Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 28 (2007) 1252–1262
Pattern Recognition Letters
www.elsevier.com/locate/patrec

3D free-form object recognition in range images using local surface patches

Hui Chen, Bir Bhanu *

Center for Research in Intelligent Systems, University of California, Riverside, CA 92521, USA

Received 20 April 2006; received in revised form 25 January 2007
Available online 28 February 2007

Communicated by G. Sanniti di Baja

Abstract

This paper introduces an integrated local surface descriptor for surface representation and 3D object recognition. A local surface descriptor is characterized by its centroid, its local surface type and a 2D histogram. The 2D histogram shows the frequency of occurrence of shape index values vs. the angles between the normal of reference feature point and that of its neighbors. Instead of calculating local surface descriptors for all the 3D surface points, they are calculated only for feature points that are in areas with large shape variation. In order to speed up the retrieval of surface descriptors and to deal with a large set of objects, the local surface patches of models are indexed into a hash table. Given a set of test local surface patches, votes are cast for models containing similar surface descriptors. Based on potential corresponding local surface patches candidate models are hypothesized. Verification is performed by running the Iterative Closest Point (ICP) algorithm to align models with the test data for the most likely models occurring in a scene. Experimental results with real range data are presented to demonstrate and compare the effectiveness and efficiency of the proposed approach with the spin image and the spherical spin image representations.
© 2007 Elsevier B.V. All rights reserved.

Keywords: 3D object recognition; Local surface patch; Model indexing; Free-form surface registration; Range images

1. Introduction

3D object recognition, an important research field of computer vision and pattern recognition, involves two key tasks: object detection and object recognition. Object detection determines if a potential object is present in a scene and its location; object recognition determines the object ID and its pose (Suetens et al., 1992). Researchers have done an extensive research on recognizing objects from 2D intensity images. It has been challenging to design a system based on 2D intensity images which can handle problems associated with changing 3D pose, lighting and shadows effectively. The 3D data collected by a range sensor can provide geometric information about objects which is less sensitive to the above imaging problems. As a result, the design of a recognition system using 3D range data has received significant attention over the years.

In 3D object recognition, the key problems are how to represent free-form surfaces effectively and how to match the surfaces using the selected representation. In the early years of 3D computer vision (Besl and Jain, 1985; Chin and Dyer, 1986), the representation schemes included Wire-Frame, Constructive Solid Geometry (CSG), Extended Gaussian Image (EGI), Generalized Cylinders and planar faces (Bhanu, 1984; Faugeras and Hebert, 1986). Early work mainly dealt with the polyhedral objects. It segmented curved surfaces into planar surfaces. However, the planar patch is not the most suitable representation for free-from surfaces and researchers have used a number of representations, including B-Splines (Bhanu

* Corresponding author. Tel.: +1 9097873954; fax: +1 9097873188.
E-mail addresses: hchen@vislab.ucr.edu (H. Chen), bhanu@vislab.ucr.edu (B. Bhanu).

0167-8655/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2007.02.009

and Ho, 1987), surface curvatures, superquadrics (Solina and Bajcsy, 1990) and deformable models to recognize free-form objects in range images (Campbell and Flynn, 2001). Other recent surface representations include the splash representation (Stein and Medioni, 1992), the point signature (Chua and Jarvis, 1997), the spin image (Johnson and Hebert, 1999), the surface point signature (Yamany and Farag, 1999), the harmonic shape image (Zhang and Hebert, 1999), the spherical spin image (Correa and Shapiro, 2001), the 3D point's "fingerprint" (Sun and Abidi, 2001) and the 3D shape contexts and harmonic shape contexts (Frome et al., 2004).

In this paper, we introduce an integrated local surface descriptor for 3D object representation. We calculate the local surface descriptors only for the feature points which are in the areas with large shape variation measured by shape index (Dorai and Jain, 1997). Our approach starts from extracting feature points in range images, then defines the local surface patch at each of the feature points (Chen and Bhanu, 2004). Next we calculate local surface properties of a patch. These properties are 2D histogram, surface type and the centroid. The 2D histogram consists of shape indexes and angles between the normal of the feature point and that of its neighbors. The surface of a patch is classified into different types based on the mean and Gaussian curvatures of the feature point. For every local surface patch, we compute the mean and standard deviation of shape indexes and use them as indexes to a hash table. By comparing local surface patches for a model and a test image, and casting votes for the models containing similar surface descriptors, the potential corresponding local surface patches and candidate models are hypothesized. Finally, we estimate the rigid transformation based on the corresponding local surface patches and calculate the match quality between the hypothesized model and test image.

The rest of the paper is organized as follows. Section 2 introduce the related work and contributions. Section 3 presents our approach to represent the free-form surfaces and matching the surface patches. Section 4 gives the experimental results to demonstrate the effectiveness and efficiency of the proposed approach and compares them with the spin image and spherical spin image representations. Section 5 provides the conclusions.

2. Related work and contributions

2.1. Related work

Stein and Medioni (1992) used two different types of primitives, 3D curves and splashes, for representation and matching. 3D curves are defined from edges and they correspond to the discontinuity in depth and orientation. For smooth areas, splash is defined by surface normals along contours of different radii. Both of the primitives can be encoded by a set of 3D super-segments, which are described by the curvature and torsion angles of a super-segment. The 3D super-segments are indexed into a hash table for fast retrieval and matching. Hypotheses are generated by casting votes to the hash table and false hypotheses are removed by estimating rigid transformations. Chua and Jarvis (1997) used the point signature representation, which describes the structural neighborhood of a point, to represent 3D free-form objects. Point signature is 1D signed distance profile with respect to the rotation angle defined by the angle between the normal vector of the point on the curve and the reference vector. Recognition is performed by matching the signatures of points on the scene surfaces to those of points on the model surfaces. The maximum and minimum values of the signatures are used as indexes to a 2D table for fast retrieval and matching.

Johnson and Hebert (1999) presented the spin image (SI) representation for range images. Given an oriented point on a 3D surface, its shape is described by two parameters: distance to the tangent plane of the oriented point from its neighbors and the distance to the normal vector of the oriented point. The approach involved three steps: generating a spin image, finding corresponding points and verifying hypotheses. First, spin images are calculated at every vertex of the model surfaces. Then the corresponding point pair is found by computing the correlation coefficient of two spin images centered at those two points. Next the corresponding pairs are filtered by using geometric constraint. Finally, a rigid transformation is computed and a modified Iterative Closest Point (ICP) algorithm is used for verification. In order to speed up the matching process, principal component analysis (PCA) is used to compress spin images. Correa and Shapiro (2001) proposed the spherical spin image (SSI) which maps the spin image to points onto a unit sphere. Corresponding points are found by computing the angle between two SSIs. Yamany and Farag (1999) introduced the surface signature representation which is a 2D histogram, where one parameter is the distance between the center point and every surface point and the other one is the angle between the normal of the center point and every surface point. Signature matching is done by template matching.

Zhang and Hebert (1999) introduced harmonic shape images (HSI) which are 2D representation of 3D surface patches. HSIs are unique and they preserve the shape and continuity of the underlying surfaces. Surface matching is simplified to matching harmonic shape images. Sun and Abidi (2001) introduced 3D point's "fingerprint" representation which is a set of 2D contours formed by the projection of geodesic circles onto the tangent plane. Each point's fingerprint carried information of the normal variation along geodesic circles. Corresponding points are found by comparing the fingerprints of points. Frome et al. (2004) introduced two regional shape descriptors, 3D shape contexts and harmonic shape contexts, for recognizing 3D objects. The 3D shape context is the straightforward extension of 2D shape contexts (Belongie et al., 2002) and the harmonic shape context is obtained by applying the harmonic transformation to the 3D shape context. Objects are recognized by comparing the distance between the representative descriptors.

2.2. Contributions

The contributions of this paper are: (a) A new local surface descriptor, called LSP representation, is proposed for surface representation and 3D object recognition. (b) The LSP representation is compared to the spin image (Johnson and Hebert, 1999) and spherical spin image (Correa and Shapiro, 2001) representations for its effectiveness and efficiency. (c) Experimental results on a dataset of 20 objects (with/without occlusions) are presented to verify and compare the effectiveness of the proposed approach.

3. Technical approach

The proposed approach is described in Table 1. It has two stages: offline model building and online recognition.

3.1. Feature points extraction

In our approach, feature points are defined in areas with large shape variation measured by shape index calculated from principal curvatures. In order to estimate the curvature of a point on the surface, we fit a quadratic surface $f(x,y) = ax^2 + by^2 + cxy + dx + ey + f$ to a local window centered at this point and use the least square method to estimate the parameters of the quadratic surface, and then use differential geometry to calculate the surface normal, Gaussian and mean curvatures and principal curvatures (Bhanu and Chen, 2003; Flynn and Jain, 1989). We move the local window around and repeat the same procedure to compute the shape index value for other points.

Shape index (S_i), a quantitative measure of the shape of a surface at a point p, is defined by (1) where k_1 and k_2 are maximum and minimum principal curvatures, respectively

$$S_i(p) = \frac{1}{2} - \frac{1}{\pi}\tan^{-1}\frac{k_1(p) + k_2(p)}{k_1(p) - k_2(p)}. \quad (1)$$

With this definition, all shapes are mapped into the interval $[0,1]$ (Dorai and Jain, 1997). Larger shape index values represent convex surfaces and smaller shape index values represent concave surfaces (Koenderink and Doorn, 1992). Fig. 1 shows the range image of an object and its shape index image. In Fig. 1a, the darker pixels are away from the camera while the lighter ones are closer. In Fig. 1b, the brighter points denote large shape index values which correspond to ridge and dome surfaces while the darker pixels denote small shape index values which correspond to valley and cup surfaces. From Fig. 1, we can see that shape index values can capture the characteristics of the shape of objects, which suggests that shape index can be used for feature point extraction. In other words, the center point is marked as a feature point if its shape index S_i satisfies Eq. (2) within a $w \times w$ window

$$\begin{aligned} &S_i = \max \text{ of shape indexes and } S_i \geqslant (1+\alpha) * \mu, \\ &\text{or } S_i = \min \text{ of shape indexes and } S_i \leqslant (1-\beta) * \mu, \\ &\text{where } \mu = \frac{1}{M}\sum_{j=1}^{M} S_i(j) \quad 0 \leqslant \alpha,\ \beta \leqslant 1. \end{aligned} \quad (2)$$

In Eq. (2) α, β parameters control the selection of feature points and M is the number of points in the local window. The results of feature extraction are shown in Fig. 2, where the feature points are marked by red dots. From Fig. 2, we can clearly see that some feature points corresponding to the same physical area appear in both images.

Table 1
Algorithms for recognizing 3D objects in a range image

(a) For each model object
{
  Extract feature points (Section 3.1);
  Compute the LSP descriptors for the feature points (Section 3.2);
  for each LSP
  {
    Compute (μ,σ) of the shape index values and use them to index a hash table;
    Save the model ID and LSP into the corresponding entry in the hash table; (Section 3.3)
  }
}

(b) Given a test object
{
  Extract feature points (Section 3.1);
  Compute the LSP descriptors for the feature points (Section 3.2);
  for each LSP
  {
    Compute (μ,σ) of the shape index values and use them to index a hash table;
    Cast votes to the model objects which have a similar LSP (Section 3.4.1);
  }
  Find the candidate models with the highest votes (Section 3.4.2);
  Group the corresponding LSPs for the candidate models (Section 3.4.2);
  Use the ICP algorithm to verify the top hypotheses (Section 3.5);
}

(a) Algorithm for constructing the model database (offline stage). (b) Algorithm for recognizing a test object (online stage).

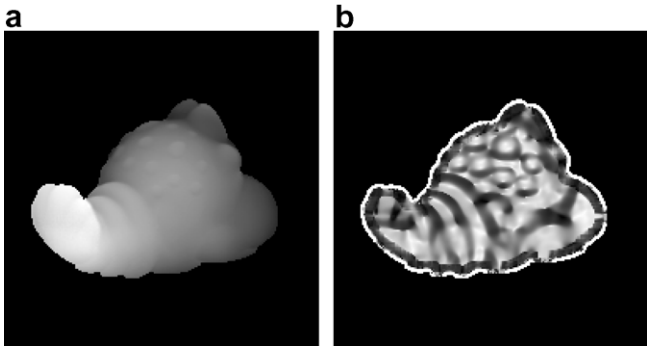

Fig. 1. (a) A range image and (b) its shape index image. In (a), the darker pixels are away from the camera and the lighter ones are closer. In (b), the darker pixels correspond to concave surfaces and the lighter ones correspond to convex surfaces.

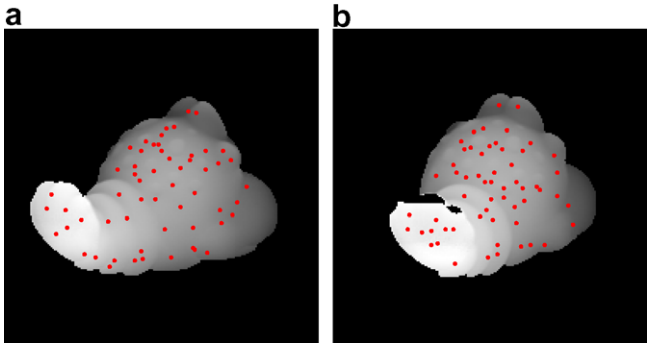

Fig. 2. Feature points location (·) in two range images, shown as gray scale images, of the same object taken at different viewpoints.

3.2. Local surface patches (LSP)

We define a "local surface patch" as the region consisting of a feature point P and its neighbors N. The LSP representation includes its surface type T_p, centroid of the patch and a histogram of shape index values vs. dot product of the surface normal at the feature point P and its neighbors N. A local surface patch is shown in Fig. 3. The neighbors N satisfy the following conditions:

$$N = \{pixels\ \ N, \|N - P\| \leqslant \epsilon_1\} \text{ and } acos(n_p \bullet n_n) < A, \quad (3)$$

where $\bullet$ denotes the dot product between the surface normal vectors n_p and n_n at the feature point P and at a neighboring point of N, respectively. The *acos* denotes the inverse cosine function. The two parameters ϵ_1 and A are important since they determine the descriptiveness of the local surface patch representation. For every point N_i belonging to N, we compute its shape index value and the angle θ between the surface normals at the feature point P and N_i. Then we form a 2D histogram by accumulating points in particular bins along the two axes based on Eq. (4) which relates the shape index value and the angle to the 2D histogram bin (h_x, v_y). One axis of this histogram is the shape index which is in the range $[0, 1]$; the other is the cosine of the angle ($\cos\theta$) between the surface normal vectors at P and one of its neighbors in N. It is equal to the dot product of the two vectors and it is in the range $[-1, 1]$. In (4), $\lfloor f \rfloor$ is the floor operator which rounds f down to the nearest integer; (h_x, v_y) are the indexes along the horizontal and vertical axes respectively and (b_x, b_y) are the bin intervals along the horizontal and vertical axes, respectively. In order to reduce the effect of the noise, we use bilinear interpolation when we calculate the 2D histogram. One example of the 2D histogram is shown as a gray scale image in Fig. 3; the brighter areas in the image correspond to bins with more points falling into them. Note that in the 2D histogram in Fig. 3 some of the areas are black since no points are falling into those bins

$$h_x = \left\lfloor \frac{S_i}{b_h} \right\rfloor, \quad v_y = \left\lfloor \frac{\cos\theta + 1}{b_y} \right\rfloor. \quad (4)$$

The surface type T_p of a LSP is obtained based on the Gaussian and mean curvatures of the feature point using Eq. (5) (Besl and Jain, 1988; Bhanu and Nuttall, 1989) where H are mean curvatures and K are Gaussian curvatures. There are eight surface types determined by the signs of Gaussian and mean curvatures given in Table 2. The centroid of local surface patches is also calculated for the computation of the rigid transformation. Note that a feature point and the centroid of a patch may not coincide.

In summary, every local surface patch is described by a 2D histogram, surface type T_p and the centroid. The 2D histogram and surface type are used for comparison of LSPs and the centroid is used for grouping corresponding LSPs and computing the rigid transformation, which will be explained in the following sections. The local surface patch encodes the geometric information of a local surface

$$T_p = 1 + 3(1 + \mathrm{sgn}_{\epsilon_H}(H)) + (1 - \mathrm{sgn}_{\epsilon_K}(K)),$$
$$\mathrm{sgn}_{\epsilon_x}(X) = \begin{cases} +1 & \text{if } X > \epsilon_x, \\ 0 & \text{if } |X| \leqslant \epsilon_x, \\ -1 & \text{if } X < \epsilon_x. \end{cases} \quad (5)$$

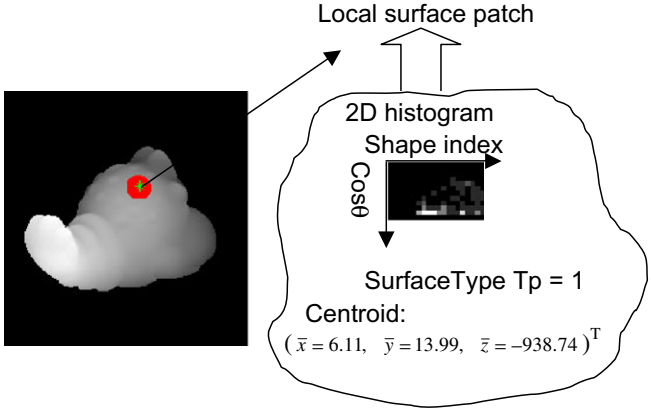

Fig. 3. Illustration of a local surface patch (LSP). Feature point P is marked by the asterisk and its neighbors N are marked by the dots. The surface type of the LSP is 1 based on Table 2.

3.3. Hash table building

Considering the uncertainty of location of a feature point, we also calculate descriptors of local surface patches

Table 2
Surface type T_p based on the signs of mean curvature (H) and Gaussian curvature (K)

Mean curvature H	Gaussian curvature K		
	$K > 0$	$K = 0$	$K < 0$
$H < 0$	Peak $T_p = 1$	Ridge $T_p = 2$	Saddle ridge $T_p = 3$
$H = 0$	None $T_p = 4$	Flat $T_p = 5$	Minimal $T_p = 6$
$H > 0$	Pit $T_p = 7$	Valley $T_p = 8$	Saddle valley $T_p = 9$

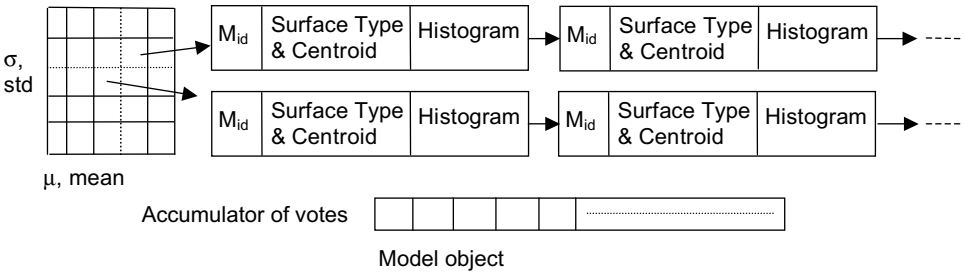

Fig. 4. Structure of the hash table. Every entry in the hash table has a linked list which saves information about the model LSPs and the accumulator records the number of votes that each model receives.

for neighbors of a feature point P. To speed up the retrieval of local surface patches, for each LSP we compute the mean $\left(\mu = \frac{1}{L}\sum_{l=1}^{L} S_i(p_l)\right)$ and standard deviation $\left(\sigma^2 = \frac{1}{L-1}\sum_{l=1}^{L}(S_i(p_l) - \mu)^2\right)$ of the shape index values in N where L is the number of points on the LSP under consideration and p_l is the lth point on the LSP. Then we use them to index a hash table and insert into the corresponding hash bin the information about the model LSPs. Therefore, the model local surface descriptors are saved into the hash table. For each model object, we repeat the same process to build the model database. The structure of the hash table is explained in Fig. 4, where every bin in the hash table has an associated linked list which saves the information of the model surface descriptors in terms of model ID, 2D histogram, surface type and the centroid; and the accumulator keeps track of the number of votes that each model obtains.

3.4. Recognition

3.4.1. Comparing local surface patches

Given a test range image, we extract feature points and get local surface patches. Then we calculate the mean and stand deviation of the shape index values in N for each LSP, and cast votes to the hash table if the histogram dissimilarity between a test LSP and a model LSP falls within a preset threshold ϵ_2 and the surface type is the same. Since a histogram can be thought of as an approximation of a probability density function, it is natural to use the $\chi^2 - divergence$ function (6) to measure the dissimilarity (Schiele and Crowley, 2000)

$$\chi^2(Q,V) = \sum_i \frac{(q_i - v_i)^2}{q_i + v_i}, \tag{6}$$

where Q and V are the two normalized histograms and q_i and v_i are the numbers in the ith bin of the histogram for Q and V, respectively.

From (6), we know the dissimilarity is between 0 and 2. If the two histograms are exactly the same, the dissimilarity will be zero. If the two histograms do not overlap with each other, it will achieve the maximum value 2.

Fig. 5 and Table 3 show an experimental validation that the local surface patch is view-invariant and has the discriminative power to distinguish shapes. We do experiments under two cases: (1) a local surface patch (LSP1) generated for an object is compared to another local surface patch (LSP2) corresponding to the same physical area of the same object imaged at a different viewpoint; a low dissimilarity ($\chi^2(\text{LSP1},\text{LSP2}) = 0.24$) is found between LSP1 and LSP2 and they have the same surface type. (2) LSP1 is compared to LSP3 which lies in a different area of the same object; the dissimilarity ($\chi^2(\text{LSP1},\text{LSP3}) = 1.91$) is high even though they happen to have the same surface type. The experimental results suggest that the local surface patch representation provides distinguishable features and it can be used for distinguishing objects. Table 3 lists the comparison of LSPs. We observe that the two similar local surface patches (LSP1 and LSP2) have close mean and standard deviation of the shape index values (compared to other combinations); they can be used for fast retrieval of local surface patches.

Table 3
Comparison results for three local surface patches shown in Fig. 5

	Mean	Std.	Surface type	$\chi^2 - divergence$
LSP1	0.672	0.043	9	$\chi^2(\text{LSP1},\text{LSP2}) = 0.24$
LSP2	0.669	0.038	9	$\chi^2(\text{LSP1},\text{LSP3}) = 1.91$
LSP3	0.274	0.019	9	

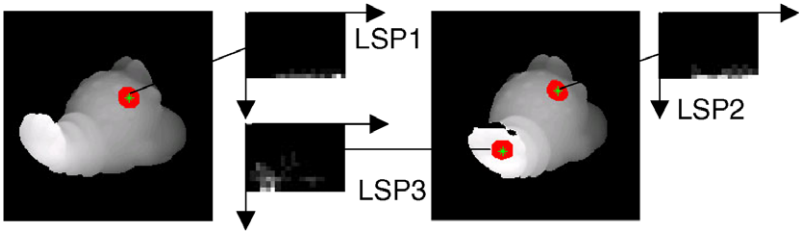

Fig. 5. Demonstration of discriminatory power of local surface patches. The 2D histograms of three LSPs are displayed as gray scale images. The axes for the LSP image are the same as shown in Fig. 3. Note that LSP1 and LSP2 are visually similar and LSP1 and LSP3 are visually different.

3.4.2. Grouping corresponding pairs of local surface patch

After voting by all the LSPs contained in a test object, we histogram all hash table entries and get models which receive the highest votes. From the casted votes, we know not only the models which get higher votes, but also the potential corresponding local surface patch pairs. Note that a hash table entry may have multiple items, we choose the local surface patch with the minimum dissimilarity and the same surface type as the possible corresponding patch. We filter the possible corresponding pairs based on the geometric constraint,

$$d_{C_1,C_2} = |d_{S_1,S_2} - d_{M_1,M_2}| < \epsilon_3, \tag{7}$$

where d_{S_1,S_2} and d_{M_1,M_2} are the Euclidean distances between centroids of the two surface patches. The constraint (7) guarantees that the distances d_{S_1,S_2} and d_{M_1,M_2} are consistent. For two correspondences $C_1 = \{S_1, M_1\}$ and $C_2 = \{S_2, M_2\}$ where S is the test surface patch and M is the model surface patch, they should satisfy (7) if they are consistent corresponding pairs. Therefore, we use the simple geometric constraint (7) to partition the potential corresponding pairs into different groups. The larger the group is, the more likely it contains the true corresponding pairs.

Given a list of corresponding pairs $L = \{C_1, C_2, \ldots, C_n\}$, the grouping procedure for every pair in the list is as follows: (a) Use each pair as a group of an initial matched pair. (b) For every group, add other pairs to it if they satisfy (7). (c) Repeat the same procedure for every group. (d) Select the group which has the largest size.

Fig. 6 shows one example of partitioning corresponding pairs into groups. Fig. 6a shows the feature point extraction results for a test object. Comparing the local surface patches with the LSPs on the model objects and querying the hash table, the initial corresponding LSP pairs are shown in Fig. 6b and c, in which every pair is represented by the same number superimposed on the test and model object images. We observe that both of true and false corresponding pairs are found. After applying the geometric constraint (7), the filtered largest group is shown in Fig. 6d and e, in which the pairs satisfying the constraint (7) are put into one group. We observe that true correspondences between the model and the test objects are obtained by comparing local surface patches, casting votes to the hash table and using the simple geometric constraint.

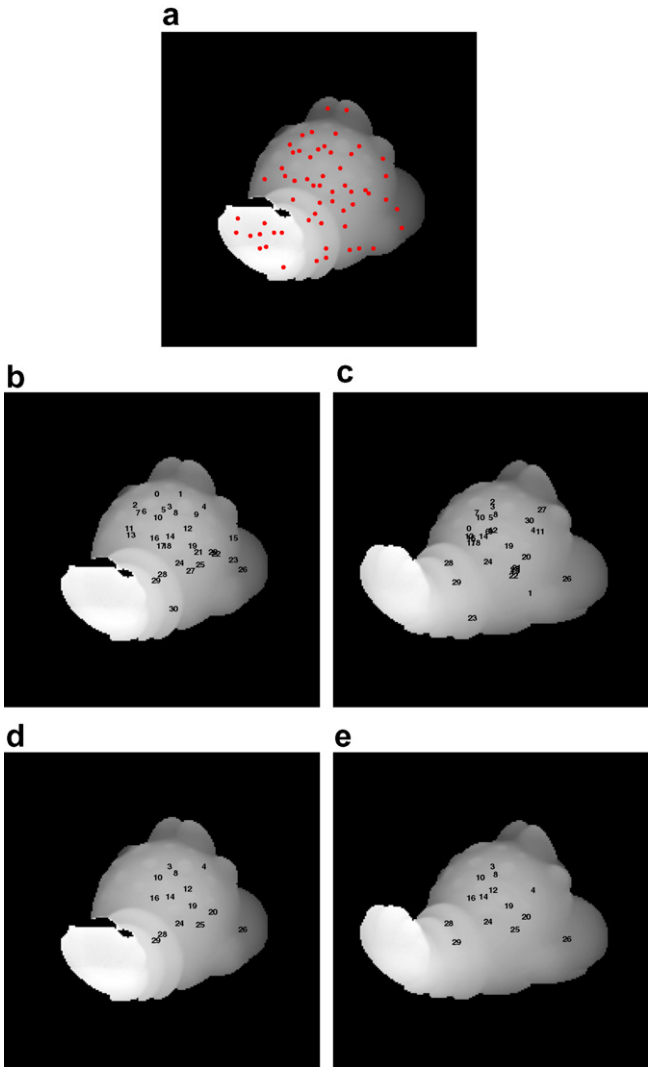

Fig. 6. An example of corresponding LSPs. (a) Feature points marked as dots on the test object. (b) Test object with matched LSPs after hashing. (c) A model object with matched LSPs after hashing. (d) Test object in (b) with matched LSPs after applying the geometric constraint (7). (e) The model object in (c) with matched LSPs after applying the geometric constraint (7).

3.5. Verification

Given the v corresponding LSPs between a model-test pair, the initial rigid transformation, which brings the model and test objects into coarse alignment, can be estimated by minimizing the sum of the squares of alignment errors $\left(\Sigma = \frac{1}{v}\sum_{l=1}^{v}|U_l - R * M_l - T|^2\right)$ with respect to the rotation matrix R and the translation vector T where U_l and M_l are the centroids of a corresponding LSP pair between the test LSP U_l and the model LSP M_l. The rotation matrix and translation vector are computed by using the quaternion representation (Horn, 1987). Given the estimate of initial rigid transformation, the Iterative Closest Point (ICP) algorithm (Besl and Mckay, 1992) determines if the match is good and to find a refined alignment between them. If the test object is really an instance of the model object, the ICP algorithm will result in a good registration and a large number of corresponding surface points between the model-test pair will be found. Since ICP algorithm requires that the test data set be a subset of the model set, we use the modified ICP algorithm proposed by Zhang, 1994 to remove outliers based on the distance distribution.

Starting with the initial transformation obtained from the coarse alignment, the modified ICP algorithm is run to refine the transformation by minimizing the distance between the randomly selected points of the model object

and their closest points of the test object. For each object in the model database, the points are randomly selected and the modified ICP algorithm is applied to those points. The same procedure with random selection of points is repeated 15 times and the rigid transformation with the minimum root mean square (RMS) error is chosen. The object at the top of the sorted list of model objects with the minimum RMS error is declared as the recognized object. In the modified ICP algorithm, the speed bottleneck is the nearest neighbor search. Therefore, the kd-tree structure is used in the implementation.

4. Experimental results

4.1. Data and parameters

We use real range data collected by Ohio State University (OSU, 1999). There are 20 objects in our database and the range image of the model objects are shown in Fig. 7. The parameters of our approach are $\epsilon_1 = 6.5$ mm, $A = \pi/3$, $\epsilon_2 = 0.75$, $\epsilon_3 = 9.4$ mm, $\alpha = 0.35$, $\beta = 0.2$, and $\epsilon_H = \epsilon_K = 0.003$. For the LSP computation, the number of bins in the shape index axis is 17 and the number of bins in the other axis is 34. The total number of LSPs calculated for the model objects is about 34,000. The average size of local surface patch is 230 pixels and the average number of pixels on an object is 11,956. We apply our approach to the single-object and the two-object scenes. The model objects and scene objects are captured at two different viewpoints. All the 20 model-test pairs are 20° apart except the pairs of object 3, 14 and 19 that are 36° apart.

We have also used a large UCR ear database (pose variation ±35°) of 155 subjects with 902 images (UCR, 2006). In addition, we have used images with large pose variation from the UND dataset (UND, 2002). We have used all three datasets (OSU, UCR, UND) to evaluate the robustness and rotation invariance of the LSP representation (see Section 4.4).

4.2. Single-object scenes

These test cases show the effectiveness of the voting scheme and the discriminating power of LSPs in the hypothesis generation. For a given test object, feature points are extracted and the properties of LSPs are calculated. Then LSPs are indexed into the database of model LSPs. For each model indexed, its vote is increased by one. We show the voting results (shown as a percentage of the number of LSPs in the scene which received votes) for the 20 objects in Fig. 8. Note that in some cases the numbers shown are larger than 100 since some LSPs may receive more than one vote. We observe that most of the highest votes go to the correct models. For every test object, we perform the verification for the top three models which obtained the highest votes. The verification results are listed in Table 4, which shows the candidate model ID and the corresponding RMS registration error. From Table 4, we observe that all the test objects are correctly recognized. In order to examine the recognition results visually, we display the model object and test object in the same image before and after the alignment for four examples. The images in Fig. 9a show test objects and their corresponding model objects before alignment; the images in Fig. 9b show test objects and the correctly recognized model objects after alignment. We observe that each model object is well aligned with the corresponding test object and the test cases with large pose variations are correctly handled. Since the proposed LSP representation consists of

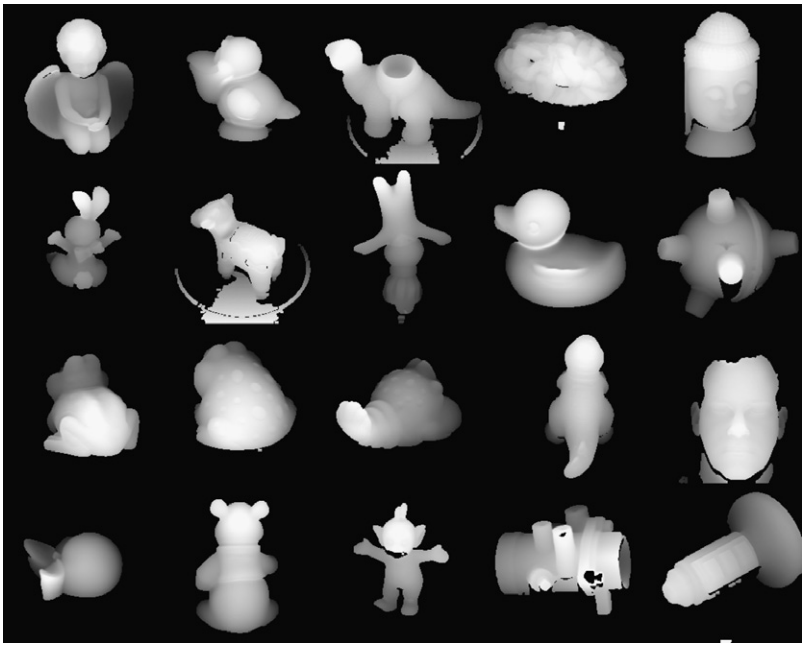

Fig. 7. The range images of objects in the model database. The object IDs (0–19) are labeled from left to right and top to bottom.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	**69**	16	69	27	36	11	40	7	33	24	49	60	33	7	20	16	4	0	51	49
1	51	116	75	44	46	30	42	40	104	53	100	**118**	79	32	38	2	67	14	65	79
2	63	56	**146**	54	61	8	76	41	78	54	82	76	49	45	39	8	52	2	53	63
3	21	21	60	**76**	44	7	44	39	55	10	55	63	71	31	18	10	31	0	10	50
4	35	44	30	25	**69**	5	12	17	67	37	67	50	19	28	7	5	25	0	35	51
5	8	20	11	5	2	**120**	5	22	5	5	31	25	17	0	0	0	25	54	5	5
6	68	49	158	52	30	6	**171**	53	76	57	73	83	78	42	39	19	45	0	105	46
7	10	62	12	20	12	55	27	**102**	60	12	82	55	57	15	0	15	40	62	30	65
8	50	61	86	53	50	26	45	36	**172**	58	113	94	93	28	24	30	57	12	65	87
9	30	32	48	13	36	3	55	3	86	**92**	26	44	32	30	17	40	46	0	63	28
10	43	85	68	80	61	48	58	81	118	38	**143**	114	75	21	31	13	68	8	85	123
11	18	86	68	47	38	33	83	76	63	29	80	**104**	90	45	40	6	63	7	63	100
12	57	72	75	79	62	22	90	88	75	31	87	127	**131**	79	24	14	61	9	61	100
13	31	75	68	27	44	10	41	27	58	27	51	65	41	**79**	6	3	51	0	27	44
14	31	51	100	37	72	6	72	17	**106**	41	89	93	51	65	96	6	48	0	55	86
15	5	65	10	5	0	25	0	15	35	15	35	25	25	10	0	**110**	60	20	25	10
16	35	64	69	41	58	19	48	42	**105**	42	83	85	42	39	21	8	103	10	37	58
17	5	43	7	0	2	53	0	25	17	7	41	43	23	7	0	2	30	**87**	0	10
18	9	30	44	11	13	13	50	21	63	26	67	59	48	26	11	13	25	0	**161**	55
19	40	49	63	44	32	7	51	47	105	43	108	87	57	47	36	18	30	2	64	**115**

Fig. 8. Voting results, shown as a percentage of the number of LSPs in the scene which received votes, for twenty models in the single-object scenes. Each row shows the voting results of a test object to 20 model objects. The maximum vote in each row is bounded by a box.

Table 4
Verification results for single-object scenes

Test objects	Results (top three matches)		
0	(**0**, **0.624**)	(2, 4.724)	(11, 1.529)
1	(11, 3.028)	(**1**, **0.314**)	(8, 3.049)
2	(**2**, **0.504**)	(10, 2.322)	(8, 2.148)
3	(**3**, **0.913**)	(12, 2.097)	(11, 1.335)
4	(**4**, **0.632**)	(8, 2.372)	(10, 1.781)
5	(**5**, **0.217**)	(17, 2.081)	(10, 3.146)
6	(**6**, **0.5632**)	(2, 3.840)	(18, 4.692)
7	(**7**, **0.214**)	(10, 2.835)	(19, 3.901)
8	(**8**, **0.426**)	(10, 1.326)	(11, 2.691)
9	(**9**, **0.459**)	(8, 2.639)	(18, 4.745)
10	(**10**, **0.263**)	(19, 2.451)	(8, 3.997)
11	(**11**, **0.373**)	(19, 3.773)	(12, 1.664)
12	(**12**, **0.525**)	(11, 1.698)	(19, 4.149)
13	(**13**, **0.481**)	(1, 1.618)	(2, 4.378)
14	(8, 2.694)	(2, 4.933)	(**14**, **0.731**)
15	(**15**, **0.236**)	(1, 2.849)	(16, 4.919)
16	(8, 3.586)	(**16**, **0.306**)	(11, 1.499)
17	(**17**, **0.252**)	(5, 2.033)	(11, 2.494)
18	(**18**, **0.395**)	(10, 2.316)	(8, 2.698)
19	(**19**, **0.732**)	(10, 2.948)	(8, 3.848)

The first number in the parenthesis is the model object ID and the second one is the RMS registration error. The unit of registration error is millimeters (mm).

histogram of shape index and surface normal angle, it is invariant to rigid transformation. The experimental results shown here verify the view-point invariance of the LSP representation.

4.3. Two-object scenes

We created four two-object scenes to make one object partially overlap the other object as follows. We first properly translated objects along the x- and y-axes, and then resampled the surface to create a range image. The visible points on the surface are identified using the Z-buffer algorithm. Table 5 provides the objects included in the four scenes and the voting and registration results (similar to the examples in Section 4.2) for the top six candidate model objects. The candidate models are ordered according to the percentage of votes they received and each candidate model is verified by the ICP algorithm. We observe that the objects in the first three scenes objects are correctly recognized and the object 12 is missed in the fourth scene since it received a lower number of votes and as a result was not ranked high enough. The four scenes are shown in Fig. 10a and the recognition results are shown in Fig. 10b. We observe that the recognized model objects are well aligned with the corresponding test objects.

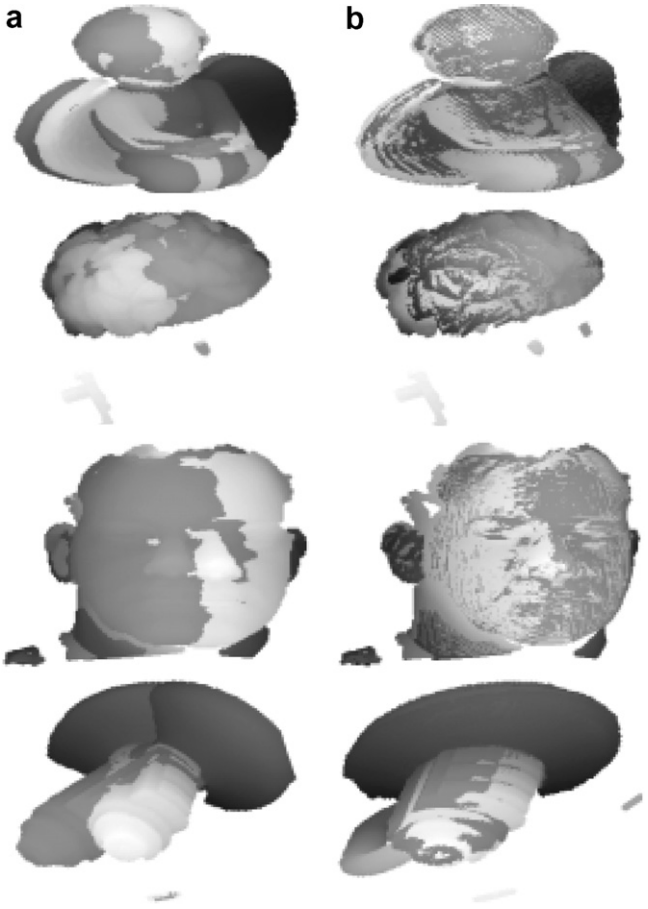

Fig. 9. Four examples of correctly recognized model-test pairs. Each row shows one example. The test objects are shaded light gray while the recognized model objects are shaded dark gray and overlaid on the test objects. (a) Model and test objects before alignment. (b) Model and test objects after alignment. For the range images of model objects, the lighter pixels are closer to the camera and the darker pixels are away from the camera. In example 1, the rotation angle is 20.4° and the axis is $[0.0319, 0.9670, 0.2526]^{T}$. In example 2, the rotation angle is 35.9° and the axis is $[-0.0304, -0.5714, -0.1660]^{T}$. In example 3, the rotation angle is 14.1° and the axis is $[0.0187, 0.2429, 0.0046]^{T}$. In example 4, the rotation angle is 36.2° and the axis is $[0.0691, 0.9724, 0.2128]^{T}$.

Table 5
Voting and registration results for the four two-object scenes shown in Fig. 10a

Test	Objects in the image	Voting and registration results for the top six matches					
Scene 0	1, 10	(**10**, **137**, **0.69**)	(**1**, **109**, **0.35**)	(11, 109, 1.86)	(2, 102, 5.00)	(12, 100, 1.78)	(19, 98, 2.14)
Scene 1	13, 16	(11, 72, 2.51)	(8, 56, 2.69)	(2, 56, 3.67)	(**13**, **56**, **0.50**)	(10, 51, 1.98)	(**16**, **48**, **0.53**)
Scene 2	6, 9	(**6**, **129**, **1.31**)	(2, 119, 3.31)	(18, 79, 3.74)	(8, 76, 2.99)	(**9**, **56**, **0.55**)	(12, 52, 1.97)
Scene 3	4, 12	(**4**, **113**, **0.81**)	(8, 113, 2.09)	(11, 88, 1.69)	(2, 86, 3.05)	(10, 81, 1.89)	(19, 74, 3.85)

The first number in the parenthesis is the model object ID, the second one is the voting result and the third one is RMS registration error. The unit of registration error is millimeters (mm).

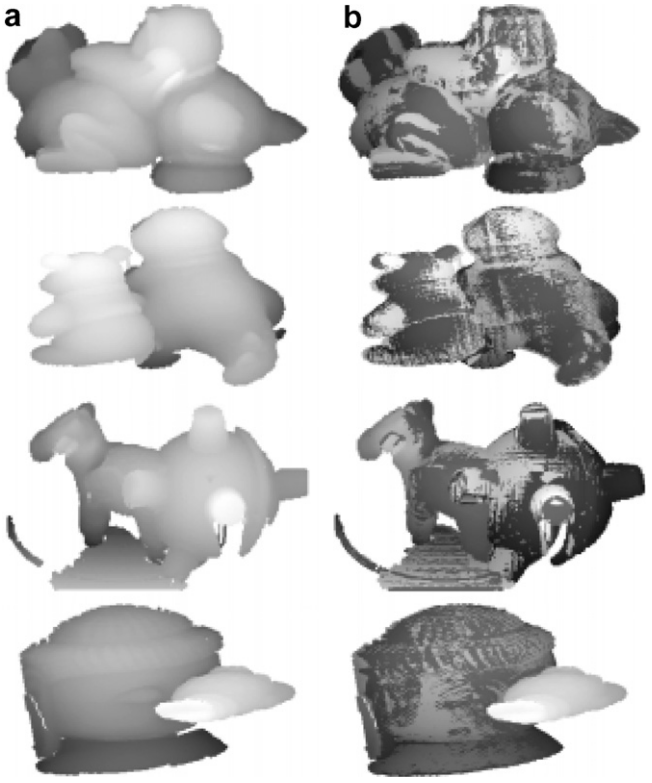

Fig. 10. Recognition results for the four two-object scenes. Each row shows one example. The test objects are shaded light gray while the recognized model objects are shaded dark gray. (a) Range images of the four two-object scenes. (b) Recognized model objects overlaid on the test objects with the recovered pose. For the range images of model objects, the lighter pixels are closer to the camera and the darker pixels are away from the camera. Note that in the last row one object is missed.

4.4. Robustness and rotation invariance of LSP representation

In order to show that the proposed LSP representation is robust and rotationally invariant, we tested it on a dataset of 3D ears collected by ourselves called the UCR dataset. The data are captured by Minolta Vivid 300 camera. The camera outputs a 200×200 range image and its registered color image. There are 155 subjects with a total of 902 shots and every person has at least four shots. There are three different poses in the collected data: frontal, left and right (within $\pm 35°$ with respect to the frontal pose). Fig. 11 shows side face range images of three people collected in the UCR dataset. The pose variations, the earrings and the hair occlusions can be seen in this figure. The dataset is split into a model set and a test set as follows. Two frontal ears of a subject are put in the model set and the rest of the ear images of the same subject are put in the test set. Therefore, there are 310 images in the model set and 592 test scans with different pose variations. The recognition rate is 95.61%.

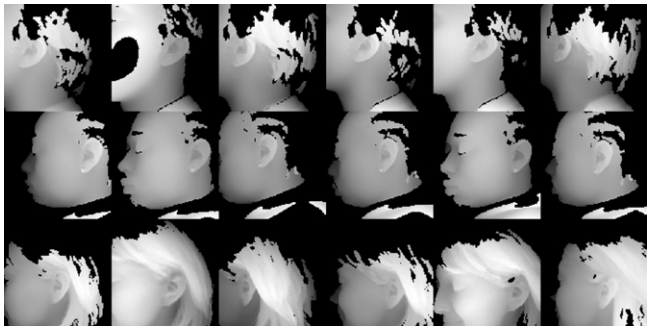

Fig. 11. Examples of side face range images of three people in the UCR dataset. Note the pose variations, the earrings and the hair occlusions for the six shots of the same person.

In addition, we also performed experiments on a subset of the UND dataset Collection G (UND, 2002), which has 24 subjects whose images are taken at four different poses, straight-on, 15° off center, 30° off center and 45° off center. Four range images of a subject with the four poses are shown in Fig. 12. For each of the straight-on ear images, we match it against rest of the images at different poses. The recognition rate is 86.11%. From the above two experiments, we conclude that the LSP representation can be used to recognize objects with a large pose variation (up to 45°).

4.5. Comparison with the spin image and the spherical spin image representations

We compared the distinctive power of the LSP representation with the spin image (SI) (Johnson and Hebert, 1999) and the spherical spin image (SSI) (Correa and Shapiro, 2001) representations. We conducted the following experiments. We take 20 model objects, compute feature points as described in Section 3.1 and calculate the surface descriptors at those feature points and their neighbors. Given a test object, we calculate the surface descriptors

Fig. 12. Four side face range images of a subject at four different poses (straight-on, 15° off, 30° off and 45° off) in the UND dataset.

Table 6
The timing in seconds for the three representations

	t_a	t_b	t_c	$\mathscr{T}$
LSP	21.46	0.8	67.16	89.42
SI	95.26	0.67	66.14	162.07
SSI	83.63	0.66	66.28	150.57

LSP denotes the local surface patch descriptor; SI denotes the spin image (Johnson and Hebert, 1999); SSI denotes the spherical spin image (Correa and Shapiro, 2001).

for the extracted feature points, find their nearest neighbors, apply the geometric constraint and perform the verification by comparing it against all the model objects. In the experiments, both of the size of the spin image and the spherical spin image are 15×15. We achieved 100% recognition rate by the three representations. However, the average computation time for the three representations are different. The total time ($\mathscr{T}$) for recognizing a single object consists of three timings: (a) find the nearest neighbors t_a, (b) find the group of corresponding surface descriptors t_b and (c) perform the verification t_c. These timings, on a Linux machine with a *AMD Opteron* 1.8 GHz processor, are listed in Table 6. We observe that the LSP representation runs the fastest for searching the nearest neighbors because the LSPs are formed based on the surface type and the comparison of LSPs is based on the surface type and the histogram dissimilarity.

5. Conclusions

We have presented an integrated local surface patch descriptor (LSP) for surface representation and 3D object recognition. The proposed representation is characterized by a centroid, a local surface type and a 2D histogram, which encodes the geometric information of a local surface. The surface descriptors are generated only for the feature points with larger shape variation. Furthermore, the generated LSPs for all models are indexed into a hash table for fast retrieval of surface descriptors. During recognition, surface descriptors computed for the scene are used to index the hash table, casting the votes for the models which contain the similar surface descriptors. The candidate models are ordered according to the number of votes received by the models. Verification is performed by running the Iterative Closest Point (ICP) algorithm to align models with scenes for the most likely models. Experimental results on the real range data have shown the validity and effectiveness of the proposed approach: geometric hashing scheme for fast retrieval of surface descriptors and comparison of LSPs for the establishment of correspondences. Comparison with the spin image and spherical spin image representations shows that our representation is as effective for the matching of 3D objects as these two representations but it is efficient by a factor of 3.79 (over SSI) to 4.31 (over SI) for finding corresponding parts between a model-test pair. This is because the LSPs are formed based on the surface type and the comparison of LSPs is based on the surface type and the histogram dissimilarity.

Acknowledgments

The authors would like to thank the Computer Vision Research Laboratory at the University of Notre Dame, for providing us their public biometrics database.

References

Belongie, S., Malik, J., Puzicha, J., 2002. Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Machine Intell. 24 (24), 509–522.

Besl, P., Jain, R., 1985. Three-dimensional object recognition. ACM Comput. Surv. 17 (1), 75–145.

Besl, P., Jain, R., 1988. Segmentation through variable-order surface fitting. IEEE Trans. Pattern Anal. Machine Intell. 10 (2), 167–192.

Besl, P., Mckay, N.D., 1992. A method of registration of 3-D shapes. IEEE Trans. Pattern Anal. Machine Intell. 14 (2), 239–256.

Bhanu, B., 1984. Representation and shape matching of 3-D objects. IEEE Trans. Pattern Anal. Machine Intell. 6 (3), 340–351.

Bhanu, B., Chen, H., 2003. Human ear recognition in 3D. Workshop on Multimodal User Authentication, 91–98.

Bhanu, B., Ho, C., 1987. CAD-based 3D object representations for robot vision. IEEE Comput. 20 (8), 19–35.

Bhanu, B., Nuttall, L., 1989. Recognition of 3-D objects in range images using a butterfly multiprocessor. Pattern Recognition 22 (1), 49–64.

Campbell, R.J., Flynn, P.J., 2001. A survey of free-form object representation and recognition techniques. Computer Vision and Image Understanding 81, 166–210.

Chen, H., Bhanu, B., 2004. 3D free-form object recognition in range images using local surface patches. Proc. Internat. Conf. Pattern Recognition 3, 136–139.

Chin, R., Dyer, C., 1986. Model-based recognition in robot vision. ACM Comput. Surv. 18 (1), 67–108.

Chua, C., Jarvis, R., 1997. Point signatures: a new representation for 3D object recognition. Internat. J. Comput. Vision 25 (1), 63–85.

Correa, S., Shapiro, L., 2001. A new signature-based method for efficient 3-D object recognition. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 769–776.

Dorai, C., Jain, A., 1997. COSMOS—A representation scheme for 3D free-form objects. IEEE Trans. Pattern Anal. Machine Intell. 19 (10), 1115–1130.

Faugeras, O., Hebert, M., 1986. The representation, recognition and locating of 3-D objects. Internat. J. Robotics Res. 5 (3), 27–52.

Flynn, P., Jain, A., 1989. On reliable curvature estimation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 110–116.

Frome, A., Huber, D., Kolluri, R., Bulow, T., Malik, J., 2004. Recognizing objects in range data using regional point descriptors. In: Proc. European Conference on Computer Vision, vol. 3, pp. 224–237.

Horn, B., 1987. Closed-form solution of absolute orientation using unit quaternions. J. Opt. Soc. Am. A 4 (4), 629–642.

Johnson, A., Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans. Pattern Anal. Machine Intell. 21 (5), 433–449.

Koenderink, J.J., Doorn, A.V., 1992. Surface shape and curvature scales. Image Vision Comput. 10 (8), 557–565.

OSU, 1999. OSU Range Image Database. URL <http://sampl.eng.ohio-state.edu/sampl/data/3DDB/RID/minolta/>.

Schiele, B., Crowley, J., 2000. Recognition without correspondence using multidimensional receptive field histograms. Internat. J. Comput. Vision 36 (1), 31–50.

Solina, F., Bajcsy, R., 1990. Recovery of parametric models from range images: The case for superquadrics with global deformations. IEEE Trans. Pattern Anal. Machine Intell. 12 (2), 131–147.

Stein, F., Medioni, G., 1992. Structural indexing: Efficient 3-D object recognition. IEEE Trans. Pattern Anal. Machine Intell. 14 (2), 125–145.

Suetens, P., Fua, P., Hanson, A., 1992. Computational strategies for object recognition. ACM Comput. Surv. 24 (1), 5–62.

Sun, Y., Abidi, M.A., 2001. Surface matching by 3D point's fingerprint. In: Proc. Internat. Conf. on Computer Vision 2, pp. 263–269.

UCR, 2006. UCR Ear Range Image Database. URL <http://vislab.u-cr.edu/>.

UND, 2002. UND Biometrics Database. URL <http://www.nd.edu/Ecvrl/UNDBiometricsDatabase.html>.

Yamany, S.M., Farag, A., 1999. Free-form surface registration using surface signatures. In: Proc. Internat. Conf. on Computer Vision, vol. 2, pp. 1098–1104.

Zhang, Z., 1994. Iterative point matching for registration of free-form curves and surfaces. Internat. J. Comput. Vision 13 (2), 119–152.

Zhang, D., Hebert, M., 1999. Harmonic maps and their applications in surface matching. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, pp. 524–530.