Seminar Report:
Model globally, match locally: Efficient and robust 3D object recognition

Julian Ost

Chair of Computer Vision & Artificial Intelligence - Technische Universität München

Abstract

This reports presents the paper "Model globally, match locally: Efficient and robust 3D object recognition" by Bertram Drost, Markus Ulrich, Nassir Navab and Slobodan Ilic from 2010. Goal of the method presented in the paper is to recognize 3D free-form objects in a point cloud and recover the pose. The approach uses the advantages of two classes of previous approaches to introduce a highly efficient and also robust and stable method, which could not be achieved by those previous methods. This is due to a mapping of the model to a sparse feature space where faster search in the scene and matching between the model and point cloud is performed.
Evaluated against other methods the paper shows the advantages in terms of accuracy, speed and the possibility to trade off both characteristics.

1 Introduction

The paper "Model globally, match locally: Efficient and robust 3D object recognition" [1] from 2010 attacks the issues of recognizing three dimensional free-form objects in point clouds captured by any sensor. Previous methods used either local point descriptors or a global model description to recognize objects in the scene. While local approaches are more efficient and accurate compared to global ones, those methods highly depend on the quality and resolution of the model and the captured scene. Nevertheless due to the advantages over global approaches, those methods were quite popular prior this one.
In this method both approaches were combined and a global model described by features is matched locally to a captured scene recognizing and returning the pose of the model. This global model description consists of so called oriented point pair features, which are grouped to map similar features from a feature space to the model. A quite efficient voting scheme is than used to locally recognize the model on a sparse 2D search space in the captured point cloud. Besides a quite efficient search, results show high recognition performance and robustness against noise, occlusion and clutter compared to state of the art methods.

2 Related Work

Local approaches were the most popular approaches and state of the art when this paper was published because of their efficiency. Despite their advantages in efficiency those approaches lack in robustness and stability. Points and their neighborhood in the scene and the model are matched by so called point descriptors of every point and the surrounding surface. Those descriptors are matched between scene the model to recover the pose. So it highly depends on the quality of the captured scene for every point. Therefore noise can be a big problem recognizing points in the scene. Also occlusion by other objects or clutter capturing the scene could minimize the chance of matching a point. Changing the area around a point leads to the following results. A smaller radius leads to a higher dependence on Noise and a bigger radius increased the dependencies on occlusion.
The second class of methods were global approaches which were not that popular because those methods were neither very precise nor fast. Those approaches do not focus on a single point and the surrounding surface but the whole point cloud. One later example is the concept of so called Surflets [5], which is the combination of to points anywhere in the point cloud and similar to the concept used here. Also some applications use a "Generalized Hough Transformation", which is

similar to the here used voting scheme. Those global applications are limited in some way e.g. to standard objects like planes, spheres or cylinders and therefore not able to capture 3D free-form objects. Other need prior segmentation of the scene to use the extracted primitives, which is computational expensive. And most of the global approaches have in common, that they recover the pose by all 6 degrees of freedom which leads to high computational efforts. Besides that global recovered poses have low precision.

3 Method

To counter the disadvantages of both approaches the method combines the advantages of those. Objects are modeled by global features robust to occlusion and noise. But matching is done afterwards in a efficient way between local points in the scene and model. The process is described below:

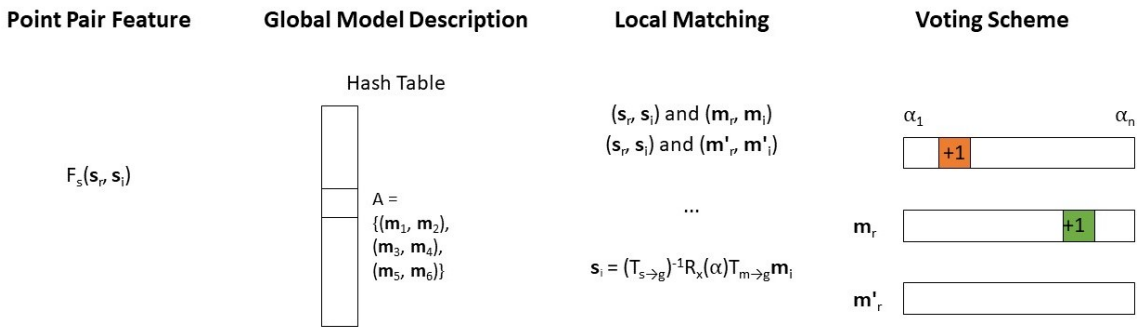

Figure 1: Overview of the method ad this section: 1. Point Pair Features; 2. Global Model Description; 3. Local Matching; 4. Voting Scheme and Clustering afterwards

The global model description consists of so called point pair features, describing a relation between any two points of the model. The features are stored in a hashing table in the off-line phase, which can be efficiently accessed. This global description is used in the on-line matching phase of the scene and the model to recover the pose through local coordinates and reduce the 6D pose problem to a 3D matching problem consisting od one point and angle.
Matched coordinates of point pairs are accumulated in a voting scheme to refine local coordinates for a point in the scene. All calculated poses are clustered and averaged, improving accuracy and return an optimal object pose.

3.1 Point Pair Feature

The scene and model are described by a finite set of oriented points, consisting of a point and a respective normal. Disadvantages of both previous approaches, relying on local surface information or expensive global methods are avoided by so called point pair features similar to surflet-pairs introduced by Wahl [5].
The point pair features describe the relative position and orientation of two oriented points with four arguments:

1. The distance as the absolute value of a distance vector d between both points
2. The Angle between the normal of the first point and the distance vector
3. The Angle between the normal of the second point and the distance vector

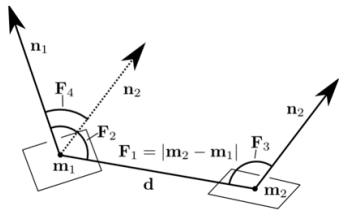

Figure 2: Point Pair of two orientated points [1]

4. The Angle between the normlas of points

$$\mathbf{F}(\mathbf{m}_1, \mathbf{m}_2) = (||\mathbf{d}||_2, \angle(\mathbf{n}_1, \mathbf{d}), \angle(\mathbf{n}_2, \mathbf{d}), \angle(\mathbf{n}_1, \mathbf{n}_2)) \tag{1}$$

The above introduced point pair feature shows an asymmetric property, which guarantees an unique feature for the exact order of two points.

3.2 Global Model Description

A four dimensional point pair feature $\mathbf{F}$ is calculated for any combination of two oriented points ($\mathbf{m}_\mathrm{i}$, $\mathbf{m}_\mathrm{j}$) on the model surface in a off-line phase. To access those point pair features efficiently and than recover the pose of an object through matching point pairs the global model description is stored in a hash table. Point pairs with similar features are grouped. Therefore the arguments of the feature vectors are discretized to $\mathbf{d}_\mathrm{dist} = \tau_d \cdot diam(M)$ for distances and $d_\mathrm{angle} = 2 \cdot \pi / n_\mathrm{angle}$ for angles. Choosing the parameters τ and $\mathrm{n}_\mathrm{angle}$ controls the size of the discrete feature space, which has some benefits described in the evaluation. Point pairs with similar feature vectors after discretizing are grouped together and stored in the same slot of a hash table. Each group of point pairs can be accessed in search efficiently using the discrete feature vectors as a key to the hash table. So, the global model description is a mapping from the four dimensional point pair feature space to a set of all point pairs (m_i, m_j).

3.3 Local Coordinates

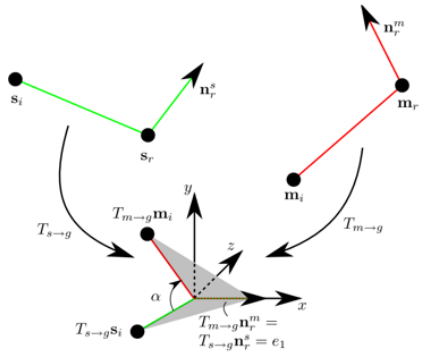

Figure 3: Transformation between model pose and object in the scene [1]

To recover the pose of an object points in the scene with are matched with model points. Therefore an arbitrary point s_r, the reference point in the scene, is picked with the goal to find the corresponding model reference point m_r and recover the pose of the model.
In a first step the point pair feature $F(s_r, s_i)$ of s_r and a random scene point s_i is calculated. The feature is than used to access the global model description and get all similar point pairs (m_r, m_i) of the model. (s_r, s_i) and (m_r, m_i) are aligned through a transformation in the following way.
To get the transformation between the pairs, the reference points of both point pairs, s_r and m_r, are translated into the origin and the respective normal vectors rotated onto the positive x-Axis of the origin frame through the transformation matrices $T_{s\to g}$ and $T_{m\to g}$.
A rotation around the x-axis with α aligns the point pairs. The complete transformation between the model and the object in the scene, given s_r, is described by α and the reference point m_r, the local coordinates, reducing the complete search for all parameters describing the pose to three parameters, compared to six in traditional approaches.
This method still needs some computational effort to calculate the angles in the on-line matching phase, which can be reduced doing part of the calculation off-line. In this method this is achieved by splitting the rotation into $\alpha = \alpha_m - \alpha_s$. The angles describe the rotation of the transformed point pair of the scene and the model into a plane defined by the x-Axis and the positive part y-Axis. While α_s is still computed on-line for every point pair, α_m can be computed and stored for every orientated point pair (m_r, m_i), while constructing the global model description

3.4 Voting Scheme

To increase accuracy and robustness of the local coordinate $(\mathbf{m}_r, \alpha)$ the reference point $\mathbf{s}_r$ is paired with any other point $\mathbf{s}_i$ of the scene and the local coordinates are calculated.
To find the optimal local coordinate a two dimensional accumulator array with rows over all model points and columns over sampled angles is created. After the rotation angle α is calculated a vote is cast for the local coordinate $(\mathbf{m}_r, \alpha)$. Voting for every point pair $(\mathbf{s}_r, \mathbf{s}_i)$ in the scene, the peaks in the array describe the optimal local coordinate and are used to recover the global pose of the model in the scene. The paper proposes to increase stability using multiple peaks above a certain threshold.

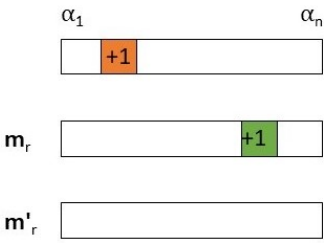

Figure 4: Accumulator Array of local coordinates with N_{points} rows and n_{angle} columns

3.5 Pose Clustering

The voting scheme described before only uses one reference point in the scene and is based on the assumption that this point lies on an objects surface. To ensure that a reference point lies on a models surface and to recognize multiple instances of an object the matching and voting is performed for multiple reference points. The optimal poses for multiple $\mathbf{s}_r$ are clustered by a translational and rotational threshold. For every cluster the votes are accumulated and local coordinates are averaged. The biggest clusters with the highest scores are returned and describe the pose of all instances of the model in the scene. Returning the biggest clusters also removes isolated poses with low scores and increases the accuracy of the method by averaging the poses of every cluster. Choosing the number of reference points can improve robustness and accuracy or increase efficiency. This trade off is shown in the evaluation.

4 Evaluation

Goal of the evaluation is to show how the method performs against sets of synthetic and real data and the up and downsides of this method. It is compared against other methods, the spin-images by Johnson and Hebert[2] and tensor matching of Mian etal. [3]. Also the effects on the method through changing the following parameters is shown:

- τ, the sampling parameter for the discrete distance
- n_angle, the number of discrete angles
- the number of reference points used for pose clustering

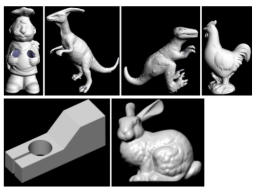

Figure 5: Models used for the evaluation [1]

The evaluation is performed on parts of the data set after Mian et al., synthetic and real data captured by Drost like the clamp shown in 5. The default parameters are set to: $\tau = 0.05$, $n_{angle} = 30 (\rightarrow \Delta\angle = 12\,\text{deg})$ and 1/3 of the scene points $|S|$ are used as reference points.
The point cloud of the model and captured scene were sub sampled again, such that the minimum distance between points $d_{dist} = \tau \cdot diam(M)$ and new normals were calculated for those points

5 Results

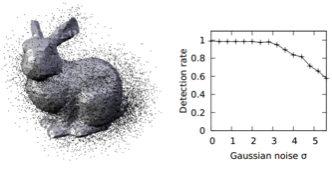

Figure 6: Noise on a object [1]

The first evaluation is performed on synthetic data against robustness against noise and occlusion. Therefore Gaussian noise is added to all points of the scene. In this experiment four different objects are used and each has been captured from 50 directions, resulting in 200 point clouds. The distribution of the noise has been chosen relative to the objects diameter and added prior to the sub sampling process. A object is counted as

detected if the error of the pose is smaller than a given threshold. The results show robustness to a certain level of noise 6.

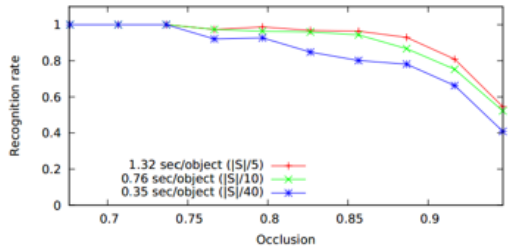

Figure 7: Recognition rate with respect to occlusion on synthetic data [1]

The second evaluation on synthetic data shows the influence of the number of reference points used for the pose clustering with respect to the degree of occlusion and clutter. Therefore the method acted on 50 scene with four to nine objects placed in each. Those influences are defined as follows:

- Occlusion: $1 - \frac{\text{model surface area in the scene}}{\text{total model surface area}}$
- Clutter: $1 - \frac{\text{model surface area in the scene}}{\text{total surface area of scene}}$

Using the method with $|S|/5$ reference points recognized nearly 90% of all objects, only highly occluded ones were missed. Setting $|S|/40$ the method only recognizes 77% of all objects, but more than 80% for over 15% visible objects were detected and the computation was 4 times faster. So changing number of the reference points used for pose clustering lowers recognition rate but increases speed. A trade off between both is possible through this parameter. The comparison of the method with the methods Tensor Matching, using multidimensional table representations of the model (Mian et al.[3]), and Spin images, matching surfaces represented as a rotational projection around normal vectors (Johnson and Herbert [2]), was performed on real data of 50 scenes by Mian w.r.t. clutter and occlusion.
Besides comparing to both other methods, the influence and benefit of varying the sampling rate τ_d can be shown 8. For $\tau = 0.025$ this method performed alike the tensor matching and both a lot better than spin images. The recognition rate of this method increased slightly and performed as fast as the method of Mian et al. Increasing τ to 0.04 the method still performed significantly better than the spin images for objects which were less than 15% occluded but worse compared to tensor matching. Nevertheless the matching is performed 40 times faster, which shows the main advantage of this method: It is possible to trade between speed and recognition rate.
The presented method has been analyzed qualitatively on actual captured data by Drost to show useability in real scenarios e.g. robotics. The Experiments on a self-build laser scanning setup showed accurate results for object manipulation despite a lot of clutter and occlusion.

6 Summary

The paper introduced an efficient, stable and accurate method to find free-form 3D objects in point clouds, which is independent from local surface information and also very fast matching through locally reduced search space. Also better recognition rates where achieved in comparison to traditional local and global approaches. A main advantage of this method is a possible trade between accuracy and speed through several parameters.
Although the method was introduces in 2010 a recent paper by Vidal [4] from 2018 shows the best performance for pose estimation in 3D point clouds with this method and successors. Further improvement in computational speed through a faster implementation in C++ and palatalization e.g. of calculating local coordinates and several reference points has been proposed. Accuracy could also be improved by redefining the poses with e.g. ICP.

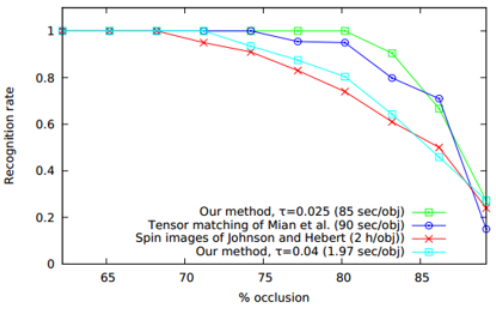

Figure 8: Recognition rate w.r.t occlusion on real data compared to Tensor Matching, Spin Images and different parameter τ [1]

References

[1] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3D object recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 998–1005. IEEE, 6/13/2010 - 6/18/2010.

[2] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[3] A. Mian, M. Bennamoun, and R. Owens. On the Repeatability and Quality of Keypoints for Local Feature-based 3D Object Retrieval from Cluttered Scenes. *International Journal of Computer Vision*, 89(2-3):348–361, 2010.

[4] Joel Vidal, Chyi-Yeu Lin, Xavier Lladó, and Robert Martí. A Method for 6D Pose Estimation of Free-Form Rigid Objects Using Point Pair Features on Range Data. *Sensors (Basel, Switzerland)*, 18(8), 2018.

[5] E. Wahl, U. Hillenbrand, and G. Hirzinger. Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification. In *Fourth international conference on 3-D digital imaging and modeling*, pages 474–481. IEEE, 2003.