Image matching with higher-order geometric features

C. Paramanand* and A. N. Rajagopalan

Image Processing and Computer Vision Laboratory, Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai—600 036, India
**Corresponding author: paramanand@gmail.com*

Received May 6, 2009; revised January 25, 2010; accepted January 26, 2010; posted January 26, 2010 (Doc. ID 110956); published March 18, 2010

We propose a geometric matching technique in which line segments and elliptical arcs are used as edge features. The use of these higher-order features renders feature representation efficient. We derive distance measures to evaluate the similarity between the features of the model and those of the image. The model transformation parameters are found by searching a 3-D transformation space using cell-decomposition. The performance of the proposed method is quite good when tested on a variety of images. © 2010 Optical Society of America

OCIS codes: 100.5010, 100.4995.

1. INTRODUCTION

Image matching refers to the process of locating known objects. It is necessary for many applications including industrial inspection, robot navigation, aerial image analysis, and image retrieval [1,2]. The edge maps of the model object and the image are usually represented by points [1,3], lines [4,5], circles [6], and other shape descriptors [7,8]. The technique proposed in [9] uses a new set of features called virtual circles which represent the empty space between edge points and not the actual edges. The features are compared in order to locate the presence of the model in the probe image. There are two basic approaches to find a model in an image: those based on feature correspondences which use pairing between model features and image features [6,10], and those which search for the best possible transformations within a transformation space [4,11–13]. Our focus in this paper is on the latter approach.

In Hough transform-based techniques [11,14], the transformation space is represented by a grid of bins. A large number of pairings between model and image features is generated and based on the feature correspondences, the counters of bins are incremented. Bins having a large count are considered as good candidates for matching. This approach can be inaccurate since false peaks can occur even at incorrect transformations [4]. The more popular point-matching algorithm proposed by Huttenlocher *et al.* [12] finds the transformations that minimize the Hausdorff distance between the model and image point-sets. This technique can effectively deal with outliers as well as occlusions. Many speed-up strategies have also been subsequently developed [1]. In [15], Hausdorff distance-based matching has been applied on line segment features. Line segments are treated as points in 4-D space, which reduces it to a 4-D point-matching problem under translation. The main disadvantage is that when there are breaks in line segments (due to occlusions or fragmented image data), significant errors are introduced in the location of line segments due to change in the length and the mid-point that can degrade the performance [15]. For other transformation space-based approaches to matching line segments and circles, see [16–20].

In this paper, we develop a geometric matching scheme in which images are represented using line segments and arcs [6,21]. We also derive expressions to evaluate the similarity between these higher-order features. The model transformation is considered to be 2-D translation and rotation which is typical in an industrial scenario. The model is located by searching the transformation space using cell-decomposition [22]. We would like to mention that the work described here is a significant extension over the authors' own previous works [23,24]. The experimental section of this paper contains several new results pertaining to accuracy, noise tolerance, and performance in the presence of breakpoints, occlusions and outliers. Also, we have included comparisons with the point-matching technique as well as our implementation using line segments alone as features. The novelties of our work can be summarized as follows: (i) We develop a transformation space-based matching technique using line segments and elliptical arcs. (ii) Our representation is inherently compact due to the use of higher-order features. (iii) We construct closed-form expressions for the distance measures to perform feature matching in the presence of occlusions, illumination variations, and clutter. (iv) Our method is considerably faster than point-matching and line-matching techniques.

In Section 2, we discuss feature representation. Distance measures to compare model and image features are

derived in Section 3. The matching scheme is presented in Section 4. Experimental results are given in Section 5, and we conclude with Section 6.

2. FEATURE REPRESENTATION

The motivation for using edge features instead of intensities is to reduce the amount of information to be processed by the recognition algorithm [4]. In addition, edge features are relatively more stable under illumination variations and/or pose changes.

We follow the technique proposed in one of our earlier works [23] to obtain a representation of the edge map of gray-scale images in terms of line segments and elliptical arcs. We detect points with significant local intensity change (i.e., edge points) using the Canny edge operator. Edge linking is performed using an 8-pixel neighborhood to get connected sets of edge points. Following this, critical points (which are points at which there is a change in the shape of the edges) are detected in each connected set using the curvature measure proposed in [23]. This measure evaluates the ratio of change of the tangential angle with respect to curve length. For more details, see [23]. The edge map is then divided into edge segments at the critical points. These segments are then subject to line fitting and ellipse fitting. A total least-squares approach [25] is followed for fitting the edge points to line segments. Ellipse fitting is based on the least-squares algorithm proposed by Fitzgibbon *et al.* [26] wherein the algebraic distance is minimized subject to the constraint of ellipticity. This technique is preferred because it is ellipse-specific, computationally efficient, and robust to noise. Those edge segments that can neither be fit by a line segment nor by an elliptical arc are segmented with a combination of line segments and elliptical arcs [23]. For feature representation, we store the line equation coefficients, the end point coordinates, the ellipse parameters (center, length of axes, angle), the extent of the elliptical arc, and the length of each feature (in terms of number of edge pixels).

When conditions are not very well controlled (as is usually the case in a real-world scenario), features of the same object can change across different images due to occlusions, illumination variations, or fragmented image data. In Figs. 1(a) and 1(b), we show two instances of the image of a computer mouse. We obtain their representation in terms of line segments and elliptical arcs. For purpose of depiction, we show the entire ellipse that is fitted to a curved feature. In Fig. 1(c), which corresponds to the feature representation of Fig. 1(a), we see that the mouse has been represented by one elliptical arc and a few line segments (elliptical arcs are shown in pink color and line segments are shown in green color). On the other hand, in Fig. 1(d), which corresponds to the feature representation of Fig. 1(b), we observe that the edge segments between points P_a and P_b, and those between points P_b and P_c have been represented by two different elliptical arcs. Note that this representation is different from that of Fig. 1(c). This is because the edge points between P_a and P_b, and the points between P_b and P_c are separated by the cable (unexpectedly) running over the mouse in Fig. 1(b) and are linked as separate edge segments. The point we wish to emphasize is that a matching algorithm should be robust to such feature distortions.

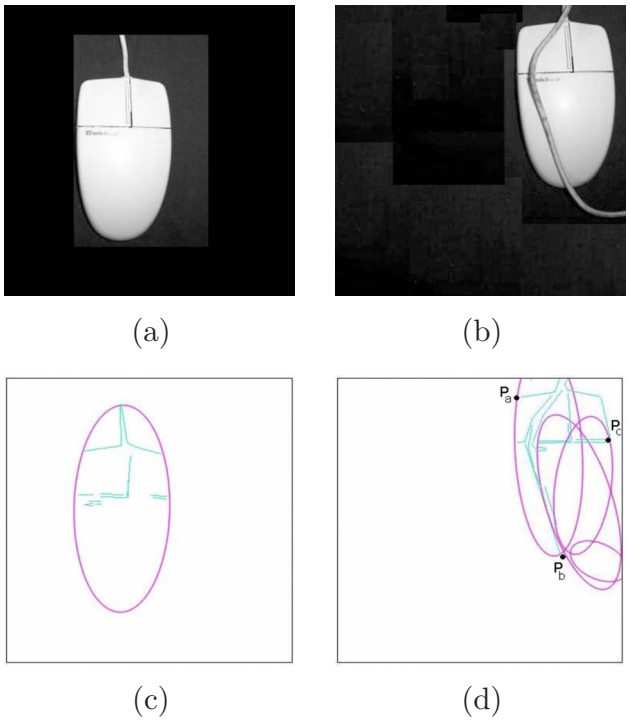


Fig. 1. (Color online) (a),(b) Images of a computer mouse. (c), (d). Feature representation of Figs. 1(a) and 1(b), respectively.

3. FEATURE MATCHING

The line segment and elliptical arc features of the model and probe images are obtained as discussed in the previous section and these features are used for matching. We need to compare model line segments with image line segments, and model elliptical arcs with image elliptical arcs. When there are breaks, a part of the model elliptical arc can also occasionally be fitted to a line segment. Hence, we need to be able to compare model elliptical arcs with line segments, too. Another effect of breaks is that the parameters of model features such as the mid-point and the end points of line segments, and also the center, the axes, and the orientation of elliptical arcs can be altered, which precludes a direct comparison of the feature parameters. To address this problem, we propose closed-form distance measures that can also handle breaks.

A. Line-to-Line Distance

Let m_l and i_l denote a model line segment and image line segment, respectively. We compare m_l and i_l using distance measure $d_l(m_l,i_l)$, which is based on the integrated squared perpendicular distance (ISPD) defined in [5]. The ISPD denoted by $d_s(m_l,i_l)$ is given by $d_s(m_l,i_l) = (\text{len}(i_l)/3)(v_1^2+v_1v_2+v_2^2)$, where v_1 and v_2 are perpendicular distances between the infinitely extended model line m_l and the two end points of the image line segment i_l, and len (i_l) denotes the length of i_l. Distance measure d_s can be used to match line segments despite breaks in them. However, it yields a low value even if the line segments m_l and i_l are far apart but are collinear [18]. To

overcome this, we modify d_s by considering the distance v_3 between the mid-points of the model and image line segments and use it to penalize collinear lines that are far apart. Distance $d_l(m_l,i_l)$ is thus given by

$$d_l(m_l,i_l)=\begin{cases} d_s(m_l,i_l), & v_3 \leqslant \tau_l \\ d_s(m_l,i_l)+v_3^2, & v_3 > \tau_l \end{cases}, \quad (1)$$

where threshold τ_l is set to half the length of the model line segment m_l, since at the correct transformation, v_3 will never be greater than $\text{len}(m_l)/2$, irrespective of the number of breaks.

B. Ellipse-to-Ellipse Distance

We propose distance measure d_e to compare a model elliptical arc m_e with an image elliptical arc i_e. Evaluating the Euclidean distance between a point and an ellipse is complicated as it involves solving a fourth-order polynomial. In contrast, the algebraic distance is simpler to compute [27]. The algebraic distance between a point and an ellipse increases as the Euclidean distance increases and is zero when the point lies on the ellipse [28]. Distance d_e is obtained by integrating the square of the algebraic distance between a point on i_e and model ellipse m_e along the image elliptical arc.

Consider a model ellipse m_e with center (h_m,k_m), major axis a_m, and minor axis b_m. The algebraic distance between a point $p_j=(x_j,y_j)$ and ellipse m_e is given by $e_j=((x_j-h_m)^2/a_m^2)+((y_j-k_m)^2/b_m^2)-1$. When the model ellipse has orientation α_m,

$$e_j=\frac{(x_j\cos\alpha_m+y_j\sin\alpha_m-h_r)^2}{a_m^2}+\frac{(-x_j\sin\alpha_m+y_j\cos\alpha_m-k_r)^2}{b_m^2}-1, \quad (2)$$

where $h_r=h_m\cos\alpha_m+k_m\sin\alpha_m$ and $k_r=-h_m\sin\alpha_m+k_m\cos\alpha_m$.

Consider an image elliptical arc i_e with center (h_i,k_i) and extending from θ_1 to θ_2 rad as shown in Fig. 2. Let a_i, b_i, and α_i denote the major axis, the minor axis and the orientation of i_e, respectively. The coordinates of a point $p_j=(x_j,y_j)$ on i_e can be written as $x_j=((a_i\cos\theta\cos\alpha_i-b_i\sin\theta\sin\alpha_i)+h_i)$ and $y_j=((a_i\cos\theta\sin\alpha_i+b_i\sin\theta\cos\alpha_i)+k_i)$. The algebraic distance between a point on the image elliptical arc and the model ellipse is then

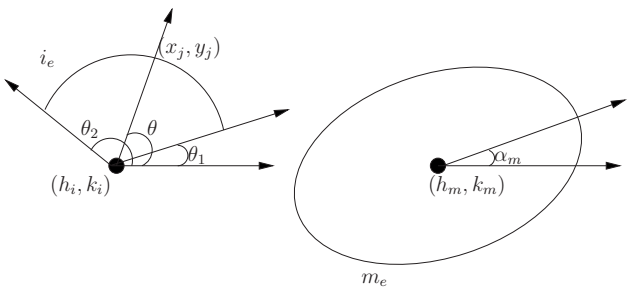


Fig. 2. Distance between model and image elliptical arcs.

$$e(\theta)=\frac{(x(\theta)-h')^2}{a_m^2}+\frac{(y(\theta)-k')^2}{b_m^2}-1, \quad (3)$$

where $x(\theta)=(a_i\cos\theta\cos\alpha_d-b_i\sin\theta\sin\alpha_d)$, $y(\theta)=(a_i\cos\theta\sin\alpha_d+b_i\sin\theta\cos\alpha_d)$, $\alpha_d=\alpha_i-\alpha_m$, $h'=(h_m-h_i)\cos\alpha_m+(k_m-k_i)\sin\alpha_m$, and $k'=-(h_m-h_i)\sin\alpha_m+(k_m-k_i)\cos\alpha_m$.

The distance measure d_e between m_e and i_e is obtained by integrating the square of the distance $e(\theta)$ from θ_1 to θ_2 as

$$d_e(m_e,i_e)=\int_{\theta_1}^{\theta_2} e^2(\theta)d\theta. \quad (4)$$

Interestingly, this yields a closed form solution for d_e as a function of model and image ellipse parameters, as given in Appendix A.

In Section 2, we noted that the dimensions of ellipses representing the same object can vary across different images. Figure 3(a) shows a model consisting of an ellipse and Fig. 3(b) shows an image in which the model ellipse has been broken into four smaller arcs. In the representation shown in Fig. 3(c), we see that the image elliptical arcs differ in dimensions from the original model due to breaks in the ellipse. The translation at which distance d_e between the model and image ellipses is a minimum is shown in Fig. 3(d), which is indeed correct. Thus, the distance measure d_e can be used to match elliptical arcs even when there are breaks.

C. Ellipse-to-Line Distance

We next derive distance measure d_{e_l} to compare model elliptical arcs and image line segments. Consider a line segment with end points (x_1,y_1) and (x_2,y_2), and whose equation is $y=mx+g$. Also, consider an elliptical arc with center (h_m,k_m), major axis a_m, minor axis b_m, and orien-

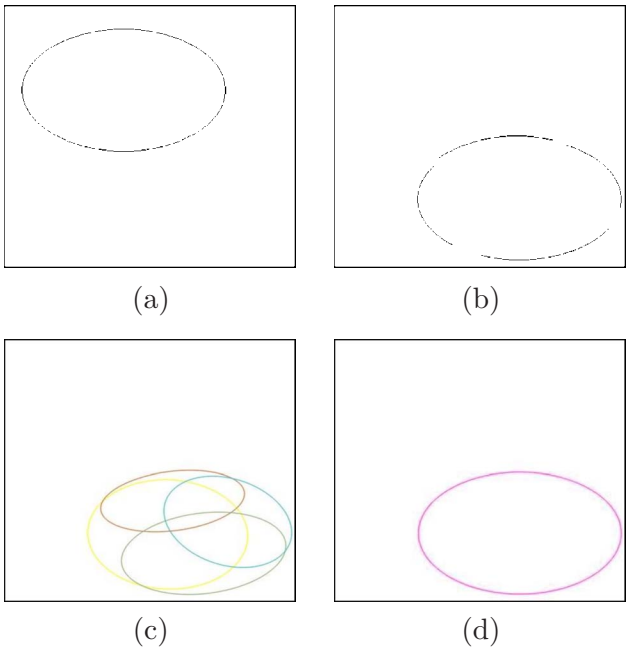


Fig. 3. (Color online) (a) Model ellipse. (b) Target image containing a distorted instance of the model. (c) Feature representation of Fig. 3(b). (d) Model located in the target image.

tation α_m rad. A point on the image line segment can be written as $p_j=(x_j,y_j)=(x_j,mx_j+g)$. Then, the algebraic distance between the model ellipse and a point on the image line segment is

$$e_{j_l}=\frac{(ux_j+g\sin\alpha_m-h_r)^2}{a_m^2}+\frac{(vx_j+g\cos\alpha_m-k_r)^2}{b_m^2}-1, \tag{5}$$

where $h_r=h_m\cos\alpha_m+k_m\sin\alpha_m$, $k_r=-h_m\sin\alpha_m+k_m\cos\alpha_m$, $u=\cos(\alpha_m)+m\sin(\alpha_m)$, and $v=m\cos(\alpha_m)-\sin(\alpha_m)$. The square of the algebraic distance e_{j_l} is integrated along the length of the line segment to get the distance measure d_{e_l} as

$$d_{e_l}=\int_{x=x_1}^{x=x_2}\left\{\frac{(ux+g\sin\alpha_m-h_r)^2}{a_m^2}+\frac{(vx+g\cos\alpha_m-k_r)^2}{b_m^2}-1\right\}^2 dx. \tag{6}$$

On solving Eq. (6), we obtain a closed-form expression for d_{e_l} in terms of the line segment and ellipse parameters as given in Appendix A.

When the equation of the line segment i_l is of the form $x=g'$, the expression for d_{e_l} can be obtained by expressing the squared algebraic distance in terms of the variable y instead of x, and by integrating it.

4. LOCATING THE MODEL

Let the feature sets of the given image and the model be denoted by I and M, respectively. Let I_L and M_L denote the set of all line segments of the image and the model, and I_E and M_E denote the set of all elliptical arcs of the image and the model, respectively. Then, $I=I_L\cup I_E$ and $M=M_L\cup M_E$. The distance measure D between the feature sets of the model and the image when a transformation $t(\cdot)$ is applied on the model can be defined as

$$D(t(M),I)=\max\{\max_{m_l\in M_L}[\min_{i_l\in I_L}\{d_l(t(m_l),i_l)\}],\ \max_{m_e\in M_E}[\min_{i_e,i_l\in I}\{d_e(t(m_e),i_e),d_{e_l}(t(m_e),i_l)\}]\}. \tag{7}$$

Distance $D(t(M),I)$ will take a low value only when every transformed model feature is close to an image feature. However, in practical situations, because of occlusions and outliers, only a fraction of the model features may match the image features. Hence, $D(t(M),I)$ as defined in Eq. (7) cannot be used directly to match images.

We use the length of the features (in terms of number of pixels) as a cue for matching the model and probe image. Let τ_n denote the fraction of the number of model edge pixels N_m (obtained from the edge detector) to assert the presence of the model. Let I_f denote the set of image features that are close to the transformed model features when a transformation $t(\cdot)$ is applied on the model; i.e., $I_f=\{i\in I\,|\,\exists m\in M$ for which $d(t(m),i)<\tau_d\}$. Here, τ_d is the distance threshold and d can be d_l, d_e, or d_{e_l} depending on whether m and i are line segments or elliptical arcs. The number of image pixels that match the model is given by $N_{fi}=\Sigma_{i\in I_f}\text{len}(i)$, where len denotes the length of a feature. For $t(\cdot)$ to qualify as an acceptable transformation, the condition to be satisfied is

$$\frac{N_{fi}}{N_m}>\tau_n. \tag{8}$$

This condition is similar to the forward criterion used in point-matching [1].

When a transformation $t(\cdot)$ is applied on the model features M and the criterion in Eq. (8) is satisfied, we define a modified measure $D_{\text{mod}}(t(M),I)$. Consider the function $\Delta(m)$ defined on a model feature m as

$$\Delta(m)=\min_{i\in I}d(m,i), \tag{9}$$

where d can be d_l, d_e, or d_{e_l} depending on m and i. For each model feature m, the quantity $\Delta(m)$ gives the distance to the closest feature in I. Let M_f denote the set of model features that are matched to image features when a transformation $t(\cdot)$ is applied; i.e., $M_f=\{m\in M\,|\,\Delta(t(m))<\tau_d\}$. Then, we define $D_{\text{mod}}(t(M),I)$ as

$$D_{\text{mod}}(t(M),I)=\max_{m\in M_f}\Delta(t(m))=\max_{m\in M_f}\min_{i\in I}d(t(m),i). \tag{10}$$

Note that distance $D_{\text{mod}}(t(M),I)$ is analogous to the forward partial Hausdorff distance [1], as it enables matching a fraction of model features. While evaluating D_{mod}, we consider only those model features that are close to at least one of the image features, whereas in $D(t(M),I)$ all the model features are considered.

For computational efficiency, we use cell-decomposition [22] to search the discretized 3-D transformation space and find the transformation for which the distance between features is minimum. Initially, the entire transformation space is regarded as an interesting cell. At each step, we divide the interesting cells into eight sub-cells of equal size, apply the transformation corresponding to the center of a sub-cell on the model features, evaluate the distance between features, and check whether the condition in Eq. (8) holds for a large value of τ_d (initially, as described in the experimental section). Only those sub-cells that satisfy Eq. (8) are likely to contain the actual transformation and are regarded as interesting. These cells are further sub-divided into eight smaller cells. The threshold τ_d is gradually reduced as the cell size decreases. In order to determine I_f, every model feature is to be compared with every image feature. However, this task can be effectively done by restricting the comparison of features based on their locations. For a particular model m, we evaluate the distance $d(t(m),i)$ with only those image features i which lie in a region centered around the midpoint of the transformed model. The distances with respect to other image features are set to a high value without incurring the cost of evaluating them. The area of the region is restricted by cell size and the descriptively rich line segments and elliptical model feature parameters. We would like to emphasize that this step is important for quickly eliminating uninteresting cells and results in considerable savings in computations. For further speed-up, when measuring the distance for model ellipti-

cal arcs, instead of computing d_e and d_{e_l}, we use the algebraic distance between the end points of the elliptical arc/line segment and the model ellipse [Eq. (2)] as the distance measure. The process of cell-division is continued recursively until the size of an interesting cell is sufficiently small (till the area of the translation grid is 200 pixels and the angular extent is 1°), whence we stop dividing it and evaluate the distance for all the transformations corresponding to that cell. In this step, we use the actual ellipse distance measure given by Eqs. (A1) and (A2) (in Appendix A) to accurately find the optimum transformation. When multiple instances of the same model are present, all those cells that contain the actual transformations of the model are regarded as interesting during cell-decomposition. Finally, we pick those transformations for which the distance $D_{\text{mod}}(t(M),I)$ is minimized and Eq. (8) also holds.

5. EXPERIMENTAL RESULTS

For purpose of validation, we tested our algorithm on both real and synthetic data. For real examples, we captured images containing different objects in such a way that the model object underwent only translation and/or in-plane rotation motion. Models were obtained by manually cropping specific regions from edge images. All the experiments were run on a P-IV PC with a 2 GHz processor and 256 MB RAM.

We first present results on real images. The proposed algorithm was tested for performance when the model was to be located in a cluttered environment. In the first experiment, we selected the model and probe images from a stereo pair of the Middlebury dataset [29]. The model was arbitrarily chosen from the left stereo image, and its edge map and feature representation are shown in Fig. 4(a). The aim was to locate the instance of the model in the right stereo image shown in Fig. 4(b). The model consisted of 525 edge pixels and was represented by one ellipse and 17 line segments. There were 8626 edge pixels in the target image and these were represented by 30 ellipses and 368 line segments as shown in Fig. 4(c). We note that there are many irrelevant features that can obscure the true features of the model. The images were matched using the proposed scheme with threshold $\tau_n = 0.9$. The optimum transformation parameters of the model were correctly determined as can be seen from Fig. 4(d), where the transformed model features are overlaid on image edges for the purpose of depiction. The rotation angle was found to be zero degrees, as should be the case for this translational stereo pair.

We next applied our method to locate the instance of a model tube in a pharmaceutical image [Fig. 5(b)]. The model edge map and feature representation are shown in Fig. 5(a). Note that the model tube has undergone translation and rotation in the image of Fig. 5(b). There were 553 pixels in the model and these were represented by one elliptical arc and seven line segments [Fig. 5(a)]. The image in Fig. 5(b) had 6810 edge points and these were represented by four elliptical arcs and 140 line segments [Fig. 5(c)]. The threshold τ_n was set as 0.9. The model was correctly located in the image by our method as seen in Fig. 5(d) which shows the transformed model features overlaid on the image edges.

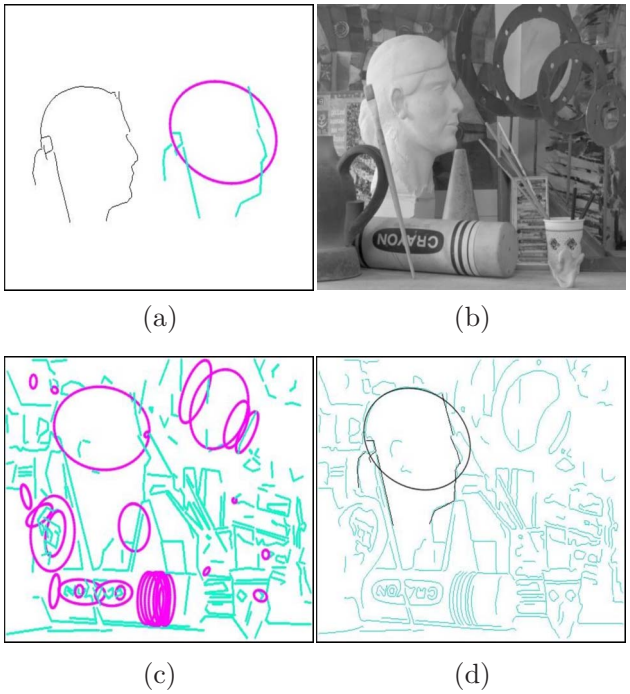


Fig. 4. (Color online) (a) Model image and its features. (b) Given image. (c) Image features. (d) Model correctly located in the image.

In yet another experiment, the objective was to locate the model table-tennis racket [whose gray-scale image and feature representation are shown in Fig. 6(a)] in the image shown in Fig. 6(b). We observe that the racket in Fig. 6(b) is occluded by different objects and its illumina-

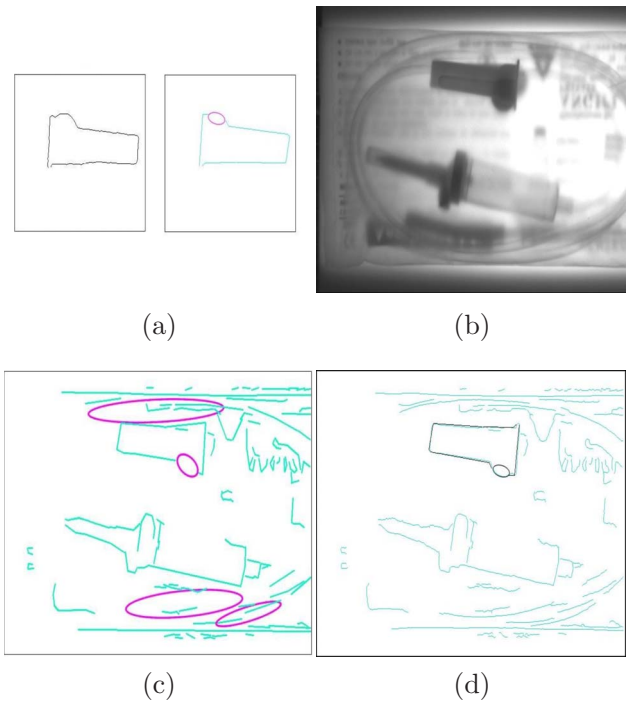


Fig. 5. (Color online) (a) Tube model. (b) Given image. (c) Image features. (d) Output result.

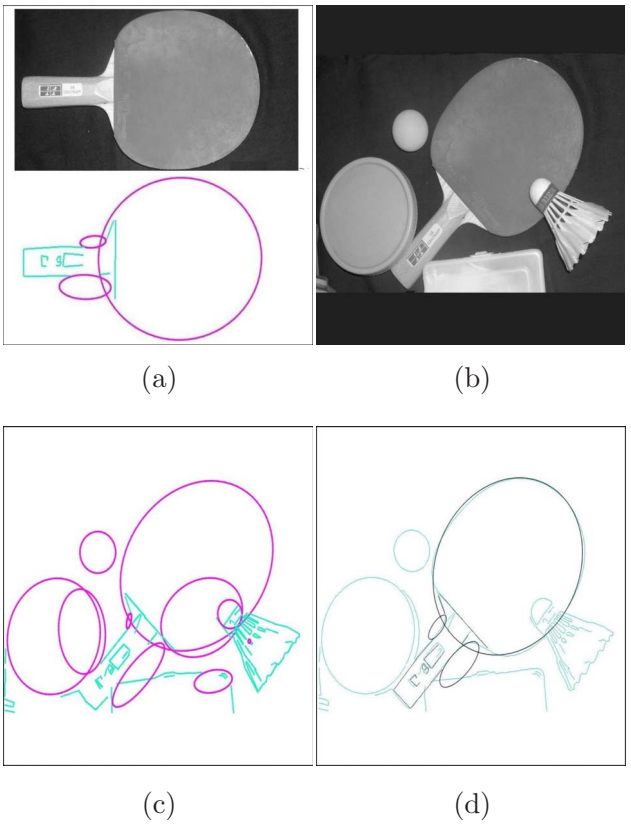


Fig. 6. (Color online) (a) Model racket and its features. (b) Given image. (c) Image features. (d) Racket located in the target image.

tion is also different from the model racket of Fig. 6(a). The model racket consisted of 1724 edge points represented by 18 line segments and three elliptical arcs [Fig. 6(a)]. There were 4771 edge points in the probe image and these were represented by 105 line segments and 10 elliptical arcs [Fig. 6(c)]. The model and image features were matched for $\tau_n = 0.80$. In Fig. 6(c), we observe that the curved region of the racket is split into two elliptical arcs. In the region around the handle, we note that some of the line segments are broken or are missing due to occlusions. Despite these distortions, the algorithm correctly locates the model racket in the probe image as shown in Fig. 6(d).

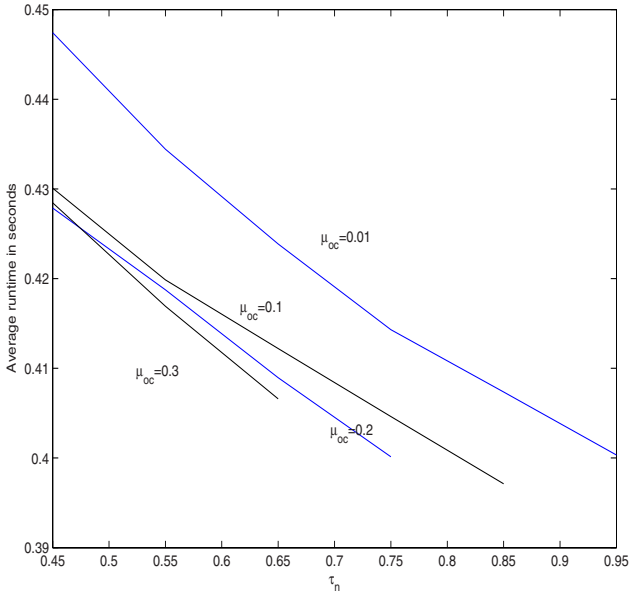
We also tested our algorithm on 20 other real images and compared its performance with the standard point-matching technique (PMT) given in [1]. In our implementation of PMT, we obtained the distance transform of the image using the OpenCv library function 'cvDistTransform.' Through cell-decomposition, we found the transformation that minimizes the forward Hausdorff distance between the model points and image points. The CPU run-times for both the algorithms were calculated using the C function 'clock'. We found that the run-time for our algorithm was between 0.3 and 0.7 seconds, while it was 3 to 6 seconds for PMT. The number of cells examined while determining the model translation and rotation parameters was between 4000 and 8000 for the proposed algorithm. In comparison, the number of cells for PMT was much higher (between 25,000 and 50,000).

To check our implementation of PMT, we placed synthetically generated models in 50 probe images of size 512×512 pixels such that the models underwent only translation. The number of cells examined in the 2-D transformation space by our implementation of PMT was between 2000 and 3000 which is comparable to what has been reported in the literature [1]. Note that the number of cells for translation alone is much smaller as compared to the case when both translation and rotation are present.

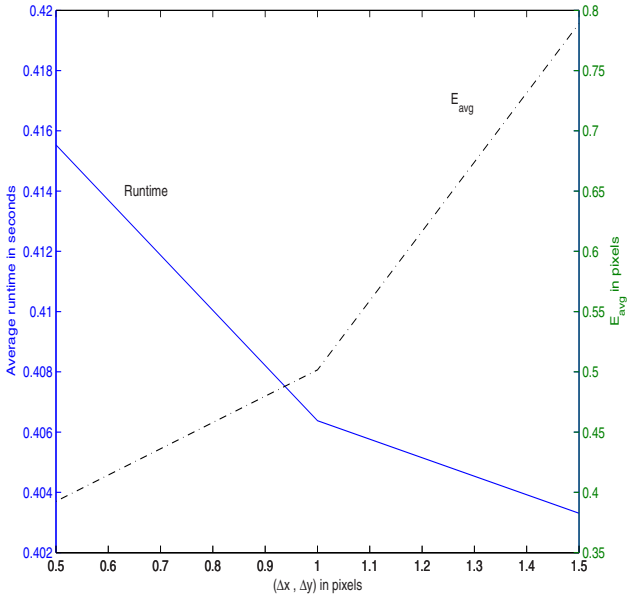
For conducting statistical tests, we generated a large and competitive synthetic dataset of 1550 probe images using 20 different models. Model images of various shapes were manually created on a grid of size 380×320 pixels. Probe images of size 1080×880 pixels were generated by placing a rotated and translated instance of the model. The angles of rotation varied from 0 to 360° in steps of 45°. In addition to the model, different line segments and elliptical arcs having arbitrary parameters, and manually generated shapes were added at random locations in the image in order to simulate breaks due to occlusions, clutter, and illumination variations. The degree of outliers varied from 100% to 600%. The locations of these additional objects were uniformly distributed throughout the image.

Using a subset of images from this dataset, we performed studies of our method under different conditions including breakpoints, occlusions, clutter, and noise. We also studied the effect of resolution of the transformation space and threshold τ_n on speed and accuracy. The threshold τ_d was empirically determined as $\tau_d = V_{cell}/300$, where V_{cell} is the 3-D volume of an interesting cell. Note that τ_d decreases as cell decomposition progresses. The run-time and accuracy were evaluated for different levels of distortion. In order to quantify the accuracy of the model transformation t_{opt} estimated by our algorithm, we define an error measure $E_{\text{m}} = (\Sigma_{p_m \in M_p} d_2(t_{\text{opt}}(p_m), I_p))/(\#(M_p))$; where p_m denotes a model edge point; M_p and I_p denote the set of all the edge points of the model and clean probe image, respectively; $d_2(t_{\text{opt}}(p_m), I_p)$ denotes the Euclidean distance between the transformed model point and its closest image point, and $\#(M_p)$ denotes the total number of model points. The average value of E_m when computed over the subset of images is denoted by E_{avg}.

We first studied the performance of our algorithm in the presence of breakpoints and under different levels of occlusions. On every line segment and elliptical arc of a clean probe image, a breakpoint was created within the feature at a random location. To simulate the effect of occlusions, a fraction of the number of pixels of the feature was removed around every breakpoint. To the resultant image, a fixed number of outliers was added. For each level of occlusion μ_{oc}, we varied the threshold τ_n and evaluated the run-time on each of the probe images. For every image, the model was located with error E_{avg} less than 0.45 pixels. The average run-time is plotted against the threshold value τ_n in Fig. 7(a). Different curves in the plot correspond to different occlusion fractions as indicated. The curve corresponding to occlusion level $\mu_{oc} = 0.01$ denotes the case of a single breakpoint. For each level of occlusion, we observe that the run-time decreases as the threshold τ_n is increased. For $\mu_{oc} = 0.01$, we observe that the run-time decreases by about 10% as τ_n is increased from 0.45 to 0.95. This is because a higher value of τ_n denotes a stricter condition, which ensures that



(a)



(b)

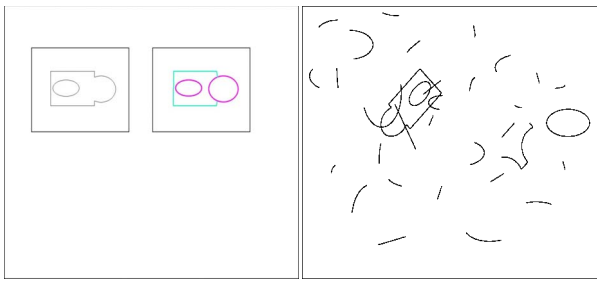
Fig. 7. (Color online) (a) Performance as a function of τ_n. (b) Speed–accuracy trade-off.

fewer cells are treated as interesting in cell-decomposition. However, the maximum value of τ_n that can be used for an image is dependent on the extent of occlusion it has undergone. For an occlusion level of μ_{oc}, the maximum value of τ_n is chosen as $\tau_n = 0.95 - \mu_{oc}$ after providing a 5% margin for error in edge detection. We also evaluated the performance for more than one breakpoint. Even after introducing three breakpoints per feature, it was found that the average run-time increased only marginally to 0.411 seconds (as compared to 0.404 seconds for one breakpoint per feature) for $\tau_n = 0.9$.

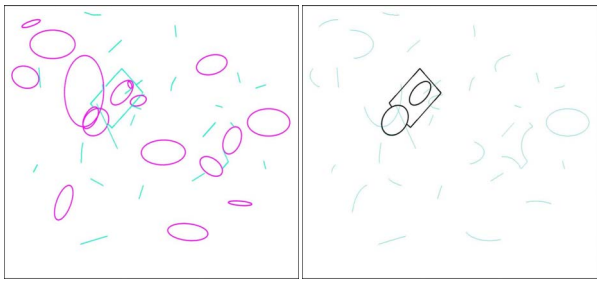
In the following experiment, we study the trade-off between speed and accuracy. Figure 7(b) shows plots of the average run-time (on the left y-axis) and the average error (on the right y-axis) for different resolutions $(\Delta x, \Delta y)$ for translation. The resolution of the angle-axis $(\Delta\alpha)$ was kept fixed at 0.5° because poor angular resolution throws up too many interesting cells. We observe that as the translation resolution is changed from 0.5 to 1.5 pixels, the average error correspondingly increases from 0.4 to 0.78 pixels. Since we adopt cell-decomposition, the reduction in run-time is very marginal.

As a representative example of the synthetic dataset, Fig. 8(a) shows a model consisting of 872 points and its feature representation using two elliptical arcs and five line segments. An image generated with this model is shown in Fig. 8(b). The 3467 points of the image were represented by 17 elliptical arcs and 30 line segments as shown in Fig. 8(c), where we observe that there are breaks in the model elliptical arcs due to intersection with other shapes. Despite this distortion, the proposed matching algorithm correctly located the model in the image as shown in Fig. 8(d).

We studied the performance of our method in the presence of noise by considering the effect of (i) image noise and (ii) noisy model parameters. We have already shown in an earlier experiment [Fig. 7(a)] that our model is robust to changes in model parameters due to breakpoints and occlusions. When independent additive white Gaussian (AWG) noise was added to the original gray-scale image, we found that our algorithm is quite robust and can tolerate image noise levels up to a standard deviation of 20. We also conducted experiments by introducing independent AWG noise to the estimated parameters of line segments and ellipses. The tolerance level for each parameter was found to be as follows: line endpoints (3.0), ellipse center (5.0), length of axes (5.0) and orientation (2.0), where the number in parentheses represents the maximum permissible standard deviation of noise for that parameter.



(a) (b)



(c) (d)

Fig. 8. (Color online) (a) Model and its features. (b) Probe image. (c) Image features. (d) Model correctly located in the image.

We ran the proposed method as well as the point-matching algorithm over all the 1550 images in our synthetic dataset in groups of ten images at a time. Based on the performance analysis plots of Fig. 7, we chose $\Delta_x=\Delta_y=0.5$ pixels, $\Delta_\alpha=0.5°$, and $\tau_n=0.95-\mu_{oc}$. The E_{avg} for the proposed algorithm over the entire dataset was found to be 0.41 pixels which is comparable to that of PMT. Figure 9(a) shows a comparison plot for the average number of cells examined per group of ten images. We observe that the number of cells examined by the proposed algorithm is significantly smaller than for the point-matching method. Figure 9(b) shows the corresponding average run-times. The run-time averaged over all the images was just 0.317 seconds for our algorithm, while it was 3.8 seconds for PMT. We also implemented our algorithm with line segments *alone* as features. With just line features, the curved edges had to be represented by several line segments and this approximation was generally not identical in the model and probe images. Consequently, the distance thresholds used in our algorithm had to be relaxed to arrive at the model transformation. The number of cells to be examined increased due to the relaxed thresholds. The average run-time was 1.5 seconds using only line features. This was mainly due to an increase in the number of features and number of cells examined. This value is slightly higher than the average run-time of 1.08 seconds reported for the line matching technique in [16].

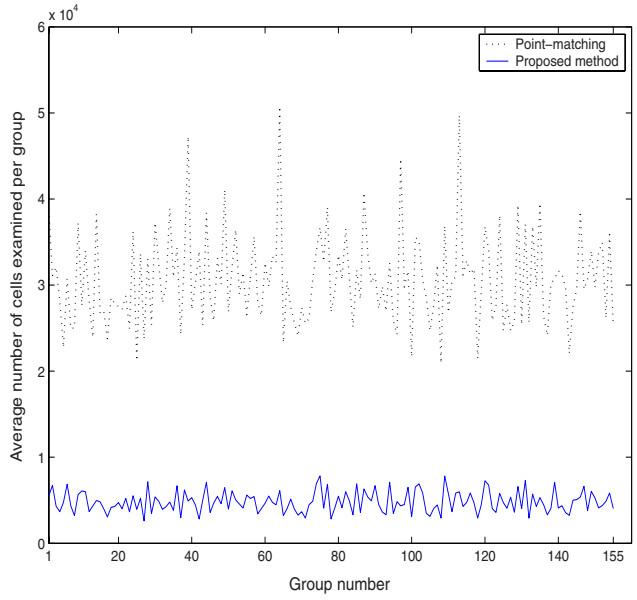


Fig. 9. (Color online) Performance comparison over 1550 images. (a) Number of cells evaluated. (b) CPU run-time per group for the proposed and the point-matching methods.

The performance of our method on all these images clearly demonstrates its effectiveness. The use of higher-order features results in a better description of images, as very few features are involved. The number of cells examined for matching is quite lower when compared with the standard PMT and the line matching scheme (our algorithm with just line features). Our method is about eight-to-ten times faster than the standard point-matching algorithm.

6. CONCLUSIONS

The main contribution of this paper was to propose a transformation space-based technique for matching images using line segments and elliptical arcs. Our approach provides a compact and rich description of images in terms of higher-order features. We formulated distance measures for comparing the features of the model and the image. Our method used cell-decomposition for efficiently searching the transformation space to find the optimal model transformation parameters. The algorithm was tested on several real and synthetic images. The run-time for the proposed technique is significantly lower than that of the standard point-matching algorithm.

APPENDIX A: DISTANCE MEASURES

In this appendix, we give closed-form expressions for distance measures d_e and d_{e_l} in Section 3. The variables described here are as defined earlier in Subsections 3.B and 3.C.

1. Ellipse-to-Ellipse Distance

The expression for d_e after solving the integral in Eq. (4) can be shown to be

$$d_e = \frac{I_{x_4} - 4h'I_{x_3} + 6h'^2 I_{x_2} - 4h'^3 I_{x_1}}{a_m^4} + \frac{I_{y_4} - 4k'I_{y_3}}{b_m^4} + \frac{6k'^2 I_{y_2} - 4k'^3 I_{y_1}}{b_m^4} - \frac{2(I_{x_2} - 2h'I_{x_1} + h'^2(\theta_2 - \theta_1))}{a_m^2} - \frac{2(I_{y_2} - 2k'I_{y_1} + k'^2(\theta_2 - \theta_1))}{b_m^2} + \frac{I_{x_2y_2} + k'^2 I_{x_2} - 2k'I_{x_2y_1}}{a_m^2 b_m^2} - \frac{2h'I_{x_1y_2} - 2h'k'^2 I_{x_1} + 4h'k'I_{x_1y_1} + h'^2 I_{y_2} - 2h'^2 k'I_{y_1}}{a_m^2 b_m^2}$$

$$+(\theta_2-\theta_1)\left(\frac{h'^4}{a_m^4}+\frac{k'^4}{b_m^4}+1+\frac{h'^2k'^2}{2a_m^2b_m^2}\right) \tag{A1}$$

The terms $I_{x_py_q}=\int_{\theta_1}^{\theta_2}x^p(\theta)y^q(\theta)d\theta$, $I_{x_p}=\int_{\theta_1}^{\theta_2}x^p(\theta)d\theta$, and $I_{y_q}=\int_{\theta_1}^{\theta_2}y^q(\theta)d\theta$. On expansion, we get

- $I_{x_4}=a_i^4c^4(\alpha_d)I_{c_4s_0}-4a_i^3c^3(\alpha_d)b_is(\alpha_d)I_{c_3s_1}+6a_i^2c^2(\alpha_d)b_i^2s^2(\alpha_d)I_{c_2s_2}-4b_i^3s^3(\alpha_d)a_ic(\alpha_d)I_{c_1s_3}+b_i^4s^4(\alpha_d)I_{c_0s_4}$, where $c(\alpha_d)$ and $s(\alpha_d)$ denote $\cos(\alpha_d)$ and $\sin(\alpha_d)$, respectively, and $I_{c_ps_q}=\int_{\theta_1}^{\theta_2}c^p(\theta)s^q(\theta)d\theta$, for $p,q=0,1,2,3,4$. For example, $I_{c_1s_3}=\int_{\theta_1}^{\theta_2}c(\theta)s^3(\theta)d\theta=(s^4(\theta_2)-s^4(\theta_1))/4$.
- I_{y_4} can be obtained from I_{x_4} by replacing $c(\alpha_d)$ and $s(\alpha_d)$ with $s(\alpha_d)$ and $-c(\alpha_d)$, respectively.
- $I_{x_3}=a_i^3c^3(\alpha_d)I_{c_3s_0}-3a_i^2c^2(\alpha_d)b_is(\alpha_d)I_{c_2s_1}+3b_i^2s^2(\alpha_d)a_ic(\alpha_d)I_{c_1s_2}-b_i^3s^3(\alpha_d)I_{c_0s_3}$.
- I_{y_3} is obtained from I_{x_3} by replacing $c(\alpha_d)$ and $s(\alpha_d)$ with $s(\alpha_d)$ and $-c(\alpha_d)$, respectively.
- $I_{x_2}=a_i^2c^2(\alpha_d)I_{c_2s_0}-2a_ic(\alpha_d)s(\alpha_d)I_{c_1s_1}+b_i^2s^2(\alpha_d)I_{c_0s_2}$.
- $I_{y_2}=a_i^2s^2(\alpha_d)I_{c_2s_0}+2a_is(\alpha_d)c(\alpha_d)I_{c_1s_1}+b_i^2c^2(\alpha_d)I_{c_0s_2}$.
- $I_{x_1}=a_ic(\alpha_d)I_{c_1s_0}-b_is(\alpha_d)I_{c_0s_1}$.
- $I_{y_1}=a_is(\alpha_d)I_{c_1s_0}+b_ic(\alpha_d)I_{c_0s_1}$.
- $I_{x_2y_2}=a_i^4c^2(\alpha_d)s^2(\alpha_d)I_{c_4s_0}+2b_ia_i^3c^3(\alpha_d)s(\alpha_d)I_{c_3s_1}+2a_ic(\alpha_d)b_i^3s^3(\alpha_d)I_{c_1s_3}+b_i^4c^2(\alpha_d)s^2(\alpha_d)I_{c_0s_4}-2b_ic(\alpha_d)a_i^3s^3(\alpha_d)I_{c_3s_1}-4a_i^2c^2(\alpha_d)b_i^2s^2(\alpha_d)I_{c_2s_2}-2a_is(\alpha_d)b_i^3c^3(\alpha_d)I_{c_1s_3}+a_i^2b_i^2I_{c_2s_2}(c^4(\alpha_d)+s^4(\alpha_d))$.
- $I_{x_1y_1}=a_i^2c(\alpha_d)s(\alpha_d)I_{c_2s_0}+a_ib_ic^2(\alpha_d)I_{c_1s_1}-a_ib_is^2(\alpha_d)I_{c_1s_1}-b_i^2c(\alpha_d)s(\alpha_d)I_{c_0s_2}$.
- $I_{x_1y_2}=a_i^3s^2(\alpha_d)c(\alpha_d)I_{c_3s_0}-a_i^2b_is^3(\alpha_d)I_{c_2s_1}+a_ib_i^2c^3(\alpha_d)I_{c_1s_2}-b_i^3c^2(\alpha_d)s(\alpha_d)I_{c_0s_3}+2a_i^2b_ic^2(\alpha_d)s(\alpha_d)I_{c_2s_1}-2b_i^2a_is^2(\alpha_d)c(\alpha_d)I_{c_1s_2}$.
- $I_{x_2y_1}$ can be obtained by replacing the terms $c(\alpha_d)$ and $s(\alpha_d)$ in $I_{x_1y_2}$ by $s(\alpha_d)$ and $-c(\alpha_d)$, respectively.

2. Ellipse-to-Line Distance

On solving the integral in Eqn. (6), the expression for d_{e_l} turns out to be

$$\begin{aligned}
d_{e_l}=&\frac{\{(ux_2+gs(\alpha_m)-h_r)^5-(ux_1+gs(\alpha_m)-h_r)^5\}}{5ua_m^4}\\
&+\frac{1}{5vb_m^4}\{(vx_2+gc(\alpha_m)-k_r)^5-(vx_1+gc(\alpha_m)-k_r)^5\}\\
&-\frac{2}{3ua_m^2}\{(ux_2+gs(\alpha_m)-h_r)^3-(ux_1+gs(\alpha_m)-h_r)^3\}\\
&-\frac{2}{3vb_m^2}\{(vx_2+gc(\alpha_m)-k_r)^3-(vx_1+gc(\alpha_m)-k_r)^3\}\\
&+\{x_2-x_1\}+\frac{1}{2a_m^2b_m^2}\left\{u^2v^2\left(\frac{x_2^5-x_1^5}{5}\right)+\left(\frac{x_2^4-x_1^4}{4}\right)\right.\\
&\times[(2u^2vc(\alpha_m)g+2uv^2gs(\alpha_m))-(2u^2k_rv+2h_ruv^2)]\\
&+[(u^2g^2c^2(\alpha_m))+(u^2k_r^2)-(2u^2k_rgc(\alpha_m))+(v^2g^2s^2(\alpha_m))\\
&+(h_r^2v^2)+(4uvg^2c(\alpha_m)s(\alpha_m))-(4uk_rvgs(\alpha_m))\\
&-(4uvh_rgc(\alpha_m))+(4h_ruk_rv)-(2h_rgv^2s(\alpha_m))]\\
&\times\left(\frac{x_2^3-x_1^3}{3}\right)+[(g^4s^2(\alpha_m)c^2(\alpha_m))+(k_r^2g^2s^2(\alpha_m))\\
&-(2k_rg^3s^2(\alpha_m)c(\alpha_m))+(h_r^2g^2c^2(\alpha_m))+(h_r^2k_r^2)\\
&-(2k_rgh_r^2c(\alpha_m))-(2h_rg^3c^2(\alpha_m)s(\alpha_m))-(2h_rgk_r^2s(\alpha_m))\\
&+(4k_rh_rg^2c(\alpha_m)s(\alpha_m))](x_2-x_1)+[(2vg^3s^2(\alpha_m)c(\alpha_m))\\
&-(2k_rvg^2s^2(\alpha_m))-(2k_rh_r^2v)+(2vh_r^2gc(\alpha_m))\\
&+(2ug^3c^2(\alpha_m)s(\alpha_m))+(2uk_r^2gs(\alpha_m))\\
&-(4k_rug^2c(\alpha_m)s(\alpha_m))-(4h_rvg^2c(\alpha_m)s(\alpha_m))\\
&-(2h_rug^2c^2(\alpha_m))-(2h_ruk_r^2)+(4h_ruk_rgc(\alpha_m))\\
&\left.+(4k_rvh_rgs(\alpha_m))]\left(\frac{x_2^2-x_1^2}{2}\right)\right\}.
\end{aligned} \tag{A2}$$

ACKNOWLEDGMENTS

We thank the reviewers for their useful comments and suggestions.

REFERENCES

1. W. J. Rucklidge, *Efficient Visual Recognition Using the Hausdorff Distance* (Lecture Notes in Computer Science, Springer-Verlag, 1996).
2. B. Zitova and J. Flusser, "Image registration methods: a survey," Image Vis. Comput. **21**, 977–1000 (2003).
3. M. T. Goodrich, J. S. B. Mitchell, and M. W. Orletsky, "Approximate geometric pattern matching under rigid motions," IEEE Trans. Pattern Anal. Mach. Intell. **21**, 371–379 (1999).
4. T. M. Breuel, "Geometric aspects of visual object recognition," Ph.D. dissertation (Massachusetts Institute of Technology, 1992).
5. J. R. Beveridge and E. M. Riseman, "How easy is matching 2D line models using local search?" IEEE Trans. Pattern Anal. Mach. Intell. **19**, 564–579 (1997).
6. W. E. L. Grimson, "On the recognition of curved objects," IEEE Trans. Pattern Anal. Mach. Intell. **11**, 632–643 (1989).

7. H. J. Wolfson, "On curve matching," IEEE Trans. Pattern Anal. Mach. Intell. **12**, 483–489 (1990).
8. P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," Int. J. Comput. Vis. **61**, 55–79 (2005).
9. H. S. Alhichri and M. Kamel, "Virtual circles: a new set of features for fast image registration," Pattern Recogn. Lett. **24**, 1181–1190 (2003).
10. N. Ayache and O. D. Faugeras, "HYPER: a new approach for the recognition and positioning of two-dimensional objects," IEEE Trans. Pattern Anal. Mach. Intell. **8**, 44–54 (1986).
11. J. L. Mundy and A. J. Heller, "The evolution and testing of a model-based object recognition system," in *Proceedings of IEEE Conference on Computer Vision* (IEEE, 1990) p. 268.
12. D. Huttenlocher, D. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," IEEE Trans. Pattern Anal. Mach. Intell. **15**, 850–863 (1995).
13. M. Hagedoorn and R. C. Veltkamp, "Reliable and efficient pattern matching using an affine invariant metric," Int. J. Comput. Vis. **31**, 203–225 (1999).
14. G. Stockman, "Object recognition and localization via pose clustering," Comput. Vis. Graph. Image Process. **40**, 361–387 (1987).
15. X. Yi and O. I. Camps, "Line-based recognition using a multidimensional Hausdorff distance," IEEE Trans. Pattern Anal. Mach. Intell. **21**, 901–916 (1999).
16. T. M. Breuel, "Implementation techniques for geometric branch-and-bound matching methods," Comput. Vis. Image Underst. **90**, 258–294 (2003).
17. C. Guerraa and V. Pascucci, "Line-based object recognition using Hausdorff distance: from range images to molecular secondary structures," Image Vis. Comput. **23**, 405–415 (2005).
18. L. K. Shark, A. A. Kurekin, and B. J. Matuszewski, "Development and evaluation of fast branch-and-bound algorithm for feature matching based on line segments," Pattern Recogn. **40**, 1432–1450 (2007).
19. R. Yang and Y. Gao, "Line-based affine invariant object location using transformation space decomposition," in *Proceedings of International Conference on Pattern Recognition* (IEEE, 2006) pp. 646–649.
20. T. M. Breuel, "On the use of interval arithmetic in geometric branch and bound algorithms," Pattern Recogn. Lett. **24**, 1375–1384 (2003).
21. W. Wan and J. A. Ventura, "Segmentation of planar curves into straight-line segments and elliptical arcs," Graph. Models Image Process. **59**, 484–494 (1997).
22. T. M. Breuel, "Fast recognition using adaptive subdivisions of transformation space," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 1992), p. 445.
23. C. Paramanand and A. N. Rajagopalan, "An efficient representation of digital curves with line segments and elliptical arcs," in *Proceedings of IET International Conference on Visual Information Engineering* (IET, 2006), pp. 470–475.
24. C. Paramanand and A. N. Rajagopalan, "Efficient geometric matching with higher-order features," in *Proceedings of IEEE Conference on Pattern Recognition* (IEEE, 2008) pp. 1–4.
25. T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing* (Prentice Hall, 2000).
26. A. W. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least squares fitting of ellipses," IEEE Trans. Pattern Anal. Mach. Intell. **21**, 476–480 (1999).
27. Z. Zhang, "Parameter estimation techniques: A tutorial with application to conic fitting," Image Vis. Comput. **15**, 59–76 (1997).
28. R. P. Millane, M. E. Fitzsimons, M. Qi, and A. Haider, "Analysis of gravel river beds using three-dimensional laser scanning," Proc. SPIE **63160**, 63160B.1–63160B.10 (2006).
29. D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2007), p. 1.