

A Local Density Shape Context Algorithm for Point Pattern Matching in Three Dimensional Space

You ZHOU, Panpan CUI, Yizhang WANG, Limin WANG, Sen YANG, Xu ZHOU

Abstract: Three dimensional space point pattern matching technology shows significant usage in many scientific fields. It is a great challenge to match pairwise with rigid transformation in three dimensional space. In this paper, we propose an effect of Local Density Shape Context algorithm (LDSC). In LDSC, the point local density is firstly used for cutting down the negative impacting on extracting the feature descriptor. And the optimization of pairwise matching is firstly used in LDSC for improving the effectiveness. To demonstrate the performance of LDSC, we conduct experiments on synthetic datasets and real world datasets. The experimental results indicate that LDSC outperforms the three compared classical methods in most cases. LDSC is robust to outliers and noise.

Keywords: point local density; point pattern matching; shape context

1 INTRODUCTION

Point pattern matching plays important roles in many scientific and application fields, such as fingerprint recognition and iris recognition [1]. It is also an indispensable technology in computer vision, image matching and moving object recognition [2]. The goal of the point pattern matching problem is to find a spatial optimal correspondence from images [3]. Feature points are extracted from images and used for finding matching pattern. Point pattern matching is a NP hard problem. And there is not available prior information between point patterns. Outliers and noise are included in point sets within the extracting procedure which increases the complexity of solving the problem. The study of point pattern matching in three dimension space has become a challenging task [4].

The rest of this paper is organized as follows. In Section 2, we review related work. In Section 3, we present the proposed novel Local Density Shape Context algorithm (LDSC). We report experimental results with discussions in Section 4. Finally, we conclude the paper in Section 5.

2 RELATED WORK

Aiming at matching point pattern with different rigid transformation, many matching algorithms have been proposed in different methodologies and used in various fields. The existing algorithms are divided into five types and discussed in the following subsections.

2.1 Spectra Context Based Algorithms

Spectra Context algorithm is firstly introduced in [5], they used the matching rate of point pairwise to construct a proximity matrix, and then get final matching result by using singular value decomposition of the proximity matrix. Spectra Context algorithm is easy to implement, but it cannot work well when large angle rotation exists in point pattern. Shapiro et al. put forward to get the proximity matrix inside the point sets and get better result [6]. Carcassoni et al. combined the proximity matrix with EM algorithm for improvement [7]. These improved

methods enhance the robustness of Spectra Context algorithm. However, this kind of algorithms can only be used for two dimension point pattern matching and cannot work well in three dimensional situation.

2.2 Biological Model Based Algorithms

A kind of biological model based algorithms was proposed in the last decade and such as Particle Swarm Optimization algorithm [8, 9] and Genetic Algorithm [10] were used to find the corresponding relationship between two point sets. Ansari et al. proposed a Genetic Algorithm based method, the correspondence between point sets is regarded as strings, the strings are updated iteratively according to genetic options (selection, cross, mutation) [10]. Particle Swarm Optimization model is also used for matching point pattern, the solution space is directly represented by transform parameters [8]. This kind of algorithm can be applied to the n-dimension point pattern matching, but with the dimension increasing, it is easy to fall into local optimal value. High computational cost is needed, which means time consumption algorithm.

2.3 Iterative Optimization Algorithms

Iterated Closest Point (ICP) is the best known point pattern matching algorithm [11], it assigns a correspondence based on the nearest neighbor relationships, then uses the estimated transformation to obtain new correspondence iteratively until algorithm converges. ICP is sensitive to dispersed distribution of points, and the initial position of two point sets is required close to each other. To overcome the shortcomings of ICP, improved algorithms were proposed. For example, LM-ICP introduces nonlinear optimizer in iteration [12]. TPS-RPM is an algorithm based on soft assignments and deterministic annealing [13]. This kind of algorithms is sensitive to the initial position.

2.4 Shape Context Based Algorithms

The Shape Context algorithm is firstly put forward by Belongie et al. [14]. They used the spatial distribution of point sets as features of each point, but it is not good for

dealing with large angle rotation between two point sets. However, this kind of algorithms is designed for two dimensional problem and cannot work well in three dimension situation [15, 16]. Positive Bipartite Graph Shape Context (PSC) algorithm was proposed especially for three dimensional point pattern matching problem [17]. The space division and feature descriptor for three dimensional space were firstly introduced in PSC. However, PSC does not deal well with low density points and sensitive to outliers. The performance of PSC declines greatly because of the influence of low density points when extracting point features.

2.5 Other Algorithms

Some other kind of algorithms was proposed in the last decade, such as methods based on quadratic programming [18, 19], Markov random field [20, 21] and Geometric invariant features [22, 23] etc.

In this paper, we propose a novel Local Density Shape Context algorithm based on the local points density for three dimensional point pattern matching. The local point density and optimization of pairwise matching are firstly proposed in LDSC. LDSC is robust to outliers and noise, it is also a fast and efficient algorithm. The detailed description of LDSC is introduced in section 3.

3 DENSITY SHAPE CONTEXT ALGORITHM

3.1 Problem Description

A general definition of pattern matching about two point sets is defined as Definition 1. The point pattern matching algorithm is to find a transformation T between P and Q .

Definition 1

Given two point sets P and Q , where $P = \{P_i : P_i \in R^N, i = 1, 2, \dots, n\}$ and $Q = \{Q_j : Q_j \in R^N, j = 1, 2, \dots, m\}$, $m \neq n$. Point pattern matching algorithm is to find an assignment $T : P_k \rightarrow Q_k$, where $P_k \subseteq P$ and $Q_k \subseteq Q$, T is similarity transformation. The best matching result is to ensure the matching error minimized between $T(P_k)$ and Q_k .

3.2 The Detailed Steps of LDSC Algorithm

LDSC is proposed for robust matching point pattern in three dimensional space. It uses local point density and optimization of pairwise matching. In order to understand the steps of the LDSC easily, the psuedocode of our method is shown in Algorithm 1.

Algorithm 1: The Detailed Steps of LDSC Algorithm

Input: two point sets P and Q .

Output: pairwise relation $rel = \{(p_i, q_j)_v : v = 1, 2, \dots, k\}$.

- 1: eliminate low density points in P and Q (see 3.3).
 - 2: calculate feature descriptors of each point in P and Q using shape context descriptor (see 3.4).
 - 3: calculate initial pairwise relationship rel_i using Kuhn-Munkres algorithm [24] according to feature similarity (see 3.5).
-

4: $i = 0$.

5: **Repeat**

6: $i = i + 1$

7: calculate transformation matrix U_i using Kabsch algorithm [25] according to rel_i ;

8: transform Q_i , where $Q_{i+1} = Q_i * U_i$.

9: calculate $Qvalue_i$ using Eq. (17).

10: remove pairwise $(p_i, q_j)_v$ with maximum Euclidean distance in rel_i , calculate $Qvalue_{i+1}$.

11: **Until** $Qvalue_i < Qvalue_{i+1}$.

13: transform Q_1 according to rel_i and optimize solution (see 3.6), return rel_i

3.3 Eliminate Low Density Point

Low density point negatively impacts point feature descriptor while using shape context descriptor, especially it is redundant point. Low density points are eliminated from point set in LDSC. Given a n points set P , we use L_i as the variable of low density point, which is defined in Eq. (1), where p_i is low density point when L_i is 1, ρ_i is the local density of p_i and defined in Eq. (2), d_c is the cutoff distance and defined in Eq. (3). In these equations, where d_{ij} is the Euclidean distance of p_i and p_j , $d_{ij} = \|p_i - p_j\|$, $\chi(x) = 1$ if $x < 0$ and $\chi(x) = 0$ otherwise, pct is a user-defined parameter used to determine the size of d_c , we define $S_{ij} = \|p_i - p_j\|$, $i \leq j$ and arrange the elements of S in a descending order to form a vector $s = (s_1, s_2, \dots, s_{n*(n-1)/2})$, and $\rho_{pct*n*(n-1)/2}$ is calculated in the same way.

$$L_i = \begin{cases} 1, & \rho_i < \rho_{pct*n*(n-1)/2} \\ 0, & otherwise \end{cases} \quad (1)$$

$$\rho_i = \sum_{j=1}^n \chi(d_{ij} - d_c), i \neq j \quad (2)$$

$$d_c = s_{pct*n*(n-1)/2} \quad (3)$$

3.4 The Shape Context Descriptor

In LDSC, the shape context descriptor is used to calculate features of each point. For any point p_i , p_i is moved to the origin, the spatial distribution of other points p_j is used as the feature of point p_i . The detailed processes are shown in the following subsection. The method of calculating descriptor is the same for P and Q .

3.4.1 Translation

For any point $p_i \in P$, in order to facilitate to determine the spatial distribution of the remaining points in point set P , we first translate the point set P and ensure that p_i is translated to the origin. The coordinate of other point p_j is calculated by Eq. (4).

$$P_j = P_j - P_i, i \neq j \quad (4)$$

3.4.2 Rotation

In three dimensional space, three necessary points are needed to uniquely determine the rotation transformation. For any point p_i , the other two points are necessary to be chosen, the relative position of these two points does not change when the point set rotates. We use M_i and C_i as these two points for point p_i , the definitions are shown in Eq. (5) and Eq. (6), where μ is a user defined parameter which influences the robust ability of outlier, $\mu = 2$ is used in this paper. The coordinates of p_i and p_j are updated by Eq. (4).

$$M_i = \frac{\sum_{i=1}^m p_i}{m} \quad (5)$$

$$C_i = \sum_{j=1}^m \frac{p_j}{|p_j|^\mu}, i \neq j \quad (6)$$

The transformations are taken by calculating the angle ∂ between vector $p_i M_i$ and XOY plane according to Eq. (7). We rotate clockwise ∂ degrees around X-axis to put M_i on XOY plane. β is the angle between $p_i M_i$ and X-axis right direction which is defined in Eq. (8). We rotate P clockwise with β degrees around Z-axis and put M_i on X-axis. According to Eq. (9), we calculate angle γ between $p_i C_i$ and XOY plane, then we rotate P with γ degrees around Z-axis and put C_i on XOY plane.

$$\partial = \arctan \frac{M_{iz}}{M_{iy}}, \text{ if } M_{iy} < 0, \text{ set } \partial = \partial + \pi \quad (7)$$

$$\beta = \arctan \frac{M_{iy}}{M_{ix}}, \text{ if } M_{ix} < 0, \text{ set } \beta = \beta + \pi \quad (8)$$

$$\gamma = \arctan \frac{C_{iz}}{C_{iy}}, \text{ if } C_{ix} < 0, \text{ set } \gamma = \gamma + \pi \quad (9)$$

3.4.3 Three Dimensional Space Division

We divide three dimensional space into multiple regions by using two parameters D_plane and $D_spheres$. The XOY plane is divided into isometric sectors regions, where $D_plane = 30^\circ$ in this paper as shown in Fig. 1a. Then we extend all sectors vertically to the entire spherical area as shown in Fig. 1b, $D_spheres$ is the number of spheres and $D_spheres = 6$ in this paper. The feature descriptor of each point p_i is a 144 dimensions vector.

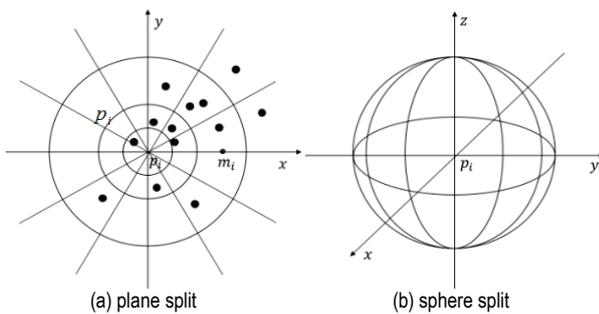


Figure 1 Three Dimensional Space

In order to get the values of feature descriptor in each dimension, we calculate the number of points in each region. For each point p_j , $\mathcal{G}_{p_i p_j}$ is the angle between $p_i p_j$ and XOY plane based on Eq. (10). We use $R(p_j)$ to calculate the regions of point p_j which is defined in Eq. (11). $f(x)$ is used to calculate the value of point p_j in which is located spherical area as shown in Eq. (12), where $[x]$ is downwards. $g(x)$ is used to calculate the value of point p_j in which are located regions sectors as shown in Eq. (13). $Avg(p_i)$ is the average Euclidean distance between p_i and other points p_j which is defined in Eq. (14).

$$\mathcal{G}_{p_i p_j} = \arctan \frac{p_{jy}}{p_{jx}} + \begin{cases} \pi, (p_{jx} \leq 0 \ \& \ p_{jy} \geq 0) \\ 2\pi, (p_{jx} \geq 0 \ \& \ p_{jy} \leq 0) \\ 0, (otherwise) \end{cases} \quad (10)$$

$$R(p_j) = \begin{cases} f\left(\frac{2 * \|p_i p_j\|}{Avg(p_i)}\right) + g\left(\frac{\mathcal{G}_{p_i p_j}}{D_plane}\right), \text{ if } p_{iz} \geq 0 \\ f\left(\frac{2 * \|p_i p_j\|}{Avg(p_i)}\right) + g\left(\frac{\mathcal{G}_{p_i p_j}}{D_plane}\right) + 72, \text{ otherwise} \end{cases} \quad (11)$$

$$f(x) = \begin{cases} \pi / D_plane * [x], \text{ if } x < D_spheres \\ \pi / D_plane * (D_spheres - 1), \text{ otherwise} \end{cases} \quad (12)$$

$$g(x) = \begin{cases} x, \text{ if } x = [x] \\ x + 1, \text{ otherwise} \end{cases} \quad (13)$$

$$Avg(p_i) = \frac{\sum_{j=1}^m \|p_i p_j\|}{m}, i \neq j \quad (14)$$

The similarity of two points p_i and q_j is defined in Eq. (15), where K is the number of divided regions and 144 in this paper, $O_k(x)$ is the value of the k^{th} region of point x . The smaller w_{ij} means more similarity between p_i and q_j .

$$w_{ij} = w(p_i, q_j) = \sum_{k=1}^K [O_k(p_i) - O_k(q_j)]^2 \quad (15)$$

3.5 Point Pairwise Matching

A similarity matrix between P and Q can be calculated by w_{ij} . To find the best matching result is equivalent to finding the best matching relation, which can guarantee S is the minimum value shown in Formula (16), q_j is the corresponding point of p_i .

$$S = \sum_{i=1}^n w(p_i, q_j) \quad (16)$$

For the convenience of solving the problem, we assume that edge $E(p_i, q_j)$ exists between any pairwise in P and Q , and w_{ij} is the weight of $E(p_i, q_j)$, so the problem is degenerated to the minimum weighting of a bipartite graph. For the minimum weight matching, we use Kuhn-Munkres [24] algorithm to find the point pattern matching result.

3.6 Point Pairwise Matching Optimization

We use maximal *Qvalue* as the best matching result. The equation of *Qvalue* is Eq. (17), where d is the number of pairwise in rel_i , $|x|$ is the number of points in set x , *RMSD* is root mean square deviation and defined in Eq. (18), where s is scale, r is rotation, t is translation. The number of pairwise points and *RMSD* are taken into account in *Qvalue*, which means good result with bigger d and smaller *RMSD*.

$$Qvalue = \frac{d^2}{1 + RMSD * |P| * |Q|} \tag{17}$$

$$RMSD = \sqrt{\frac{\sum_{i=1}^d ||s \cdot r \cdot p_i + t - q_i||^2}{d}} \tag{18}$$

In each iterative step, we calculate Euclidean distance of rel_i pairwise and arrange in ascending order. The pairwise with maximal Euclidean distance is removed or does not depend on *Qvalue*. The optimization is trying to find the maximal *Qvalue*. After iterative procedure, we get a point pattern matching result rel_i and maximal *Qvalue*.

4 EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we use synthetic datasets and real world datasets to assess the effectiveness of the proposed LDSC. We take three algorithms as baselines: PSC [15], ICP [9] and TPS-RPM [11]. We use Accuracy as the evaluation criterium and describe it in subsection 4.1. We give a detailed method of synthetic datasets generation. The accuracy results are shown in figures and tables in subsections 4.2 and 4.3. We conducted these experiments on the same device (RAM 8G, NVIDIA GeForce GTX 1050 Ti).

4.1 Algorithm Evaluation Criteria

In all experiments, the target point set Q is generated from template point set P by rigid transformation. The ground truth of matching pairwise is known in advance. The Accuracy is defined as Eq. (19), where the $Number(x)$ is a function for calculating pairwise number of set x , $SetG$ is a pairwise set of ground truth, $SetM$ is an algorithm output pairwise set, \cap is intersection.

$$Accuracy = \frac{Number(SetM \cap SetG)}{Number(SetG)} \tag{19}$$

4.2 Synthetic Datasets

We use three synthetic datasets cases to evaluate the performance of four mentioned algorithms. For three cases, the template point sets of each datasets are generated with the same pattern. The noise and outlier are generated following the corresponding parameters and are described in each case part. Case1 and Case2 are used for

illustrating noise measurement of four algorithms. Case3 is used for illustrating outlier measurement.

Given two point sets P and Q , P is the template point set, Q is the target point set and generated by transformation of P . The transformation includes translation, rotation and scale. The number of points of P and Q is the same and take 100. Each point is randomly generated in three axis. The range of values in each axis is $[-100, 100]$. The parameters of transformation are $t_x, t_y, t_z, s, r_x, r_y$ and r_z, t_x, t_y and t_z are the translation in x -axis, y -axis and z -axis separately. The range of translation is $[-100, 100]$. s is scale factor. The range of s is $[0.5, 3]$. r_x, r_y and r_z are three rotation angles in three axes separately. The range of rotation is $[-180^\circ, 180^\circ]$. In each case, we generate 100 synthetic datasets and demonstrate the average results.

4.2.1 Case1: Different Noise Percentage Experiments

In this case, for the noise percentage of point set Q is chosen different value from 0 to 100 percentage. We randomly choose percentage points in Q and add noise to these points. The perturb range of noise is $[-5, 5]$ in three axes. The result is shown in Fig. 2, where the column is accuracy and the row is noise percentage. LDSC and PSC achieve good results on increasing noise percentage. ICP and TPS-RPM cannot work well for point pattern matching in three dimensional space. The performance of LDSC is almost a slight decline and superior to PSC. It proves that LDSC algorithm has better ability to resist noise than other algorithms.

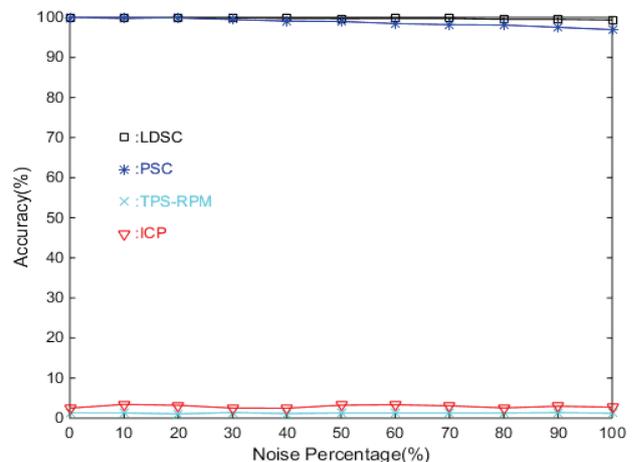


Figure 2 Accuracy of algorithms on different noise percentage datasets

4.2.2 Case 2: Different Noise Level Factor Experiments

In this case, noise is added to all points in point set Q . The noise measurement is decided by the noise level factor. We add noise in three axis, the range of noise is $[-f * 1.5, f * 1.5]$, where f is noise level factor. The range of f value is taken as $[0, 10]$ in this case. The average accuracy of four algorithms is shown in Fig. 3, where the column is accuracy and the row is noise level factor. This case indicates that LDSC algorithm works better than other algorithms.

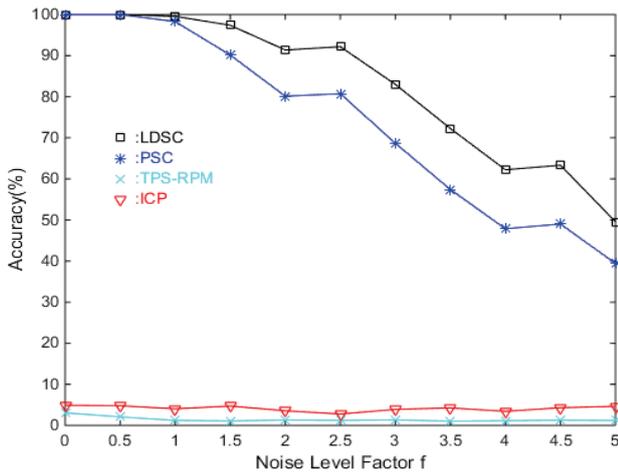


Figure 3 Accuracy of algorithms on different noise level factor datasets

4.2.3 Case 3: Different Outlier Percentage Experiments

In this case, an amount of outliers is put into target point set Q . Therefore, the point number of P and Q is different. Outliers are randomly generated, the scopes of three coordinates are chosen in $[-200, 200]$. Outlier percentage is the number of outliers divides number of points. The relationship between n and m is shown as Eq. (20), where n is point number of P and m is point number of Q . The average accuracy of four algorithms is shown in Fig. 4, where the column is accuracy and the row is outlier percentage. These experimental results demonstrate that LDSC could get more than 60 correct matching rates even if there are 50 percentage outliers. LDSC is more robust on outlier than other algorithms.

$$m = n * (1 + outlier\ percentage) \tag{20}$$

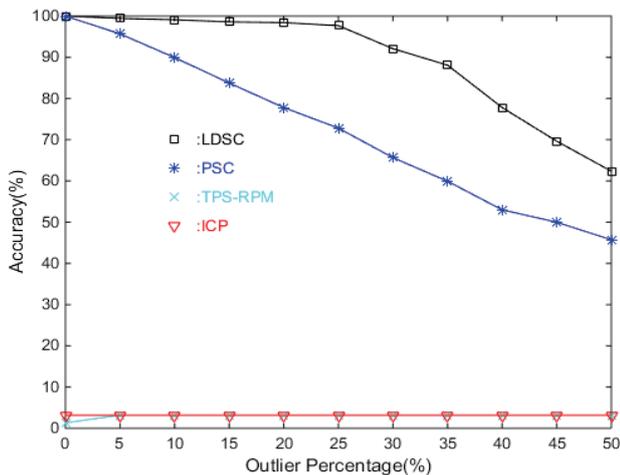


Figure 4 Accuracy of algorithms on different outlier percentage datasets

4.3 Real World Datasets

We use three real world datasets test performances of four algorithms in accuracy and time consumption. These three datasets are consisted of three models with different shapes. First model represents a Bunny, the second model represents a Cow and the third model represents a Wolf. We did six experiments on three real world datasets.

Three of six are rigid transformation experiments. The detailed parameters are shown in Tab. 1.

Table 1 Transform parameters

Model	Rotation			Translation			Scale
	r_x	r_y	r_z	t_x	t_y	t_z	s
Bunny	12.6	15.1	26.3	25	-8	0.9	0.8
Cow	21.3	17.1	15.1	0.6	0.9	0.2	0.9
Wolf	22.9	5.7	16.2	5.5	-7.5	-5.2	0.85

The other three experiments are rigid transformation with outliers which are randomly generated. The transform parameters are the same, as shown in Tab. 1. In addition, we add 60 outliers to the target point set. The original point set is seen as template point set, and the target point set is generated by rigid transformation of the template point set. In this paper, the initial temperature parameter of TPS-RPM and ICP is slightly bigger than the largest distance of all point pairs, the other parameters are the same as the original algorithms.

Table 2 Accuracy on rigid transformation real world datasets

Model	Method	Rigid transformation			
		Total point pairs	Correct matching point pairs	Accuracy (%)	Time (s)
Bunny	LDSC	1889	1889	100	291.95
	PSC	1889	1889	100	198.26
	ICP	1889	19	1.00	119.67
	TPS-RPM	1889	9	0.47	391.68
Cow	LDSC	1022	980	95.89	44.10
	PSC	1022	974	95.30	34.26
	ICP	1022	10	0.97	31.69
	TPS-RPM	1022	981	95.98	173.7
Wolf	LDSC	1448	1448	100	130.99
	PSC	1448	1446	99.86	100.64
	ICP	1448	1157	79.90	96.33
	TPS-RPM	1448	2	0.13	545.04

Table 3 Accuracy on rigid transformation real world datasets with outliers

Model	Method	Rigid Transformation+60 Outliers			
		Total Point Pairs	Correct Matching Point Pairs	Accuracy (%)	Time (s)
Bunny	LDSC	1889	1786	94.54	334.654
	PSC	1889	908	48.06	319.236
	ICP	1889	24	1.27	128.194
	TPS-RPM	1889	7	0.37	395.043
Cow	LDSC	1022	980	95.89	95.522
	PSC	1022	736	72.01	36.333
	ICP	1022	2	0.19	31.921
	TPS-RPM	1022	2	0.19	223.51
Wolf	LDSC	1448	1448	100	162.553
	PSC	1448	1316	90.8	106.956
	ICP	1448	1448	0.21	101.452
	TPS-RPM	1448	2	0.13	497.585

The registration results of four algorithms are shown in Fig. 5 and Fig. 6. In Fig. 5, the target point set is generated by rigid transformation. In Fig. 6, the outlier points are put into target point set. The detailed information of accuracy is recorded in Tab. 2 and Tab. 3 respectively. LDSC gives the best matching results. In addition, the accuracy of TPS-RPM based on deterministic annealing and thin plate splines sometimes varies greatly.

The accuracy of four algorithms is the average value of hundred experiments. In Tab. 2, the accuracy of ICP and TPS-RPM is low and changes greatly, and they are

sensitive to the distribution of outlier points. LDSC performs the best results. In Tab. 3, in the case of redundant points, ICP and TPS-RPM almost do not work,

the performance of PSC declines significantly, LDSC always maintains a high accuracy, the results demonstrate that LDSC is robust.

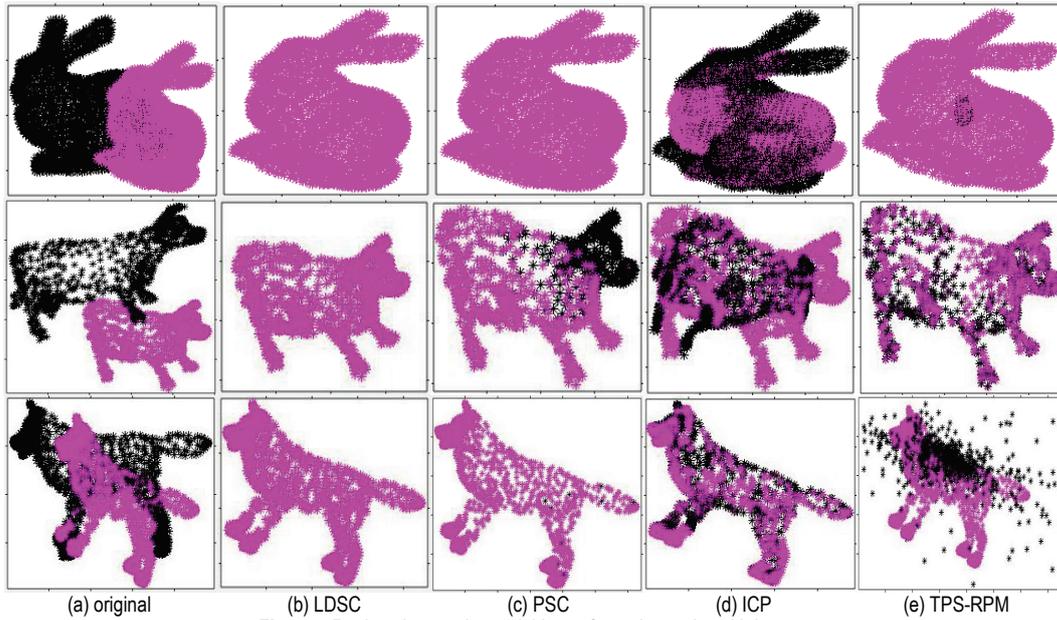


Figure 5 Registration results on rigid transformation real world datasets

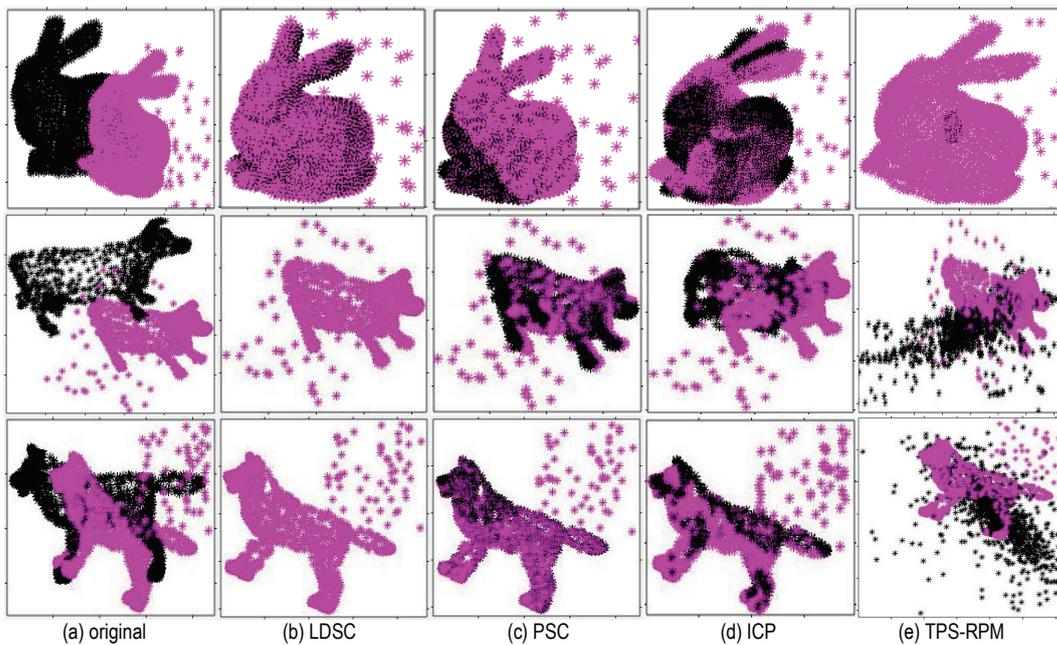


Figure 6 Registration results on rigid transformation real world datasets with outliers

5 CONCLUSION

In this paper, we proposed the effect of Local Density Shape Context algorithm, namely LDSC. We carried out experiments on a lot of synthetic datasets and real world datasets to assess the effectiveness of LDSC. We mainly make the following contributions: (1) we firstly use the local point density for cutting down the negative impact on extracting the feature descriptor. (2) the optimization of pairwise matching is also firstly used in LDSC for improving the effectiveness.

LDSC works well on point pattern matching in three dimensional space. LDSC is a powerful and promising tool and we believe that it can be applied to many

research fields. Multiple point pattern matching should be further studied in our future work.

ACKNOWLEDGEMENTS

This research is supported in part by the National Science Foundation of China Grant No. 61772227 and Science & Technology Development Foundation of Jilin Province under the grant No. 20160101259JC, 20180201045GX.

6 REFERENCES

- [1] Tang, J., Shao, L., & Zhen, X. (2014). Robust point pattern matching based on spectral context. *Pattern Recognition*, 47(3), 1469-1484. <https://doi.org/10.1016/j.patcog.2013.09.017>
- [2] Goshtasby, A. A. (2012). *Image Registration: Principles, Tools and Methods*. Springer Publishing Company, Incorporated. <https://doi.org/10.1007/978-1-4471-2458-0>
- [3] Besl, P. J. & McKay, N. D. (1992, April). Method for registration of 3-D shapes. *Sensor Fusion IV: Control Paradigms and Data Structures / International Society for Optics and Photonics*, 1611, 586-607.
- [4] Ma, J., Zhao, J., & Yuille, A. L. (2016). Non-rigid point set registration by preserving global and local structures. *IEEE Transactions on Image Processing*, 25(1), 53-64. <https://doi.org/10.1109/TIP.2015.2467217>
- [5] Scott, G. L. & Longuet-Higgins, H. C. (1991). An algorithm for associating the features of two images. *Proc. R. Soc. Lond. B*, 244(1309), 21-26. <https://doi.org/10.1098/rspb.1991.0045>
- [6] Shapiro, L. S. & Brady, J. M. (1992). Feature-based correspondence: an eigenvector approach. *Image & Vision Computing*, 10(5), 283-288. [https://doi.org/10.1016/0262-8856\(92\)90043-3](https://doi.org/10.1016/0262-8856(92)90043-3)
- [7] Carcassoni, M. & Hancock, E. R. (2003). Spectral correspondence for point pattern matching. *Pattern Recognition*, 36(1), 193-204. [https://doi.org/10.1016/S0031-3203\(02\)00054-7](https://doi.org/10.1016/S0031-3203(02)00054-7)
- [8] Zhou, Y., Wei, W., Yu, C., Sun, Y., Xing, C., Shen, W., & Shi, X. H. (2011). A Munkres Particle Swarm Optimization Algorithm for Point Pattern Matching. *International Conference on Human Health and Biomedical Engineering*. 1164-1167. <https://doi.org/10.1109/HHBE.2011.6029033>
- [9] Yin P. Y. (2006). Particle swarm optimization for point pattern matching. *Journal of visual communication and image representation*, 17(1), 143-162. <https://doi.org/10.1016/j.jvcir.2005.02.002>
- [10] Ansari, N., Chen, M. H., & Hou, E. S. H. (1990). Point pattern matching by a genetic algorithm. *Industrial Electronics Society, IECON '90, Conference of IEEE*, 2, 1233-1238. <https://doi.org/10.1109/IECON.1990.149313>
- [11] Besl, P. J. & Mc Kay, H. D. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 239-256. <https://doi.org/10.1109/34.121791>
- [12] Fitzgibbon, A. W. (2003). Robust registration of 2D and 3D point sets. *Image & Vision Computing*, 21(13), 1145-1153. <https://doi.org/10.1016/j.imavis.2003.09.004>
- [13] Chui, H. & Rangarajan, A. (2003). A new point matching algorithm for non-rigid registration. *Computer Vision & Image Understanding*, 89(2), 114-141. [https://doi.org/10.1016/S1077-3142\(03\)00009-2](https://doi.org/10.1016/S1077-3142(03)00009-2)
- [14] Belongie, S., Malik, J., & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4), 509-522. <https://doi.org/10.1109/34.993558>
- [15] Gui, Y., Su, A., & Du, J. (2013). Point-pattern matching method using SURF and Shape Context. *Optik-International Journal for Light and Electron Optics*, 124(14), 1869-1873. <https://doi.org/10.1016/j.ijleo.2012.05.037>
- [16] Liu, Y., Lan, C., Li, C., Mo, F., & Wang, H. (2016). S-AKAZE: An effective point-based method for image matching. *Optik-International Journal for Light and Electron Optics*, 127(14), 5670-5681. <https://doi.org/10.1016/j.ijleo.2016.03.072>
- [17] Pang, K., Liang, Y., Zhou, Y., Wang, K., & Wang, S. (2015). A directional algorithm based on shape context for 3D point pattern matching. *Journal of Computational Information Systems*, 11(7), 2355-2365.
- [18] Lian, W., Liang, Y., Pan, Q., Chen, Y., & Zhang, H. (2006). A Clustering and Quadratic Programming Based POCS Algorithm for Point Matching. *Intelligent Control and Automation, WCICA 2006. The Sixth World Congress IEEE on*, 2, 9791-9794.
- [19] Lian, W., Zhang, L., Liang, Y., & Pan, Q. (2010). A quadratic programming based cluster correspondence projection algorithm for fast point matching. *Computer Vision & Image Understanding*, 114(3), 322-333. <https://doi.org/10.1016/j.cviu.2009.12.001>
- [20] Caetano, T. S., Caelli, T., Schuurmans, D., & Barone, D. A. C. (2006). Graphical models and point pattern matching. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 28(10), 1646-1663. <https://doi.org/10.1109/TPAMI.2006.207>
- [21] Zhao, X., Wang, S., & Ding, X. (2007). A dynamic generating graphical model for point-sets matching. *Applied Mathematics & Computation*, 185(2), 1109-1119. <https://doi.org/10.1016/j.amc.2006.07.031>
- [22] Zass, R. & Shashua, A. (2008). Probabilistic graph and hypergraph matching. *Computer Vision and Pattern Recognition, CVPR 2008. IEEE Conference on*, 1-8. <https://doi.org/10.1109/CVPR.2008.4587500>
- [23] Lee, J. H. & Won, C. H. (2010). Topology preserving relaxation labeling for nonrigid point matching. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 33(2), 427-432.
- [24] Bourgeois, F. & Lassalle, J. C. (1971). An extension of the Munkres algorithm for the assignment problem to rectangular matrices. *Communications of the ACM*, 14(12), 802-804. <https://doi.org/10.1145/362919.362945>
- [25] Kabsch, W. (1976). A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Foundations and Advances*, A32, 922-923. <https://doi.org/10.1107/S0567739476001873>

Contact information:

You ZHOU, Doctor and Associate Professor
College of Computer Science and Technology, Jilin University
Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education
2699 Qianjin Street, Changchun, 130012, China
E-mail: zyou@jlu.edu.cn

Panpan CUI, Master Student
College of Computer Science and Technology, Jilin University
2699 Qianjin Street, Changchun, 130012, China
E-mail: ppancui@163.com

Yizhang WANG, PhD
College of Computer Science and Technology, Jilin University
2699 Qianjin Street, Changchun, 130012, China
E-mail: wyzhang2012@sina.cn

Limin WANG, Doctor and Professor
School of Management Science and Information Engineering,
Jilin University of Finance and Economics
3699 Jingyue Street, Changchun, 130117, China
E-mail: wlm_new@163.com

Sen YANG, Master Student
College of Computer Science and Technology, Jilin University
2699 Qianjin Street, Changchun, 130012, China
E-mail: Yangsen15@jlu.edu.cn

Xu ZHOU, Doctor and Associate Professor
(Corresponding author)
Center for Computer Fundamental Education, Jilin University
2699 Qianjin Street, Changchun, 130012, China
E-mail: zhouxu@jlu.edu.cn