Fast 3D object recognition of rotationally symmetric objects

Rui Pimentel de Figueiredo, Plinio Moreno, and Alexandre Bernardino *

Institute for Systems and Robotics, Instituto Superior Tcnico, Lisboa, Portugal
{ruifigueiredo,plinio,alex}@isr.ist.utl.pt

Abstract. In this paper we extend a recent approach for 3D object recognition in order to deal with rotationally symmetric objects, which are frequent in daily environments. We base our work in a recent method that represents objects using a hash table of shape features, which in the case of symmetric objects contains redundant information. We propose a way to remove redundant features by adding a weight factor for each set of symmetric features. The removal procedure leads to significant computational savings while keeping the recognition perfomance. The experiments show recognition time improvements up to 300x with respect to state-of-the-art methods.

1 Introduction

3-D object recognition plays a role of major importance in the robotics field. Many applications, such as object grasping and manipulation, critically depend on visual perception algorithms. These, must be robust to cluttered environments and to sensor noise, as well as fast enough for real-time operation, in order for the robot to correctly interact with the surrounding environment. During the last decades, several methods have been proposed to solve the object recognition problem, but it is still a very challenging task and many research efforts continue to be made. Due to recent technological advances in the field of 3-D sensing, range sensors provide 3-D points with reasonable quality and high sampling rates, sufficient for efficient shape-based object recognition. In recent past, Drost *et al.* [1] proposed an approach which extracts description from a given object model, using point pair features, encoding the geometric relation between oriented point pairs. The matching process is done locally using an efficient voting scheme (see Fig. 3) similar to the Generalized Hough Transform (GHT) [2]. Their method is robust to sensor noise and outperforms other feature-based state-of-the-art methods like Spin Images [3] and Tensors [4], both in terms of computational speed in terms, robustness to occlusion and clutter.

In this paper we introduce an important extension to [1] for dealing efficiently with rotationally symmetric objects, which are common in many daily tasks (e.g.

* This work is supported by the European Community's 7th Framework Programme, grant agreement First-MM-248258 and by project FCT [PEst-OE/EEI/LA0009/2011].

kitchenware objects like cups, glasses, cans, plates). We drastically reduce the computational effort of [1] when dealing with this kind of objects.

Next section overviews the Drost *et al.* object recognition and pose estimation algorithm. Then, in section 3 we propose a methodology to efficiently deal with rotational symmetries. Lastly, in section 4 we show results that validate our approach.

2 Method Overview

The basic units to describe surface shape are surflets [5] $\mathbf{s} = (\mathbf{p}, \mathbf{n})$, where $\mathbf{p}$ represents sample points in the surface and $\mathbf{n}$ are the associated surface normals. Let M be the set of all model surflets, $M = \{\mathbf{s}_i^m, i = 1..N\}$ and let S be the set of all scene surflets, $S = \{\mathbf{s}_i^s, i = 1...N\}$.

The recognition process consists in matching scene surflet pairs $(\mathrm{s}_r^s, \mathrm{s}_i^s)$ to model surflet pairs $(\mathrm{s}_r^m, \mathrm{s}_i^m)$. Being $\mathbf{s}_r$ and $\mathbf{s}_i$ two surflets, the Point Pair Feature (PPF) $\mathbf{F} \in F \subset \mathbb{R}^4$ is defined as a 4-tuple composed by the distance between the reference, $\mathbf{p}_r$, and secondary, $\mathbf{p}_i$, points and the angle between the normal of the reference point $\mathbf{n}_r$ and the vector $\mathbf{d} = |\mathbf{p}_i - \mathbf{p}_r|$, the angle between the normal of the secondary point $\mathbf{n}_i$ and $\mathbf{d}$ and the angle between $\mathbf{n}_r$ and $\mathbf{n}_i$ as illustrated in Fig. 1. This could be formally described by

$$\mathbf{F} = \mathrm{PPF}(\mathbf{s}_r, \mathbf{s}_i) = (f_1, f_2, f_3, f_4) = (\|\mathbf{d}\|, \angle(\mathbf{n}_r, \mathbf{d}), \angle(\mathbf{n}_i, \mathbf{d}), \angle(\mathbf{n}_r, \mathbf{n}_i)) \quad (1)$$

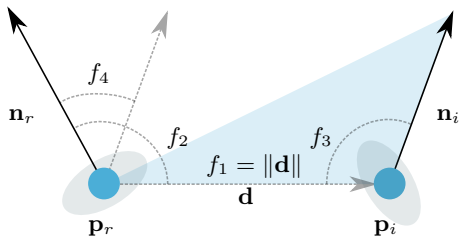


Fig. 1. Point Pair Feature

In section 2.1 the off-line model description creation is briefly described. Then, in section 2.2 we describe the on-line object recognition and pose estimation step.

2.1 Model Description

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 model surflet pairs, $A = \{(\mathbf{s}_r^m, \mathbf{s}_i^m), r \neq i\}$, which has cardinality $|A| = N \times (N - 1)$.

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 PPF 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.

2.2 Pose Estimation

A set of reference surflets on the scene $R_s \subset S$ is uniformly sampled from S and each of them is paired with all the other surflets on the scene. The number of reference points is given by $|R_s| = \xi |S|$ where $\xi \in [0; 1]$ is the reference points sampling ratio control parameter. For each scene surflet pair $(\mathbf{s}_r^s, \mathbf{s}_i^s) \in S^2$, $\text{PPF}(\mathbf{s}_r^s, \mathbf{s}_i^s)$ is computed and set of similar model surflet pairs is retrieved from the hash table. From every match between a scene surflet pair $(\mathbf{s}_r^s, \mathbf{s}_i^s) \in S^2$ and a model surflet pair $(\mathbf{s}_r^m, \mathbf{s}_i^m) \in M^2$, one is able to compute the rigid transformation that aligns the matched model with the scene. This is done first by computing the transformations $\mathbf{T}_{m\to g}$ and $\mathbf{T}_{s\to g}$ that align $\mathbf{s}_r^m$ and $\mathbf{s}_r^s$, respectively, to the object reference coordinate frame x axis, and secondly by computing the rotation α around the x axis that aligns $\mathbf{p}_i^m$ with $\mathbf{p}_i^s$. The transformation that aligns the model with the scene is then computed considering the ensuing expression:

$$\mathbf{T}_{m\to s} = \mathbf{T}_{s\to g}^{-1} \mathbf{R}(\alpha) \mathbf{T}_{m\to g} \tag{2}$$

In detail, the transformations $\mathbf{T}_{m\to g}$ and $\mathbf{T}_{s\to g}$ translate $\mathbf{p}_r^m$ and $\mathbf{p}_r^s$, respectively, to the reference coordinate frame origin and rotates their normals $\mathbf{n}_r^m$ and $\mathbf{n}_r^s$ onto the x axis. After applying these two transformations, $\mathbf{p}_i^m$ and $\mathbf{p}_i^s$ are still misaligned. The transformation $\mathbf{R}(\alpha)$ applies the final rotation needed to align these two points. The previous reasoning is depicted in Fig. 2. The transformation expressed in eq. (2) can be parametrized by a surflet on the model and a rotation angle α. In [1], this pair $(\mathbf{s}_r^m, \alpha)$ is mentioned as the *local coordinates* of the model with respect to reference point $\mathbf{s}_r^s$.

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. A vote is placed in the accumulator array by incrementing the position correspondent to the local coordinates $(\mathbf{s}_r^m, \alpha)$, by 1 (see Fig. 3). After pairing $\mathbf{s}_r^s$ with all $\mathbf{s}_i^s$, 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.

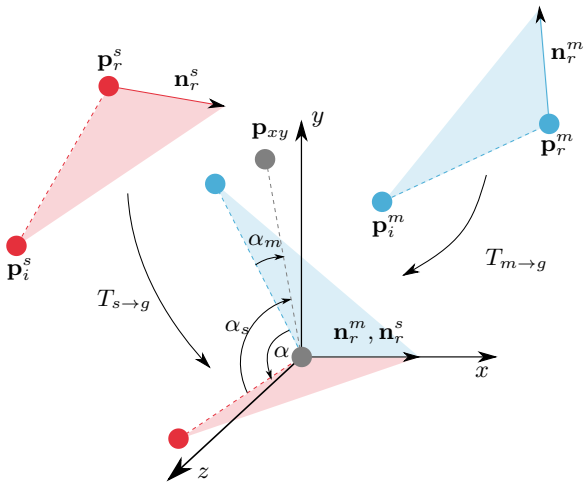


Fig. 2. Pose acquisition by surflet pair aligment

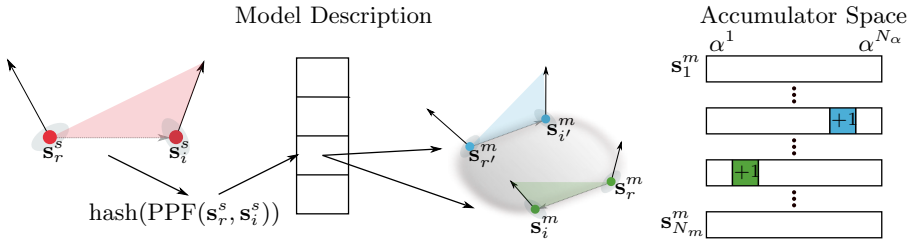


Fig. 3. Feature Matching and Voting Scheme

3 Dealing with Rotational Symmetry

We consider an object to be rotationally symmetric if its shape appearance is invariant to rotations around a given axis of symmetry (see Fig. 4). In order to efficiently deal with this kind of objects, we incorporate a strategy that reduces drastically the size of the model description D, by discarding redundant surflet pairs, thus increasing dramatically the recognition runtime performance. To accomplish this, a Euler angle representation [6], 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 the aligning transformation has a very low translation $\mathbf{t}$ and if the roll and pitch rotational components, ϕ_{roll} and ϕ_{pitch} respectively, are close to 0, then this surflet pair corresponds to a rotation around the symmetry axis. Thus,

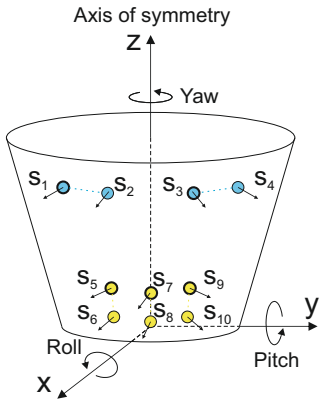


Fig. 4. An example of a rotationally symmetric object model. All illustrated surflet pairs have similar discrete feature. In the figure, pairs represented with similar color are redundant.

the surflet pair is redundant and therefore discarded.

$$\phi_{\text{roll}} < \phi_{\text{th}} \quad \text{and} \quad \phi_{\text{pitch}} < \phi_{\text{th}} \quad \text{and} \quad \|\mathbf{t}\| < t_{\text{th}}\text{diam}(M) \tag{3}$$

where $\text{diam}(M)$ is the maximal distance between model points. The weight w of the homologous surflet pair, stored in the hash table, is then incremented by 1. This process is clearly illustrated in Fig. 5.

Due to the fact that the sampled model point clouds are not perfect, *i.e.*, only approximate the true shape of the object, the α_{th} and d_{th} thresholds must take into account these sampling inperfections. Higher thresholds increase the number of jointly represented surflet pairs but reduce the stringency with which we consider two given model surflet pairs redundant.

By representing redundant features jointly we decrease the number of feature matches thus decreasing the computations during the voting process. Each feature match contributes with a weight equal to the model feature weight, w, instead of 1. The peaks in the accumulator – originated by redundant surflet pairs – which were previously scattered throughout the local coordinates are now concentrated at single local coordinates. This is the result of keeping only one surflet pair, *i.e.*, one local coordinate, representing all the respective discarded redundant ones which correspond to different local coordinates.

Before clustering, we collapse all poses that only differ on the yaw component, *i.e.*, redundant hypotheses, to a single pose. This is achieved by means of an additional step – after knowing the final transformation $\mathbf{T}_{m\to s}$ (see equation (2)) – which removes the rotational component around the object axis of symmetry, *i.e.*, $\phi_{yaw} = 0$, ensuring that all redundant poses are gathered in the same cluster, therefore allocating less resources and reducing the number of computations during the pose clustering step.

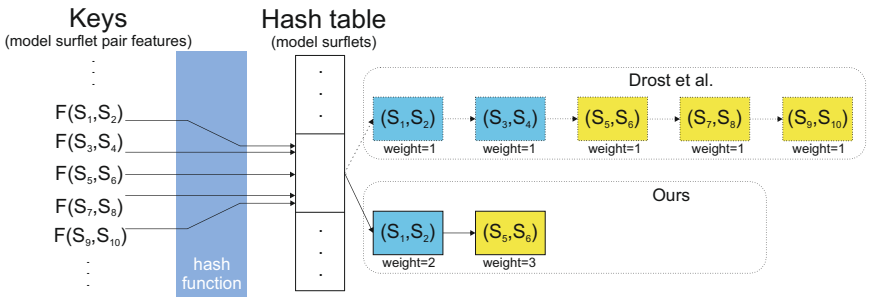
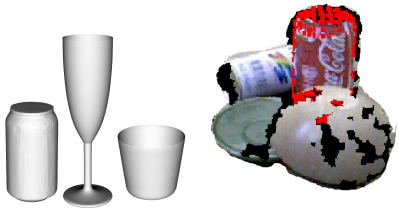


Fig. 5. Example of surflet pairs with similar feature stored in the same slot of the hash table, during the creation of the object model description.

4 Results

To evaluate the performance gains of the proposed strategies to handle rotationally symmetries efficiently, in the presence of noisy visual sensors, we created an experimental scenario similar to the one referred in [1]. In this experimental scenario the models library comprises only one model at a time, since we were interested in evaluating the quality of the poses recovered by the algorithms. With this purpose, we generated 200 synthetic scenes containing a single instance of a given model from the ROS household objects library (see Fig. 6(a)) [7], on a random pose. Before the downsampling step, each scene was corrupted by different levels of additive gaussian noise, with standard deviation σ proportional to the model diameter diam(M). By using synthetically generated scenes, we were able to compare the algorithm pose results with a known ground truth.



(a) (b)

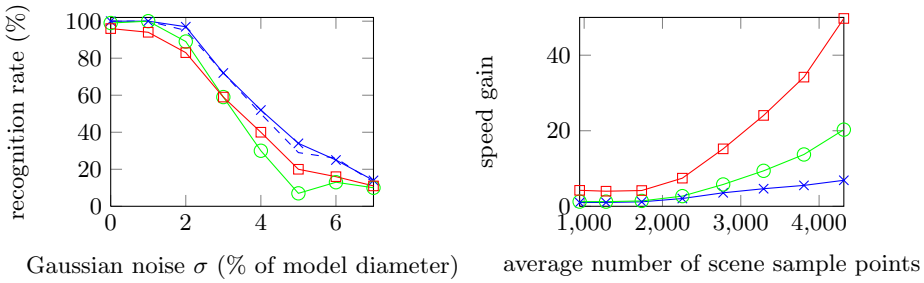
Fig. 6. (a) ROS Household object models. From left to right: coke can, champagne glass and cup. (b) Our method correctly detecting a Coca-Cola can. Figure best seen in color.

During recognition we chose 5% of the scene points as reference points by setting ξ to 0.05. A higher percentage would increase the robustness to noise

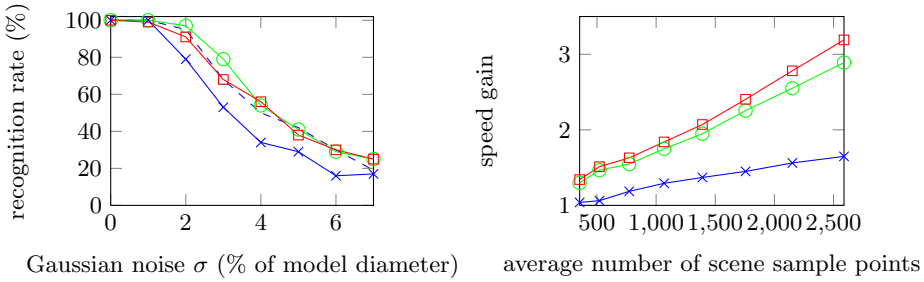
but also the recognition runtime. We considered three different pose thresholds (ϕ_{th} and t_{th}) to jointly represent features considered redundant. Fig. 7 shows recognition performance results and speed gains for all the considered models and thresholds. When t_{th} and ϕ_{th} are both set to 0 (blue $\times$ markers), no features are jointly represented. Therefore the computational savings are only due to collapsing of pose hypotheses around the axis of rotational symmetry, during the pose clustering step. As we increase the pose thresholds t_{th} and ϕ_{th}, we are able to jointly represent more features and hence have computational savings not only on the clustering but also on the matching step. For the tests with the cup model and pose thresholds set to $t_{\text{th}} = 0.025$ and $\phi_{\text{th}} = 6\,^\circ$ (red $\square$ markers), we were able to discard 93.17% surflet pairs during the creation of the model description, and reduce the number of computations during pose recognition. As shown in Fig. 7, the recognition rate drops slightly 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 300 times faster than [1]. However, the number of jointly represented surflet pairs depends heavily on the object geometric configuration. For objects whose shape has a smaller radius relative to the axis of symmetry, and also lower surflet density on the surface, less performance gains can be achieved. For the tests comprising the champagne glass model we were only able to discard 55.33% surflet pairs (with $t_{\text{th}} = 0.025$ and $\phi_{\text{th}} = 6\,^\circ$) during the creation of the model description, and achieve no more than 3.5 times speed improvements during recognition relatively to [1]. Fig. 6(b) shows qualitative results of our method with real data, in a cluttered scenario. Overall, we were able to obtain major improvements on recognition speed without significant cost on recognition performance.

References

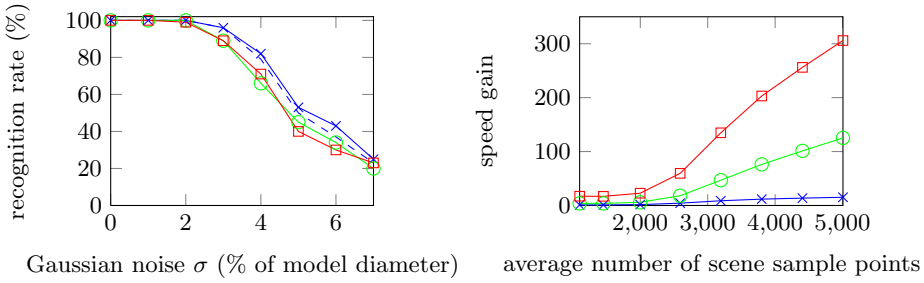
1. B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," *IEEE Transactions on Computer Vision and Pattern Recognition (CVPR)*, pp. 998 – 1005, June 2010.
2. D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.
3. 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*, pp. 433–449, 1999.
4. A. S. Mian, M. Bennamoun, and R. Owens, "Three-dimensional model-based object recognition and segmentation in cluttered scenes," *IEEE Transactions on Pattern Anal. Mach. Intell*, vol. 28, pp. 1584–1601, 2006.
5. E. Wahl, U. Hillenbrand, and G. Hirzinger, "Surflet-pair-relation histograms: A statistical 3D-shape representation for rapid classification," *3D Digital Imaging and Modeling, International Conference on*, p. 474, 2003.
6. J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., 1989.
7. M. Ciocarlie, "Household objects database," accessed 19-July-2012. [Online]. Available: http://www.ros.org/wiki/household_objects_database



(a) Coke can model.



(b) Champagne glass model.



(c) Cup model.

Fig. 7. Comparison results of our approach (continuous lines) against the original method of Drost *et al.* (dashed lines), with $\xi = 0.05$; Left: Recognition rate (%). Right: Time performance gain $\frac{\text{Drost } et\ al.\ \text{runtime}}{\text{Our runtime}}$.
Parameters: $t_{\text{th}} = 0, \phi_{\text{th}} = 0\,^\circ$ (blue ×), $t_{\text{th}} = 0.005, \phi_{\text{th}} = 1.2\,^\circ$ (green ○), $t_{\text{th}} = 0.025, \phi_{\text{th}} = 6\,^\circ$ (red □)