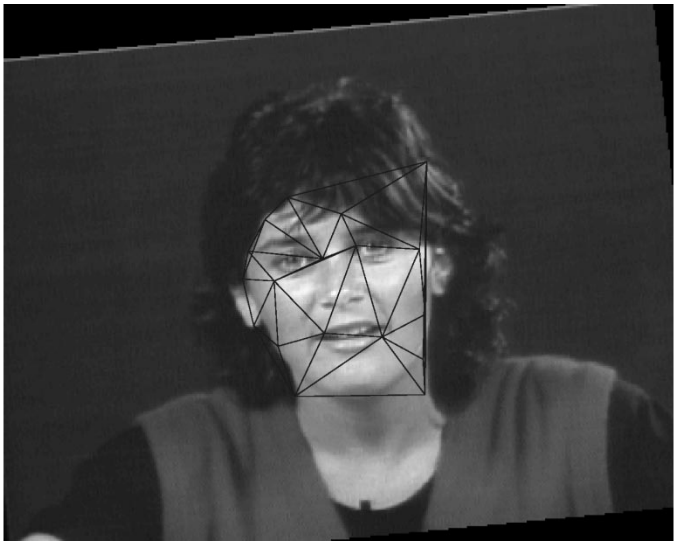


Fig. 16. Tracking result of Miss America with affine transform.

Miss America video sequence. The experiment focuses on rotating and scaling; therefore, $g = 0$ and $h = 0$ and no golden section searching is involved. Other affine transform parameters are $\lambda_1 = 0.9$, $\lambda_2 = 1.1$, $\phi = -0.05$, and $\tau = -0.05$. Fig. 15 is the original frame and Fig. 16 is the new image after the affine transform. Fig. 16 also shows the tracked result. One may notice that the deformed mesh follows the affine transform well. This experiment proves that our affine transform invariant representation is effective.

When the translation distances $g$ and $h$ are not zero while small, we may rely on the golden section algorithm to track them. But the translation distance which can be tracked by the golden section algorithm is inversely proportional to the affine transform. The larger the affine transform, the smaller the trackable translation. When $h$ and $g$ are larger than the trackable translation, the case is like the object recognition in [22]. Similarly, we therefore define a set of exhaustive search regions with its center at the coordinates of $[kG, lH]$ where $(k, l)$ are integers and $(G, H)$ is the size of the regions. $(G, H)$ is inversely proportional to the deformation. Then, our method can be applied to all the regions. The maximum similarity of all the regions is the final tracking result. The computation cost for the exhaustive search is increased by multiple times which is equal to the number of the regions.

## VIII. Conclusions

This paper has described an object tracking method using 2-D Gabor wavelets, a 2-D mesh, and a 2-D golden section algorithm. By using the GWT coefficients to represent the local features, one is able to track objects in a video sequence with large deformations or in which objects undergoes affine transforms—conditions under which many existing motion estimation methods fail. Because the GWT coefficients represent the local features centered on feature points, they are robust to deformations and transformations. By randomly selecting feature points based on their GWT energy, our approach represents an object more efficiently than the traditional uniform grid-based methods or edge-based methods developed in computer vision. In comparison with the methods recently developed for video object tracking such as block or mesh based approaches, our method provides a more comprehensive representation of the object since every feature point is selected based on the Gabor wavelet transform which reveals both spatial and temporal information of the feature point. These feature points provide the possibility of constructing a rich, complete, and multi-resolution representation of the object.

Another advantage of the technique is that during the feature point selection, the background is easily removed as its GWT coefficients are usually small. The global placement of a feature is represented by an area vector invariant to the affine transform, which is a reasonable transform model explaining object appearances in consecutive frames. In general, it is difficult to apply a gold-section algorithm directly to a similarity function as one cannot guarantee that the similarity function is unimodal. However, by representing the object using the amplitude of the GWT coefficients, which are related to local features, we note that similarity function is close to unimodal. Thus, the golden section algorithm can be applied in a search for the object to save computation time. Finally, our experimental results show the effective advantages of our proposed method.

## References

[1] P. van Beek, A. Murat Tekalp, N. Zhuang, I. Celasun, and M. Xia, “Hierarchical 2D mesh representation, tracking and compression for object-based video,” *IEEE Trans. Circuits Syst. Video Technol., Special Issue*, vol. 9, pp. 353–369, Mar. 1999.

[2] S. C. Han and J. Woods, “Spatiotemporal subband/wavelet coding of video with object-based motion information,” in *Proc. ICIP’97*, vol. II, 1997, pp. 629–632.

[3] B. Gunsel, A. Tekalp, and P. van Beek, “Content-based access to video object: Temporal segmentation, visual summarization, and feature extraction,” *Signal Process.*, vol. 66, pp. 261–280, Apr. 1998.

[4] Y. Altunbasak and A. Murat Tekalp, “Occlusion-adaptive, content-based mesh design and forward tracking,” *IEEE Trans. Image Processing*, vol. 6, pp. 1270–1280, Sept. 1997.

[5] C. Stiller, “Object oriented video coding employing dense motion fields,” in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. V, 1994, pp. 273–276.

[6] P. van Beek and A. Tekalp, “Object-based video coding using forward tracking 2-D mesh layers,” in *Proc. SPIE, Visual Communications and Image Processing ’97*,, vol. 3024, Feb. 1997, pp. 699–710.

[7] Y. Deng, D. Mukherjee, and B. S. Manjunath, “NeTra-V: Toward an object-based video representation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 616–627, Sept. 1998.

[8] H. Musmann, M. Hotter, and J. Ostermann, “Object-oriented analysis-synthesis coding of moving images,” *Signal Process.: Image Commun.*, vol. 1, pp. 117–138, Oct. 1989.

[9] *Overview of the MPEG-4 Standard*, ISO-IEC/JTC1/SC29/WG11, June 2000.

[10] *Overview of the MPEG-7 Standard*, ISO-IEC/JTC1/SC29/WG11, June 2000.

[11] M. G. Ramos and S. S. Hemami, “Eigenfeatures coding of videoconferencing sequences,” in *Proc. Visual Communications and Image Processing ’96*, Orlando, FL, Mar. 1996.

[12] D. Wang, “Unsupervised video segmentation based on watersheds and temporal tracking,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 539–545, Sep. 1998.

[13] C. Gu and M.-C. Lee, “Semiautomatic segmentation and tracking of semantic video objects,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 572–584, Sep. 1998.

[14] F. Bremond and M. Thonnat, “Tracking multiple nonrigid objects in video sequences,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 585–591, Sept. 1998.

[15] D. Gabor, “Theory of communication,” *J. Inst. Elect. Eng.*, vol. 93, no. 26, pp. 429–459, 1946.

[16] S. Marcelja, “Mathematical description of the responses of simple cortical cells,” *J. Opt. Soc. Amer.*, vol. 70, pp. 1297–1300, 1980.

[17] J. G. Daugman, “Two-dimensional spectral analysis of cortical receptive field profile,” *Vis. Res.*, vol. 20, pp. 847–856, 1980.

[18] ——, “Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters,” *J. Opt. Soc. Amer.*, vol. 2, no. 7, pp. 1160–1169, 1985.

[19] T. S. Lee, “Image representation using 2D Gabor wavelets,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 959–971, Oct. 1996.

[20] B. S. Manjunath and W. Y. Ma, “Texture features for browsing and retrieval of image data,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 837–842, Aug. 1996.

[21] A. Jain and G. Healey, “A multiscale representation including opponent color features for texture recognition,” *IEEE Trans. Image Processing*, vol. 7, pp. 124–128, Jan. 1998.

[22] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Wurtz, and W. Konen, “Distortion invariant object recognition in the dynamic link architecture,” *IEEE Trans. Comput.*, vol. 42, pp. 300–311, Mar. 1993.

[23] X. Wu and B. Bhanu, “Gabor wavelet representation for 3-D object recognition,” *IEEE Trans. Image Processing*, vol. 6, pp. 47–64, Jan. 1997.

[24] S. Mallat, *A Wavelet Tour of Signal Processing*. New York: Academic, 1998.

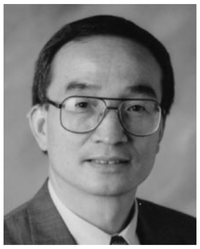
[25] R. P. Wurtz, “Object recognition robust under translation, deformations, and changes in background,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 7, pp. 769–775, Jul. 1997.

[26] B. S. Manjunath, C. Shekhar, and R. Chellappa, “A new approach to image feature detection with applications,” *Pattern Recognit.*, vol. 31, pp. 627–640, 1996.

[27] Z. Wang and J. Ben-Arie, “SVD and log–log frequency sampling with Gabor kernels for invariant pictorial recognition,” in *Proc. 1997 IEEE Int. Conf. Image Processing (ICIP’97)*, vol. 3, Oct. 1997, pp. 162–165.

[28] G. E. Forsythe, M. A. Malcolm, and C. B. Moler, *Computer Methods for Mathematical Computations*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

**Chao He** received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 1995, and the M.S. degree in electrical engineering from The Ohio State University (OSU), Columbus, in 2000. He is currently pursuing the Ph.D. degree at OSU. His research interest includes audio/video compression and streaming.



**Yuan F. Zheng** (S'82–M'86–SM'90–F'97) received the M.S. and Ph.D. degrees in electrical engineering from The Ohio State University, Columbus, in 1980 and 1984, respectively. His undergraduate education was received at Tsinghua University, Beijing, China in 1970.

From 1984 to 1989, he was with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC. Since August 1989, he has been with The Ohio State University (OSU), where he is currently Professor and Chairman of Electrical Engineering. His research interests include two aspects. One is in robotics which includes multiple robots coordination, legged and wheeled mobile robots, human–robot coordination, personal robotics, and special robots for bio-applications. The other is in wavelet transform for multimedia compression. Current efforts for the latter focus on content-based compression, 3-D wavelet transformation, video object tracking, and content-based retransmission in communications. He is currently on the Editorial Board of *Autonomous Robots*, an associate editor of the *International Journal of Intelligent Automation and Soft Computing*, and on the Editorial Board of the *International Journal of Intelligent Control and Systems*.

Prof. Zheng was Vice-President for Technical Affairs of the IEEE Robotics and Automation Society from 1996 to 1999. He was an associate editor of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION between 1995–1997. He was the Program Chair of the 1999 IEEE International Conference on Robotics and Automation, held in Detroit, MI, on May 10–15, 1999.

**Stanley C. Ahalt** (M'86) received the B.S.E.E. and M.S.E.E. degrees from the Virginia Polytechnic Institute and State University, Blacksburg, in 1978 and 1980, respectively, and the Ph.D. degree in electrical engineering from Clemson University, Clemson, SC, in 1986.

Since 1987, he has been with the Department of Electrical Engineering, The Ohio State Unievrsity, Columbus, where he is currently a Professor. During 1980 and 1981, he worked at Bell Telephone Laboratories, where he developed industrial data products. During 1996, he was on sabbatical leave from OSU as a visiting professor at the Universidad Polytecnica de Madrid, Spain. His research interests include high performance computing, pattern recognition and data compression, with applications to automatic target recognition, image compression, and video annotation. He has published over 90 archival journal papers, conference papers, and book chapters in these areas.

Dr. Ahalt is a member of the International Neural Network Society and the IEEE Circuits and Systems, and Signal Processing societies. He is a former editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS, and was the recipient of the 1997 OSU Lumley Research Award and the 2000 OSU College of Engineering Research Award.