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

Chao He, Jianyu Dong, Yuan F. Zheng and Stanley C. Aihalt
Department of Electrical Engineering
The Ohio State University
Columbus, OH 43210
Emails:{hec,zheng}@ee.eng.ohio-state.edu

Abstract—This paper presents a new method for tracking an object in a video sequence which uses a 2-D Gabor wavelet transform (GWT), a 2-D mesh, and a 2-D golden section algorithm. An object 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. The amplitudes of the GWT coefficients of a feature point are then used as the local feature. 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. 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 video sequence, 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.

Keywords—Object tracking, Feature points, Gabor wavelet, Mesh, Golden section.

I. INTRODUCTION

DIGITAL video processing such as video indexing, video manipulation, and video compression is changing from frame-based approaches to object-based or content-based approaches ([1],[2],[3],[4]). 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 and indexing of the video content at the object level.

Object tracking, which attempts to locate, in successive frames, all objects which appear in the current frame, is essential for purposes of processing. The most straightforward approach to this task is to consider objects as rectangular blocks and use traditional block matching algorithms [5]. However, since objects may have irregular shapes and deformations in different frames, video spatial segmentation and object temporal tracking can be combined ([6],[7]). In [7], a supervised I-frame segmentation and unsupervised P-frame tracking algorithm based on motion estimation is proposed, and in [6], 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.

In ([1],[3],[4]) an object is represented by a set of nodes and connecting line segments which usually form a nonuniform mesh. 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 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.

Gabor functions and wavelets, originally proposed by Gabor [8], have achieved impressive results when used for texture and object recognition. Gabor wavelets are very suitable for representing local features based on its useful properties: (a) Gabor wavelets are the best time and frequency localized wavelets, (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 ([9],[10],[11]). Gabor functions have been used for image processing in a number of ways, such as the complete image representation [12]. In ([13],[14]) the mean and standard deviation of the GWT coefficients have been used successfully in texture browsing and retrieval. In [15], the magnitude and the phase of the coefficients of the GWT, along with a uniform, elastic, and deformable grid of sampling points have been used for 3-D object recognition. The Gabor wavelets are used for local feature representation and the grid is used for encoding both signal energy and structural information of an object model. Then, a flexible matching algorithm based on simulated annealing is applied to detect or recognize the object.

In this paper, we combine the 2-D Gabor wavelets idea in [15] and the 2-D mesh idea in ([1],[3],[4]) for object tracking in the object-based or content-based video processing. The amplitudes of the GWT coefficients instead of the local intensity in ([1],[3],[4]) are used to represent local features, and a nonuniform 2-D mesh instead of a uniform grid in [15] is used to represent global features. Since our technique is applied to ordinary video sequence instead of the infrared imagery in [15], we present three new methods here based on the characteristics of the ordinary video sequence: (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 [13]:

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

$$G(u, v) = \exp \left\{ -\frac{1}{2} \left( \frac{|uW - vN|^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2} \right) \right\}. \quad (2)$$

where $\sigma_x$ and $\sigma_y$ are the standard deviations 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 deviations of $G(u, v)$ along the u and v axes, respectively. The Gabor function is well known for its optimal time-frequency localization [16].

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

$$g_{mn}(x, y) = a^{-m} g(x', y'), \quad a > 1, m = 1 : M, n = 1 : N. \quad (3)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = a^{-m} \begin{bmatrix} \cos \theta_n & \sin \theta_n \\ -\sin \theta_n & \cos \theta_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \quad \theta_n = n\pi/N. \quad (4)$$

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 size or the bandwidth of the Gabor wavelets is also controlled by $a^{-m}$ in order to keep the energy of the Gabor wavelets constant.

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.

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

$$\hat{I}(x, y, m, n) = \int I(x', y') g_{mn}(x - x', y - y')dx' dy'. \quad (5)$$

Referring to Eq. (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 [15]:

$$L(x, y) = \{abs(\hat{I}(x, y, m, n)), m = 1 : M, n = 1 : N\}. \quad (6)$$

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

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, some researchers have used a uniform grid placed on the object and the nodes of the grid are then selected as the feature points [15].

Another concern is that different points have different levels of importance in object representation. For example, in a noisy environment, some noise-contaminated points may be ill-suited for use in the object representation. To avoid selection of these points, an amplitude threshold can be used [17]. 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 [18]. The point with the maximum value of this function in a local neighborhood is selected as a feature point $\{(x, y)\}$.

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.

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.

We propose a new selection scheme to avoid these problems. 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 that region. The result is similar to [4] but our scheme is much simpler. We describe this process as follows.
First, we define the energy of the GWT coefficients of point \((x, y)\) as:

\[
E(x, y) = \sum_{m=1}^{M} \sum_{n=1}^{N} |\hat{f}(x, y, m, n)|^2.
\]

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

\[
S(x, y) = f_s(E(x, y)).
\]

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 [(9),(10),(11)]. 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.

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 \(S_r(x, y)\):

\[
S_r(x, y) = S(x, y)R(x, y).
\]

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 \(T\) depending on the desired number of the feature points is set for \(S_r(x, y)\), and those points with \(S_r(x, y) > T\) will be selected as feature points:

\[
\{(x_j, y_j)\} = \{(x, y) | S_r(x, y) > T\}.
\]

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

\[
P((x, y) \in \{(x_j, y_j)\}) = \frac{1}{2} - \text{erf}(\frac{T}{S(x, y)}).
\]

where \(\text{erf}(x)\) is the error function and defined as \(\frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2} dt\). The points with higher energy are more likely to be selected as \(\text{erf}(x)\) is an increasing function.

All points are evaluated equally. Figure 1 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 [15] 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 are selected by the grid approach 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.

IV. Global Feature

After calculating the GWT and selecting the feature points, 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 \(\{(x_j(n), y_j(n))|n = 1 : N_p\}\) distributed on the object. First we select a new point \((x_j(i), y_j(i))\) from \(i=1\) to \(N_p\) and connect this point to all old points \(\{(x_j(j), y_j(j))|j = 1 : i - 1\}\) 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. We use \(N_t\) and \(N_t\) to denote the number of the line segments and triangles retained after the procedure is completed, respectively. Figure 1 shows an example of this mesh generating procedure.

In [1],[4], a small region is defined for every node to limit the mesh’s deformation but inside the region there is no constraint. In [15], deformed grid is represented by the lengths of the line segments and angles between them. But this representation is variant to the affine transform. 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, \(G = [a(1), a(2), ..., a(N_t)]\), where \(a(i)\) is the area of the \(i\)th triangle.

The affine transform can be modeled by the following formula [19]:

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

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 \(G_a = [a_a(1), a_a(2), ..., a_a(N_t)]\), \(a_a(i)\) 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) \cite{19}, the matrix $C$ can be decomposed into three parts:

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

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 one major idea behind our tracking algorithm.

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. We basically follow the method in \cite{15}. The two representations should have the same structure such as $N_p, N_t$, and $N_i$, 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. Therefore, the final similarity measure is the weighted sum of the local and global similarities. The local feature similarity, $S_l$, is the average of individual local feature similarities between each pair of corresponding feature points and the individual local feature similarity $s_l(i)$ between the local features of corresponding feature points $(x_{f_a}(i), y_{f_a}(i))$ in the object and $(x_{f_m}(i), y_{f_m}(i))$ in the model is defined as \cite{15}:

$$
s_l(i) = \frac{L_0(i) \cdot L_M(i)}{\|L_0(i)\| \times \|L_M(i)\|} \min\left(\frac{\|L_M(i)\|}{\|L_0(i)\|}, \frac{\|L_M(i)\|}{\|L_0(i)\|}\right),
$$

where $L_M(i)$ is the local feature vector of $(x_{f_m}(i), y_{f_m}(i))$ in the model, and $L_0(i)$ is the local feature vector of $(x_{f_a}(i), y_{f_a}(i))$ 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 average of $s_l(i)$ for all feature point pairs:

$$
S_l = \frac{1}{N_p} \sum_{i=1}^{N_p} s_l(i).
$$

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, $G_M = (a_m(1), a_m(2), \ldots, a_m(N_t))$, and the object’s area vector, $G_O = (a_o(1), a_o(2), \ldots, a_o(N_t))$:

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

where $k$ is a parameter used to balance the sensitivity of the local and global similarities which depends on the deformation of the object.

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

$$
S = \alpha S_l + (1 - \alpha) S_g
$$

where $\alpha$ is a weighting parameter which can be selected through experiment.

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 \cite{15}. Another popular method is to search the mesh nodes independently \cite{3}, \cite{4}.

Here we propose a 2-D golden section searching algorithm to find the object in the next frame. Our method utilizes the global information so that the object is tracked efficiently, and it is fast because of the golden section algorithm.

In a one dimensional optimization problem, the golden section algorithm can be used to find a global maximum if the function is unimodal \cite{20}. Suppose that a function is defined on $[0, 1]$, and $\tau$ is the unique maximum of $f(x)$ on $[0, 1]$. If $f(x)$ strictly increases for $x < \tau$ and strictly decreases for $x > \tau$, it is called unimodal. The golden section search is based on Fibonacci numbers which are defined as:

$$
F_0 = F_1 = 1, F_2 = F_0 + F_1, F_3 = F_1 + F_2, \ldots, F_k = F_{k-1} + F_{k-2}.
$$

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 $\tau$. 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 $k \rightarrow \infty$, $\frac{F_k}{F_{k-1}} = \rho \approx 0.618$, so the algorithm is called the golden section algorithm.

Here we expand the above golden section algorithm from 1-dimensional to 2-dimensional. Suppose that a function $f(x,y)$ is defined on a rectangle, $a_k b_k c_k d_k$, as shown in Figure 2, and $(x, y)$ is the unique maximum of $f(x, y)$. If $f(x, y)$ decreases when the distance between $(x, y)$ and $(x, y)$ 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 $a_{k,j}g_kd_k$. 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. Figure 3 shows an example of the similarity function between the first and fifth frame of the Miss America video sequence.

The maximum point of $S_{MO}(x, y)$ is the coarse position of the object in the next frame, and 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 [13]. For every feature point, we simply move it around in a local searching region to find its best matching position.

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 video sequences. The parameters of the Gabor wavelets are selected by the experiments, and for convenience, we express these parameters in the Fourier domain. Our experiments show that using more than 4 frequencies and 5 orientations does not improve the performance significantly, but increase the computational complexity. Consequently, we limit our experimental parameters accordingly. The highest frequency is $3\pi/4$ and the scaling ratio is $\sqrt{2}$. The five orientations are $0, \pi/5, 2\pi/5, 3\pi/5$ and $4\pi/5$. The standard deviation, $\sigma_w$, along the long axis of the Gabor wavelet with the highest frequency is $\pi/3$. The scaling ratio of the standard deviation is also $\sqrt{2}$. The ratio between the standard deviation of the long and short axes is 2.

Although the analytic Gabor wavelet extends to infinity, its value converges to zero and its energy is concentrated near the center. Therefore, we truncate the Gabor wavelets using a rectangular window to obtain finite truncated Gabor wavelets. The size of the window is a trade-off between the computational complexity, and the precision of the transform, and depends on the selection of the parameters of the Gabor wavelets. We choose a window size $19 \times 19$ which is sufficiently large when used with the above selected parameters. In the similarity measure, we choose $\alpha = 0.5$ and $k$ between 5 and 10.

In the first searching step, the size of the searching area in the golden section algorithm is limited in order to keep the similarity function unimodal. We select the size to be less than $37 \times 37$ which is two times the window used for the truncation of the Gabor wavelets. In the second refinement step, the searching rectangle is $11 \times 11$. 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.

We use the Miss America sequence to test the ability of the algorithm to track object deformations and shifts. 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. Figure 4 shows the first frame of the sequence. Figure 5 shows the tracking result in the last frame. The solid mesh is the tracking result of our proposed method.
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. 4. The first frame of the Miss America sequence.

Fig. 5. The tracking result of the 100th frame of the Miss America sequence tracked by the proposed method.

VIII. CONCLUSIONS

This paper has described a powerful new scheme for object tracking using a 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 frequency information of the feature point. These feature points provide the possibility of constructing a rich, complete, and multi-resolution representation of the object. Our experimental results show the effective advantages of our proposed method.

REFERENCES