Fast performance 3D object recognition: Dealing with rotationally symmetric objects

Rui Pimentel de Figueiredo
ruifigueiredo@isr.ist.utl.pt

Plinio Moreno
plinio@isr.ist.utl.pt

Alexandre Bernardino
alex@isr.ist.utl.pt

Instituto de Sistemas e Robótica
Instituto Superior Técnico
Lisboa, PT

Abstract

In this paper we extend a recent approach for 3D object recognition in order to deal with rotationally symmetric objects. These are frequent in daily environments and can be recognized with significant computational savings if symmetrical features are jointly represented. We show improvements up to 120x with respect to state-of-the-art methods.

1 Introduction

In recent past, Drost *et al*. [3] proposed an approach which extracts description from a given object model, using point pair features [6], which encode the geometric relation between oriented point pairs. The matching process is done locally using an efficient voting scheme similar to the Generalized Hough Transform (GHT) [1]. Their method is robust to sensor noise and outperforms other feature-based state-of-the-art methods like Spin Images [4] and Tensors [5], both in terms of robustness to occlusion and clutter and in terms of computational speed.

In this paper we introduce an important extension to [3] for dealing efficiently with rotationally symmetric objects, which are common in many environments (e.g. kitchenware objects like cups, glasses, cans, plates). We drastically reduce the computational effort of [3] when dealing with this kind of objects.

2 Method Overview

Each object model is represented by a set of points and associated surface normals, i.e. surflets [6]. Let M be the set of all model surflets, $M = \{s_i^m, i = 1..N\}$ – upper indices m and s will be further used to distinguish model from scene, respectively. An object description suitable for object recognition and pose estimation is created through the analysis of all possible permutations of surflet pairs. Let A be the set of all surflet pairs, $A = \{(s_r^m, s_t^m), r \neq t\}$, which has cardinality $|A| = N \times (N-1)$.

2.1 Model Description

For each surflet pair (s_r, s_t), we compute a descriptive 4-element feature vector as illustrated in figure 1. This could be formally described by the following expression:

$$F(s_r, s_t) = (f_1, f_2, f_3, f_4) = (\|d\|, \angle(n_r, d), \angle(n_t, d), \angle(n_r, n_t)) \quad (1)$$

The data structure used to represent the model description is a hash table for quick retrieval, in which the key value is given by the discrete point pair feature while the mapped value is the respective surflet pair. Since one key could be associated with several model surflet pairs, each slot of the hash table contains a list of surflet pairs with similar discrete feature.

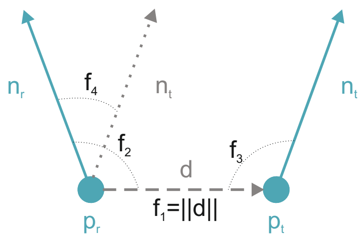

Figure 1: Point pair feature descriptor

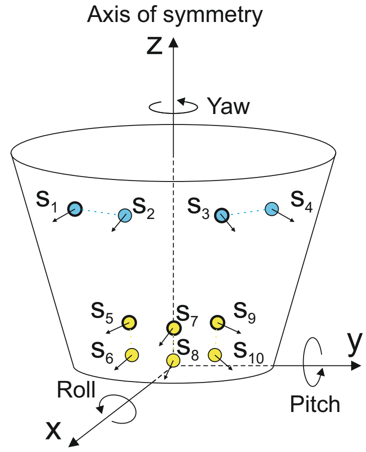

Figure 2: An example of a rotationally symmetric object model. All illustrated surflet pairs have similar discrete feature. In the figure, pairs represented with similar color are redundant.

2.1.1 Dealing with rotational symmetry

In order to efficiently deal with rotationally symmetric objects, we incorporate a strategy that reduces drastically the size of A, by discarding redundant surflet pairs, thus increasing dramatically the recognition runtime performance. To accomplish this, a Euler angle representation [2], is used to describe orientation. In our work we chose the X-Y-Z Euler representation since we assume that the object axis of symmetry is aligned with the z axis of the object reference coordinate frame. During the creation of the model description, for each surflet pair, we compute the transformation with respect to the object model reference frame (see section 2.2) that aligns it with each similar pair already stored in the hash table. If there is at least one pair for which the alignment transformation has no translation and no roll and pitch components on the rotation as expressed on eqs. (2) and (3) respectively, then, this surflet pair corresponds to a rotation of the other (homologous) around the symmetry axis, and is therefore redundant and discarded.

$$d < d_{\text{th}} \quad (2)$$

$$\alpha_{\text{yaw}=0} < \alpha_{\text{th}} \quad (3)$$

The weight of the homologous surflet pair, stored in the hash table, is then incremented by 1. This process is clearly ilustrated in figures 2 and 3.

2.2 Pose Estimation

A set of reference surflets on the scene $R_s \subset S$ – where S is the set of all scene surflets, $S = \{s_i^s, i = 1..N\}$ – is randomly chosen and each of them is paired with all the other surflets on the scene. For each scene surflet pair $(s_r^s, s_t^s) \in S^2$ we compute a point pair feature $F(s_r^s, s_t^s)$ and then, using the extracted feature, we obtain a set of model surflet pairs whose feature is similar to it. From every match between a scene surflet pair $(s_r^s, s_t^s) \in S^2$ and a model surflet pair $(s_r^m, s_t^m) \in M^2$, we are able to extract the rigid transformation that aligns the matched model with the scene. This is done by first computing the transformations $T_{m\to g}$ and $T_{s\to g}$ that align s_r^m and s_r^s, respectively, to the object reference coordinate frame x axis, and secondly the rotation α around the x axis that aligns p_t^m with p_t^s. The final transformation that aligns the model with the scene is then computed considering the ensuing expression:

$$T_{m\to s} = T_{s\to g}^{-1} R(\alpha) T_{m\to g} \quad (4)$$

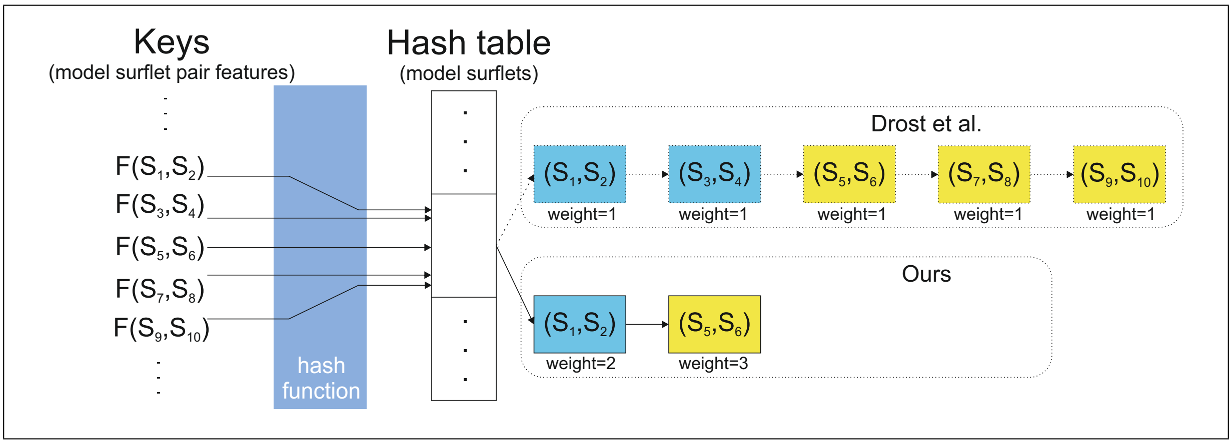

Figure 3: Example of surflet pairs with similar feature stored in the same slot of the hash table, during the creation of the object model description.

The transformations $T_{m\rightarrow g}$ and $T_{s\rightarrow g}$ translate p_r^m and p_r^s, respectively, to the reference coordinate frame origin and rotates their normals n_r^m and n_r^s onto the x axis. After applying these two transformations, p_t^m and p_t^s are still misaligned. The transformation $R(\alpha)$ applies the final rotation needed to align these two points.

The transformation expressed in eq. (4) can be parametrized by a surflet on the model and a rotation angle α. In [3], this pair (s_r^m, α) is mentioned as the *local coordinates* of the model with respect to s_r^s.

2.2.1 Voting Scheme

This method uses a voting scheme similar to the GHT for pose estimation. For each scene reference surflet, a two-dimensional accumulator array that represents the discrete space of local coordinates is created. The number of rows, N_m, is the same as the number of model sample surflets $|M|$, and the number of columns N_{angle} is equal to the number of sample steps of the rotation angle α.

The voting procedure goes as follows: considering a given reference surflet s_r^s on the scene surface S, we pair it with every other surflet $s_t^s \in S$. For each resulting surflet pair we search on the model surface for similar surflet pairs, with the aim of finding where it might be in the model. This is done by querying the model descriptor for surflet pairs with similar feature. The computed feature $F(s_r^s, s_t^s)$ is used as an index to the model hash table and a list of matched surflet pairs, with similar feature, is returned. For every match (s_r^m, s_t^m) the rotation angle α is computed and a vote is placed in the accumulator array by incrementing the position correspondent to the local coordinates (s_r^m, α), by the *weight* of the matched model surflet pair. After pairing s_r^s with all s_t^s, the highest peak – i.e. the position with more votes – in the accumulator corresponds to the optimal local coordinate. In the end, all retrieved pose hypotheses whose position and orientation do not differ more than a predefined threshold are clustered together.

To deal with symmetry, before clustering, we collapse all redundant hypotheses to a single pose. This additional step removes the rotational component around the object axis of symmetry, i.e. yaw, ensuring that all redundant poses are gathered in the same cluster, therefore allocating less resources and reducing the number of computations.

3 Results

To evaluate the quality of the poses recovered by the algorithm and its runtime performance, we generated 200 synthetic scenes containing a single instance of a symmetric cup. Each scene was then corrupted by different levels of additive Gaussian noise, with standard deviation proportional to the model size. By using synthetically generated scenes, we were able to compare the algorithm pose results with a known ground truth. During recognition we chose 5% of the scene points as reference points. A higher percentage would increase the robustness to noise but also the recognition runtime. A recovered pose was considered to be correct if the error relative to the ground truth pose was smaller than diameter$(M)/10$ for the position and 12^{o} for the orientation. Thresholds from expressions (2) and (3) were set to 12^{o} and diameter$(M)/40$, respectively. The method was implemented in C++ and the experiment was run on a single core of a dual-core 2.6 GHz computer with 4GB of RAM. We were able to discard near 93% surflet pairs during the creation of the model description, and reduce the number of computations during pose recognition. As shown in figure 4, the recognition rate drops lightly for high levels of noise due to sampling effects, but the recognition time performance increases significantly. For $|S| \approx 5000$, our method achieves a recognition time 120 times faster than [3].

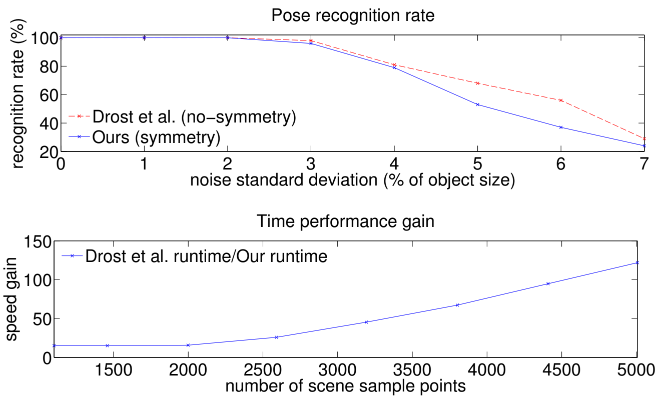

Figure 4: Comparison results of our method against the method of Drost *et al.*, with $|R_s| = 0.05\,|S|$ reference points.

4 Acknowledgements

This work was supported by the FCT project [PEst-OE/EEI/LA0009/2011].

References

[1] D H Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[2] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., 1989.

[3] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 998 –1005, June 2010.

[4] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 433–449, 1999.

[5] Ajmal S. Mian, Mohammed Bennamoun, and Robyn Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE Trans. Pattern Anal. Mach. Intell*, 28:1584–1601, 2006.

[6] Eric Wahl, Ulrich Hillenbrand, and Gerd Hirzinger. Surflet-pair-relation histograms: A statistical 3d-shape representation for rapid classification. *3D Digital Imaging and Modeling, International Conference on*, 0:474, 2003.