8

Applications of the Proposed Techniques

The techniques we proposed in Chapter 3 and Chapter 5 are general techniques in computer vision and pattern recognition. For example, the local surface patch (LSP) representation can be used to recognize 3D general objects; the global-to-local registration with the optimization framework can handle non-rigid shape registration. We discuss the applications of these two techniques in this chapter with several non-ear examples.

8.1 LSP Representation for General 3D Object Recognition

The LSP representation introduced in Chapter 5 for ear recognition is a general surface descriptor applicable to any 3D objects. In this subchapter, we propose a general 3D object recognition system using the LSP representation. 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 the votes for potential corresponding local surface patches the candidate models are hypothesized. The verification is performed by aligning models with the test data for the most likely models that occurs in a scene.

8.1.1 Local Surface Patch

Similar to the LSP representation introduced in Chapter 5, the surface type T_p of a LSP is obtained based on the Gaussian and mean curvatures of a feature point using equation (8.1) where H is the mean curvature and K is the Gaussian curvature [106, 107]. There are eight surface types determined by the signs of Gaussian and mean curvatures given in Table 8.1. A local surface patch is shown in Figure 8.1.

$$Tp = 1 + 3(1 + sgn_{\epsilon_H}(H)) + (1 - sgn_{\epsilon_K}(K))$$
$$sgn_{\epsilon_x}(X) = \begin{cases} +1\, if & X > \epsilon_x \\ 0 \quad if & |X| \le \epsilon_x \\ -1\, if & X < \epsilon_x \end{cases} \tag{8.1}$$

In this chapter, feature points are defined in areas with large shape variations as measured by the shape index values calculated from the principal curvatures. The shape index values are calculated as described in Section 5.1.

Figure 8.2 shows the range image of an object and its shape index image. In Figure 8.2(a), the darker pixels are away from the camera while the lighter ones are closer. In Figure 8.2(b), 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 Figure 8.2, 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 extraction of feature points. In other words, the center point is marked as a feature point if its shape index S_i satisfies equation (8.2) within a $w \times w$ window placed at a pixel in an image.

$$\begin{aligned} S_i &= \text{max of shape indexes and } S_i \ge (1+\alpha) * \mu \\ \text{or } S_i &= \text{min of shape indexes and } S_i \le (1-\beta) * \mu \\ \text{where } \mu &= \frac{1}{M}\sum_{j=1}^{M} S_i(j) \quad 0 \le \alpha, \beta \le 1. \end{aligned} \tag{8.2}$$

In equation (8.2) 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 Figure 8.3 where the feature points are marked by red dots. From Figure 8.3, we can clearly see that some feature points corresponding to the same physical area appear in both images.

Table 8.1. Surface type Tp based on the signs of Mean curvature (H) and Gaussian curvature (K).

Mean Curvature H	$K > 0$	$K = 0$	$K < 0$
$H < 0$	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 $T_p = 9$

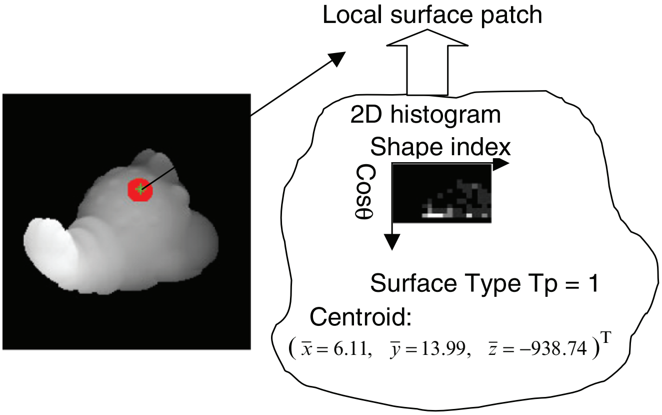
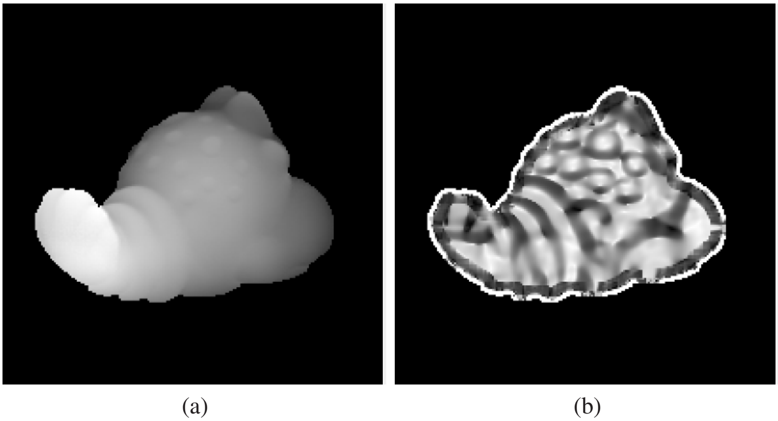


Fig. 8.1. Illustration of a local surface patch (LSP). Feature point P is marked by the asterisk and its neighbors N are marked by the dots. The surface type of the LSP is 1 based on Table 8.1.



(a) (b)

Fig. 8.2. (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.

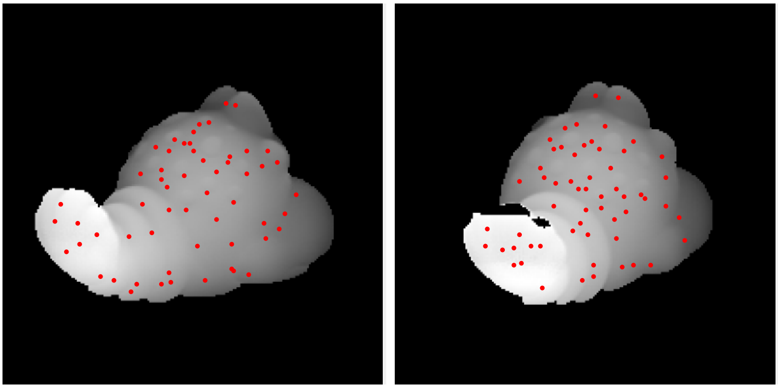


Fig. 8.3. Feature points location (·) in two range images, shown as gray scale images, of the same object taken at different viewpoints.

8.1.2 Hash Table Building

Considering the uncertainty of location of a feature point, we also calculate descriptors of local surface patches for neighbors of a feature point P. To speed up the retrieval of local surface patches, for each LSP we compute the mean ($\mu = \frac{1}{L}\sum_{l=1}^{L} S_i(p_l)$) and standard deviation ($\sigma^2 = \frac{1}{L-1}\sum_{l=1}^{L}(S_i(p_l) - \mu)^2$) of the shape index values in a neighborhood around the feature point where L is the number of points on the LSP under consideration such that the angle between the surface normal at the feature point and its neighboring points is small. p_l is the l^{th} point on the LSP. We use LSPs to index a hash table and insert into the corresponding hash bin the information about the model LSPs. Thus, 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 Figure 8.4, 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 receives.

8.1.3 Hypotheses Generation

Given a test range image, we extract feature points and get local surface patches. Then we calculate the mean and stand deviation of shape index, and cast votes to the hash table if the histogram dissimilarity falls below a preset threshold ϵ_2 and the surface type is the same. The dissimilarity between the two histograms is evaluated by equation (5.3).

After voting, we histogram all hash table entries and get models which received the top three highest votes. By casting votes, we not only know which models get higher votes, but also we know the potential corresponding local surface patch pairs. Note that a hash table entry may have multiple items, we choose the local surface patch from the database with minimum dissimilarity and the same surface type as the possible corresponding patch. We filter the possible corresponding pairs based on the geometric constraints (5.4). Figure 8.5 shows one example of partitioning corresponding pairs into groups. It shows the grouping results for an object imaged at two different viewpoints. Figure 8.5(a) shows the feature point extraction result for the test object.

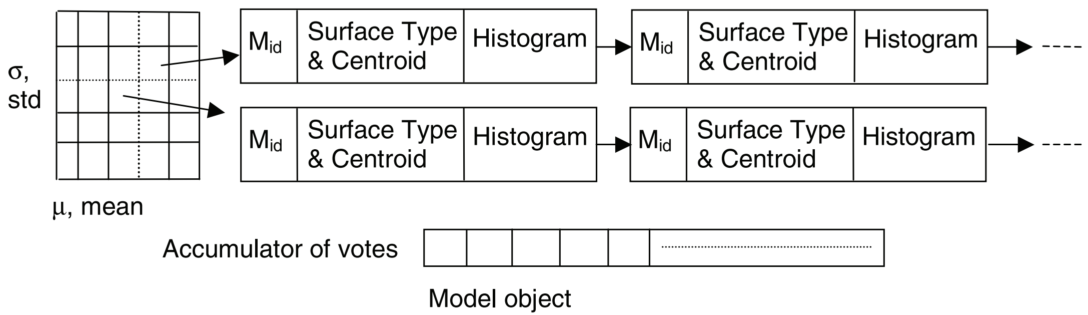


Fig. 8.4. Structure of the hash table. Every entry in the hash table has a linked list which saves information about the model LSPs and the accumulator records the number of votes that each model receives.

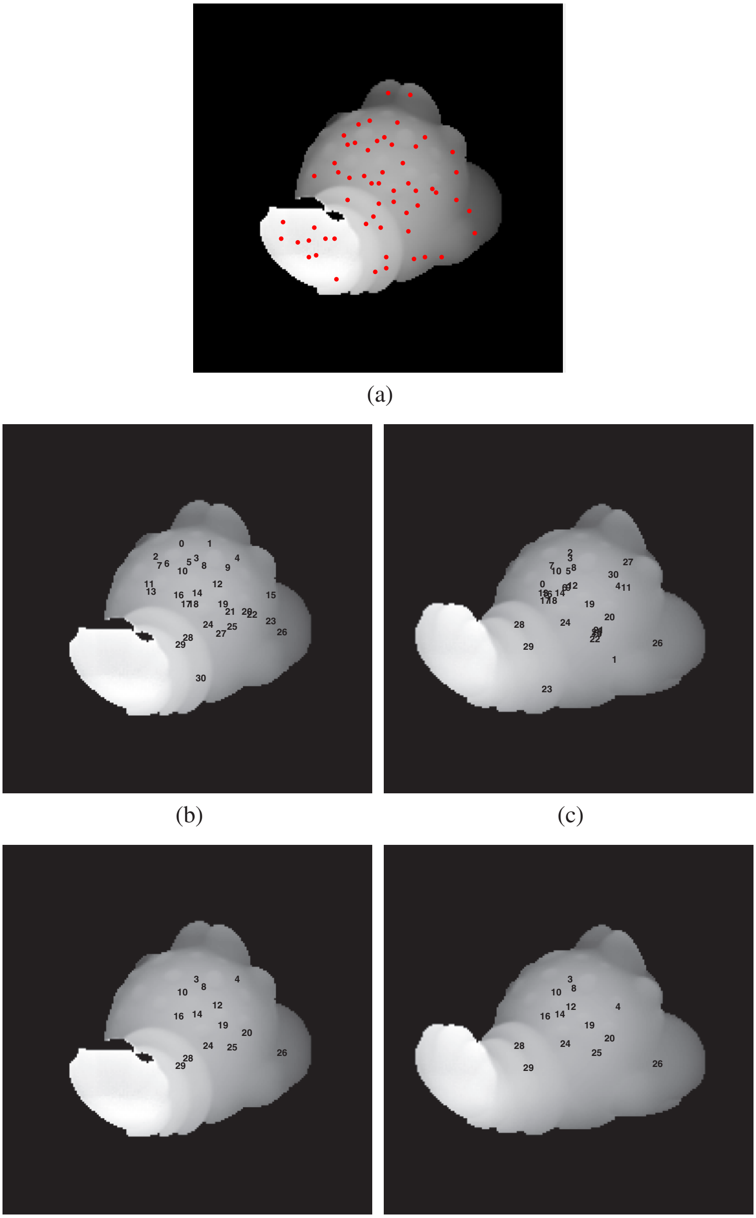
Comparing the local surface patches with LSPs on the model object and casting votes to the hash table, the initial corresponding LSP pairs are shown in Figure 8.5(b), in which every pair is represented by the same number superimposed on the test and model images. We observe that both the true and false corresponding pairs are found. After applying the simple geometric constraint (5.4), the filtered largest group is shown in Figure 8.5(c), in which the pairs satisfying the constraint (5.4) 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.

8.1.4 Verification

Given the corresponding LSPs between a model-test pair, the initial rigid transformation, which brings the model and test objects into a coarse alignment, can be estimated by minimizing the sum of the squares of theses errors ($\Sigma = \frac{1}{n}\sum_{l=1}^{n} |S_l - R*M_l - T|^2$) with respect to the rotation matrix R and the translation vector T. The rotation matrix and translation vector are computed by using the quaternion representation [76].

Given the estimate of initial rigid transformation, the purpose of iterative closest point (ICP) algorithm [63] is to determine if the match is good and to find a refined alignment between them. If the probe ear is really an instance of the gallery ear, the ICP algorithm will result in a good registration and a large number of corresponding points between gallery and probe ear surfaces will be found. Since ICP algorithm requires that the test data set is a subset of the model set, we use the modified ICP algorithm proposed by Zhang [70] 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 control points of the model object and their closest points of the test objecct. For each object in the model database, the control points are randomly selected and the modified ICP is applied to those points. We repeat the same procedure 15 times and choose the rigid transformation with the minimum root mean square (RMS) error. The object in the model database with the minimum RMS error is declared as the recognized object. In the



(a)

(b)

(c)

(d)

(e)

Fig. 8.5. 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 Figure 8.5(b) with matched LSPs after applying the geometric constraint (5.4). (e) The model object in Figure 8.5(c) with matched LSPs after applying the geometric constraint (5.4).

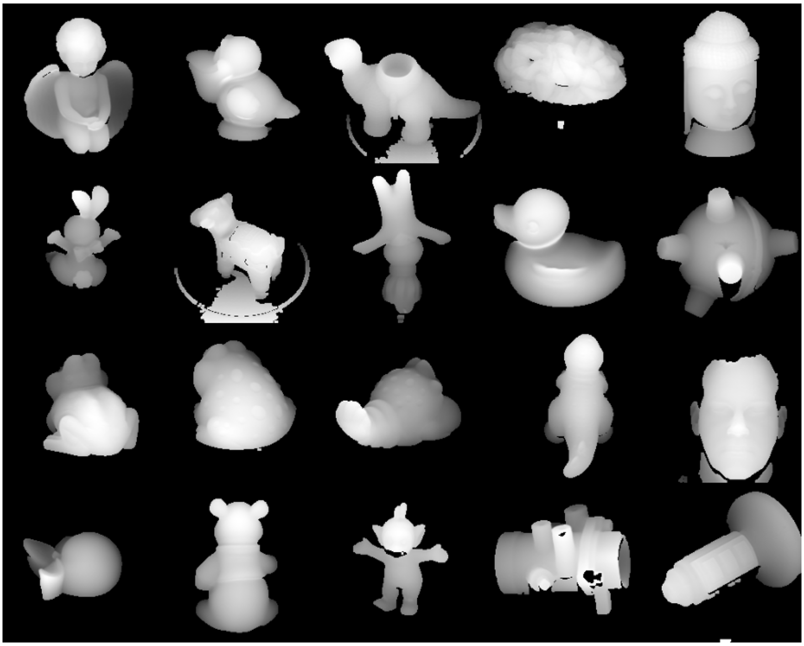


Fig. 8.6. The range images of objects in the model database. The object IDs (0 to 19) are labeled from left to right and top to bottom.

modified ICP algorithm, the speed bottleneck is the nearest neighbor search. Therefore, the kd-tree structure is used in the implementation.

8.1.5 Experimental Results

Data and Parameters

We use real range data collected by Ohio State University.[1] There are 20 objects in our database and the range image of the model objects are shown in Figure 8.6. 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 $\epsilon_H = \epsilon_K = 0.003$. Note that A, ϵ_1, ϵ_3 parameters are the same as in Chapter 5. The new parameters in this chapter are ϵ_2, ϵ_H, ϵ_K, α and β. The number of bins in the shape index axis is 17 and the number of bins in the other axis (cos θ) is 34. The total number of LSPs calculated in the model objects is about 34,000. The average size of local surface patch is 230 pixels. We apply our approach to the single-object and

[1] http://sampl.eng.ohio-state.edu/sampl/data/3DDB/RID/minolta/

two-object scenes. The model objects and scene objects are captured at two different viewpoints.

Single-Object Scenes

In this test case, we show the effectiveness of the voting scheme and the discriminating power of LSP 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 twenty objects in Figure 8.7. 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 8.2, which shows the candidate model ID and the corresponding RMS registration error. From Table 8.2, 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 Figure 8.8 (top figure) show test objects and their corresponding model objects before alignment. The images in Figure 8.8 (bottom figure) 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 (up to $45°$) are correctly handled. Since the proposed LSP representation consists of histogram of shape index and surface normal angle, they are invariant to rigid transformation. The experimental results verified the view-point invariance of the LSP representation.

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 were identified using the Z-buffer algorithm. Table 8.3 provides the objects included in the four

	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. 8.7. 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 8.2. Verification results for single-object scenes. 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 in millimeters (mm).

Test objects	Results (Top 3 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)

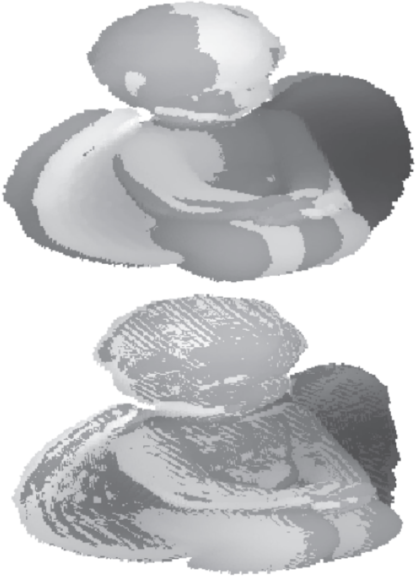


Fig. 8.8. Four examples of correctly recognized model-test pairs. Each row shows one example. The test objects are shaded light gray while the recognized model objects are shaded dark gray and overlaid on the test objects. The top figure shows the model and test objects before alignment, and the bottom one shows the 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 shown here, the rotation angle is 20.4° and the axis is $[0.0319, 0.9670, 0.2526]^T$.

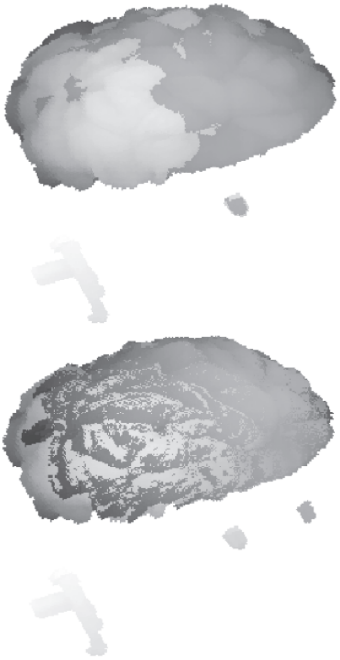


Fig. 8.8. Figure 8.8 Continued, Example 2. The rotation angle is 35.9° and the axis is $[-0.0304, -0.5714, -0.1660]^T$.

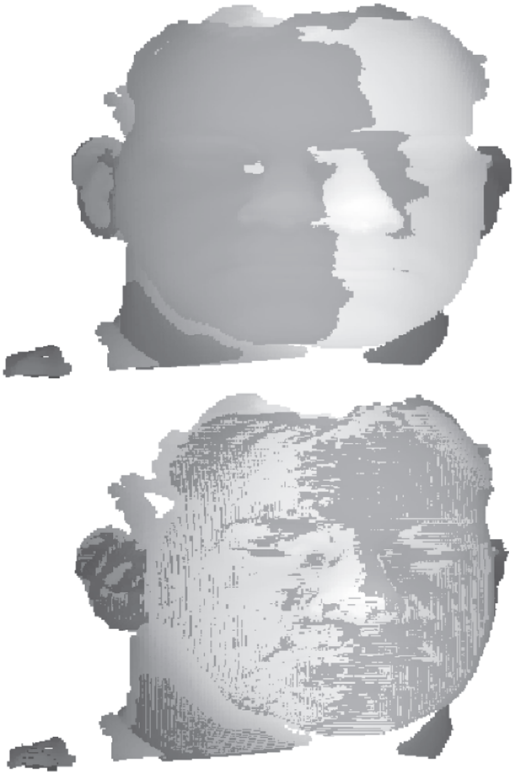


Fig. 8.8. Figure 8.8 Continued, Example 3. The rotation angle is 14.1° and the axis is $[0.0187, 0.2429, 0.0046]^T$.

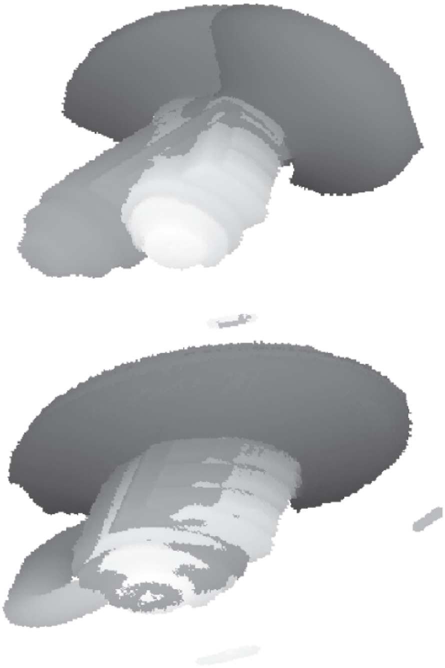


Fig. 8.8. Figure 8.8 Continued, Example 4. The rotation angle is 43.6° and the axis is $[0.6855, -0.6150, -1.3909]^T$.

Table 8.3. Voting and registration results for the four two-object scenes shown in Figure 8.9(a). 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 in millimeters (mm).

Test	Objects in the image	Voting and registration results for top 6 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)
Scnee 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)

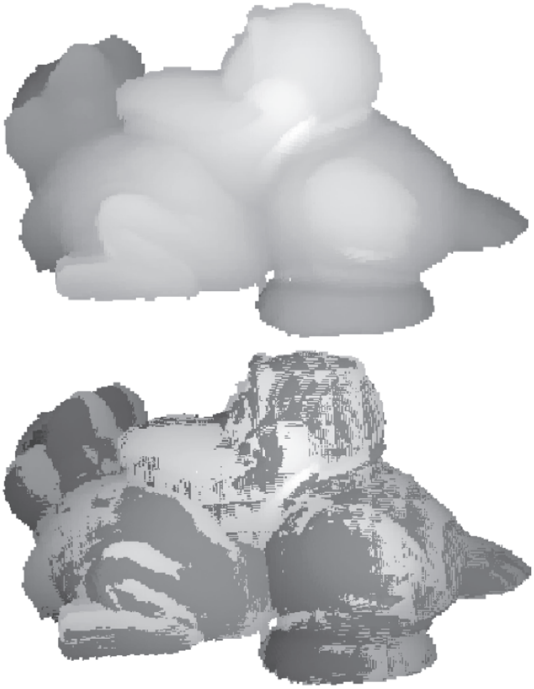


Fig. 8.9. Recognition results for the four two-object scenes. Example 1 is shown here. The test objects are shaded light gray while the recognized model objects are shaded dark gray. The top figure shows the range images of the four two-object scenes, and the bottom one shows the recognized model objects 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 Example 4 one object is missed.

Fig. 8.9. Figure 8.9 Continued, Example 2.

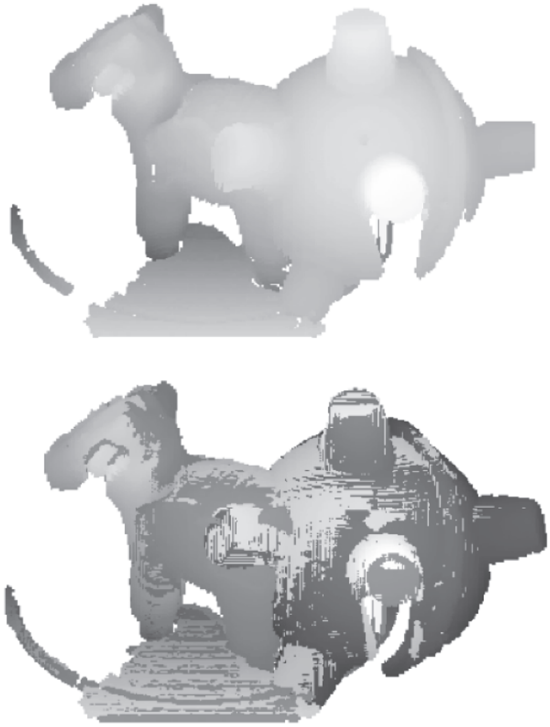


Fig. 8.9. Figure 8.9 Continued, Example 3.

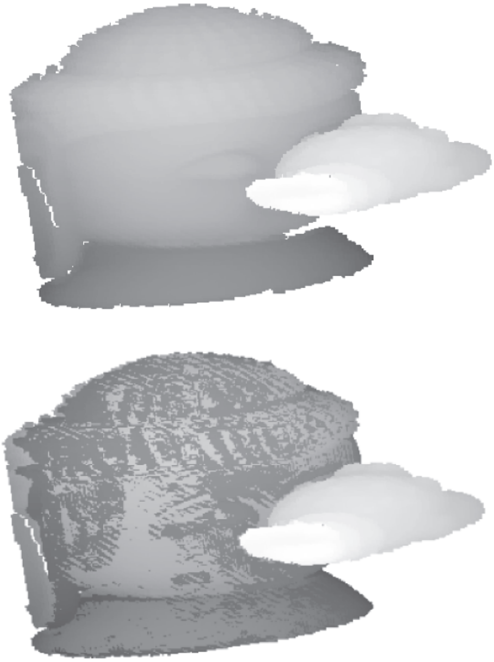


Fig. 8.9. Figure 8.9 Continued, Example 4.

scens and the voting and registration results 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 the vote it obtained was not ranked on the top. The four scenes are shown in Figure 8.9 (top figure). The recognition results are shown in Figure 8.9 (bottom figure). We observe that the recognized model objects are well aligned with the corresponding test objects.

8.1.6 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) [3] and the spherical spin image (SSI) [4] representations. We conducted the following experiments. We take 20 model objects, compute feature points as described in Section 8.1.1 and calculate the surface descriptors at those feature points and their neighbors. Given a test object, we calculate the surface descriptors 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 ($\mathcal{T}$) for recognizing a single object consists of three timings:

- find the nearest neighbors t_a;
- find the group of corresponding surface descriptors t_b;
- perform the verification t_c.

These timings, on a Linux machine with an *AMD Opteron* 1.8 GHz processor, are listed in Table 8.4. 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.

8.1.7 Discussions

From the experimental results, we observe that the integrated local surface descriptor (LSP) is really effective for surface representation and

Table 8.4. The timing in seconds for the three representations. LSP denotes the local surface patch descriptor; SI denotes the spin image [3]; SSI denotes the spherical spin image [4].

	t_a	t_b	t_c	$\mathcal{T}$
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

3D object recognition. For fast retrieval of surface descriptors, the generated LSPs for all models are indexed into a hash table. 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 aligning 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 3.89 (over SSI) to 4.37 (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.

8.2 Global-to-Local Non-Rigid Shape Registration

Registration of non-rigid shapes is an important issue in computer vision and it has drawn an increasing attention due to its wide range of applications related to recognition, tracking and retrieval. The shape registration problem can be stated as follows: Given two shapes, a model shape M and a target shape S, find the best transformation that assigns any point of M a corresponding point in S and minimizes the dissimilarity between the transformed shape $\hat{M}$ and S. Therefore, there are two problems to be resolved: the correspondence, and the transformation.

The non-rigid shape registration is much harder since it has more degrees of freedom than the rigid shape registration problem. Recently researchers have come up with different approaches to solve the non-rigid shape registration problem [72, 108–111]. Chui and Rangarajan [72] presented an optimization based approach, TPS-RPM (thin plate spline-robust point matching) algorithm, to jointly estimate the correspondence and non-rigid transformations between two point-based shapes. Belongie et al. [108] proposed a descriptor called *shape context* to find correspondences by minimizing the overall shape context distances and the TPS transformation is iteratively solved. Guo et al. [109] described a joint clustering and diffeomorphism estimation algorithm that can simultaneously estimate the correspondence and fit the diffeomorphism between the two point sets (a diffeomorphism is a invertible function that maps one differentiable manifold to another such that both the function and its inverse are smooth). Paragios et al. [110] introduced a simple and robust shape representation (distance functions) and a variational framework for global-to-local registration in which a linear motion model and a local deformation field are incrementally recovered. Zheng and Doermann [111] presented a relaxation labeling based point matching algorithm for aligning non-rigid shapes. The point matching is formulated as a graph matching problem to preserve local neighborhood structures in which the point is a node and neighboring points are connected by edges.

As compared to these approaches, we decompose the non-rigid shape registration into a two-step procedure:

- the global similarity transformation that brings the model shape and target shape into a coarse alignment;
- the local deformation that deforms the transformed model shape to the target shape [112].

For the first step, feature based registration is employed; in the second step the local deformation is formulated as an optimization problem to preserve the structure of the shape model.

8.2.1 Global Similarity Registration

The task of the global registration is to find a similarity transformation between M and S that includes three parameters (s, θ, T); a scale factor s, a rotation θ and a translation vector $T = (T_x, T_y)$.

$$S = s \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} M + \begin{bmatrix} T_x \\ T_y \end{bmatrix} \tag{8.3}$$

Once the corresponding pairs (M_i, S_i) are known, the three parameters can be estimated by minimizing the sum of square distance between the transformed M_i and S_i.

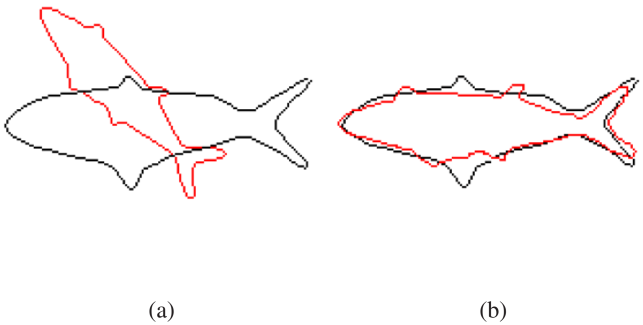
The shape context descriptor [108] is used to find the correspondence only. Considering the shape contexts of two point a and b, the cost C_{ab} of matching the two points is evaluated by χ^2 test statistic since shape contexts are histograms that can approximate probability distributions.

$$C_{ab} = C(a,b) = \frac{1}{2}\sum_{i=1}^{K} \frac{(h_a(i) - h_b(i))^2}{h_a(i) + h_b(i)} \tag{8.4}$$

where $h_a(i)$ and $h_b(i)$ denote the K-bin normalized histograms at points a and b respectively. Given the sets of costs $C(a,b)$ between pairs a on the model shape and b on the target shape, the optimal correspondence is found by minimizing the sum of individual matching costs. This is solved by a bipartite matching algorithm with the one-to-one matching constraint [108]. In our case, the configuration of the model shape is fixed and we resample the points from the contour of the model shape. Then we compute shape context descriptors to find the correspondences of the model shape in the target shape to calculate the global similarity transformation, while in [108] the non-linear TPS transformation is solved iteratively by warping the model shape to the target and recovering correspondences. Figure 8.10 shows one example of the global similarity registration result; Figure 8.10 shows the model shape and the target shape before alignment; Figure 8.10 (a) shows the transformed model shape is superimposed on the target shape after alignment. The three parameters are $s = 1.28, \theta = 45.3°, T = (-36.4, 77.9)$ pixels. We observe that the model shape is roughly aligned with the target shape.

8.2.2 Local Deformation

After the model shape is brought into coarse alignment with the target shape through the global registration, it needs to deform to the target



(a) (b)

Fig. 8.10. Global similarity registration. The model shape is shown in red and the target shape is shown in black). (a) Initial condition. (b) Global similarity registration result.

shape with the structure of the shape model preserved. The local deformation is model by the thin plate spline (TPS) transformation that is widely used in shape matching. As described in Section 3.2.3, we can achieve this task by minimizing the proposed cost function,

$$\begin{aligned} E(\mathbf{x},\mathbf{y}) &= E_{img}(\mathbf{x},\mathbf{y}) + \gamma E_D(\mathbf{x},\mathbf{y}) \\ &= \sum_{i=1}^{n} g(|\nabla I_m(x_i,y_i)|) + \frac{1}{2}\gamma(\mathbf{x}^T K\mathbf{x} + \mathbf{y}^T K\mathbf{y}). \quad (8.5) \end{aligned}$$

As derived in Chapter 3.2.3, the coordinates of $(\mathbf{x},\mathbf{y})$ can be iteratively updated by

$$\begin{aligned} \mathbf{x}_t &= (\gamma K + \alpha I)^{-1}\Big(\alpha\mathbf{x}_{t-1} + \\ &\quad \sum_{i=1}^{n} \frac{1}{(1+|\nabla I_{step}^{t-1}(x_i,y_i)|)^2} \frac{\partial|\nabla I_{step}^{t-1}(x_i,y_i)|}{\partial\mathbf{x}}\Big) \\ \mathbf{y}_t &= (\gamma K + \alpha I)^{-1}\Big(\alpha\mathbf{y}_{t-1} + \\ &\quad \sum_{i=1}^{n} \frac{1}{(1+|\nabla I_{step}^{t-1}(x_i,y_i)|)^2} \frac{\partial|\nabla I_{step}^{t-1}(x_i,y_i)|}{\partial\mathbf{y}}\Big). \quad (8.6) \end{aligned}$$

Figure 8.11 shows an example of deforming the model shape (dude) to the target shape at iteration 0, 25, 50, 75, 100, and 125. We observe that the model shape is brought more closer to the target shape during the deformation.

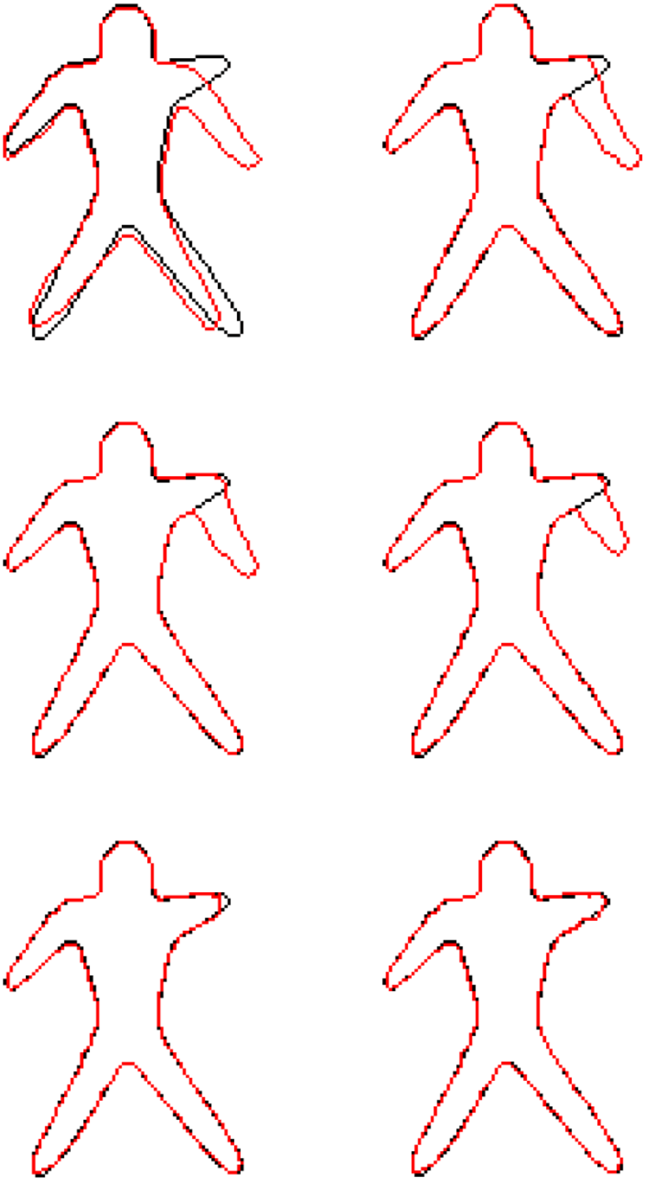


Fig. 8.11. Deformation of the transformed model shape to the target shape at iteration 0, 25, 50, 75, 100, and 125. The model shape is show in red and the target shape is shown in black.

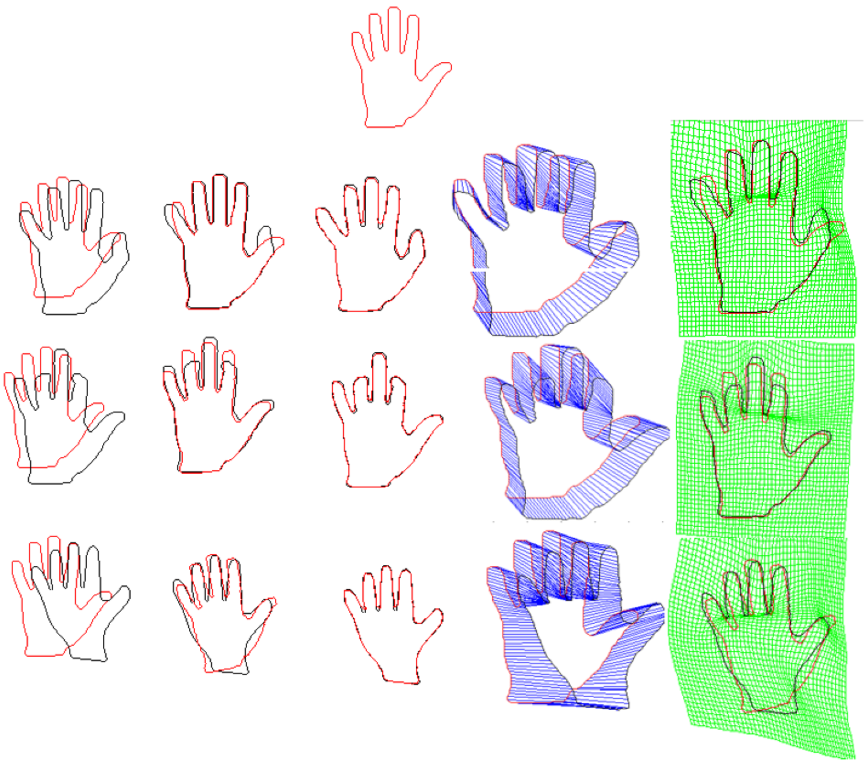


Fig. 8.12. Global-to-local registration results for the shape of hand. The model shape is shown in red and the target shape is shown in black. The first row shows the model shape. The second, third, and fourth rows show examples with different initial conditions. For rows 2–4, the first column shows the initial condition; the second column shows global registration; the third column shows local deformation; the fourth column shows correspondences, and the fifth column shows grid deformation.

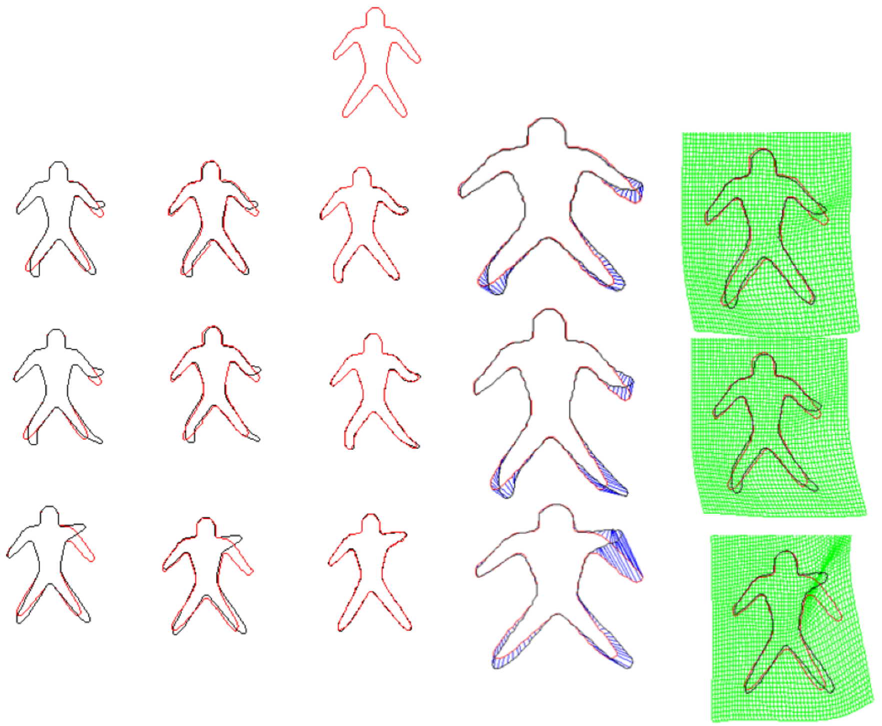


Fig. 8.13. Global-to-local registration results for the shape of dude. The model shape is shown in red and the target shape is shown in black. The first row shows the model shape. The second, third, and fourth rows show examples with different initial conditions. For rows 2–4, the first column shows the initial condition; the second column shows global registration; the third column shows local deformation; the fourth column shows correspondences, and the fifth column shows grid deformation.

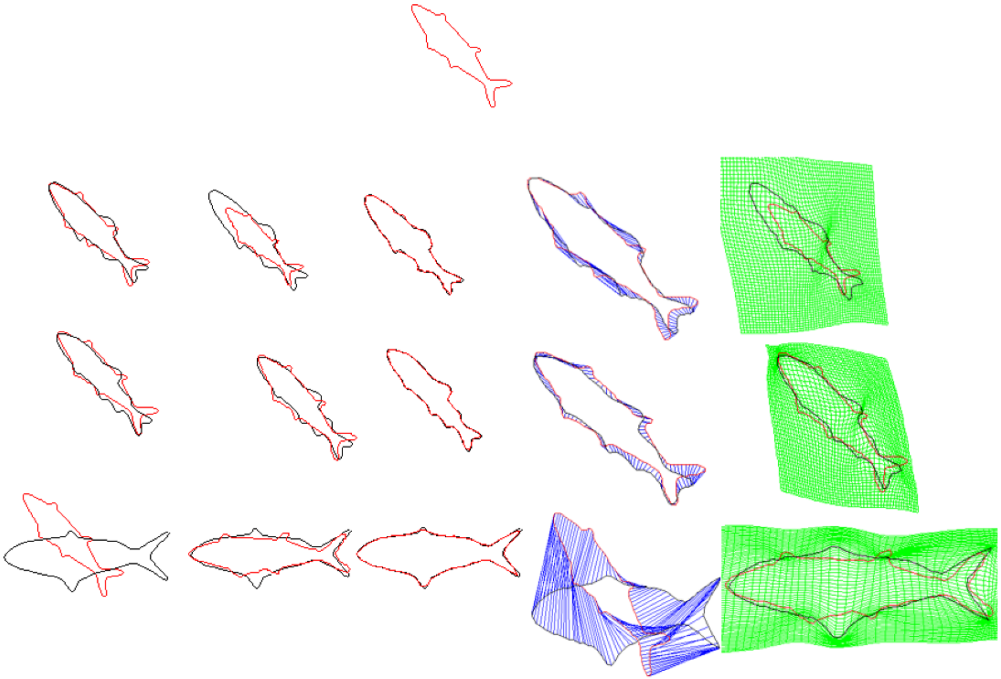


Fig. 8.14. Global-to-local registration results for the shape of fish. The model shape is shown in red and the target shape is shown in black. The first row shows the model shape. The second, third, and fourth rows show examples with different initial conditions. For rows 2–4, the first column shows the initial condition; the second column shows global registration; the third column shows local deformation; the fourth column shows correspondences, and the fifth column shows grid deformation.

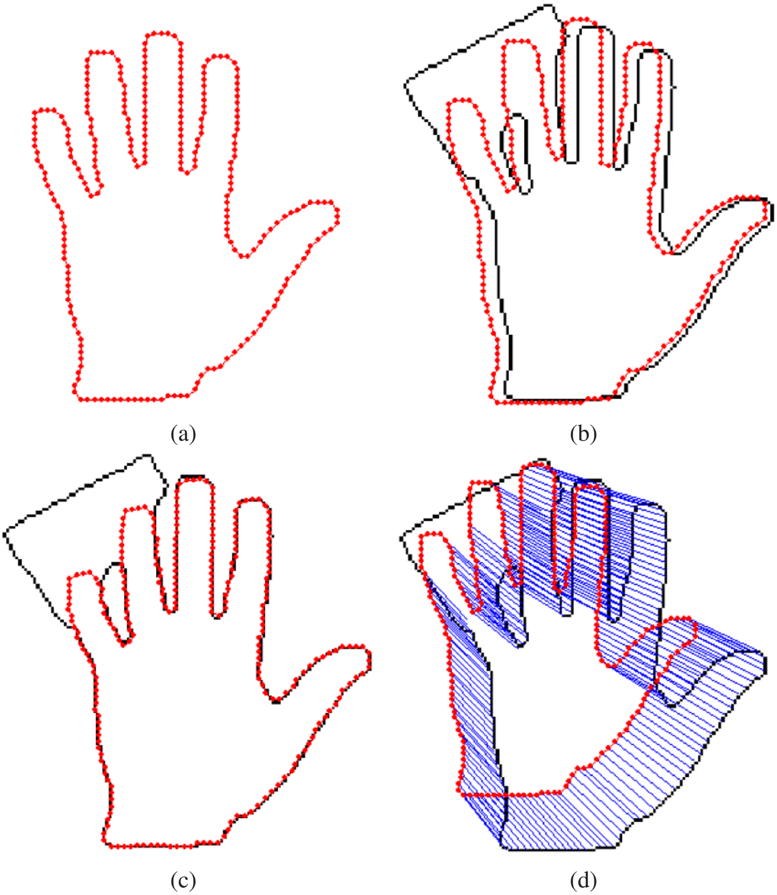


Fig. 8.15. An example of the registration of the occluded hand shape. The model shape is shown with dotted line and the target shape is shown with solid line. (a) Model shape. (b) Global registration. (c) Local deformation driven by the proposed optimization formulation. (d) Correspondences. Note that the target hand shape is occluded.

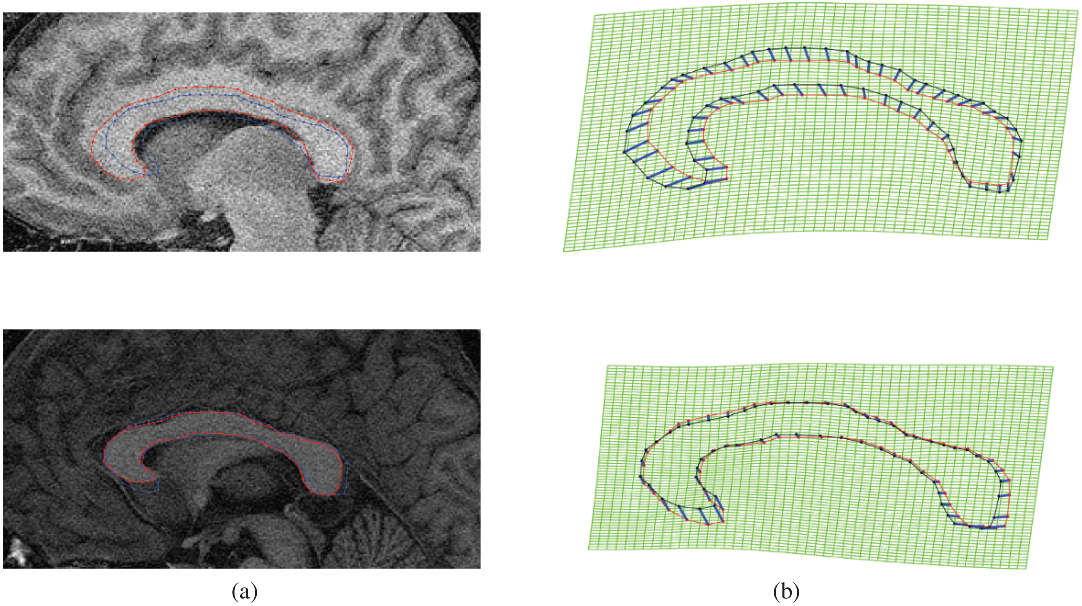


Fig. 8.16. Two example of registration of the callosum shape in MRI images. (a) The global registration result (blue line) and the local deformation result (red line). (b) Grid deformation and the correspondences connected by the blue lines. Images were obtained from Dr. Song Wang (University of South Carolina, Columbia, SC).

8.2.3 Experimental Results on Binary Images

We used the shape database collected by Brown University.[2] Figures 8.12, 8.13 and 8.14 show the non-rigid shape global-to-local registration results on three shapes of the hand, dude, and fish, respectively. The regularization term γ for shapes of the hand, dude and fish is 20, 100, 25, respectively. In the three figures, the first row shows the model shape and the columns below the model shape show the model shape and target shape before the global registration, the transformed model shape overlaid on the target shape after the global alignment, the registration results after local shape deformation, the one-to-one correspondence connected by the blue lines, and the warped regular grids by the local deformation computed from equation (3.18). From Figures 8.12, 8.13 and 8.14, we see that the model shape is successfully deformed to the target shape and the one-to-one correspondence is established.

Figure 8.15 shows the registration of the model shape with the occluded target shape. Since the shape model encodes the prior information about the shape topology, the registration of the occluded shape can be handled. We observe that the model shape is driven towards the target shape by the optimization formulation.

Quantitative evaluation: The accuracy of the proposed approach is quantified by the average Euclidean distance between the points in the transformed model shape and the correspondences in the target shape. The average distances after the local deformation for the shapes of hand, dude and fish are 0.59, 0.45, and 0.61 pixels, respectively; while the distances after global alignment are 1.84, 1.80, and 2.49 pixels. We observe that the proposed global-to-local non-rigid shape registration algorithm brings the model shape and the target shape into a good alignment.

8.2.4 Experimental Results on Gray-Level Images

The proposed formulation is also valid for gray-levle images as well as binary images. Figure 8.16 shows the registration of callosum in a human brain MRI image. Figure 8.16(a) shows the global registration result by the blue line and local deformation result by the red line; Figure 8.16(b) shows the grid deformation in which the correspondences are connected by the blue lines.

[2] http://www.lems.brown.edu/vision/

8.2.5 Discussions

We have proposed a novel global-to-local procedure for aligning non-rigid shapes. Since the structure of the model shape should be preserved under the deformation, the bending energy is incorporated into the optimization formulation as a regularization term to penalize the large shape deformation. The optimization procedure drives the initial global registration towards the target shape with the structure of the model shape preserved and finally finds one-to-one correspondence between the model shape and target shape. Experimental results on three non-rigid shapes show the effectiveness of the proposed approach.

8.3 Conclusions

In this chapter, we presented two general applications in the field of computer vision and pattern recognition. First, we presented 3D object recognition results on general 3D range images. We use the local surface patch (LSP) representation that has been used for ear recognition. The only difference in LSP representation used for ears and general 3D objects is the number of surface types. In 3D ear recognition, the surface takes three different types (concave, saddle, and convex). For the general 3D object recognition, the surface takes eight different types (peak, ridge, saddle ridge, flat, minimal, pit, valley, and saddle valley).

We also presented a global-to-local registration scheme for the precise alignment of a variety of images collected from various applications. These images included hand images, dude images, fish images, and MRI images.