PERGAMON

Computers & Graphics 25 (2001) 211–222

COMPUTERS
&GRAPHICS

www.elsevier.com/locate/cag

Technical Section

Shape matching and recognition using a physically based object model

Jen-Hui Chuang*, Jin-Fa Sheu, Chien-Chou Lin, Hui-Kuo Yang

Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, 30010, Republic of China

Abstract

We present a novel approach of shape matching and recognition of 3D objects using artificial potential fields. The potential model assumes that boundary of every 3D template object of identical volume is uniformly charged. An initially small input object placed inside a template object will experience repulsive force and torque arising from the potential field. A better match in shape between the template object and the input object can be obtained if the input object translates and reorients itself to reduce the potential while growing in size. The template object which allows the maximum growth of the input object corresponds to the best match and thus represents the shape of the input object. The above repulsive force and torque are analytically tractable for an input object represented by its boundary samples, which makes the shape matching efficient. The proposed approach is intrinsically invariant under translation, rotation and size changes of the input object. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Shape orientation; Shape matching; Object recognition; Artificial potential field

1. Introduction

One of the major problems in computer vision is object recognition. Many existing algorithms simplify the problem by reducing it to a shape matching and recognition problem. In this paper, we propose a method for matching and recognition of 3D objects using a potential field model.

1.1. Shape matching and recognition of 2D objects

Although the recognition of 3D objects is of primary interest in computer vision, many 2D shape matching algorithms are developed for situations which can be regarded as two dimensional. Template matching methods are presented in [1–3] for object recognition. Fourier descriptors [4,5] transform the coordinates of boundary points into a set of complex numbers for the matching. Moments of 1D functions representing segments of shape contours are used in the matching in [6–9]. Stochastic models, e.g., autoregressive models, are used for shape classification in [10–12]. In [13–15], the boundary of a region is represented by a sequence of numbers and the shape matching is accomplished by string matching. Other shape-matching methods involve finding the polar transform of the shape sample [16] or calculating the distances of the feature points from the centroid [17], etc. In [18], the shape matching is accomplished by graph matching for multilevel structural descriptions of shape samples. Multiple 1D matching processes for multiscale curvature descriptions are adopted in [19] in building shape models. Usually, the scaling of the object, the viewscale of the shape contour, and the rotation and the translation of the object need to be determined before a final matching process can take place. The matching process may involve the matching of binary images, discrete 1D data, or extracted shape features arranged into structured data.

In [20], a potential-based approach for 2D shape matching is proposed. The matching process involves the minimization of a scalar function, the potential. The

* Corresponding author. Tel.: + 886-3-573-1979; fax: + 886-3-572-1490.

E-mail address: jchuang@cis.nctu.edu.tw (J.-H. Chuang).

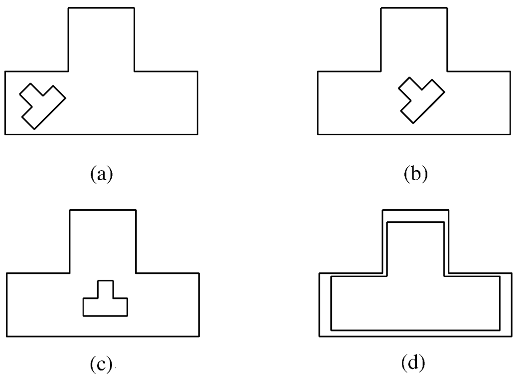

Fig. 1. The basic shape-matching procedure: (a) place a (size-reduced) template inside the input region, (b) translate and (c) rotate the template to reduce the potential, then (d) increase the size of the template.

potential model assumes that the border of any 2D region is uniformly charged. If a shape template is small in size and can be placed inside a region whose shape is to be determined, the template will experience repulsive force and torque arising from the potential field. The basic idea of the approach is to achieve a better match in shape between the template and the given region by translating and reorienting the template along the above force and torque directions, respectively, toward the configuration of the minimum potential. The template is then expanded and its configuration readjusted until the template almost touches the border of the given region (see Fig. 1). For a selected group of shape templates, the template with the largest final size is considered as the best match. Such an approach is intrinsically invariant under translation, rotation and size changes of the shape sample.

1.2. 3D object matching and recognition

The recognition of 3D objects is one of the most challenging problems in computer vision. The recognition is much harder than the 2D case because of the complexity involved with the extra degree of freedom. Instead of extending the concepts adopted in the 2D approaches presented in Section 1.1 to three dimensions, most of 3D recognition works are based on the analysis of 2D scenes.

Several recognition systems extract features from 2D images and match them to corresponding features in a database of 3D object models. The approaches turn the recognition process into a procedure of verifying each candidate hypothesis and finally ranking the verified hypotheses. In [21], some invariant features are extracted from triplet of surfaces and then constructed into an invariant feature indexing (IFI) of interpretation table. In the verification process, the extracted features are used to prune the hypotheses which are incorrect. A method based on the rotational symmetry is proposed in [22] to reduce the number of hypotheses. Further improvement is proposed in [23] based on a multiview and multigroup approach. In the BONSAI system proposed in [24], a constrained search mechanism is used. The features extracted from the CAD model are represented by an interpretation tree (IT) and the search space is pruned by pose estimation.

Other approaches of recognition of 3D objects from 2D images use different image features. In [25], triangle pairs and quadrilateral pairs are used as features. In [26], edges of a 3D object are described by second-order equations and a transformation from a feature of a 3D object model to a 2D image feature is established. With such a transformation, proper alignment between a transformed 3D model and a possible image of the model can be obtained. In [27], the edges of a polyhedral 3D model are described by a weighted graph and the matching process is carried out using the Laplacian matrix. In [28], probabilistic indexing is used to determine whether there is a correspondence between projections of a 3D model and a 2D image. In [29], Fourier transform of the boundary description of a 3D model is first obtained, which is then used in the matching through clustering.

Several physically based computer vision methods have been proposed for the representation and recognition of 3D objects using the finite element method. In [30], in order to define virtual forces which deform an object to fit a set of data points, springs are attached to pairs of corresponding object and data points. The shape matching is achieved by identifying the equilibrium displacement of the springs. In [31], a dynamic balloon model represented by using a triangular mesh is driven by an outward inflation force, while the vertices in the mesh are linked to neighboring vertices through springs to simulate the surface tension and to keep the shell smooth, until the mesh elements reach the object surface. The system includes an adaptive local triangle mesh subdivision scheme that results in an evenly distributed mesh. In [32], a 3D object is regarded as an isolated conductor. By assuming a piecewise constant charge density for each triangle of a triangular mesh, the segmentation is achieved by identifying surface concavities by tracing local charge density minima.

1.3. The proposed approach

In this paper, the potential-based shape-matching approach presented in [20] is extended to three dimensions using a generalized potential model presented in [33]. The goal is to develop a potential-based 3D shape-matching algorithm which, as its 2D counterpart, can correctly and efficiently perform the matching without knowing the exact information about the location, orientation and size of the input object. Furthermore, the proposed shape-matching scheme is not based on any hypothesis of feature correspondence. Therefore, feature extraction of an input object which is required for a structured object representation is not needed.

In Section 2, the generalized potential model is briefly described. According to such a model, the repulsion between two 3D objects, one with polyhedral description and the other represented by point samples on its surface, can be evaluated analytically. In Section 3, a potential-based shape-matching algorithm using these analytic results is developed. Some computer implementation details of the matching algorithm are also presented. Simulation results for the shape matching of some 3D objects are presented in Section 4. Section 5 presents some concluding remarks.

2. Generalized potential fields in the 3D space

In [33], it is shown that the Newtonian potential, being harmonic in the 3D space, cannot prevent a charged object point from running into another object whose surface is uniformly charged. This is because the value of such a potential function is finite at the continuously charged surface. Subsequently, generalized potential models are proposed to assure collision avoidance between 3D objects. The potential function is inversely proportional to the distance between two point charges to the power of an integer and, as reviewed next, the potential and thus its gradient due to polyhedral surfaces can be calculated analytically. The shape-matching approach proposed in this paper will use these results to evaluate the repulsion between template and input objects.

Consider a planar surface S in the 3D space as shown in Fig. 2; the direction of its boundary, ΔS, is determined with respect to its surface normal, $\hat{\mathbf{n}}$, by the right-hand rule, $\hat{\mathbf{u}} \times \hat{\mathbf{l}} = \hat{\mathbf{n}}$, where $\hat{\mathbf{u}}$ and $\hat{\mathbf{l}}$ are along the (outward) normal and tangential directions of ΔS, respectively. For the generalized potential function, the potential value at $\mathbf{r}$ is defined as

$$\int_S \frac{\mathrm{d}S}{R^m}, \quad m \geqslant 2, \tag{1}$$

where $R = |\mathbf{r}' - \mathbf{r}|$, $\mathbf{r}' \in S$, and integer m is the *order* of the potential function. The basic procedure to evaluate the potential at $\mathbf{r}$ is similar to that outlined in [34] for the evaluation of the Newtonian potential ($m = 1$) and can be summarized as follows:

(i) Write the integrand of the potential integral over S as surface divergence of some vector function.
(ii) Transform the integral into the one over ΔS based on the surface divergence theorem.
(iii) Evaluate the integral as the sum of line integrals over edges of ΔS.

Related geometric quantities associated with an edge C_i of S in the plane containing S, Q, are shown in Fig. 3 for $\mathbf{r}' \in C_i$. Without the loss of generality, it is assumed that

$$d \overset{\Delta}{=} \hat{\mathbf{n}} \cdot (\mathbf{r} - \mathbf{r}') > 0 \tag{2}$$

which is equal to the distance from $\mathbf{r}$ to Q.

For (i), we have (see [33])

$$\frac{1}{R^m} = \nabla_S \cdot (f_m(R)\mathbf{P}), \tag{3}$$

where $\mathbf{P}$ is the position vector of $\mathbf{r}'$ with respect to the projection of $\mathbf{r}$ on Q, $\mathbf{r}_Q$, and

$$f_m(R) = \begin{cases} \dfrac{\log R}{R^2 - d^2}, & m = 2, \\ \dfrac{-1}{(m-2)R^{m-2}(R^2 - d^2)}, & m \neq 2. \end{cases} \tag{4}$$

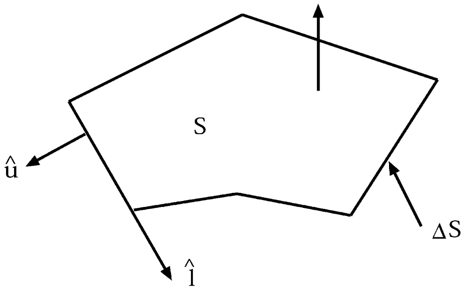

Fig. 2. A polygonal surface S in the 3D space.

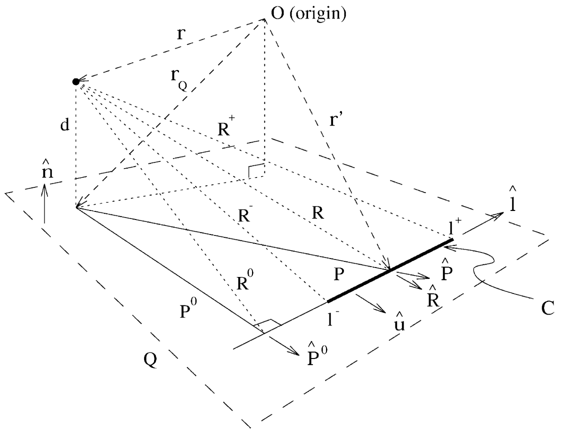

Fig. 3. Geometric quantities associated with a point, an edge C_i (subscript i is omitted) of S shown in Fig. 2 and the plane Q containing S.

Note that if $\mathbf{r}_Q$ is inside $S, f_m(R)$ will become singular for some $\mathbf{r}'' = \mathbf{r}_Q$, i.e., $R = d$. Let S_ε denote the intersection of S and a small circular region on Q of radius ε and centered at $\mathbf{r}_Q$, the potential due to S can be evaluated as

$$\begin{aligned}\int_S \frac{\mathrm{d}S}{R^m} &= \lim_{\varepsilon\to 0}\left[\int_{S-S_\varepsilon} \nabla_s \cdot (f_m(R)\mathbf{P})\,\mathrm{d}S + \int_{S_\varepsilon}\frac{\mathrm{d}S}{R^m}\right] \\ &= \int_{\Delta S} f_m(l)\mathbf{P}\cdot\hat{\mathbf{u}}\,\mathrm{d}l + \lim_{\varepsilon\to 0}\left[\int_0^\alpha\int_0^\varepsilon \frac{p\,\mathrm{d}p\,\mathrm{d}\theta}{(p^2+d^2)^{m/2}}\right] \\ &= \sum_i \mathbf{P}_\mathbf{i}^0\cdot\hat{\mathbf{u}}_\mathbf{i}\int_{C_i} f_{m,i}(l_i)\,\mathrm{d}l_i + g_m(\alpha), \end{aligned} \tag{5}$$

where

$$f_{m,i}(l_i) = f_m\left(R = \sqrt{l_i^2 + d^2 + (P_i^0)^2}\right), \tag{6}$$

$$g_m(\alpha) = \begin{cases} \alpha \log d, & m = 2, \\ \dfrac{\alpha}{(m-2)d^{m-2}}, & m > 2, \end{cases} \tag{7}$$

P_i^0 is the distance between $\mathbf{r}_Q$ and C_i, l_i is measured from the projection of $\mathbf{r}$ on C_i along the direction of $\hat{\mathbf{l}}_i$, and α is the angular extent of the circumference of S_ε lying inside S as $\varepsilon \to 0$. For example, $\alpha = 2\pi$ if $\mathbf{r}_Q$ is inside S, $\alpha = \pi$ if $\mathbf{r}_Q$ is on an edge of S and α is equal to the angle between two edges if $\mathbf{r}_Q$ is a vertex of S where the two edges are connected. (For simplicity, the subscript i is omitted whenever it is appropriate.)

Since $f_m(l)$ is a rational function for even m's when $m \neq 2$ and is rationalizable for odd m's (see [35]), the line integrals can always be evaluated in closed form except for $m = 2$. For example, if $P^0 \neq 0$, then we have[1]

$$\begin{aligned}\int_C f_3(l)\,\mathrm{d}l &= \int_{l^-}^{l^+} f_3(l)\,\mathrm{d}l \\ &= \frac{1}{P^0 d}\left[\tan^{-1}\frac{l^- d}{P^0 R^-} - \tan^{-1}\frac{l^+ d}{P^0 R^+}\right]. \end{aligned} \tag{8}$$

with R^- and R^+ equal to the distances from $\mathbf{r}$ to the two end points of C, respectively. Thus, the repulsive force exerted on a point charge due to S can be found analytically by evaluating the gradient of the following function[2]

$$\Phi(x, y, z) = \frac{1}{z}\tan^{-1}\frac{xz}{y\sqrt{x^2+y^2+z^2}}, \tag{9}$$

at some (x, y, z)'s.

For the potential-based 3D shape matching, the evaluation of the repulsion between two 3D objects involves the calculation of the repulsion between pairs of polygons; each pair has a polygon from the template object and the other from the input object. For continuously charged object surfaces, it is obvious that the direct calculation of the potential between two polygons requires a quadruple integral. To simplify the mathematics, the input object is approximately represented by a set of point samples on its surface in this paper. The repulsion between two 3D objects, in forms of repulsive force and torque, can then be estimated in closed form through superposition using the above analytical expressions. The repulsion will be used in the shape-matching process discussed next.

3. The potential-based shape matching

In this paper, a potential-based approach to 3D shape matching is proposed. The matching process is a direct extension of the 2D approach presented in [20]. While the matching procedure for 2D shapes starts with a shape template placed inside an input object, the 3D matching considered in this paper is performed by first placing input data (which may be obtained from a partial object surface) inside a shape template.

3.1. The shape-matching process

For the shape-matching process, the following operations are performed with an input object placed inside a template object (see Fig. 4). Since the potential function, and its gradient, will increase indefinitely (e.g., $g_m(\alpha)$ in Eq. (7) will diverge as $d \to 0$ if $\alpha \neq 0$) as a point of the input object approaches the boundary of the template object, the input object will be confined inside the template object throughout the matching process if sufficient samples are taken from the surface of the former.

Step 1: Potential minimization through translations

(a) Compute the total force between the template object and the input object. Find the minimum potential position of the latter along the above force direction.

(b) Repeat (a) until two consecutive executions of (a) result in negligible difference in the two minimum potential positions found.

Step 2: Potential minimization through rotations

(a) With the centroid of the input object chosen as the rotation center, compute the total repulsive torque between the template object and the input object. Find the angular position of minimum potential for the input object by rotating it with the rotation axis aligned with the above torque direction.

(b) Repeat (a) until an execution of it results in negligible difference in the angular position of the input object.

[1] Detailed discussions concerning some special cases for the potential calculation, e.g., for $P^0 = 0$ or $d = 0$, etc., as well as the associated collision avoidance property can be found in [33].

[2] In this paper, only $m = 3$ is considered. The increase of the value of m will increase the contribution to the potential due to nearest boundary points of an object. In any case, the matching should perform equally well since no collision between the input object and the template is allowed.

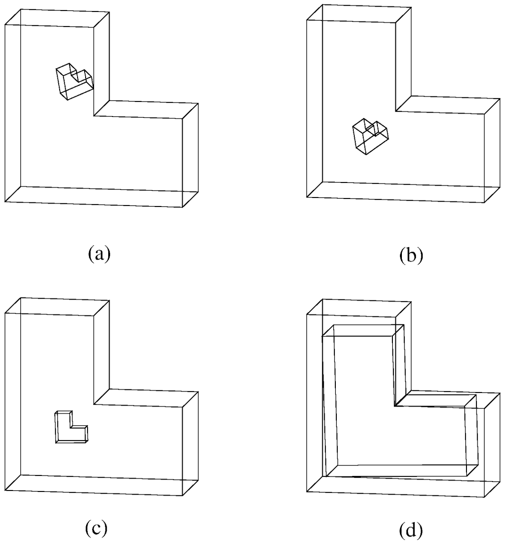

Fig. 4. The basic shape-matching procedure: (a) place a (size-reduced) input object inside a template object, (b) translate and (c) rotate the input object to reduce the potential, then (d) increase the size of the input object.

Step 3: Scaling

With the centroid of the input object fixed in space, and with the constraint that the object remains inside the template object, find the maximum size of the input object.

Step 4: End the matching process if the execution of Step 3 results in negligible size change of the input object; otherwise, go to Step 1.

3.2. Implementation of the algorithm

3.2.1. Implementation of Step 1

Since the input object is located inside the template object and a contact between them will result in an infinite potential value, the minimum potential position always exists along the initial force direction found in Step 1(a). (Situations involving multiple local minima in potential will be discussed later in this section.) For the computer implementation, the minimal potential position is identified efficiently by performing a gradient-based binary search using the projection of the repulsive force along the initial force direction. The precision in specifying the location of the input object is chosen to be 0.1% of the length of the template object.[3]

Although a minimal potential position along the initial force direction can be found with an execution of Step 1(a), it is possible to reduce the potential further by searching for minimum potential configuration of the input object in a different direction (ideally orthogonal to the previous search direction) in the 3D space. In Step 1(b) this is done by recalculating the total repulsive force then repeating Step 1(a). The search process ends when the object translation due to two consecutive executions of Step 1(a) is negligible, i.e., less than 0.1% of the length of the template object.

3.2.2. Implementation of Steps 2 and 3

While the repulsive force is used to translate the input object to a position of minimum potential in Step 1, if the input object is allowed to rotate, it is possible to reduce the potential further. (The choice of the object centroid as the rotation center is mainly for the convenience of the computer implementation and will have little effect on the shape matching results.) Therefore, the minimum potential object orientation is determined in Step 2 of the proposed shape-matching algorithm. Because the degree of freedom of object rotation in searching for such an orientation is comparable to that discussed in the previous section for object translation, the minimization procedure is similar to that of Step 1. The precision in specifying the 1D potential minimum in Step 2(a) is chosen to be within 0.5° of the object angular position.

The process of maximizing the size of the input object in Step 3 is similar to the processes discussed above for finding the object configuration of minimum potential. The center of expansion of the object is conveniently chosen as its centroid. The associated binary search is carried out with a finite number of iterations determined by the closest distance allowed between the input object and the template object. For the simulation results presented in this paper, the closest distance is chosen to be 0.1% of the length of the template object.

3.2.3. Initial object configurations

For the process of finding the minimum potential configuration of an input object, the result corresponds to a local minimum of the potential function. In general, depending on the initial object configuration, such a result may not correspond to the global minimum. For example, consider the two rectangles shown in Fig. 5. As the inner rectangle rotates with respect to its centroid, the potential will have local minima at four different angular positions with two of them do not correspond to the global minimum in the potential value. In order to resolve such a problem, multiple initial object configurations are tried out for each of the shape matching problems considered in this paper.

For the implementation of the proposed approach, it is assumed that the input object is initially placed near the centroid of a template object. Subsequently, the object is

[3] The precision in specifying the object position and orientation are arbitrarily chosen such that the configuration change due to further minimization of the potential is not noticeable.

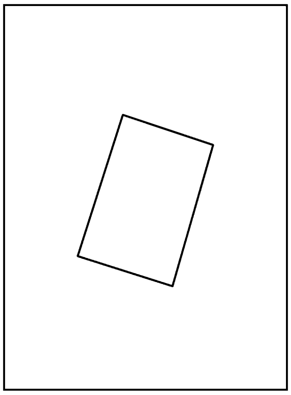

Fig. 5. With its centroid chosen as the rotation center, the inner rectangular will experience local minima in the potential value at four angular positions.

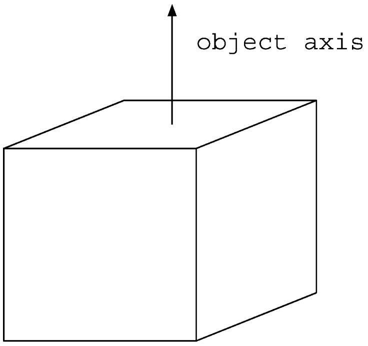

Fig. 6. The arbitrarily chosen object axis is passing through the object centroid and is parallel to the z-axis.

rotated into different orientations each used in a separate shape matching process. For simplicity, the centroid of the input object is chosen as the rotation center and the line containing the centroid and parallel to the z-axis is chosen as the axis of the object (see Fig. 6). With the axis initially pointed to vertex A of the tetrahedron shown in Fig. 7, the object is rotated such that the axis is pointed to three other vertices B, C, and D. For each of the four axis directions, the object is rotated by 0, 120, and 240° with respect to the object axis, respectively, to generate three initial object orientations. Therefore, there are 12 different initial orientations totally. Initial object configurations thus generated seems to work well for the shape-matching problems considered in this paper,[4] as presented next.

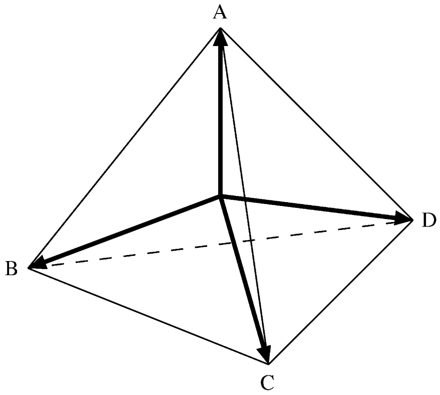

Fig. 7. Four directions of object axis used to generate 12 initial object orientations.

4. Simulation results

We now present some experimental results. The potential-based shape-matching system was tested using databases of 3D object models obtained from:

- Washington State University, NETLIB scientific data repository.
- University of South Florida's vision research group.
- Michigan State University's PRIP Lab.

The object models comprise not only simple polyhedra but also curved objects and objects with complex geometry. Fig. 8 shows images of some of the object models. A text file which contains lists of vertices, polygons, and surfaces of the associated polyhedral representation can be obtained for each object model.

In the following experiments, shape templates are represented by complete polyhedral surfaces given in the databases. Simulation will first be performed for "ideal" input objects, i.e., point samples of an input object are obtained by directly using the above list of vertices. Non-ideal cases will then be considered which include missing vertices due to a partial view of an object and the perturbation in vertex locations. Finally, simulation will be performed to show the effect of sampling.

4.1. "Ideal" case

In this section, we first choose nine simple polyhedra from the databases as shape templates and pick four of them as input objects. Figs. 9(a)–(d) show the shape-matching results for the four input objects, respectively. In each of the figures, the matching results shown for each template object include (1) one of the initial input object configurations, (2) the final (size maximized) input object configuration and (3) the template object whose configuration remains unchanged throughout the

[4] To improve the efficiency of the proposed approach, it will be more desirable if less initial configurations can be used while the matching result will not be affected. Such an important issue is currently under investigation.

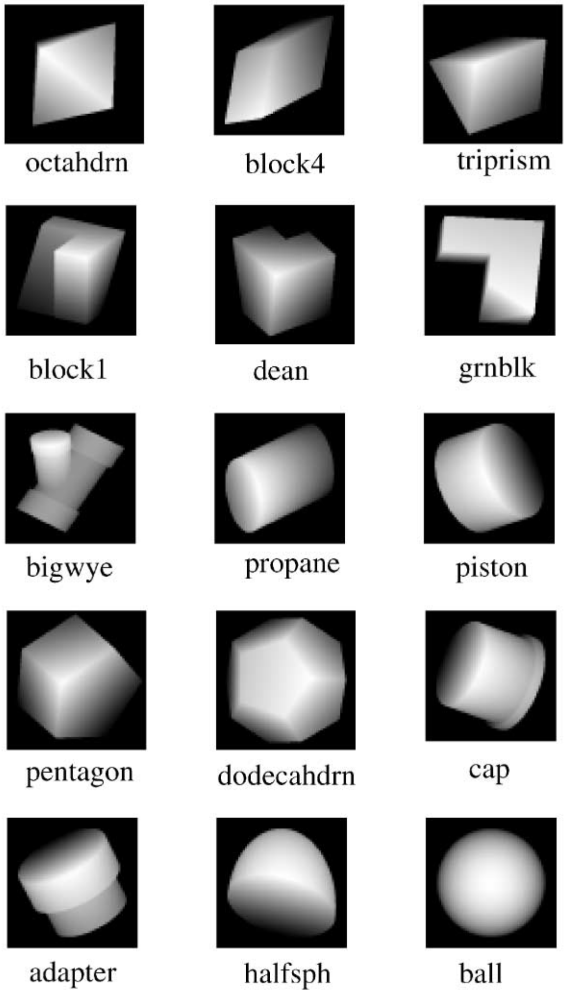

Fig. 8. Images of some object models.

matching process. The best match which correctly identifies a shape template for one of the input objects is marked with a "V". The CPU times spent on a Sun4u Sparc Ultra-1 workstation for the best matches shown in Figs. 9(a)–(d) are equal to 7.6, 32.0, 38.2, and 34.7 s, respectively. (For these relatively simple shape templates and input data, an execution of the matching process presented in Section 3.2 takes less than 45 s.) Note that for each pair of shape template and input object, the time spent in the matching is determined not only by the number of primitives representing the two but also by their shapes. For example, while more primitives are used in obtaining the best match in Fig. 9(b) than that in Fig. 9(c), the former in fact takes less time to calculate.

The proposed shape-matching method is also tested for more complex situations involving curved objects. Fig. 10 shows the matching results for *propane* and *piston*, respectively. Because of the similarity in their shapes, the same number of configuration and size adjustments described in Section 3.2 are carried out in deriving the best matches. The CPU time spent in calculating the two best matches are equal to 30.1 and 513.5 s, respectively. The ratio of them $(513.5/30.1 = 17.1)$ is very close to the ratio of the products of the number of the vertices and the number of polygons of the two objects $(\frac{386\times 384}{128\times 66} = 17.5)$. To further demonstrate the dependency of the computation time on the amount of input data, Fig. 11 shows the CPU time spent in the shape matching if different numbers of vertices of *propane* are used. It is readily observable that the former is a linear function of the latter.

In general, the input data may not be acquired with perfect conditions. For example, the data may be obtained with some noise contaminations, or part of the data may not be available due to a partial view of the object. Simulations of some of these non-ideal cases are given next.

4.2. Shape-matching results for some non-ideal cases

4.2.1. Shape-matching based on a partial view

To simulate the shape matching for input objects with incomplete surface descriptions, e.g., due to a partial view, it is assumed that only the vertices of a portion of the polygons of a shape model are available as input data. Fig. 12 shows partial object surfaces of some shape models and the corresponding shape-matching results. Similar results for objects with more complex boundary descriptions are shown in Fig. 13. Table 1 shows the computation times as well as the amounts of different boundary elements,[5] and the percentages of these numbers with respect to those of the complete object boundary, of the input objects shown in Figs. 12 and 13.

For each of the input objects which has been considered earlier in the previous section for shape-

Table 1
Boundary elements and CPU times for examples shown in Figs. 12 & 13

	Vertices/polygons (%)	CPU time (s)
Dean	8/3(73/36)	2.66
Dodecahdrn	16/6(80/50)	2.38
M-104	17/11(77/55)	5.24
M-110	20/9(71/41)	4.91
M-112	47/26(72/62)	29.88
Propane	97/33(76/50)	26.62
Piston	259/192(67/50)	381.76

[5] Although vertices, not polygons, of an input object are used in shape matching, the percentage of the latter will be a good estimate of the former for an evenly and densely sampled object surface.

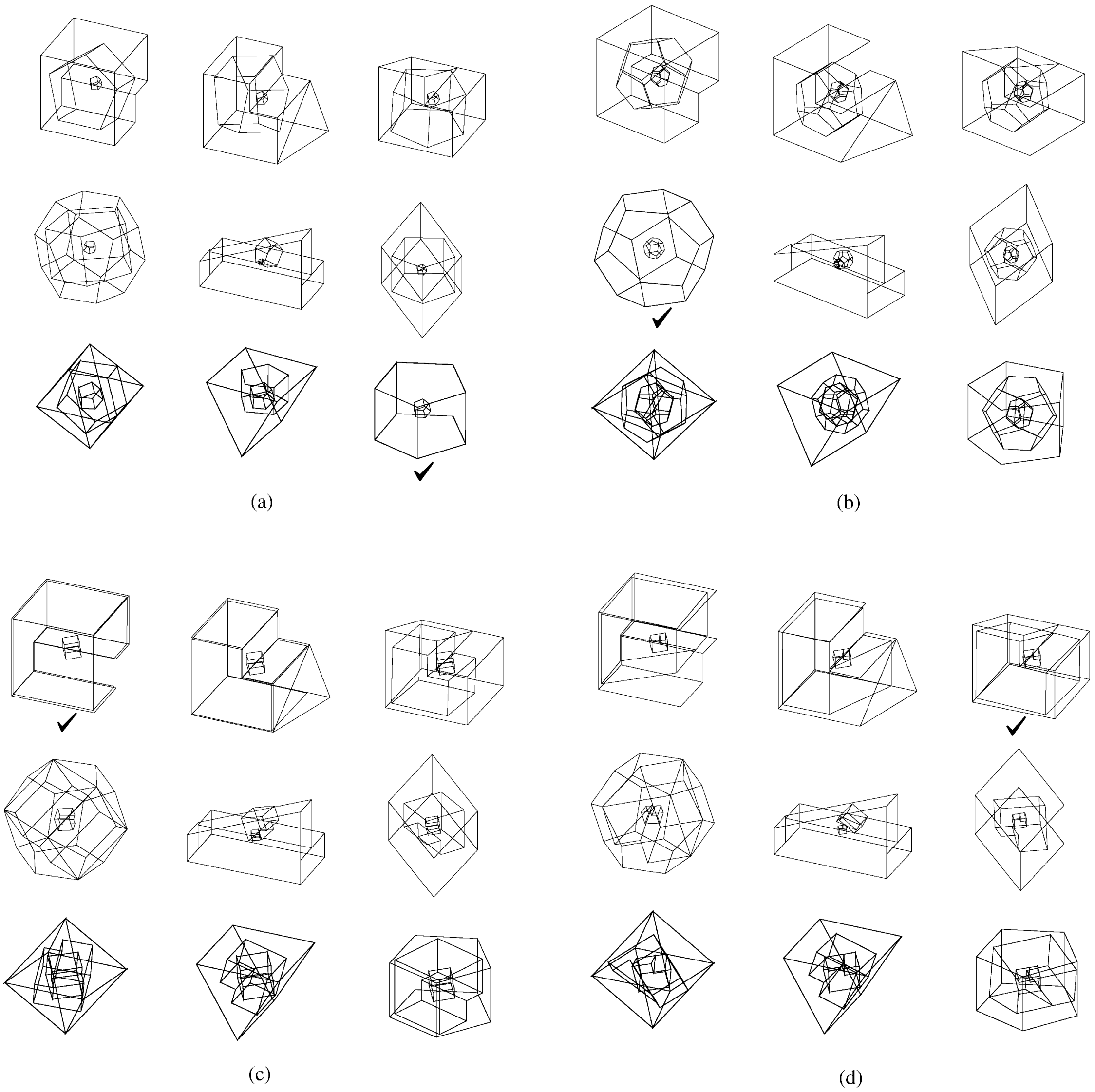

Fig. 9. Shape-matching results obtained for four input objects.

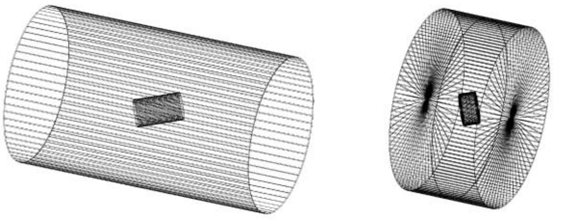

Fig. 10. Shape-matching results for *propane* (left) and *piston* (right).

matching, there is a reduction in the CPU time in this case due to a reduction of the amount of boundary elements (vertices) used. However, the former is not proportional to the latter in general since the effective shape of the input object is changed.

4.2.2. Input data with noise contamination

To simulate the shape matching for input data with data acquisition errors, noises are added to the locations of point samples of the input object. To just demonstrate the basic idea, the noise condition is simplified by considering only the perturbation in one dimension. For each of the two cylindrical objects shown in Fig. 10, perturbations are introduced by adding noises uniformly distributed between ± 5 and ± 10%, respectively, of the coordinates of object vertices measured from its centroid along the direction of the cylinder's axis. Fig. 14 shows

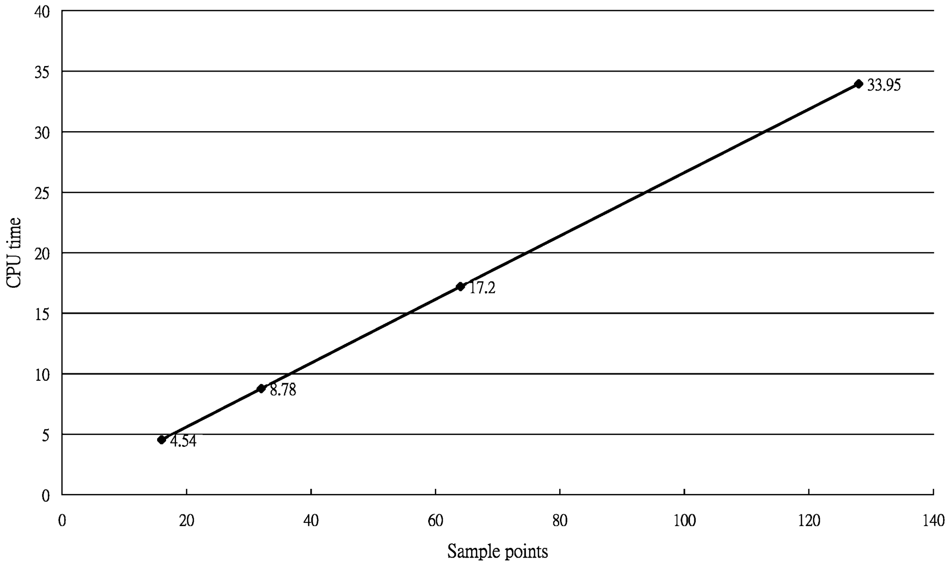

Fig. 11. The CPU time spent in the shape-matching for *propane* as a function of the amount of input data.

the shape-matching results with the above additive noises. While these results are consistent to those obtained for the noise-free case, the noises result in a small drop in the final size of the input object.

4.2.3. *Effects of sampling*

Intuitively, in order to properly represent an object with point samples on its surface, it would be desirable that these samples are dense enough and have a near-uniform distribution. Nevertheless, for the shape-matching examples considered in the previous sections, the vertices of the corresponding shape models, or a subset of such vertices, are used directly as simulated input data (point samples) of an object. Despite the satisfactory shape-matching results presented so far, there are certain occasions when using such data will cause a problem.

Fig. 15(a) shows an unsuccessful shape-matching example for a cap-shaped object whose boundary consists of two cylindrical surfaces of different cross sections, the rim connecting them, and two circular bases. While there are 128 vertices along each circumference of the two circular bases, there are 383 vertices located at the rim. Due to such a highly uneven distribution of point samples, the matching process yields a final, minimal potential, configuration of the input object which moves the above (heavily sampled) rim near the central region of the shape template, and in turn pushes one of the bases against the template's boundary.[6]

To examine the effect of sampling further on an empirical basis, the following simple adjustment to the above problem is considered: only a subset of the vertices, i.e., the 256 vertices on the circumferences of the two bases, are used as input data. Fig. 15(b) shows the shape-matching result thus obtained. One can see easily that the aforementioned problem is not presented here.

4.2.4. *Discussion*

In the previous sections, the proposed shape-matching method is shown to work for certain non-ideal cases. However, it is easy to see that, like other shape-matching approaches, such an approach will fail to work at some point as the situation gets worse. For example, depending on the viewing angle or the severity of an occlusion, input data obtained from partial object surfaces may not convey enough information regarding the object shape that the shape-matching process will generate an erroneous result. Similar arguments also apply to the noise condition.

On the other hand, it is possible for the proposed approach to use raw data of an input object, in the form of object points, without first establishing a structural relationship among the data points if the points have a dense or sparse, fairly uniform distribution and cover enough shape features of the object. For input data not possessing such desirable attributes, the processes of (i) fitting (polyhedral) surfaces to the raw data, and (ii) obtaining near uniformly distributed point samples from these surfaces may be necessary before desirable object samples can be obtained.[7] Such issues

[6] While the vertices are uniformly distributed along each of the above circumferences, this is not true for the 383 vertices at the rim. Thus, a tilt can be observed from the shape-matching result.

[7] Ideally, the process should perform a "coarse feature extraction" to get "meaningful" samples.

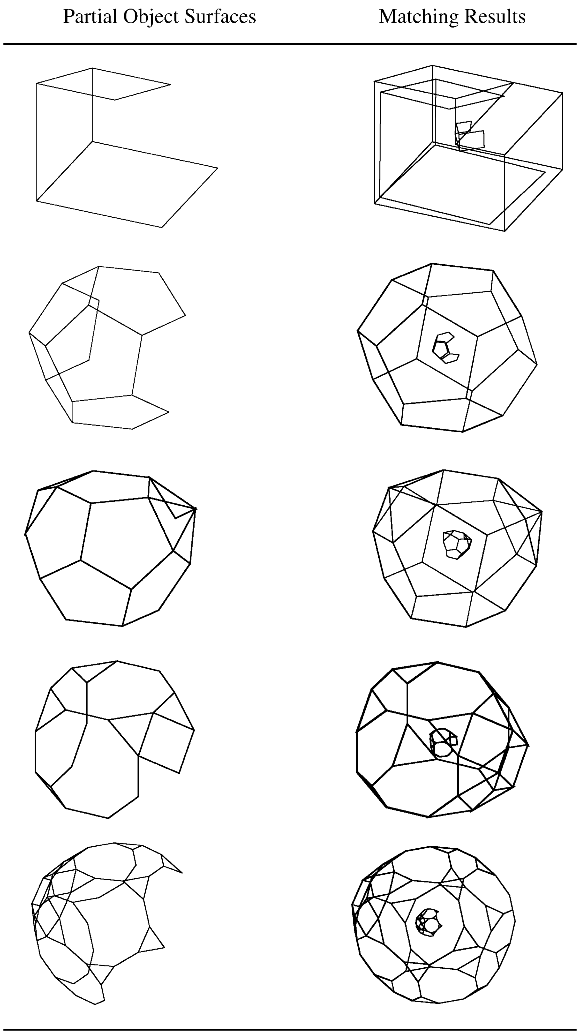

Fig. 12. Matching results using partial views of, from top to bottom, *dean*, *dodecahdrn*, *M*-104, *M*-110, and *M*-112.

have been considered by several researchers, e.g., those in [31].

Fig. 16(a) shows a 240 × 240 range image of *adapter*. Since very dense (about 40,000) point samples are obtained for the visible surfaces of the object, the above (i) and (ii) are not performed. Instead, in order to reduce the computation complexity, the set of samples are downsized by randomly selecting about 300 of them. Figs. 16(b) and (c) illustrate the top and side views of the shape-matching result, respectively, using these subset of samples directly. Fig. 16(d) shows the matching result using the range image of the object.

There are certain types of objects for which the proposed isotropic scaling scheme will not perform perfectly. For example, for *bigwye* shown in Fig. 8, an input object

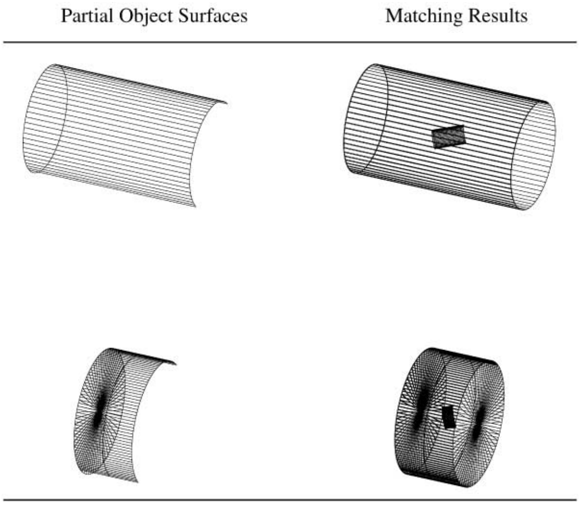

Fig. 13. Matching results using partial views of *propane* and *piston*.

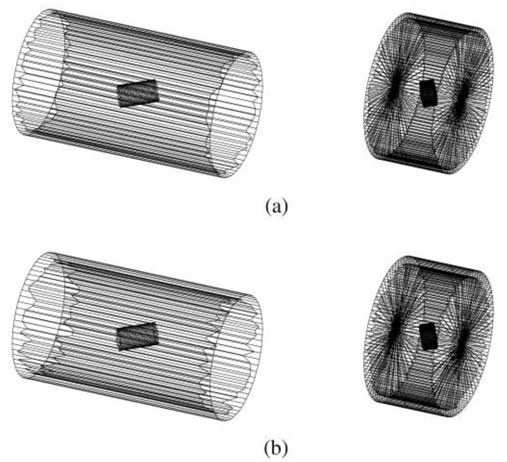

Fig. 14. Matching results with (a) 5% and (b) 10% noise contamination in the input data.

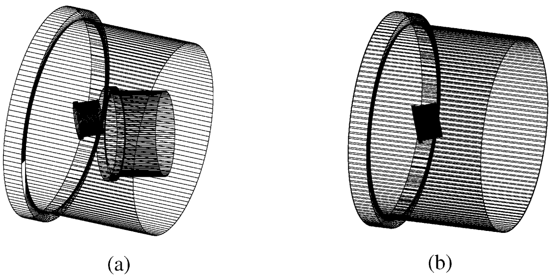

Fig. 15. (a) An unsuccessful shape-matching result due to a highly uneven distribution of input data. (b) The result obtained by using a subset of the input data.

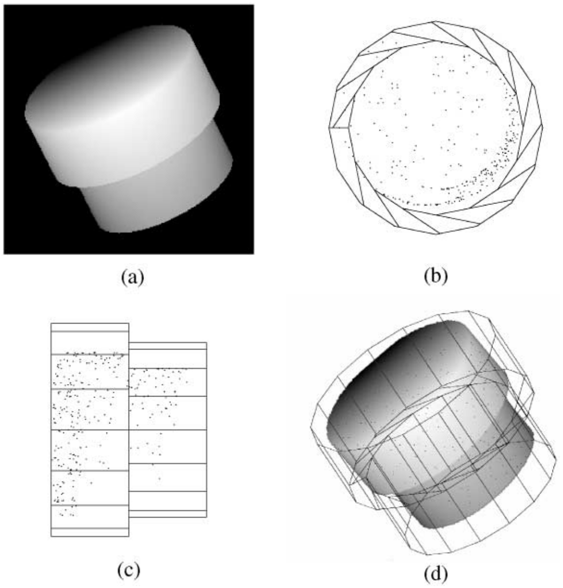

Fig. 16. Shape-matching result for range data (see text).

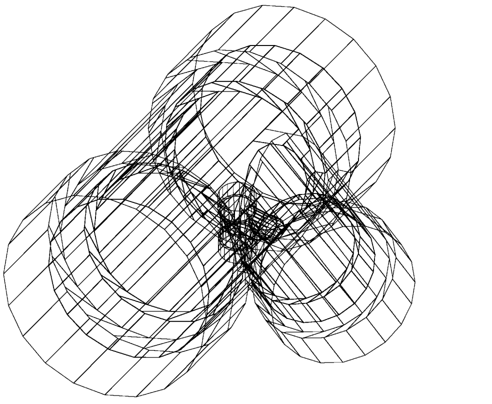

Fig. 17. Another unsuccessful shape-matching result.

of identical shape will not grow fully after the matching process, as shown in Fig. 17. (As a simpler 2D example of such a problem, consider the process of growing a size-reduced I-shaped object in the corresponding 2D-shaped model.) More involved scaling scheme will need to be developed to overcome such a problem.

5. Conclusions

A novel approach of shape-matching and recognition of 3D objects using generalized potential fields is proposed in this paper. The potential-based model assumes that 3D object boundaries are uniformly charged. An initially small input object placed inside a template object can translate and reorient itself to reduce the repulsive potential. If the input object is allowed to increase its size in template objects of different shapes, the template object which allows the largest final size of the input object will correspond to the best match. Such an approach is intrinsically invariant under translation, rotation and size changes of the input object.

For an input object represented by its boundary samples, the above process of potential reduction can be carried out efficiently since the resultant potential gradients, in the forms of repulsive force and torque exerted on the object, are analytically tractable. Simulation results show that the potential-based shape-matching scheme has the time complexity as a linear function of the boundary samples and works to some extent for a partial view of an object. Moreover, due to the nature of the potential model, such a scheme performs satisfactorily against noise contamination in the input data. The proposed shape-matching scheme is not based on any hypothesis of feature correspondence. Therefore, feature extraction for a structured object representation is not needed.

Acknowledgements

This work was supported by National Science Council, Republic of China, under grants NSC86-2213-E-009-105 and NSC87-2213-E-009-065.

References

[1] Perkins WA. A model based vision system for industrial parts. IEEE Transactions on Computers 1978;21:126–43.

[2] Rummel P, Beutel W. Workpiece recognition and inspection by a model-based scene analysis system. Pattern Recognition 1984;17:141–8.

[3] Ayache N, Faugeras OD. HYPER: a new approach for the recognition and positioning of two-dimensional objects. IEEE Transactions on Pattern Analysis and Machine Intelligence 1986;8:44–54.

[4] Zahn CT, Roskies RZ. Fourier descriptors for plane closed curves. IEEE Transactions on Computers 1972;21: 269–81.

[5] Persoon E, Fu KS. Shape discrimination using Fourier descriptors. IEEE Transactions on Systems Man and Cybernetics 1977;7:170–9.

[6] Hu MK. Visual pattern recognition by moment invariants. IEEE Transactions on Information Theory 1962;8: 179–87.

[7] Maitra S. Moment invariants. Proceedings of IEEE 1979;67:697–9.

[8] Reddi SS. Radial and angular moment invariants for image identification. IEEE Transactions on Pattern Analysis and Machine Intelligence 1981;3:240–2.

[9] Leu JG. Computing a shape's moments from its boundary. Pattern Recognition 1991;24:949–57.

[10] Dubois SR, Glanz FH. An autoregressive model approach to two-dimensional shape classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 1977;8: 55–66.

[11] Kashyap RL, Chellappa R. Stochastic models for close boundary analysis: representation and reconstruction. IEEE Transactions on Information Theory 1981;27: 627–37.

[12] Das M, Paulik MJ, Loh NK. A bivariate autoregressive modeling technique for analysis and classification of planar shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 1990;12:97–103.

[13] Freeman H, Davis LS. A corner finding algorithm for chain coded curves. IEEE Transactions on Computers 1977;26:297–303.

[14] Tsai WH, Yu SS. Attributed string matching with merging for shape recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 1985;7:453–62.

[15] Maes M. Polygonal shape recognition using string-matching techniques. Pattern Recognition 1991;24:433–40.

[16] Blumenkrans A. 2-dimensional object recognition using a 2-dimensional polar transform. Pattern Recognition 1991;24:879–90.

[17] Chang CC, Hwang SM, Buehrer DJ. A shape recognition scheme based on relative distances of feature points from the centroid. Pattern Recognition 1991;24:1053–63.

[18] Shapiro LG, Haralic RM. Structural descriptions and inexact matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 1981;3:504–19.

[19] Ueda N, Suzuki S. Learning visual models from shape contours using multiscale convex/concave structure matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 1993;15:337–52.

[20] Chuang J-H. A potential-based approach for shape-matching and recognition. Pattern Recognition 1996;29(3): 463–70.

[21] Flynn PJ, Jain AK. 3D object recognition using invariant feature indexing of interpretation tables. Image Understanding 1992;55(2):119–29.

[22] Flynn PJ. 3D object recognition with symmetric models: symmetry extraction and encoding. IEEE Transactions on Pattern Analysis and Machine Intelligence 1994;16(8): 814–8.

[23] Mao J, Flynn PJ, Jain AK. Integration of multiple feature groups and multiple views into a 3d object recognition system. Computer Vision and Image Understanding 1995; 62(3):309–25.

[24] Flynn PJ, Jain AK. BONSAI: 3D object recognition using constrained search. IEEE Transactions on Pattern Analysis and Machine Intelligence 1991;13(10):1066–75.

[25] Wong KC, Cheng Y, Kittler J. Recognition of polyhedral using triangle pair features. IEE Proceedings-I 1993;140(1): 72–85.

[26] Huttenlocher DP, Ullman S. Recognition solid objects by alignment with an image. International Journal of Computer Vision 1990;5:195–212.

[27] Horaud R, Sossa H. Polyhedral object recognition by indexing. Pattern Recognition 1995;28(12):1855–70.

[28] Olson CF. Probabilistic indexing for object recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 1995;17(5):518–22.

[29] Arbter K, Snyder WE, Burkhardt H, Hirzinger G. Application of affine-invariant Fourier descriptors to recognition of 3D objects. IEEE Transactions on Pattern Analysis and Machine Intelligence 1990;12(7):640–7.

[30] Pentland A, Sclaroff S. Closed-form solutions for physically based shape modeling and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 1991;13(7):715–29.

[31] Chen Y, Medioni G. Surface description of complex objects from multiple range images. Proceedings of IEEE Conference on CVPR, 1994. p. 153–8.

[32] Wu K, Levine MD. 3D parts segmentation using simulated electrical charge distributions. IEEE Transactions on Pattern Analysis and Machine Intelligence 1997; 19(11):1223–35.

[33] Chuang J-H. Potential-based modeling of three-dimensional workspace for obstacle avoidance. IEEE Transactions on Robotics and Automation 1998;14(5):778–85.

[34] Wilton DR, Rao SM, Glisson AW, Schaubert DH, Al-Bundak OM, Butler CM. Potential integrals for uniform and linear source distributions on polygonal and polyhedral domains. IEEE Transactions on Antennas and Propagation 1984;AP-32(3):276–81.

[35] Bers L, Karal F. Calculus. New York: Holt, Rinehart and Winston, 1976.