A Registration Method for 3D Point Clouds with Convolutional Neural Network

Shangyou Ai, Lei Jia, Chungang Zhuang(✉), and Han Ding

School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, People's Republic of China
cgzhuang@sjtu.edu.cn

Abstract. Viewpoint independent 3D object pose estimation is one of the most fundamental step of position based vision servo, autopilot, medical scans process, reverse engineering and many other fields. In this paper, we presents a new method to estimate 3D pose using the convolutional neural network (CNN), which can apply to the 3D point cloud arrays. An interest point detector was proposed and interest points were computed in both source and target point clouds by region growing cluster method during offline training of CNN. Rather than matching the correspondences by rejecting and filtering iteratively, a CNN classification model is designed to match a certain subset of correspondences. And a 3D shape representation of interest points was projected onto an input feature map which is amenable to CNN. After aligning point clouds according to the prediction made by CNN, iterative closest point (ICP) algorithm is used for fine alignment. Finally, experiments were conducted to show the proposed method was effective and robust to noise and point cloud partial missing.

Keywords: CNN · Point clouds · Point detector · Registration · Rigid transformation

1 Introduction

The scope of object pose estimation ranges from medical data process to automation in industry. For example, position based vision servo (PBVS) [1] is one of the two basic approaches in the field of visual servo control, and it necessitates the pose of the robot with respect to a specific coordinates prior to be known before subsequent execution. The variation of illumination conditions, background clutter, and occlusion makes conventional image-based techniques ineffective. Since 3D LIDAR scanner is far more accessible in recent years, one can obtain the 3D point cloud of an object much easier than before. It becomes very attractive to do the registration work for 3D point clouds as well as for images [2–5]. Among plenty of approaches, iterative closest points (ICP) [6] is a well-known method to solve the registration problem numerically. However, it always suffers from the local minima because of the non-convex characteristic and the iteration nature of the ICP approach. [4] Provides a globally optimal ICP solution based on branch-and-bound method, in exchange of time consumption. Here we focus on coarse registration method to provide initial transformation before using ICP.

Y. Huang et al. (Eds.): ICIRA 2017, Part III, LNAI 10464, pp. 377–387, 2017.
DOI: 10.1007/978-3-319-65298-6_35

Some research focuses on intelligence method for point cloud processing, e.g. convolutional neural networks (CNN). In most of cases, CNN deals with the feature maps which have intuitive interpretation, like the image [7]. In order to deal with point clouds using CNN, an elaborate feature map for point clouds have to be generated. In [8], a Hough accumulator is designed associated with every points in 3D point cloud for normal estimation, and the image-like structure of the accumulator is amenable to CNN.

Since massive unstructured point clouds are difficult to find the point-to-point correspondences between target and source point clouds, point detectors are always designed for reduction of computation complexity [9]. Interest points are selected by detectors according to a specific criterion, which is invariant to rigid transformation. And a correspondence is identified by point descriptor if the similarity between two points greater than a threshold. Many research focuses on design distinctive point descriptor for 3D point clouds [3]. In [10], the geodesic graph model (GGM) was proposed, the method utilized the fact that geodesic-like distance is an invariant structure feature during non-rigid deformation.

Once the interest points are detected, a typical method for estimating the transformation is Random Sample Consensus (RANSAC) [11]. The RANSAC method estimates a transformation for a given set of correspondences iteratively, and yields to the best one that eliminates most of outliers. In this paper, instead of using RANSAC for transformation estimation, we treat the correspondence matching problem as a classification task using CNN. Owing to the effectiveness of our designed point detector, only a few points were efficient for transformation estimation. As mentioned before, a new feature map associated with interest points is also derived to be fed to CNN. After matching the correspondences predicted by CNN, singular value decomposition (SVD) is used for transformation estimation. And ICP is used as fine registration method.

2 Methodology

2.1 Registration Problem

Given two 3D point clouds, addressed as source point cloud S and target point cloud T respectively (source point cloud is available as reference, and target point loud is often acquired by a 3D scanner). We want to find a rigid transformation $\mu(R, p)$ which minimize the error E:

$$E(R,t) = \sum\nolimits_{i=1}^{N} \|(Rs_i + p) - t_i\|^2 \tag{1}$$

where the set $\{(s_i, t_i)\,\text{with}\, s_i \in S,\, t_i \in T,\, i \in 1 \cdots N\}$ forms the correspondences between source and target point clouds. In the cases that different number of points in two point clouds (e.g. partial missing in the target point cloud), only a part of matches are expected, and a rejection scheme is sometimes desirable that discards the points without counterparts [9]. In addition, accurate pair-wise matching for all the points is infeasible in practice due to the high cardinality of point clouds.

3D interest point detectors are always designed to reduce the complexity in correspondences matching [12–14]. The consistency of detected interest points should be guaranteed with the presence of noise and outliers during rigid transformation, i.e., the point detector have to be as discriminative as possible to keep the local shape information invariant to rigid transformation and robust to other disturbances. Interest points are detected in the source and target point cloud respectively, and we obtain interest points set $P^S = \{p_1^S, \ldots, p_{K_S}^S\} \subset S$, $P^T = \{p_1^T, \ldots, p_{K_T}^T\} \subset T$.

A correspondence is identified usually by using point descriptors which describe local neighborhood of each interest point [15]. A correspondence (a, b) hold if:

$$\|S(D(a)) - S(D(b))\| > \tau \tag{2}$$

where D is the descriptor function that mapping local neighborhood of a point to a set of scalar, S is a similarity measurement function, and τ is a predefined threshold. We do not require the correspondences identification by descriptor function in this paper, only interest points are required to match correspondences.

We proposed the interest point detector to denoise the information underlying in point cloud transformation. To this end, a region growing clustering is implemented to ensure the consistency of detected interest points.

2.2 Point Detector

A set of interest points are detected to represent the pose of point cloud. As a premise, sampling strategy is implemented to select the salient points in the model of the point cloud, thereby the original point cloud is represented by small number of points, a region growing clustering is carried out and the interest points are designed to be the center of the clusters with most amount of points. Details are described as follows.

First, we down-sample the source and target point clouds respectively by the strategy of choosing the salient points with a significance metric proposed by [12].

For every point in the point cloud, the covariance matrix is computed according to its K nearest neighbors:

$$COV(p_i) = \sum_{j=1}^{K} (p_j - p_i)(p_j - p_i)^T \tag{3}$$

the smallest eigenvalue of $COV(p_i)$ was chosen to be the significance assigned to each points, which measures the variance of its neighborhood. And the salient points were selected to be the top $\eta_s \times n$ points among all the points in terms of their significance. Here η_s is the sampling rate and n is the total number of points. Second, the salient points were gathered into clusters by a region growing method [16]. A seed point is randomly chosen which had not been clustered, and its neighbor points are gathered into a cluster. Intra-cluster distance threshold T_i was set to keep differences between points in clusters subtle, and inter-cluster distance threshold T_c was set to avoid difference between clusters too small, thus decrease the ambiguity for correspondences matching. Finally, the interest points were set to be centers of the L clusters with most number of points. Note that while increasing the number of interest points can increase

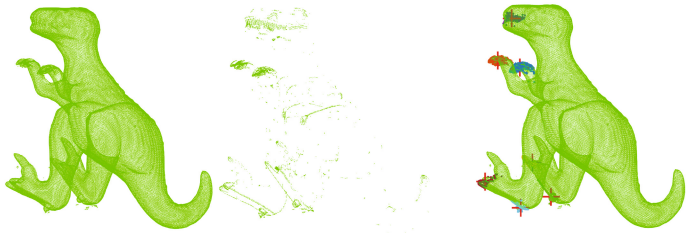


Fig. 1. Visualization of interest point detection. From left to right: Origin point cloud; salient points selected; clusters by region growing clustering (shown in colorful blobs) and detected 7 interest points (shown in red crosses). (Color figure online)

the robustness of pose representation to noise and occlusion, but also increase the cost of computation during registration period. Figure 1 shows the process of interest point detection.

2.3 Matching with CNN

Because the results of region growing clustering will deviate a bit based on the initial state of iteration and other disturbances, the final clusters result will not be identical for the same point cloud in every experiment. Then a deterministic algorithm which sort interest points into a canonical order for correspondence matching will not work. Therefore, we proposed the CNN classification model to achieve automatic correspondences matching. The representation of internal relationship of interest points is set to be the input feature map of CNN. Since the internal relationship between interest points are invariant to rigid transformation, the CNN helps to recover the complex mapping from the representation of interest points to correct correspondences.

As for source and target point clouds, interest points were computed in the source set $\{p_1^S \ldots p_{K_S}^S\}$ and target set $\{p_1^T \ldots p_{K_T}^T\}$ respectively. In the training step, K_T interest points in the source set were randomly selected, and for the chosen K_T points, the corresponding selection is assigned to a given category, which will be set as the training set target of the CNN. The categorical procedure is specified as follows.

Assume that every detected points in the target set can be matched to a detected point in the source set, there will be a total number of $C_{K_S}^{K_T}$ possible combinations. Then the category of one possible combinations is assigned to one of the $C_{K_S}^{K_T}$ selections, and the mapping from the selection to the category is trivial. Note that though the rapid growth of possible combinations against the number of interest points will increase the complexity of computation incredibly, relatively small number of interest points are chosen in practice (at least three for rigid transformation) make it feasible for point cloud registration.

Instead of input the raw point coordinates to the neural networks, the weighted adjacency matrix of interest points is computed for the input feature map of CNN. Regarding interest points as vertexes in a complete graph, the K_T interest points were

then mapped to a weighted adjacency matrix M_t using the Euclidian distance between interest points as weight:

$$M_t = (m_{ij})_{K_T \times K_T},\ m_{ij}^T = \sqrt{(p_i^T - p_j^T)^2} \tag{4}$$

Figure 2 illustrates the matching procedure with CNN.

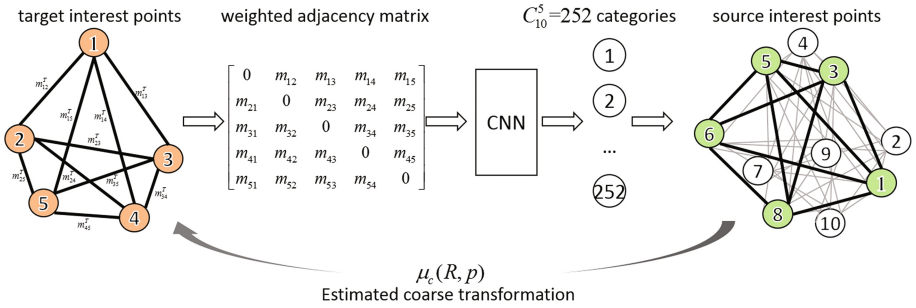


Fig. 2. Illustration of our proposed CNN matching process. 10 and 5 interest points are detected in the source point cloud and target point cloud respectively. Here m_{ij}^T indicates the weight in target cloud. The prediction made by CNN is a set of source interest points. We consider the graph as undirected graph and $m_{ij}^T = m_{ji}^T$.

The reason for this procedure is twofold. First, in order to match the correspondences between the source and target point cloud, the feature map should be invariant to rigid transformational, and robust to noise and outliers, and Euclidian distance between points meet the requirements. Second, the dataset can be transfer from raw arrays of coordinates into an organized feature map, which is amenable for CNN. And taking advantages of local conjunctions detection and shared weights of CNN, the point correspondences which woven in a tangle way originally can be found correctly by the information encoded in the weighted complete graph.

2.4 Pose Estimation

In the online registration step, the weighted adjacency matrix was computed from target set by the same pipeline. Applying the prediction of CNN, a set of points in the source set are assigned as correspondences. And the least-square error transformation is estimated by the SVD method [17], which is used to associate correspondences by all the possible permutation, and select the one with least error in (1).

After coarse registration computed by SVD, a fine registration is performed by implementing the ICP method.

3 Experiments

3.1 Region Growing Cluster

We choose the Stanford happy Buddha and a valve model for evaluation of our proposed method. Figure 3(b) and (c) show the results of region growing clustering. The sampling rate for choosing the salient point is set to be 20%, and the thresholds T_i,T_c for clustering are defined according to the range R_d of point cloud.

$$R_d = \max \sqrt{(p_i^S - p_j^S)^2} \tag{5}$$

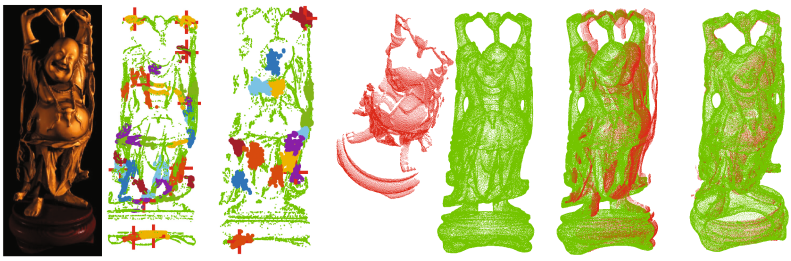


Fig. 3. Registration result of happy Buddha. (a) Model demonstration. (b) and (c) Interest points detection of source and target point cloud respectively. Colorful blobs show the result of top 15 clusters, the final interest points were shown in red crosses after rejection. Note that three correspondences can be found, since interests points can be found in the same region of source and target point cloud. (d) Initial state before registration, target and source point cloud were shown in red and green respectively. (e) Estimated coarse registration. (f) Estimated fine registration using ICP. (Color figure online)

Here we set $T_i = (1/30)R_d$, $T_c = (1/10)R_d$. Figures 3(b) and (c) show the results of clustering for happy Buddha, clusters are shown in colorful blobs. The target point cloud was scanned by laser scanner, and only the points in the front of the model were present. A rejection scheme is implemented to reject clusters according to two parameters for any cluster. First one $\phi_j = \sum_{i=1}^{L} (p_i - p_j)^2$ which measures the total distance from other clusters for cluster j, the second is $\psi_j = \max \sqrt{(p_m - p_n)^2}$ with $m, n \in j$, which indicates the diameter of cluster j. We compute the two parameters for all clusters, and reject the clusters that the ratio ψ_j/ϕ_j are larger than others. After the rejection, final detected interest points were supposed to be the most distinguishable points, and were shown in red crosses in Figs. 3(b) and (c).

3.2 CNN Architecture

The CNN classification model with architecture is shown in Fig. 4. The input to the network is $K_T \times K_T$ weighted adjacency matrix of interest points, we choose $K_T = 3$ here for preliminary experiment.

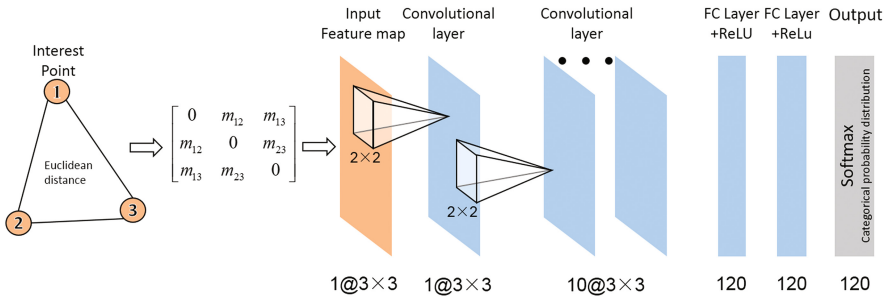


Fig. 4. Architecture of the CNN classification model. Layer's size may be changed according to the input size $K_T \times K_T$.

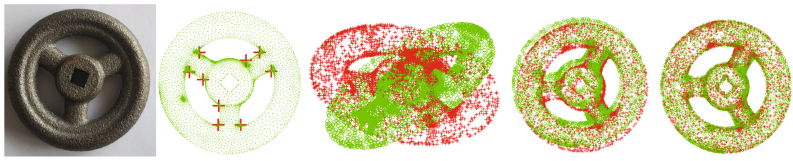
The first hidden layer convolves 10 filters of kernel size 2×2 with stride 1 and zero padding 1 with the input feature map, and then apply a rectified linear unit (abbreviated as ReLu).

The second layer is a convolutional layer of kernel size 2×2 with stride 1 and zero padding 1. Two fully-connected layer with 120 neurons is followed behind, and the final layer is a softmax layer.

The training data was generated from source point cloud by randomly choosing 10000 weighted adjacency matrices with corresponding categories mentioned in Sect. 2.3. A normalization is also implemented to reduce influence of the variations of scale of point clouds. We implement normalization by multiplying input feature map by a constant α inverse proportional to R_d so that $\alpha R_d = 200$.

3.3 Performance Analysis and Results

The proposed CNN model was trained from scratch, and after applied the proposed method to the test data, matching accuracy achieves 91%. Figures 3 and 5 shows the registration result of proposed method on both happy Buddha and the valve model. Since the number of interest points is relatively small, predicting correspondences from the CNN require less than 0.1 s on a 3.3 GHz Core i5 machine with 8 GB memory. We test the samples which have correct prediction by CNN, and reach the final RMS error 0.0082 (divided by R_d achieves relative error 4.1%) and angular-axis error 0.0837 in average without fine registration. Accuracy can be improved conceivably by increasing the number of interest points, in exchange for more time consumption and complexity of the CNN model. Tables 1 and 3 present an example of detected interest points in happy Buddha and valve model respectively, the prediction made by CNN is No.2,



(a) (b) (c) (d) (e)

Fig. 5. Registration results of valve model. (a) Model demonstration. (b) Interest points detected, shown in red crosses. (c) Initial state before registration, source and target point cloud were shown in '+' and 'o' respectively. (d) Estimated coarse registration. (e) Estimated fine registration using ICP. (Color figure online)

Table 1. Interest points detected in happy Buddha example.

Source	Position	Target	Position
1	(–0.0190, 0.0245, –0.0188)	1	(0.0224, 0.1132, –0.0093)
2	(0.0210, 0.1166, –0.0098)	2	(0.0300, 0.2395, –0.0138)
3	(–0.0074, 0.1880, 0.0004)	3	(–0.0241, 0.0563, 0.0259)
4	(0.0282, 0.1722, –0.0232)		
5	(–0.0312, 0.1151, –0.0233)		
6	(0.0257, 0.2421, –0.0166)		
7	(–0.0112, 0.0885, –0.0179)		
8	(–0.0199, 0.0566, 0.0200)		
9	(–0.0029, 0.0580, 0.0215)		
10	(–0.0190, 0.2448, –0.0188)		

Table 2. Weighted adjacency matrices in happy Buddha example. (Predicted points are the No.2, No.6, and No.8 points in the source set of Table 1).

Source	Target
$\begin{bmatrix} 0 & 0.126 & 0.078 \\ 0.126 & 0 & 0.194 \\ 0.078 & 0.194 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0.127 & 0.081 \\ 0.127 & 0 & 0.195 \\ 0.081 & 0.195 & 0 \end{bmatrix}$

No.6, and No.8 points in the source set for happy Buddha, No.1, No.6, No.8 points for valve model. Tables 2 and 4 present the corresponding matrices of target set and predicted points in source set. Comparing with the ground truth rigid transformation, the computed transformation using SVD is 0.0077 for RMS error, 0.0923 for angular-axis error.

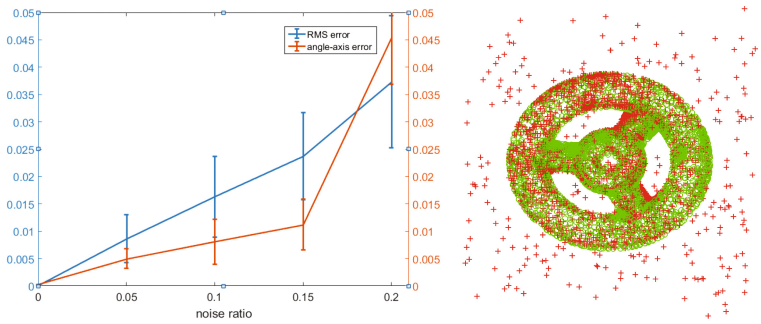
Research points out that point detectors may have the drawback of being sensitive to noise [5]. Experiments have been conducted on the valve model. We randomly generate considerable number of noise in the bounding box of point cloud and Fig. 6 shows the linear growth of error against noise. The experiments indicate that with the help of CNN, correspondences matching using local interest points can be robust to noise.

Table 3. Interest points detected in valve model example.

Source	Position	Target	Position
1	(–18.39, 4.877, 0.1504)	1	(–15.55, 3.789, 11.92)
2	(–12.17, 1.412, 0.1009)	2	(1.141, –6.646, –16.45)
3	(–13.84, 13.76, –0.1177)	3	(20.81, –1.991, 2.022)
4	(19.02, 5.172, –0.0981)		
5	(9.177, 10.65, –0.0247)		
6	(13.69, 13.85, –0.0095)		
7	(–4.979, –10.49, –0.0814)		
8	(–5.225, –19.22, 0.0657)		
9	(5.080, –19.26, –0.0089)		

Table 4. Weighted adjacency matrices in valve model. (Predicted points are the No.1, No.6, and No.8 points in the source set of Table 3).

Source	Target
$\begin{bmatrix} 0 & 33.31 & 38.10 \\ 33.31 & 0 & 27.46 \\ 38.10 & 27.46 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 34.53 & 38.12 \\ 34.53 & 0 & 27.38 \\ 38.12 & 27.38 & 0 \end{bmatrix}$



(a) (b)

Fig. 6. Result of test on sensitivity to noise. (a) RMS error and angle-axis error against noise in average, bars indicate the range of error. (b) Registration result with 20% of noise.

4 Conclusion

We proposed a 3D point cloud registration method, with convolutional neural network for correspondences matching. In this method, only interest points are required to be detected and no requirement for correspondences identification by point descriptors. The feature map of the CNN is the weighted adjacency matrix of complete graph generated by detected interest points. Experimental results show the effectiveness of our proposed method. This method presents a new potential application of CNN in

correspondences matching, where limitless ground truth data can be generate to be fed into CNN, and a set of interest points which are detected in target point cloud can be matched to the correct counterparts. Our future research includes utilizing other local descriptions, feature map representation, and some strategies focusing on rejecting interest points.

Acknowledgements. This work is partially supported by the National Natural Science Foundation of China (51375309).

References

1. Chaumette, F., Hutchinson, S.: Visual servo control. I. basic approaches. IEEE Robot. Autom. Magvol **13**(4), 83–90 (2006)
2. Jiang, J., Cheng, J., Chen, X.: Registration for 3-D point cloud using angular-invariant feature. Neurocomputing **72**, 3839–3844 (2009)
3. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3D Registration. In: IEEE International Conference on Robotics Automation, pp. 1848–1853 (2009)
4. Yang, J., Li, H., Jia, Y.: Go-ICP: solving 3D registration efficiently and globally optimally. IN: IEEE International Conference on Computer Vision, pp. 1457–1464 (2013)
5. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: efficient and robust 3D object recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 998–1005 (2010)
6. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992)
7. Miao, S., Wang, Z.J., Liao, R.: A CNN regression approach for real-time 2D/3D registration. IEEE Trans. Med. Image **35**(5), 1352–1363 (2016)
8. Boulch, A., Marlet, R.: Deep learning for robust normal estimation in unstructured point clouds. In: Eurographics Symposium on Geometry Processing, pp. 281–290 (2016)
9. Diez, Y., Roure, F., Llado, X., Salvi, J.: A qualitative review on 3D registration methods. ACM Comput. Surv. **47**(3), 45 (2015)
10. Qian, D., Chen, T., Qiao, H.: A new algorithm for non-rigid point matching using geodesic graph model. In: International Conference on Mechatronics and Automation, pp. 1174–1180 (2015)
11. Papazov, C., Burschka, D.: An efficient RANSAC for 3D object recognition in noisy and occluded scenes. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010. LNCS, vol. 6492, pp. 135–148. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19315-6_11
12. Y. Zhong.: Intrinsic shape signatures: A shape descriptor for 3D object recognition. International Conference on Computer Vision Workshop 3D representation Recognition, pp. 689–696 (2010)
13. 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. Vision **89**(2–3), 348–361 (2008)
14. Chen, H., Bhanu, B.: 3D free form object recognition in range images using local surface patches. Pattern Recogn. Lett. **28**(10), 1252–1262 (2007)

15. Salti, S., Tombari, F., Stefano, L.D.: A performance evaluation of 3D keypoint detectors. In: IEEE International Conference on 3D Imaging, Modeling, Processing, Visualization, and Transmission, pp. 236–243 (2011)
16. Pratt, W.K.: Digital Image Processing, 4th edn., pp. 590–595. Wiley, LosAltos (2007)
17. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-D point sets. IEEE Trans. Pattern Anal. Machine Intell. **9**, 698–700 (1987)