DPN-LRF: A Local Reference Frame for Robustly Handling Density Differences and Partial Occlusions

Shuichi Akizuki(✉) and Manabu Hashimoto

Graduate School of Science and Technology, Chukyo University,
101-2, Yagoto-Honmachi, Showa-ku, Nagoya, Aichi, Japan
{akizuki,mana}@isl.sist.chukyo-u.ac.jp

Abstract. For the purpose of 3D keypoint matching, a Local Reference Frame (LRF), a local coordinate system of the keypoint, is one important information source for achieving repeatable feature descriptions and accurate pose estimations. We propose a robust LRF for two main point cloud disturbances: density differences and partial occlusions. To generate LRFs that are robust to such disturbances, we employ two strategies: normalizing the effects of point cloud density by approximating the surface area in the local region and using the dominant orientation of a normal vector around the keypoint. Experiments confirm that the proposed method has higher repeatability than state-of-the-art methods with respect to density differences and partial occlusions. It was also confirmed that the method enhances the reliability of keypoint matching.

1 Introduction

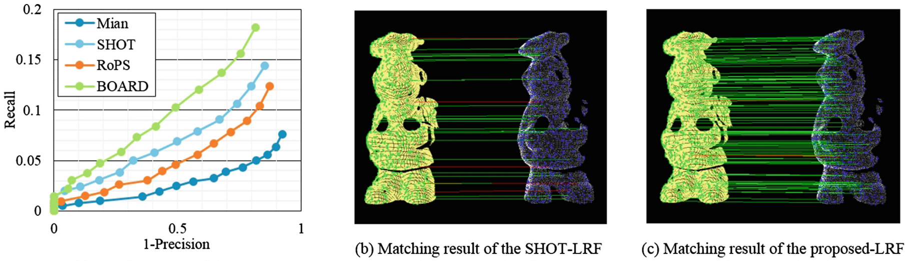
3D object detection and localization using point clouds are basic tasks for robots working in human living environments. However, the performance of vision systems for achieving such tasks is sometimes disturbed by two types of conditions. One is the difference in point cloud density between an object model and a target object in the input scene, because the distance between a range sensor and the target object is unknown. The other is the occurrence of missing point clouds, because in this case objects frequently occlude each other in the input scene.

Model-based methods are among the types of methods typically used for 3D object recognition. Such methods generally have main four modules, one each for:

1. Extracting keypoints from the object model and the input scene
2. Describing 3D features
3. Finding corresponding points between the object model and the input scene
4. Estimating position and pose of the object model in the input scene.

After the module 1, a Local Reference Frame (LRF), a local coordinate system of the keypoint, is calculated. The LRF consists of three orthogonal unit 3D vectors. A rigid transformation that aligns corresponding LRFs can be determined as the pose parameter of the object model in the input scene. In addition,

G. Bebis et al. (Eds.): ISVC 2015, Part I, LNCS 9474, pp. 878–887, 2015.
DOI: 10.1007/978-3-319-27857-5_78



(a) Recall vs. 1-Precision curve

(b) Matching result of the SHOT-LRF and the SHOT descriptor

(c) Matching result of the proposed-LRF and the SHOT descriptor

Fig. 1. 3D keypoint matching performance. (a) Recall vs. 1-precision curve. Keypoints were matched to each other by using a SHOT descriptor and four types of LRFs. (b), (c) Example matching results for SHOT-LRF and the proposed LRF with the SHOT descriptor. Green/red lines show correct/incorrect correspondences.

most of the 3D features [1,2] are calculated by dividing support regions into multiple cells according to each axis direction of the LRF, and the geometric relations of point clouds in the cells are converted to a feature vector. In this way, a repeatable LRF can enhance not only the accuracy of pose estimation, but also the reliability of keypoint matching. Figure 1 (a) shows recall vs. 1-precision curves we obtained in a keypoint matching experiment. In this experiment, we used a pair of point clouds with different densities and missing region.

We used a SHOT descriptor and four types of LRFs: Mian [3], SHOT [4], RoPS [2], and BOARD [5]. Note that the only difference among the curves is due to the methods used to calculate the LRF. Replacing the LRF confirmed the keypoint matching performance changed. Figure 1 (b) and (c) show keypoint matching results; (b) shows those obtained with SHOT-LRF and (c) shows those obtained with the proposed LRF. Many more corresponding points were obtained for the (c) results than for the (b) results.

These results led us to conclude that we should select a suitable LRF by considering the disturbances that occur in the application environment. The purpose of our research is to achieve a repeatable LRF for changing point cloud density and occlusions.

2 Related Work

There are two types of LRFs. One uses the Eigenvector of a covariance matrix generated by a point cloud within the support region centered in the keypoint. The methods described in [3,6] are the simplest of these. With these methods, Eigenvectors are assigned as each of the LRF axes. However, the directions of Eigenvectors have sign ambiguities that decrease the uniqueness of the LRF. In order to solve this problem, the SHOT [4] determines the major direction by counting the number of 3D points existing on both sides of the support regions.

RoPS [2] and DosSants [7] have been used to tackle the problem of point cloud density. With these methods, a normalized scatter matrix is used for generating the LRF. This matrix is normalized by the total area of the local surface, which is approximated as the summation of the area of each mesh. However, the repeatability of the direction of each axis will decrease when point clouds within the support region are partially missed due to occlusion. This is because in such cases the point cloud distribution will be changed. The problem with these Eigenvector-based methods is that all points within the local region contribute to the calculation of the axis direction.

The other LRFs calculate three axes one by one [1,5,8–12]. These methods commonly assign the normal vector of a local surface as the z-axis of the LRF because the direction of it has high repeatability. Therefore, the method used to calculate the x-axis characterizes each LRF. In the methods proposed in [9,10], an arbitrary point within the support region is projected onto the tangent plane of the z-axis. A vector that indicates the projected point from the origin of the z-axis is assigned as the x-axis of the LRF. However, the uniqueness of the x-axis direction is not so high because it is difficult to obtain the repeatable projected point.

The BOARD [5] has solved this problem by detecting the point that has the most inclined normal vector respective to the z-axis, and it is projected onto the tangent plane. This method is robust to occlusions because it estimates whether the most inclined normal vector exists or not in the missing region. The Mesh HoG [12] determines the dominant orientation of the polar angle on the tangent plane of the z-axis by using the polar histogram of the surface gradient. This LRF is also robust to occlusions because "dominant" information has been used for generating the x-axis. However, it has been reported in [2] that the repeatability of LRF decreases when point clouds have different densities. Therefore, there is no method that is robust to density differences and occlusions at the same time.

3 DPN-LRF: Dominant Projected Normal LRF

3.1 Overview

To deal with the effects of differences in point cloud density and those of occlusions, the proposed LRF consists of two strategies. The first is to take into account the weighting factor for the area of the local surface mesh as reported in [2]. This is one good solution for solving the problem of point cloud density. It also uses information about the area of the local surface. We also propose a method for quickly calculating the weighting factor for the area of the local surface mesh. The other strategy is to determine an x-axis that is robust to partial occlusions. Since the proposed method assigns the Dominant Projected Normal vector as the x-axis of the LRF, we named this method DPN-LRF for short.

3.2 Method for Generating the DPN-LRF

The DPN-LRF at the keypoint $\mathbf{p}$ and its normal vector $\mathbf{n}$ is calculated by using point cloud $\{\mathbf{p}_0, \mathbf{p}_2, ..., \mathbf{p}_N\}$ and its normal vectors $\{\mathbf{n}_0, \mathbf{n}_1, ..., \mathbf{n}_N\}$ within the support region centered in the $\mathbf{p}$. Figure 2 shows an overview of the method used to generate the DPN-LRF.

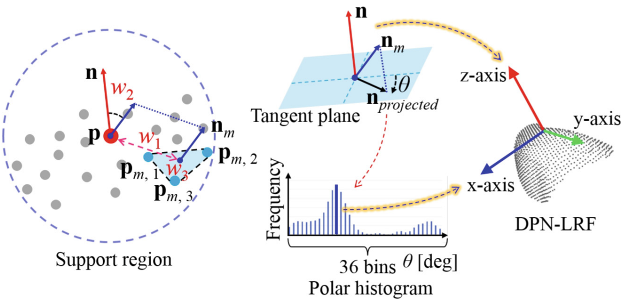


Fig. 2. Overview of the method used to generate DPN-LRF. At left is a point cloud within the support region centered on the keypoint (red point). At center top is the tangent plane of the z-axis of the DPN-LRF; θ shows orientation of the projected normal vector. At center bottom is the polar histogram. At right is the DPN-LRF on the point cloud within the support region.

z-axis: This axis is defined as the Eigenvector corresponding to the smallest Eigenvalue of covariance matrix generated from point cloud around keypoints. This vector has sign ambiguity, but this problem can easily be solved by determining that the positive direction is the direction towards the viewpoint. This vector corresponds to the normal vector $\mathbf{n}$ of each point. We used point cloud within 10 mr ("mr"=mesh resolution) p for calculating covariance matrix.

x-axis: This axis is generated as the dominant orientation of the polar direction on the tangent plane of the z-axis centered in the $\mathbf{p}$. Here, the orientation θ of the projected normal vector $\mathbf{n}_{projected}$ onto the tangent plane is voted to a polar histogram that has 36 bins (covering 360 degrees). To deal with aliasing, each vote is smoothed by using Gaussian distribution. In this histogram, a direction that has a high voted value is assigned as the x-axis direction. To generate a highly accurate axis, Bilinear interpolation is applied by using neighboring bins. Because this process uses histogram-formed data, the generated x-axis becomes stable even if point clouds are missed due to occlusions. The voted value is calculated by multiplication of the following three weighting factors.

w_1 is the weighting factor for normalizing density differences in point clouds. In the corresponding points of the object model and the input scene, the local surface areas within the support regions are the same even if the point densities differ. The voting value is normalized by using the local surface area.

First of all, point cloud is converted to triangle meshes $\{t_0, t_1, ..., t_M\}$. Here, the m-th mesh consists of $\mathbf{p}_{m,1}$, $\mathbf{p}_{m,2}$, and $\mathbf{p}_{m,3}$, and the normal vector $\mathbf{n}_m$ of t_m is calculated as $(\mathbf{n}_{m,1} + \mathbf{n}_{m,2} + \mathbf{n}_{m,3})/3$. The voting process is performed for each triangle. The weighting value w_1 of the m-th mesh is described by

$$w_{m,1} = \frac{(\mathbf{p}_{m,2} - \mathbf{p}_{m,1}) \times (\mathbf{p}_{m,3} - \mathbf{p}_{m,1})}{\sum_{m=0}^{M} (\mathbf{p}_{m,2} - \mathbf{p}_{m,1}) \times (\mathbf{p}_{m,3} - \mathbf{p}_{m,1})} \,. \tag{1}$$

Since a larger mesh represents the local shape more strongly, a large mesh gets a large weighting value. w_2 represents the distance between the keypoint and the mesh and is described by

$$w_{m,2} = r - \left\| \frac{(\mathbf{p}_{m,1} + \mathbf{p}_{m,2} + \mathbf{p}_{m,3})}{3} - \mathbf{p} \right\| . \tag{2}$$

In general, a mesh that is distant from the keypoint is sometimes affected by the presence of clutter. Therefore, by decreasing the weighing value of such normal vectors, the effects of noise can be decreased. w_3 represents the stability of the projected normal vector and is defined by

$$w_{m,3} = 1 - \mathbf{n} \cdot \mathbf{n}_m . \tag{3}$$

Here, $\cdot$ represents a dot product. When $\mathbf{n}_m$ is steeply inclined towards $\mathbf{n}$, the polar direction of the projected vector becomes stable. Therefore, the weighing value of the normal vector should become large.

y-axis: This is the perpendicular axis of the z-axis and the x-axis. Therefore, it is calculated by z-axis × x-axis.

3.3 Fast DPN-LRF Generation from Unorganized Point Cloud

By taking the area of each triangle mesh into account, the LRF can obtain robustness for differences in point cloud density. Unfortunately, triangle meshes are not always available, because unorganized point clouds are generally captured by the range sensor. Therefore, for generating the DPN-LRF, it is necessary to generate triangle meshes as an additional processing. Moreover, many 3D features do not require triangle meshes for computing. In practical use, it is preferable to generate the LRF without triangle meshes.

In this subsection, we propose a method to calculate the DPN-LRF without triangle meshes. In particular, the area of a triangle mesh is approximated as the distance between $\mathbf{p}_n$ and its nearest neighbor. (1) is replaced with

$$w'_{n,1} = \frac{\|\mathbf{p}_{nearest} - \mathbf{p}_n\|}{\sum_{n=0}^{N} \|\mathbf{p}_{nearest} - \mathbf{p}_n\|} . \tag{4}$$

Here, $\mathbf{p}_{nearest}$ represents the nearest point of $\mathbf{p}_n$. Also, (2) and (3) are respectively replaced

$$w'_{n,2} = r - \|\mathbf{p}_n - \mathbf{p}\|, \tag{5}$$

$$w'_{n,3} = 1 - \mathbf{n} \cdot \mathbf{n}_n . \tag{6}$$

The computational cost of the above modified equations is lower than that of the equations described in Sect. 3.2. Therefore, we named the LRF calculated using (4) – (6) “Fast DPN-LRF”.

4 Experiments

4.1 Performance for Point Density

To evaluate the various methods’ performance for density differences in point clouds, we performed an experiment in which LRF repeatability was calculated on corresponding points. In this experiment, we used the object models provided by [3]. For simulating a point cloud captured by a range sensor, we applied the Hidden Point Removal (HPR) operator [13] to each model. We define these models as M and present them in Fig. 3.

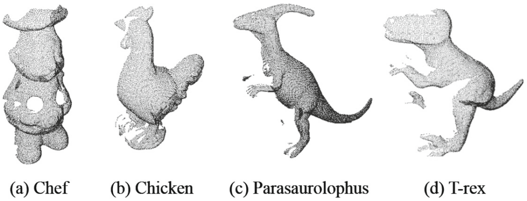
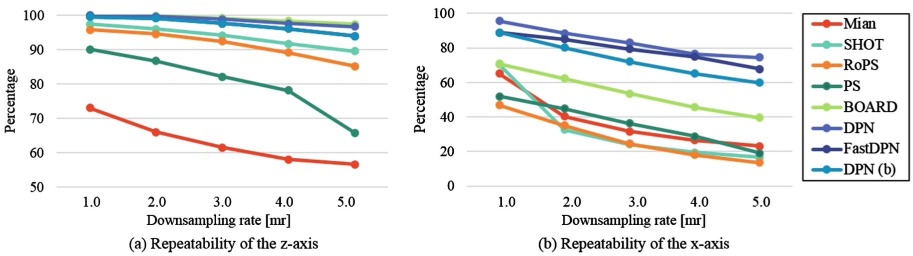

(a) Chef (b) Chicken (c) Parasaurolophus (d) T-rex

Fig. 3. Overview of the object models M.

Sparse point clouds S were generated by adding Gaussian noise with a standard deviation of 0.1 mr and then downsampled to 1.0 – 5.0 mr. We randomly extracted 1,000 keypoints from each model. The corresponding point in the S is determined as the nearest point to the keypoint of the M.

Figure 4 shows the percentage of LRF that has error within 10 [deg]. In this experiment, we compared our methods with the Mian, the SHOT, the RoPS,the PS-LRF and BOARD methods. In order to estimate the effect of the proposed weighting factors w_1, w_2, and w_3, we also compared the performance of the DPN (b). This method do not use weighting factors when the DPN-LRF is generated. All methods are implemented in C++, and we used the Point Cloud Library [14]. In this experiment, we tested different values of support radius r (5, 10, 15, and 20 mr). r = 20 mr is best for all method except for z-axis of the PS. Oprimal radius of this method was 5 mr.



(a) Repeatability of the z-axis (b) Repeatability of the x-axis

Fig. 4. The relationship between the angular error of z, the x-axis of LRF, and the point cloud density. The reliability of the z-axis and x-axis are shown in (a) and (b). The vertical axis represents the percentage of LRF that has error within 10 degrees.

In terms of z-axis repeatability, all methods achieved highly accurate estimation except for the Mian, with which some of the z-axes generated were directed toward the opposite side. When the downsampling rate was changed from 1.0 to 5.0 mr, the performance of the BOARD decreased by 2.7 %, while that of DPN and Fast DPN decreased by 5.7 %. This shows the z-axis repeatability is almost equal for BOARD-LRF and the proposed methods.

In terms of x-axis repeatability, the performance of the previous methods significantly decreased. The decrease was 54.0 % for SHOT and 31.6 % for BOARD, but was only 21.1 % for DPN and the Fast DPN. Moreover, for a 5.0 mr downsampling rate, the repeatability of DPN was 8.1 % higher than that of DPN (b). This confirmed that the proposed weighting factors helped to increase the repeatability of the axis direction when the point cloud density was different.

4.2 Performance for Partial Occlusions

To evaluate the methods' performance for partially missed point clouds within the support region, we calculated the relationship between the number of missing points and the LRF repeatability. In this experiment, we used M as the model data. The scene data was one-sided point clouds generated by using rotated object models and the HPR operator. Gaussian noise was added to each target. The range of rotation was 0 – 90 [deg] and the pitch was 10 [deg]. We randomly extracted keypoints $\mathbf{p}_m$ and $\mathbf{p}_s$ from the model and the scene, respectively. Corresponding points were determined as those that satisfied $|\mathbf{p}_m - \mathbf{R}\mathbf{p}_s| < 2.5[mr]$, where $\mathbf{R}$ is the rotation matrix. By using this process, we were able to generate partially missing point clouds in the support region of corresponding points. Figure 5 shows the relationship between the percentage of angular error of LRF within 10 [deg] and the missing rate of point clouds calculated by $missing\ rate = |N_M - N_S|/max(N_M, N_S)$. Here, N_M and N_S respectively represent the number of points within the support regions of $\mathbf{p}_m$ and $\mathbf{p}_s$. A higher value means that a lot of points have been missed. In this experiment, we calculated the angular error e of LRF by

$$e = \arccos\left(\frac{trace(\mathbf{L}_S\mathbf{L}_M^{-1}) - 1}{2}\right)\frac{180}{\pi}\ . \tag{7}$$

This equation proposed by [2]. $\mathbf{L}_S$ and $\mathbf{L}_M$ represent corresponding LRFs and are formed as 3×3 matrix. If the corresponding LRFs have no error, $trace(\mathbf{L}_S\mathbf{L}_M^{-1})$ will become the identity matrix. In this case, the value e takes zero.

In this experiment, we used over 20k corresponding points and tested different values of support radius r (5, 10, 15, and 20 mr). The results for the highest (optimal) radius for each method are 20, 20, 15, 20, 20, 20, and 20 in order from 'Mian' to 'Fast DPN'.

The stability of the Eigenvector direction is not so high when point clouds within the support region are partially missed. As a result, the repeatability of the Eigenvector-based methods Mian, SHOT, and RoPS significantly decreased when the missing rate was high. In contrast, PS and BOARD has a function to deal with the missed regions and so its repeatability was higher than that of

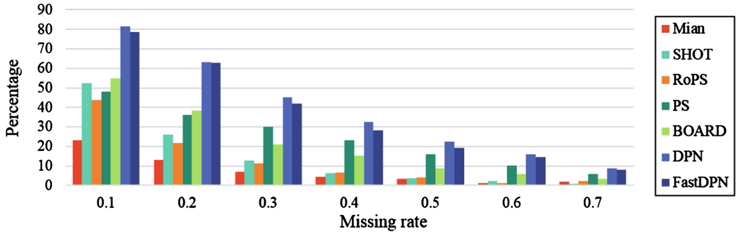


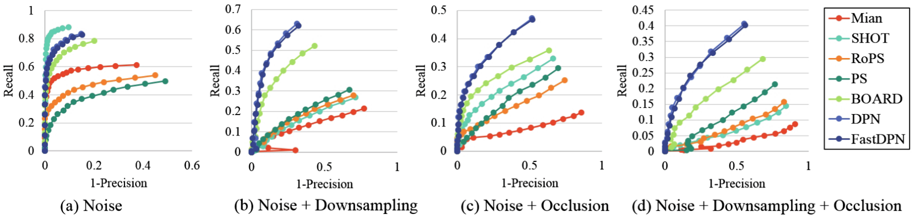
Fig. 5. Relationship between the angular error of LRF and the missing rate. Vertical axis represents percentage of LRF has error within 10 degree.

Eigen vector-based method. For every missing rate, it was confirmed that the repeatability of the proposed method is higher than that of the other methods.

4.3 Performance Evaluation for Keypoint Matching

The results obtained from the aforementioned experiments confirmed that the proposed LRF has higher repeatability than any of the other LRFs to which it was compared. Therefore, in the application of keypoint matching for object localization, it can be expected that the reliability of matching will be enhanced by replacing any of the previous LRF with the proposed one. In this subsection, we report the performance obtained for each method by recall vs. 1-precision curves as recommended in [15].

We used point clouds with four types of conditions: (a) Gaussian noise with a standard deviation of 0.1 *mr*, (b) Gaussian noise and 3.0 *mr* downsampling, (c) Gaussian noise and occlusion, and (d) Gaussian noise, downsampling and occlusion. In order to eliminate keypoint detector errors, we used 1,000 randomly extracted corresponding points. Occluded corresponding points are generated by method used in Sect. 4.2. The results obtained are shown in Fig. 6.



(a) Noise (b) Noise + Downsampling (c) Noise + Occlusion (d) Noise + Downsampling + Occlusion

Fig. 6. Recall vs. 1-precision curves for four controlled conditions.

Under condition (a), the best results were obtained with SHOT and the results obtained with BOARD, DPN, and Fast DPN were also good. This confirmed that these LRFs are robust to noise. Under condition (b), the SHOT

Table 1. Computational time of each LRF.

Method	Mian [3]	SHOT [4]	RoPS [2]	PS [9]	BOARD [5]	DPN	Fast DPN
T [msec]	0.12	0.19	0.60	0.24	0.18	0.25	0.27

curve dropped significantly because when the downsampling rate was 3.0 mr, the x-axis repeatability of SHOT was less than 25 % (see Fig. 4 (b)). In contrast, the BOARD, DPN, and Fast DPN performances were relatively higher than those of other LRFs. Under conditions (c) and (d), all methods showed lower performance than they did under conditions (a) and (b). The DPN and Fast DPN performances were also relatively higher than those of the other LRFs.

In the all conditions, curves of the DPN-LRF and Fast DPN-LRF are almost equal. Therefore, it was confirmed that Eq. (4) is a practical solution for approximation of the area of meshes.

4.4 Processing Time Comparison

We calculated the processing time in an experiment for generating LRF. In this experiment, we measured the average processing time of the LRF at a keypoint for all methods. The results are shown in Table 1. This experiment is performed on a desktop with an Intel CORE$^{\text{TM}}$i7 860 CPU and 12GB RAM.

Since RoPS and DPN require mesh generation and the mesh generation computation time depends on algorithms, we did not include these methods in the experiment. Although we were able to improve the Fast DPN processing time by approximating the area of meshes, it was still slightly slower than that of the other methods. However, this is not a problem because it requires more LRFs to get the same number of correct corresponding points by using previous LRFs.

5 Conclusion

We have proposed a novel Local Reference Frame (LRF) for robustly handling two main point cloud disturbances: density differences and partial occlusions. The method comprises two types, Dominant Projected Normal (DPN)-LRF and Fast DPN-LRF. Experiment results confirmed that the method is more robust to differences in point cloud density and partially missing clouds than other methods to which it was compared. In a keypoint matching experiment, we also confirmed that the reliability of matching was enhanced by replacing previous LRF with the proposed one. This suggests that a good combination of LRF and 3D features will provide even more reliable recognition results. Accordingly, in future work, we will consider ways of achieving such a combination.

Acknowledgements. This work was partially supported by Grant-in-Aid for Scientific Research (C) 26420398.

References

1. Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J.: Recognizing objects in range data using regional point descriptors. In: Pajdla, T., Matas, J.G. (eds.) ECCV 2004. LNCS, vol. 3023, pp. 224–237. Springer, Heidelberg (2004)
2. Guo, Y., Sohel, F.A., Bennamoun, M., Lu, M., Wan, J.: Rotational projection statistics for 3D local surface description and object recognition. Int. J. Comput. Vis. **105**, 63–86 (2013)
3. Mian, A.S., Bennamoun, M., Owens, R.A.: On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. Int. J. Comput. Vis. **89**, 348–361 (2010)
4. Tombari, F., Salti, S., di Stefano, L.: Unique signatures of histograms for local surface description. In: European Conference on Computer Vision, pp. 356–369 (2010)
5. Petrelli, A., di Stefano, L.: On the repeatability of the local reference frame for partial shape matching. In: IEEE International Conference on Computer Vision, pp. 2244–2251 (2011)
6. Zhong, Y.: Intrinsic shape signatures: a shape descriptor for 3d object recognition. In: Proceedings of the International Conference on Computer Vision Workshops, pp. 689–696 (2009)
7. dos Santos, T.R., Franz, A.M., Meinzer, H., Maier-Hein, L.: Robust multi-modal surface matching for intra-operative registration. In: IEEE International Symposium on Computer-Based@Medical Systems, pp. 1–6 (2011)
8. Stein, F., Medioni, G.G.: Structural indexing: Efficient 3-d object recognition. IEEE Trans. Pattern Anal. Mach. Intell. **14**, 125–145 (1992)
9. Chua, C.S., Jarvis, R.: Point signatures: a new representation for 3D object recognition. Int. J. Comput. Vis. **25**, 63–85 (1997)
10. Sun, Y., Abidi, M.A.: Surface matching by 3D point's fingerprint. In: ICCV, pp. 263–269 (2001)
11. Novatnack, J., Nishino, K.: Scale-Dependent/Invariant local 3D shape descriptors for fully automatic registration of multiple sets of range images. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 440–453. Springer, Heidelberg (2008)
12. Zaharescu, A., Boyer, E., Varansi, K., Horaud, R.: Surface feature detection and description with applications to mesh matching. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 373–380 (2009)
13. Katz, S., Tal, A., Basri, R.: Direct visibility of point sets. ACM Trans. Graph. **26**, 24 (2007)
14. Rusu, R.B., Cousins, S.: 3d is here: Point cloud library (PCL). In: IEEE International Conference on Robotics and Automation, ICRA, IEEE (2011)
15. Ke, Y., Sukthankar, R.: PCA-SIFT: a more distinctive representation for local image descriptors. In: CVPR (2), pp. 506–513 (2004)