Multiple Object Detection for Pick-and-Place Applications

Paolo Piccinini*, Andrea Prati◇, Rita Cucchiara*
*D.I.I., ◇Di.S.M.I. - University of Modena and Reggio Emilia
*Via Vignolese, 905 - Modena, Italy – ◇Via Amendola, 2 - Reggio Emilia, Italy

Abstract

This paper presents a novel approach for detecting multiple instances of the same object for pick-and-place automation. The working conditions are very challenging, with complex objects, arranged at random in the scene, and heavily occluded. This approach exploits SIFT to obtain a set of correspondences between the object model and the current image. In order to segment the multiple instances of the object, the correspondences are clustered among the objects using a voting scheme which determines the best estimate of the object's center through mean shift. This procedure is compared in terms of accuracy with existing homography-based solutions which make use of RANSAC to eliminate outliers in the homography estimation.

1 Introduction

Computer vision and pattern recognition techniques have been widely used in the past for industrial applications and especially for robot vision. In many fields of industry, indeed, there is the need to automate the *pick-and-place* process of *picking* up objects, possibly performing some tasks, and then *placing* down them on a different location. Most of the pick-and-place systems are basically composed of robotic systems and sensors. These sensors are in charge of driving the robot arms to the right 3D location (and possibly orientation) of the next object to be picked up, according to the robot's degrees of freedom. The placing points are usually predetermined and sensors are rarely used to guide the place phase. Conversely, object picking can be very complicated if the scene is not well structured and constrained.

Most of the picking systems consider the case of well separated objects, well aligned on the belt and with a synchronized grasping of the objects. In this case, simple photocells can be used to trigger the picking phase. However, there are several applications in which this approach will be insufficient, since forcing the objects to stay well separated and aligned on the belt will waste space and time of the process. Moreover, there can be objects which needs to and are convenient to be kept in bins, for saving time and/or for hygienic reasons, as shown in Fig. 1. In this scenario, (high resolution) cameras should be used, together with specific machine vision algorithms.

In the case the objects are positioned at random inside a bin, a container or even at random on a belt/shelf, this problem is called *bin picking* [6]. A vision-based bin-picking system presents several challenges: (1) it should be capable to work with every type of object of different dimension;

(a) (b)

Figure 1: Examples of complex situations for multiple object segmentation.

(2) objects could be very complex, with many faces, reflective surfaces or they could be packaged in transparent flowpacks; (3) these applications often require very high working frequency, reaching easily requirements of some hundreds of pieces per minute; (4) in bin picking applications, objects are very cluttered and the many (self-)occlusions make the object only partially visible (see Fig. 1). Sometimes matching between image and 3D CAD models is provided; this way is often not affordable since 3D models are not always available and their acquisition is expensive and time consuming; moreover, time constraints would make unsuitable complex fitting of 3D models, which also need to be invariant to projective 3D transformations.

The approach described in this paper is meant to tackle all these points by proposing a feature-based segmentation technique capable to segment multiple occluded objects. When objects are complex, reflective, low-textured and heavily occluded, very few distinctive feature points can be extracted from the image. Having few features to be matched with the original sample, segmentation of multiple instances of the object is not straightforward. Thus, this work proposes the use of a voting scheme to cluster the matched feature points among the different instances. Once clustered, these points vote for *principal points* of the object, i.e. points which characterize and delimit the object shape, such as the four corners in the case of a rectangular-like shape. These points can be used as both delimiters of the object for the segmentation and picking points, depending on the end effector of the robot.

2 Related Works

From the applications' point of view, the scientific literature in vision-based pick-and-place is very profuse, even though quite outdated. For instance, the pioneering work [7] exploits image processing techniques to determine the grasping points for picking up unknown everyday objects.

Basic techniques are exploited: thresholding is used to segment objects and moments are computed to determine the location of the center of gravity and the orientation of the main inertial axis. In another work, Rahardja and Kosaka [6] propose the use of stereo vision to perform bin picking of industrial complex objects. Simple visual features, such as region area, eccentricity, and gray scale mean value, are adopted for object recognition and pose estimation.

Unfortunately, these approaches adopt so simple image processing techniques that in complex scenes containing multiple objects, such as those reported in Fig. 1, cannot be applied. An alternative way instead of object segmentation is the detection-by-feature approach which searches for discriminative local features. This proposal is very old [4] and it is called *local-feature-focus*. This algorithm recognizes and locates partially visible 2D objects, by not performing the segmentation globally at pixel level, but on higher-level features, such as round holes and convex or concave 90° corners. The algorithm searches for a cluster of local features in a relative configuration which is characteristic of that specific object. One feature in the cluster is selected as the "focus" feature, i.e. the one with respect to which the other features are located. This approach accounts also for complex structure of features, by means of binary decision trees and feature indexed hypotheses. These methods exploited very specific features (such as round holes) that cannot be extended for whichever type of object.

Similarly, we do not exploit a global segmentation strategy. The basic idea is to find matches between the rough model of the object expressed in terms of local visual features (directly extracted by a sample image of it, and not by complicated synthetic models), and the current image. Differently to the above-mentioned papers, our approach uses very simple features, based on single-point SIFT [5] feature detector. The approach is attractive because of its generality and the proven robustness. In fact, the use of SIFT point-based features increases the possibility to find sufficient matches in order to have reliable segmentations. However, as demonstrated in [3], this approach is prone to errors in those applications where images may lack enough distinctive features. To overcome to this problem, Hess and Fren [3] propose to further improve the registration between two images by exploiting, together with the SIFT-based distrinctive features, a refined local set of features in which the SIFT matching criteria are applied on a region centred on the global keypoints.

In our approach, the spatial relationships among the matched SIFT features are then used to cluster features belonging to multiple instances of the object. The features are grouped with the mean shift clustering technique [2]. Other works follow a similar approach, such as [8], where a PCA-SIFT approach is used to identify multiple instances of the same object in real time and a voting scheme similar to ours is used, but the achieved clustering is used only for localization purposes and not for detecting overlapped objects.

3 Homography-based Segmentation

The final objective of the system is to segment as many objects as possible in cluttered scenes as those reported in Fig. 1. The 2D segmentation process, together with a laser-based method for estimating the distance from the camera, will provide coordinates of one or more grasping points for each object to be picked up.

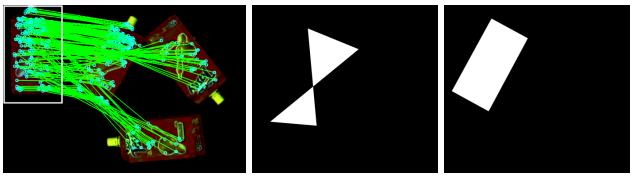
Our proposal goes through the following two phases:

1. *Feature extraction and matching*: significant features are extracted from both the object model and the current image; given a proper similarity measure, features are matched between the model and the current image, and the best correspondences are retained;

2. *Object segmentation*: given the set of correspondences, it is possible to compute a registration transform between the model and the segmented object in the current image.

The point 1 is achieved by using th SIFT feature detector and the 2NN (two nearest neighbors) heuristic proposed in [5]. The 2NN heuristic basically retains only the features F for which the ratio between the first nearest neighbor and the second nearest neighbor (wrt the Euclidean distance between the 128-long feature descriptors) is lower than a given threshold. The SIFT and 2NN heuristic have proven to be very robust in many contexts and invariant (at some extent) to rotations, scaling, noise and illumination changes.

Let us call $\mathcal{M} = \{m_1, \ldots, m_N\}$ the set of N matches found between the model M and the current image I, where each match m_i contains the (x, y) coordinates on the two reference systems: $m_i = \left\langle \left(x_i^M, y_i^M\right), \left(x_i^I, y_i^I\right) \right\rangle$. Given this set, the simplest approach for computing the registration transform between M and I (point 2 above) is to estimate the planar homography using a least squares approach. Alternatively, direct linear transform (DLT) or singular value decomposition (SVD) can be used to solve efficiently the system of linear equations obtained by the correspondences [3].



(a) Matched features (b) Homography without RANSAC (c) Homography with RANSAC

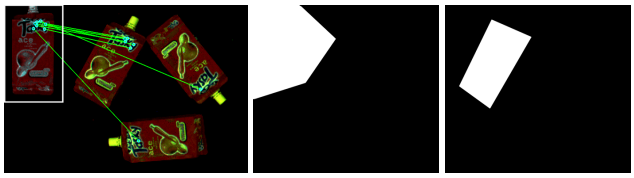
Figure 2: Wrong homography due to outliers.

Unfortunately, all these approaches are very sensitive to outliers in the set of correspondences. For instance, Fig. 2(b) shows an example of incorrect homography obtained by least square method using all the matches: it is evident that some of these matches are outliers in the estimation of the homography's parameters. A well known method to deal with the outliers is provided by RANSAC [1] which finds a set of inlier correspondences which can be used for computing the transform as described above.

The result of estimating the planar homography from the matches in Fig. 2(a) (the model is in the top-left corner, highlighted in white) with RANSAC and least square estimation is shown in Fig. 2(c). Although the result is appreciable, this approach still presents some drawbacks. The

first is that, in the case of multiple instances of the same object, the SIFT does not guarantee to find all the correspondences on the same instance. Even though RANSAC can, at a certain level, handle this situation by iteratively estimate the most consistent set of matches (as in the case of Fig. 2(c)), it is not able to cope with a large number of outliers due to the presence of multiple instances, as shown in the example of Fig. 3(b). A possible solution, described in the next section, is to cluster the matched features $\mathcal{M}$, and then perform RANSAC and least square for each cluster of features separately. In this manner, multiple homographies can be estimated, one for each detected instance of the object. Nevertheless, even though it leads to better results (as will be shown in Section 5), this approach splits the set $\mathcal{M}$ in several clusters, and this gives fewer points on which the homographies' parameters are estimated, resulting, typically, in less accurate results.

With these premises, the next section will propose an innovative solution for estimating the geometrical transform between the model and the object's instances.



(a) Matched features (b) Hom.-RANSAC all matches (c) Hom.-RANSAC clustered

Figure 3: Wrong homography due to multiple instances.

4 Segmentation of Multiple Instances

Instead of applying RANSAC on the entire set $\mathcal{M}$, we can first partition the set on S subsets $\mathcal{M}^i$ possibly containing only the features belonging to a single object/instance O^i. The clustering of features can be performed by considering, similarly to [8], the relative position of each feature with respect to the center of the object. In practice, the user defines the center of the object model. Then, given the set $\mathcal{M}$ of correspondences, the vector wrt the center in the model coordinate system is computed and stored for each match m_y. Exploiting the information about the main orientation of the feature provided by SIFT, the object's center position in the image coordinate system can be easily estimated by assuming a pure roto-translational transform (also known as *Euclidean transform*).

Given the approximation of the feature location, the noise and the simplifying assumption of Euclidean transform, the estimate of the object's center is not precise. To account for this, we cluster the center's estimates (obtained from every match $m_y \in \mathcal{M}$) by using the mean shift [2], which provides also a good estimate of the center's location. By running the RANSAC and least square on each subset $\mathcal{M}^i$, the resulting homographies are generally more accurate, as shown in Fig. 3(c), where only one of the homographies is drawn. This method will be hereinafter called *RANSAC clustered*.

However, this approach still presents the problem of being inaccurate if few matches per instance are available and/or if these matches are fairly collinear (see Fig. 4(b)), where the few matches on the middle object are collinear and the resulting homography is imprecise. Moreover, RANSAC's result is unpredictable due to its random sampling procedure, which might be a problem in industrial applications. Our proposal is to further relax the problem's conditions by assuming a complete Euclidean transform for all the pixels of the model (not only the center). This means that we only consider in-the-plane rotations and translations, not permitting the model to scale (reasonable condition if we assume that the objects are more or less at the same distance from the camera) or to rotate (much) out of the image plane. It is also worth noting that a precise segmentation is not required since our goals are to find the grasping points and to evaluate occlusions (in order to avoid the picking of covered objects).

(a) Matched features (b) Homography-RANSAC clustered (c) Our approach

Figure 4: Wrong homography due to few collinear matches.

The complete procedure can be summarized as follows:

1. during the definition of the object's model, the user can select L *principal points* $\mathcal{P} = \{P_0, \ldots, P_L\}$, where P_0 is the center of the object and the other points represent both other grasping points and points delimiting the objects, such as extrema points of the oriented bounding box;

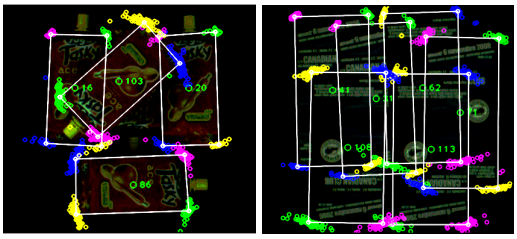
2. with the mean shift clustering, the set $\mathcal{M}$ of correspondences are partitioned in subsets $\mathcal{M}^i$ for each of the instances found in the current image;

3. for each $m_y \in \mathcal{M}^i$, the estimate for each of the L principal points is computed; let us define as $P^i_{j,y}$ the estimate obtained from match m_y of instance O^i for the principal point P_j, with $j = 1, \ldots, L$;

4. $L - 1$ mean shift algorithms are issued to find the best estimate $P^i_{j,*}$ for P_j in O^i; in this case, the mean shift is not employed for clustering, so a simpler technique (e.g., to compute the average location) should be enough; however, computational complexity of mean shift with some tens of points, as in most of our cases, is negligible;

5. the $L-1$ estimates $P^i_{j,*}$ are used to obtain the segmentation of O^i.

Fig. 4(c) shows how we can solve with our approach the problem inherent to homography. Additionally, Fig. 5 shows two examples of the result achieved with this procedure. The large green circles represent the estimates $P^i_{0,*}$ of the object's center and the values close to them are the



(a) (b)

Figure 5: Examples of the segmentation results.

number of matches assigned to that object through the clustering. The small circles in blue, magenta, yellow and green represent the estimates $P^i_{j,y}$ of the other four principal points (the extrema of the bounding box, in these examples). Note that the estimates are mixed up and fairly distributed, but the mean shift is nonetheless capable to act correctly. The white lines connect the estimates $P^i_{j,*}$ of the principal points and are the boundaries of the final segmentation. It is evident that this approach is able to segment also very occluded objects, as shown in Fig. 5(b).

5 Experimental Results

In order to evaluate our approach in challenging situations we perform extensive experimentation with three types of very diverse objects: boxes of fruit juices (see Fig. 1(a)), packs of Nutella (Fig. 1(b)) and paper flyers (Fig. 5(b)). These objects have different characteristics, such as the reflectiveness of fruit juices, the non-regular shapes of Nutella's packs or the very dark appearance of the flyers.

The accuracy of our approach can be measured by means of three different metrics: the precision/recall at *object-level*, the precision/recall at *pixel-level* and the accuracy of the *center location*. The first metrics accounts for how many correct objects are segmented (where correct segmentations are evaluated by the operator), while the second considers the pixel-by-pixel segmentation. In this case, the precision and the recall are computed with reference to all the objects to be segmented, and thus the recall tends to be very low if some objects are missed. Finally, the last metrics is more application-oriented, thinking to a pick-and-place application where the accuracy in determining the grasping point (e.g., the center) is crucial. All these measures are computed with respect to a manually-determined ground truth.

Results are summarized in Table 1. We compared our proposal with the use of homography-based segmentation by either RANSAC on all the matches (*all RS* in Table 1) or RANSAC on clustered matches (*clus RS*), as described in the previous Section. Since the RANSAC applied on all the matches finds a single instance of the object, the precision at object-level is close to 100%, but the recall at object-level is very low. Instead, the RANSAC applied to clustered matches shows a poor accuracy in identifying the center. This is due to the fact that this algorithm finds more objects than the version on all the matches (precision/recall both at object- and pixel-level are higher), but the resulting homographies are less accurate since they are estimated by fewer matches.

Fruit Juices					
	Object-level		Pixel-level		Center
	Precision	Recall	Precision	Recall	Mean
all RS	100.00%	25.00%	22.95%	23.66%	5.41 px
clus RS	91.67%	82.50%	77.43%	79.93%	18.97 px
Ours	**97.37%**	**92.50%**	**88.55%**	**87.64%**	**5,76 px**
Nutella Packs					
	Object-level		Pixel-level		Center
	Precision	Recall	Precision	Recall	Mean
all RS	100.00%	15.38%	13.23%	14.46%	6.98 px
clus RS	66.67%	33.85%	38.96%	35.82%	17.24 px
Ours	**97.84%**	**86.78%**	**82.87%**	**83.13%**	**3.86 px**
Paper Flyers					
	Object-level		Pixel-level		Center
	Precision	Recall	Precision	Recall	Mean
all RS	90.00%	16.36%	17.46%	15.72%	14.68 px
clus RS	74.00%	64.91%	71.31%	68.46%	22.27 px
Ours	**96.15%**	**90.91%**	**86.35%**	**89.39%**	**2.66 px**

Table 1: Experimental results.

Our approach outperforms the other two in all the cases and for every metrics, with average precision and recall at object-level of 97.84% and 86.78%, at pixel-level of 82.87% and 83.13%, and an average center's distance of 3.86 pixels. Additionally, our approach is much faster than the others since it avoids both the iterative procedure of RANSAC and the least square estimation. On average, our system takes about 1.2 seconds for each image to segment a number of objects between 3 and 10, while the RANSAC-based approaches take 4.41 sec. and 4.27 sec., in the case of clustered and non-clustered matches, respectively. These time performances have been obtained on a standard Windows XP PC with Core Duo at 2.4 Ghz processor and 2 GB of memory.

References

[1] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[2] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. on Information Theory*, 21(1):32–40, 1975.

[3] R. Hess and A. Fern. Improved video registration using non-distinctive local image features. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition, CVPR '07*, pages 1–8, June 2007.

[4] T. Knoll and R. Jain. Recognizing partially visible objects using feature indexed hypotheses. *IEEE Journal of Robotics and Automation*, 2(1):3–13, Mar 1986.

[5] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004.

[6] K. Rahardja and A. Kosaka. Vision-based binpicking: recognition and localization of multiple complex objects using simple visual cues. In *in 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1448–57, 1996.

[7] P. Sanz, A. Del Pobil, J. Inesta, and G. Recatala. Vision-guided grasping of unknown objects for service robots. In *Proc. of IEEE Intl Conf. on Robotics and Automation, 1998*, volume 4, pages 3018–3025, 1998.

[8] S. Zickler and M. Veloso. Detection and localization of multiple objects. In *Proc. of 6th IEEE-RAS International Conference on Humanoid Robots*, pages 20–25, Dec. 2006.