Stable Pose Estimation Using Ransac with Triple Point Feature Hash Maps and Symmetry Exploration

Ulrike Thomas
Institute of Robotics and Mechatronics (RMC)
German Aerospace Center (DLR), Wessling, Germany

Abstract

In this paper, a scene analyser is introduced which is based on Ransac (Random Sampling Consensus). This scene analysis approach is developed for robotic applications in particular, where poses of objects need to be estimated accurately that robots can grasp objects reliably. For assembly or manipulation purposes even an approximate pose estimation is not sufficient. For many objects appearance based similar poses exist, which influence the assembly strategy strongly when objects are gripped. Thus, robust pose estimation is required which is achieved by using triple point feature hash maps. This new feature vector is compared to two other feature vectors obtained from point pairs. It is shown that object poses can be estimated more precisely with roughly equal computation times with the new feature vector. Furthermore, in order to increase stability, symmetries are exploited and included into the entire scene analysis pipeline. The pipeline of the introduced scene analysis approach is illustrated and evaluated with various scenarios. The method presented here is successfully used for assembly applications.

1 Introduction

Interpretation of complex scenes and pose estimation of known objects is one of the main topics in computer vision for robotic applications. In industrial scenarios, in particular for assembly, often model data are available whereas in service scenarios model data are rare, but in every case one can think of a data base in which all these models are available. Hence the problem remains to match model information into 3D scenes. Nowadays 3D point clouds can be acquired easily by appropriate sensors like time-of-flight sensors (Swissranger SR 4000 or PMDs CamCube), Kinect-Sensor or passive stereo systems. Throughout this paper 3D point clouds are denoted by its point sets $P := \{\mathbf{p_1}, \ldots, \mathbf{p_n}\}$ and its surface normals $\mathcal{N} = \{\mathbf{n_1}, \ldots, \mathbf{n_n}\}$, which can be obtained by principle component analysis of its neighbour points. Pose estimation in general is equivalent to matching the correct model into the scene, more formally it is described by estimating the correct pose $\Theta := (\mathbf{R}, \mathbf{t})$ with $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$ which fits best into the point cloud P such that $\sum_{p_i \in M} || \mathbf{p}_i - \mathbf{p}_i^\star ||^2$ is minimized under certain constraints (e.g. collisions, physics) where $\mathbf{p}_i$ is a model point and $\mathbf{p}_i^\star$ is its closest scene point.

For pose estimation many Ranac or Prosac (progressive sampling consensus) approaches exist. Ransac is deeply investigated and can easily be implemented [4], [1]. Two key issues arise: How to draw samples and how to evaluate hypotheses. Moreover the generation of hypotheses is critical. Many feature descriptors for 6D pose estimation exist. Some of them are (fast) point feature histograms (PFH and FPFH) [9] [10], surflet pair relation tables [12] or point triplets. Their usage for pose estimation regarding accuracy is evaluated in [5]. All these features have been used widely in combination with Ransac for pose estimation in robotic applications [8], [2], [3]. For matching surfaces of broken pieces or registration of laser-scanner data from different viewpoints an approach applying the birthday attack is described in [13]. Ransac is implemented for finding shape primitives in point clouds [11]. Among the above mentioned feature descriptors spin images are introduced [6] or three-dimensional Tensors are applied for registration [7].

In this paper, a generic scene analysis approach is described. The image processing pipeline is illustrated in the next section. Section 3 discusses three different feature descriptors which are evaluated in this paper. The following section shows the Ransac step, namely the generation and evaluation of pose hypotheses. The later section contains the evaluation part, where real robotic scenarios are analysed regarding recognition rates. The paper is concluded in section 6.

2 The Image Processing Pipeline

For pose estimation and scene analysis a general approach is developed. Fig.1 illustrates the image processing chain. First the model data is hashed regarding the selected hash function and the applied feature descriptors. Among triangle meshes of object surfaces, the model data contain dense 3D surface points with their normals. The feature vectors are generated in advance. During runtime the 3D point cloud is acquired and surface normals are estimated. In this step, multiple images can either be registered and fused to a single unstructured point cloud, or one single image is used represented in each case by a 3D point cloud. At next, segmentation is applied. Here, an Euclidean algorithm is used. Ideally, each segment contains a single object. After segmentation the Ransac step follows, where each segment serves as a data pool for drawing pairs of points or triplets of points according to the used feature descriptors. Based on the gained hypotheses an evaluation step follows, where many hypotheses are rejected. This can either be done by pose clustering as in [3] or by other evaluation functions. Here cost functions are applied as filters to reject bad hypotheses. Ambiguous hypotheses are ignored and the remaining hypotheses are used to generate symmetrically equal hypotheses again, which are finally evaluated.

3 Feature Vectors

Pairs of points become very popular as feature vectors, because they can be computed very fast in point

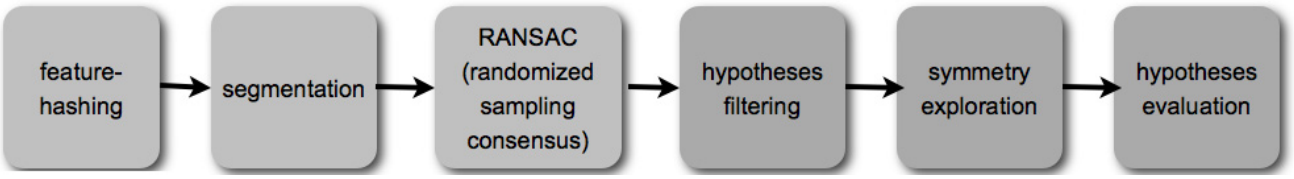

Figure 1. The image processing pipeline of the implemented scenen analyser.

clouds, where normal vectors of surfaces are available. Based on such features a hashing function is commonly applied to sort each feature vector into a bin. The more non-ambiguous such functions are, the more efficient the search for good hypotheses is. Certainly the optimum is a one to one mapping in between features and hypotheses, but this is not feasible. To this end various features and hash functions can be applied. Here, following features are evaluated and implemented in the image processing pipeline.

Surflet Pairs (SP): Fig.2 illustrates the surflet points pairs as feature vector. This vector is also used in [12], [13],[8]. Given two points denoted as $\mathbf{p}_i$ and $\mathbf{p}_j$ with their respective surface normals $\mathbf{n}_i$, $\mathbf{n}_j$ and their distance $\mathbf{d}_{ij} := \mathbf{p}_i - \mathbf{p}_j$, the feature vector is obtained by:

$$\begin{pmatrix} \| \mathbf{d}_{ij} \| \\ \angle(\mathbf{n}_i, \mathbf{d}_{ij}) \\ \angle(\mathbf{n}_j, \mathbf{d}_{ij}) \\ atan2(\ \mathbf{n}_i \cdot (\mathbf{d}_{ij} \times \mathbf{n}_j), (\mathbf{n}_i \times \mathbf{d}_{ij}) \cdot (\mathbf{d}_{ij} \times \mathbf{n}_j)) \end{pmatrix} \tag{1}$$

PF : Point Feature histograms are known from [9] and denoted as:

$$\begin{pmatrix} \| \mathbf{v} \cdot \mathbf{n}_j \| \\ \mathbf{u} \cdot \frac{\mathbf{p}_j - \mathbf{p}_i}{\|\mathbf{p}_j - \mathbf{p}_i\|} \\ atan2(\mathbf{w} \cdot \mathbf{n}_j, \mathbf{u} \cdot \mathbf{n}_j) \end{pmatrix} \tag{2}$$

where u,v,w are given by the Darboux-Frame coordinate system chosen at point $\mathbf{p}_i$[10]. If only those features are used whose point's distances are smaller than a threshold r the authors name it FPFH (fast point feature histogram). Here, in contrast, this vector is used for hashing, and we denote these features point features (PF), because of its locality.

Triple Points (TP): Point triples are evaluated in [5], but the feature vector differs from the one implemented here. The fourth and fifth dimensions are used to obtain much less ambiguity. In addition to the two given points mentioned above a third point $\mathbf{p}_k$ is chosen, hence the feature vector is denoted as:

$$\begin{pmatrix} \| \mathbf{d}_{ij} \| \\ \angle(\mathbf{n}_i, \mathbf{d}_{ij}) \\ \angle(\mathbf{n}_j, \mathbf{d}_{ij}) \\ atan2(\mathbf{n}_i \cdot (\mathbf{d}_{ij} \times \mathbf{n}_j), (\mathbf{n}_i \times \mathbf{d}_{ij}) \cdot (\mathbf{d}_{ij} \times \mathbf{n}_j)) \\ \angle(\mathbf{p}_i - \mathbf{p}_k, \mathbf{p}_j - \mathbf{p}_k) \end{pmatrix} \tag{3}$$

All three feature vectors are implemented and evaluated against each other. The first two representations are well known, whereas the third is a new feature vector used in this paper. As it can be seen in the evaluation section it improves accuracy for pose estimation and leads to much more stable results.

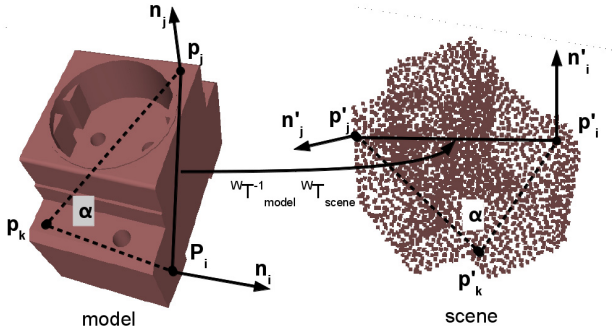

Figure 2. The feature used for hashing. According to SP, PF or TP two or three points are used to generate the feature vector.

Beside the feature vector an important issue is the mapping into the hash map. The optimal hashing will result in equally sized bins. Here the angular resolution is set to 5° for each bin and the translational resolution is set to $3mm$ regarding the sensor resolution. The maximum translational distance is obtained from the maximum distance $\epsilon_{\max}$ between two object points and a minimum value $\epsilon_{\min}$ which is restricted to a third of the $\epsilon_{\max}$. For the point feature vectors (PF) $r = \epsilon_{\min}$ is chosen as radius for the neighbours in order to avoid inserting too many point pairs.

4 Hypotheses Generation and Evaluation

Like in each typical Ransac implementation, the key steps are how to draw samples and how to evaluate hypotheses. Usually it is worse to reject hypotheses as early as possible in the processing pipeline. Thus, the non-ambiguity in the hash map is an important issue discussed in the last section and secondly drawing samples from the scene which leads to good hypotheses speeding up the entire scene analysis approach. Therefore, following steps for hypotheses generation, filtering and evaluation are implemented:

1. For all objects estimate $\epsilon_{\min}$ and ϵ_{max} which serve well as maximum distance for drawing point pairs or point triples.

2. For each segment do: Drawing one point $\mathbf{p}_i$ of the point cloud $P = \{\mathbf{p}_i \dots \mathbf{p}_n\}$ at random. Drawing a second and a third point out of a ball with

the radius given by either ϵ_{max} or by r (for point features) according to the matching model.

3. For the drawn point pairs or point triples compute the feature vectors either $f(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k)$ or $f(\mathbf{p}_i, \mathbf{p}_j)$ and try to find corresponding entries in the hash maps. Let $\mathcal{H}_k$ be a set of hypotheses for object k in the corresponding bin, then determine for each hypothesis the alignment for rigid motion with $\mathbf{R}$ and $\mathbf{t}$, see Fig.2. This pose hypothesis $\Theta := (\mathbf{R}, \mathbf{t})$ can be alignment with:

$$\mathbf{t}_\Delta = -\mathbf{p}_i^B - \frac{\mathbf{p}_j^B - \mathbf{p}_i^B}{2} + \mathbf{p}_i^A + \frac{\mathbf{p}_j^A - \mathbf{p}_i^A}{2} \quad (4)$$

and the rotational part is obtained by

$$\mathbf{R}_F^W := \left(\frac{\mathbf{d}_{ij}}{||\mathbf{d}_{ij}||} \quad \frac{\mathbf{d}_{ij} \times (\mathbf{n}_i \times n_j)}{||\mathbf{d}_{ij} \times (\mathbf{n}_i \times \mathbf{n}_j)||} \quad \frac{(\mathbf{d}_{ij} \times \mathbf{n}_i \times \mathbf{n}_j) \times \mathbf{d}_{ij}}{||(\mathbf{d}_{ij} \times \mathbf{n}_i \times \mathbf{n}_j) \times \mathbf{d}_{ij}||} \right) \quad (5)$$

By this, the estimated pose can be calculated with respect to the world's reference system by $\mathbf{R}_A^W \cdot \mathbf{R}_B^{W\,-1}$ leading to Θ, which is inserted into $\mathcal{H}_k$.

4. Now, hypotheses for all objects are collected in $\mathcal{H} := \cup_{\forall k} \mathcal{H}_k$ and evaluated with two different functions. The first one indicates how well the object matches into the scene and with the second function one obtains a quality measure how probable the hypothesis is regarding the viewpoint. The first cost function is denoted as

$$\frac{1}{\sum_{\mathbf{p}_i \in P_{seg}}} \sum_{\mathbf{p}_i \in P_{seg}} g(\mathbf{p}_i) \quad \text{with} \quad (6)$$

$$g(\mathbf{p}_i) = \begin{cases} 1 & \text{if} \quad \min_{\forall \mathbf{p}_j \in M} \{ || \mathbf{p}_j - \mathbf{p}_i || \} < \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

and the second cost function is given by assuming the view direction with $\mathbf{v}$. Then for the quality of a hypothesis $\Theta = (\mathbf{R}, \mathbf{t})$ follows

$$\frac{1}{\sum_{\mathbf{p}_j \in M | \mathbf{p}_j \cdot v > 0}} \sum_{\mathbf{p}_j \in M | \mathbf{p}_j \cdot \mathbf{v} > 0} || \mathbf{R}\mathbf{p}_j + \mathbf{t}_\Delta - \mathbf{p}_i^\star ||^2 , \quad (8)$$

where $\mathbf{p}_i^\star$ is the closest scene point.

These two functions are used as filters, where each hypothesis $h_i \in \mathcal{H}$ is rejected if it does not lead to a value above a certain threshold. All remaining hypotheses are collected in the set $\mathcal{H}_{best}$ which is sorted in descendent order $h_i \succ h_j \succ \ldots \succ h_{min}$ according to a weighted sum of both cost functions. More hypotheses are inserted into the set by exploiting the object' symmetries. Often objects with symmetrical similar poses need to be distinguished for handling and assembly. Applying only the Ransac step makes it difficult to distinguish between theses similar poses. For example the handle of the mug owns only few points in the scene, hence the basic Ransac algorithm will not find a difference between the poses where the object is rotated about its main axis. The same holds for other objects. The symmetrical hypotheses are added to the set $\mathcal{H}$ and filtered in such a way that non-ambiguous pose hypotheses, where objects do not collide survive. Moreover those hypothesis with highest evaluation values remain as estimation for the scene. Therewith more accurate hypotheses are found.

5 Results

For the first part of the evaluation the data set known from [7] is used. Analyses are provided for all illustrated scenes. The model data for the entirely used scenarios are depicted in Fig. 3. Fig.4 shows one example and plots the recognition rate according to feature vectors for the data set. The results state that the recognition rate is similar or better to the known rates, but the computation time is about 1 sec for each object, when the scene was sampled with a rate of 0.25 density.

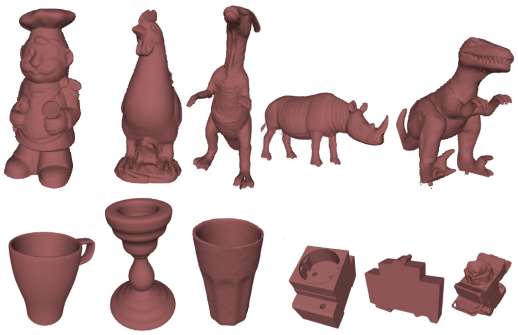

Figure 3. The upper row presents the model data from [7]. The lower row illustrates the models used here.

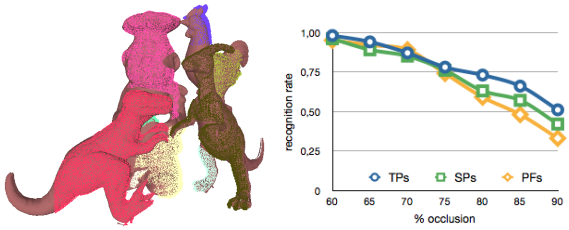

Figure 4. A quantitative analysis with the data set; left side: a recognized scene, right side the recognition rate achieved for the three different feature vectors.

As illustrated, the recognition rate is increased by using TPs. Also the benefit from this is demonstrated with a test illustrated in Fig. 5 which shows six cups in various poses. Although the objects are placed on the table the scene analysing approach estimates 6D poses. Here, symmetry exploration leads to estimated poses where the handle is matched correctly. In cases where the estimated pose is erroneous, seeking for better symmetrical hypotheses will not improve the estimated pose and the handle of the mug does not appear at the right position. Hence the number of correct can be regarded as a quality measure of estimated poses.

More scenes are depicted in Fig. 6 and Fig. 7. Tab. 1 contains their results, where the analysis is repeated 10 times. The recognition ratio is listed in the table. The quite high amount of false positive arises from a) the very similar geometries and b) from the fact that for some objects the accuracy of object models is not very high and c) from that the poses are counted as false positive if only e.g. the handle of the cup is

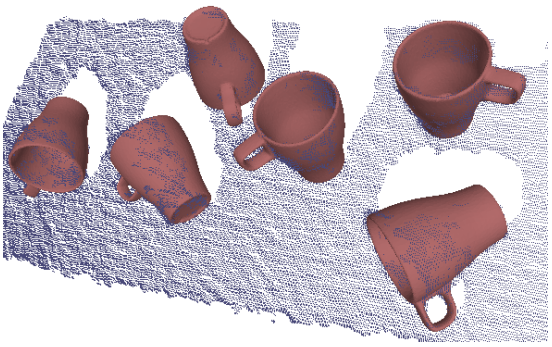

Figure 5. A: Cups placed somehow on a table.

not recognized on the right position. Altogether the recognition rates increase when Triple Point Features are used. For all experiments the number of iterations for Ransac is restricted to $20 \cdot 10^3$.

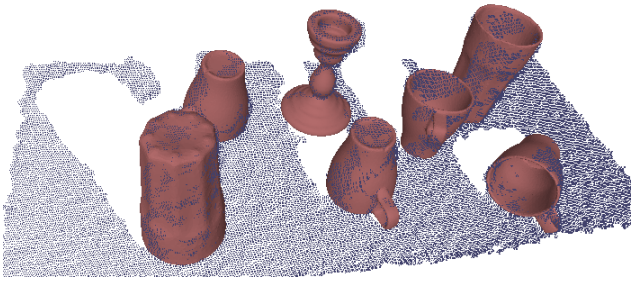

Figure 6. B: A service scenario, where three different objects are placed at random on the table.

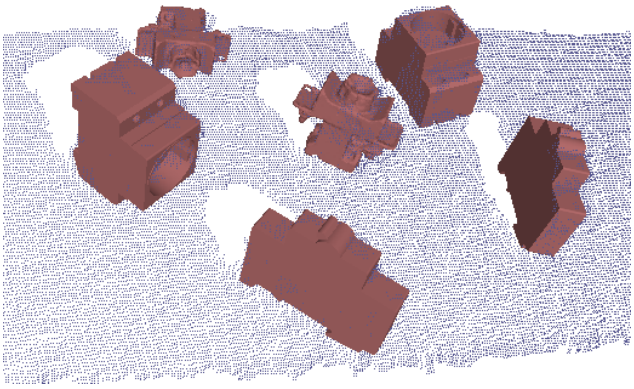

Figure 7. C: Some electrical parts, which have been grasped and plugged in an electrical cabinet assembly.

6 Conclusion

This paper presents a Ransac based scene analysis approach. For hashing feature vectors a new feature vector based on three points is introduced. In addition, symmetries are exploited to distinguish between geometrical similar but for handling purposes very different poses. The Ransac approach is robust and the exploration of similarity increases the recognition rate, which is important for robotic applications where robots have to grasp precisely. Further work will accelerate the process pipeline and moreover include texture information to distinguish between objects which only differ in their textures.

Table 1. Recognition Rate for all three examples.

Scene	PFs	SPs	TPs	$\approx \frac{time}{object}$ (TPs)
(A)	36/24	44/16	50/10	0.9 sec.
(B)	42/28	58/12	69/1	3.2 sec.
(C)	50/10	48/12	51/9	2.7 sec.

References

[1] R.C. Bolles and M. A. Fischler. A ransac-based approach to model fitting and its application to finding cylinders in range data. In *Proc. IJCAI*, pages 637–643, 1981.

[2] D. Buchholz, S. Winkelbach, and F. M. Wahl. Ransam for industrial bin-picking. In *ISR/Robotics 2010*, pages 1317–1322, 2010.

[3] B. Droste, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Effcient and robust 3d object recognition, 2010.

[4] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analy- sis and automated cartography. *CACM*, 24(6):381–395, 1981.

[5] U. Hillenbrand and A. Fuchs. An experimental study of four variants of pose clustering from dense range data. *Computer Vision and Image Understanding*, 115:1427–1448, 2011.

[6] A. E. Johnson and M. Herbert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transaction on pattern analysis and machine intelligence*, 21(5):433 – 449, 1999.

[7] A. S. Mian, M. Bennamoun, and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE Transaction on pattern analysis and machine intelligence*, 28(2):1584 – 1601, 2006.

[8] C. Papazov and D. Burschka. An efficient ransac for 3d object recognition in noisy and occluded scenes. In *ICCV*, 2010.

[9] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*, pages 3212 –3217, 2009.

[10] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3d recognition and pose estimation using the viewpoint feature histogram. In *IROS*, 2010.

[11] R. Schnabel, R. Wahl, and R. Klein. Efficient ransac for point cloud shape detection. In *Eurographics*, 2007.

[12] E. Wahl, U. Hillenbrand, and G. Hirzinger. Surflet-pair-relation histograms: a statistical 3d-shape representation for rapid classification. In *International Conference on 3-D Digital Imaging and Modeling – 3DIM 2003 IEEE Computer Society Press*, pages 474–481, 2003.

[13] S. Winkelbach, M. Rilk, C. Schoenfelder, and F. Wahl. Fast random sample matching of 3d fragments. In *PatternRecognition (DAGM 2004), Lecture Notes in Computer Science 3175*, pages 129–136, 2004.