This paper appears in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 1996

Vision-Based Bin-Picking: Recognition and Localization of Multiple Complex Objects Using Simple Visual Cues

Krisnawan Rahardja and Akio Kosaka

Robot Vision Laboratory
1285 EE Building, Purdue University
West Lafayette, IN 47907-1285, U.S.A.
e-mails: rahardja@ecn.purdue.edu, kosaka@ecn.purdue.edu

Abstract

Vision-based bin-picking is increasingly more difficult as the complexity of target objects increases. We propose an efficient solution where complex objects are sufficiently represented by simple features/cues; thus invariance to object complexity is established. The region extraction algorithm utilized in our approach is capable of providing the focus of attention to the simple cues as a trigger toward recognition and pose estimation. Successful bin-picking experiments of industrial objects using stereo vision tools are presented.

1 Introduction

The objective of this research is to develop a vision algorithm that provides sufficient information for a bin-picking robot to manipulate complex industrial objects. Typical stereo views of a workspace are shown in Fig. 1, where alternator covers are the target objects to be manipulated. In this paper, we present an algorithm that is capable of identifying such complex objects as well as estimating their 3-D pose by stereo vision technique. A sample result of the algorithm is shown in Fig. 2, where the pose-estimated manipulation landmarks (i.e. the large bearing holes and their adjacent four screw holes) are reprojected onto the original images.

For many years now, researchers have proposed various techniques for vision-based bin-picking. While there are many possible rationales to explain this circumstance, particularly in industrial applications, the facts attributed to the difficulty of implementing such systems are: a) the complexity of industrial objects; b) the lighting reflections generated by the common metallic colors on industrial objects; and c) the cluttered nature of object placements that generate mutual occlusions. At the simplest form, recognition and pose estimation of objects that are fully visible (without occlusion) can be done by some analysis on a proper binary-thresholded image [6, 14]. However, the same method will not work for randomly cluttered/stacked objects, as the setting shown in Fig. 1. An immense increase in complexity is added to the problem due to the distortion of object appearance by occlusions and exposed background features. Moreover, binary thresholding is very sensitive to lighting conditions especially on metallic colored objects. Certainly there are successful systems with more complex approaches in both 3-D and 2-D vision as the examples discussed in the following paragraphs.

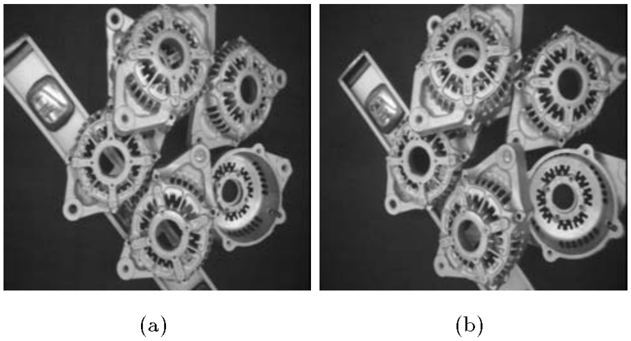
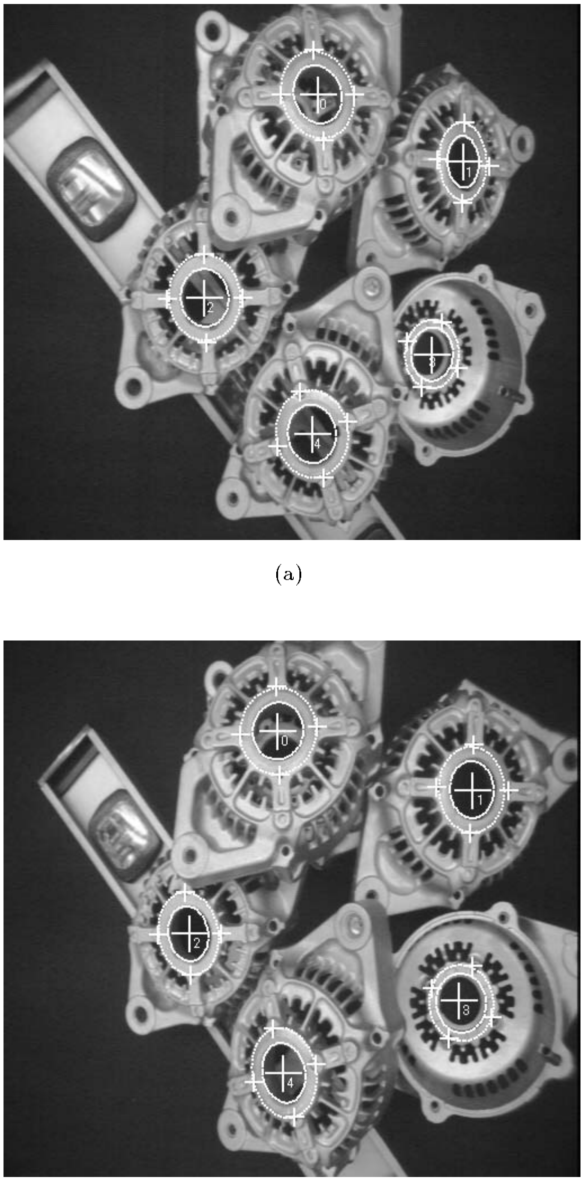


Figure 1: *Stereo views of the workspace.*

Bolles and Horaud [3] developed a recognition and pose-estimation system (3DPO) for industrial parts with *smooth/simple surfaces* using the 3-D range finder. In their work, the surface feature grouping technique was employed to generate and verify the hypotheses of object location. Wang, Kak, et al. [15] utilized a 3-D range finder to directly compute the depth and shape of articulated objects, and demonstrated



(a)

(b)

Figure 2: *The result of recognition and pose-estimation on both left (a) and right (b) images. The estimated 3-D pose of the manipulation landmarks are reprojected as the dotted ellipses to verify the accuracy of pose-estimation.*

actual bin-picking manipulations with such objects. We understand, however, that it is not easy to reconstruct the shape and surface model of complex shaped objects such as the alternator covers due to low resolution of the 3-D range finder, as well as discontinuities caused by occluded regions in the depth map.

Grimson [8] and Bolles and Cain [2] employed 2-D local features to recognize and localize cluttered objects in the scene; however their methods are limited to the case where the objects are located on a flat worktable and therefore the appearance distortion from 3-D rotations is not admissible.

Onda, et al. [11] employed stereo vision and edge feature matching technique to recognize and pose-estimate complex industrial objects. Unfortunately, due to the high computational complexity of edge-based feature interpretation and establishment of correct correspondences between stereo images, the system is constrained to smooth/linear curved objects and to limited number of objects. Obviously it is advisable for edge-based techniques not to be employed for processing complex scenes such as in Fig. 1.

In this paper, we propose a robust and efficient architecture that utilizes simple visual cues as triggers toward recognition and pose-estimation of complex objects. In the following sections, the introduction to our strategy is first presented. Next, we briefly discuss the region extraction algorithm we use, and describe how it is utilized for fast extraction of simple visual cues. We then present the recognition and pose-estimation processes of complex objects, and finally show our experimental results with real industrial objects.

2 Overall Strategy

From inspection of the alternator covers shown in Fig. 1, it is clear that even objects with such high shape complexity possess simple entities such as circular and polygonal parts. Among all the information in the image about the objects, those simple features are the most appropriate for geometric reasoning and potentially useful as *landmark* features. Furthermore, invariance to the shape complexity of the objects is established, once target objects are represented by their simplified version. We therefore define the recognition and pose-estimation of target objects as those of their simplistic landmark features – landmark features are powerful triggers for recognition as well as pose-estimation of the targets.

To carry out this concept, we require a process that segments the desired landmark features from the remaining irrelevant data from the image. For this purpose, we have derived a region extraction algorithm [13] that is based on the Split and Merge [9] framework; this is discussed in a more detailed manner in Section 4. In a nutshell, this algorithm prevents us from having to use a conventional computer vision approach (i.e. edge interpretation). Nevertheless, landmark extraction is not straightforward, since there is a possibility that landmark appearance gets deformed/distorted from the following effects: 1) occlusion from other targets; 2) rotations involving more than just the 2-D rotations on the worktable; 3) exposed background features when the target object bears openings/holes on its surface as the alternator covers shown in Fig. 1. But since the task at hand is to develop bin-picking systems, we are only concerned about the two latter sources of appearance distortion, i.e. only targets at the upper portions of the jumble can be picked up by the robot. Thus the landmark extraction process must accommodate this anticipated appearance distortions to extract landmarks of interest in spite of the occurrence of rotations, and to provide focus of attention to image areas where landmark existence is highly probable. The latter design constraint is possible because in each area where landmark existence is highly probable, there exists a *region of interest* that possesses a similar shape as the landmark of interest. To verify the hypothesis of landmark existence, popular methods of geometric reasoning can then be performed within some restricted breadth in the vicinity of the extracted regions of interest.

Now we introduce the concept of seed/supporting features as shown in the past literature by Besl [1] and Chen and Mulgaonkar [5] in 3-D and 2-D vision, respectively. In this paper, we define landmark features that are unique and easily identifiable as *seed features*; and define some other features which would be useful for both identification and pose-estimation as *supporting features*. Note that a group of multiple landmark features may constitute as a seed/supporting feature. Supporting features are needed since seed features by themselves may not provide accurate pose-estimation of the objects. This is due to the limitations of optical devices, photoelectronic/thermal noises, and digitization error that generate noisy image formation. In our experiment, for every alternator cover, the large bearing hole is defined as the seed feature and the set of four small screw holes at the perimeter of the bearing hole is defined as the supporting features as presented in the next section.

Fig. 3 shows the architecture of our solution, and the overall algorithm flow can be summarized as the following:

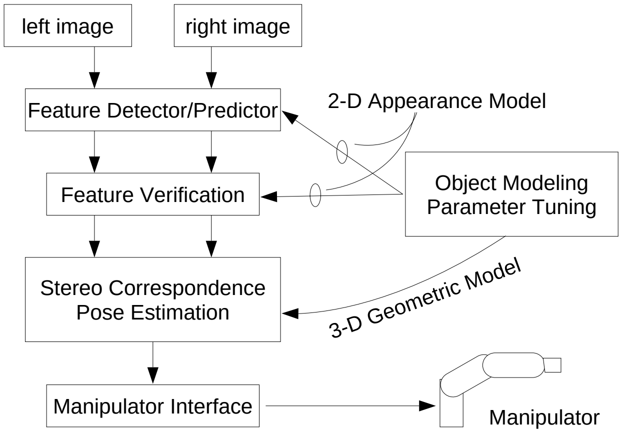


Figure 3: *System Architecture.*

1. Left and right stereo views are supplied to the system.
2. *Feature Detector/Predictor* (FDP) module gives initial estimate of visual cues by extracting regions in the images that are highly probable to be the seed and supporting features as prescribed by the *Object Modeling and Parameter Tuning* (OMPT) module.
3. In conjunction with OMPT module, *Feature Verification* (FV) module tests and decides to accept or reject each given estimate by evaluating its 2-D appearance. If the 2-D appearance of an estimate is distorted (possibly due to exposed background features, e.g. the level apparatus shown in Fig. 1), this module will then make an attempt to compensate the distortion and re-evaluate the result.
4. The *Stereo Correspondence and Pose Estimation* (SCPE) module matches the estimates from both views and determines their pose.
5. For each identification and pose estimation pair result, the *Manipulator Interface* (MI) module generates the appropriate motion commands to be supplied to the manipulator.

The following sections present more detailed descriptions of each module.

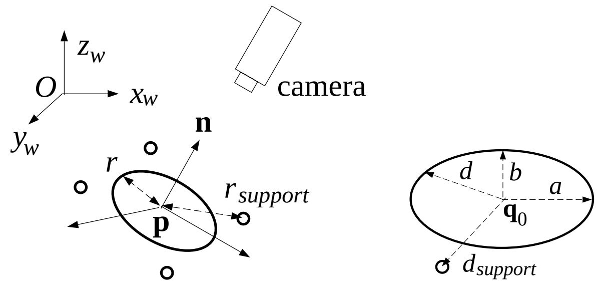
3 Object Modeling and Parameter Tuning (OMPT) Module

As presented briefly at the end of Section 2, this module assists FDP and FV modules in extracting seed and supporting features. Clearly a proper training is required to enable this module to perform the desired task. The following two subsections describe how this can be done.

3.1 Object Modeling Submodule

Assisted by human operator, the Object Modeling Submodule generates the model of target objects from a set of seed and supporting features in terms of 3-D geometry. It also analyzes the 2-D region appearance characteristics extracted by the *Feature Detector/Predictor* module and selects regions that are potential to be the prescribed seed/supporting features. Analysis is based on 2-D shape complexity and some other parametric feature descriptions (e.g. mean/standard deviation gray-scale value, etc.) as far as the task require.

In our experiment, each alternator cover is sufficiently modeled/represented by the five circular regions as shown in Fig. 4(a). The large circular bearing hole at the center is defined as the seed feature and the remaining four small screw holes on the perimeter of the bearing hole are defined as the supporting features. The relative location of a screw hole to the bearing hole and the relative position of a screw hole to another screw hole are defined in 3-D geometry. We define the center of the seed feature $\mathbf{p}$ as the object origin and the normal vector $\mathbf{n}$ perpendicular to the plane of the seed feature (or the plane created by the supporting features) as the object orientation. The system estimates the object pose as the position of the center of the seed feature $\mathbf{p} = (p_x, p_y, p_z)$ and the normal vector $\mathbf{n} = (n_x, n_y, n_z)$ with respect to the world coordinate frame $[X_w, Y_w, Z_w]$.



(a) (b)

Figure 4: *(a) The sufficient 3-D model/representation of the alternator cover. (b) The 2-D appearance analysis of a seed feature and its relative position to a supporting feature.*

In a bin-picking scenario, a manipulator can only grasp objects with stable pose; thus the system must be able to identify objects with such states. For alternator covers, the range of stable poses is parameterized by the angle θ between $\mathbf{n}$ and Z_w, and was empirically determined as $\theta \in [0^o, 45^o]$. The OMPT module utilizes this assumption to analyze the 2-D appearance of seed features.

3.2 Parameter Tuning Submodule

During training, Parameter Tuning Submodule consults a human operator in adjusting parametric descriptions necessary for the FDP module in extracting regions. This interaction is carried out by having the human operator select and eliminate candidate regions from the initial results of the FDP module in conjunction with the supplied information from the Object Modeling submodule.

At this stage, the human operator must also set an appropriate tolerance to accommodate the distortion of landmark appearance. Note that on our target objects, distorted 2-D appearance of the seed feature due to rotations can be approximated by an ellipse. Taking into account of this approximation and the tolerance to accommodate the appearance distortion from exposed background features, this module adjusts the required parameters used by the FDP module. Training stage for the system is completed when the system has obtained an approximation of optimal set of parameters for the FDP module. Sample results of this parameter adjustments can be seen in Figs. 5(e) and (f) where the FDP module successfully extracted several seed feature candidates and also several regions from the distortion tolerance.

For the type of seed features we used, several necessary parameters about ellipses were also obtained, and they are based on the facts about ellipses. Let $\mathbf{p}$ be the center of a seed feature C having the normal vector $\mathbf{n}$ with radius r, D be the projection of seed feature C onto the camera image plane, $\mathbf{q}_0$ be the centroid of region D, and $\mathbf{q}$ be all the points on the boundary of region D. Let the moment of inertia M of the region D be computed with respect to the centroid of D [14]. The distance of $\mathbf{q}_0$ to each point $\mathbf{q}$ normalized by its moment of inertia is computed as

$$d = \sqrt{(\mathbf{q} - \mathbf{q}_0)^T M^{-1} (\mathbf{q} - \mathbf{q}_0)}. \qquad (1)$$

As shown in Fig. 4(b), if the region D is an ellipse, then d should be constant over all perimeter points $\mathbf{q}$. In our experiment, to accept a region as an ellipse, the standard deviation σ_d of the distances from $\mathbf{q}_0$ to all $\mathbf{q}$ is computed, and then tested as in Eq. (2). $\sigma_{threshold}$ is obtained through simulations and trials of various centroid locations p and normal vector n angle rotations in the workspace.

$$\sigma_d \leq \sigma_{threshold} \qquad (2)$$

Empirically, it is found that the test $\sigma_{threshold} \leq 0.1$ unit performs satisfactorily. Furthermore, we also obtained the analysis of the locations of supporting features relative to the centroid of their associated seed feature. Eq. (1) is also used to compute the normalized distance $d_{support}$ of the centroid of supporting features from the centroid of their corresponding seed feature. An acceptable supporting feature must satisfy the following inequalities:

$$d_{min} \leq d_{support} \leq d_{max}, \tag{3}$$

where the constant bounds d_{min} and d_{max} were also obtained through the simulations and trials as done for the determination of $\sigma_{threshold}$.

4 Feature Detector/Predictor (FDP) Module

This module extracts image regions that are potential to be the seed and supporting features from the remaining irrelevant background. These extracted features must satisfy the parametric descriptions generated by the OMPT module. This task is a natural problem to be solved by an image segmentation algorithm. From the available algorithms, numerous appealing solutions were given as our choices. As a summary, approaches to image segmentation that are illuminating and have inspired numerous research work are Thresholding [12], Split and Merge [9], Snakes [10], and Markov Random Fields (MRF's) [7]. Our system requires

- good localization and shape preservation of seed features, since they affect the accuracy of pose-estimation;
- fast processing time, to make this system a feasible solution in industrial settings.

In addition, the system only deals with images that are taken from a controlled environment; in other words, images do not get much degraded. None of those segmentation algorithm match exactly to our design constraints; instead, our choice of segmentation method is a region extraction algorithm that greatly utilizes the convenience of edge detection.

It is known that an edge/boundary detection partitions an image into regions of smooth intensity surfaces. While in general segmentation algorithm do not give such solutions (e.g. texture segmentation), segmentation by edge-detection is a sufficient approximation to produce the required task of segmenting seed and supporting feature candidates. A clear advantage of this approach is that the shape of regions are well-preserved, especially on the surface boundaries; a useful characteristic for shape recognition.

We developed a region extraction algorithm [13] that is based on the Split and Merge [9] algorithm. The outline of this algorithm can be described as follows:

1. Canny edge detector [4] that has been proven to be superb in preserving localization of edges is applied to segment the image.
2. To close all edge contours, an edge linking routine is applied to extend dangling edge contours (especially near T-junctions) that are the effect of the *uniqueness of response criteria* of Canny operator.
3. To extract regions with closed edge-contours, perform splitting and merging based on the predicate that a region is homogeneous if edge pixel do not exist within its interior.

The product of this algorithm are regions with their complete description given by the convenient quadtree structure and the required parametric measures. To extract regions of interest, a simple comparison test between region specifications as prescribed by the OMPT module and the parametric description structure of each region is performed. Currently, we employ three parametric measures of region characteristics that are

- *shape complexity* $= \frac{perimeter^2}{area}$
- *region area*
- *gray scale mean value.*

A human operator interacts with the OMPT module to provide upper and lower bounds of each of the characteristic measures. Figs. 5 show a sample result of the region extraction algorithm applied to stereo images shown in Figs. 1. Intermediate results are shown in Figs. 5(a) and 5(b) as the result of Canny edge detection; Figs. 5(c) and 5(d) as the segmentation results; and Figs. 5(d) and 5(e) as the result of the extraction of seed feature candidates. Note that in Figs. 5 (c), (d), (e), and (f) different regions are colored differently to enhance the neighboring regions. It can be seen that the effect of the tolerance to accommodate appearance distortions as the number of extracted candidates are more than the true desired seed features.

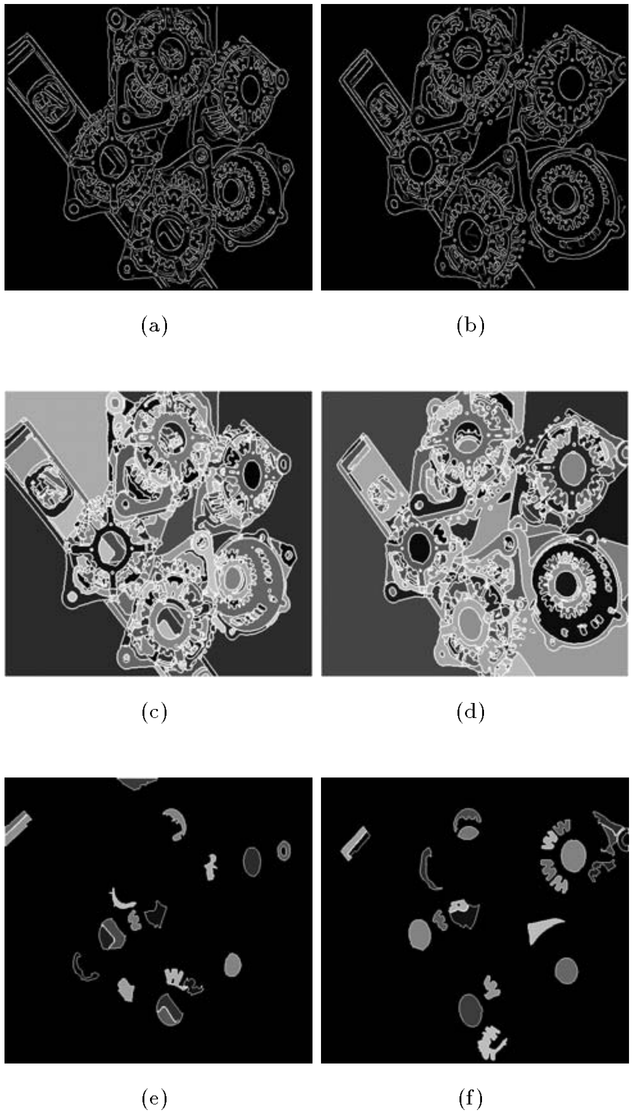


Figure 5: *Result of FDP module. (a), (b) Results of Canny edge detection applied to the stereo pair images shown in Fig. 1. (c), (d) Results of region extraction, by taking into account extracted edges shown in (a) and (b). (e), (f) Seed feature candidates are extracted from the segmented regions shown in (c) and (d), by utilizing the parametric constraints of the simplified object model.*

5 Feature Verification (FV) Module

After seed and supporting feature candidates are extracted by FDP module, FV module is responsible for accepting/rejecting those candidates as FDP module generally produce more candidates than the true desired features. This module verify the 2-D appearance of the candidates with the help of OMPT and FDP modules.

A candidate region is verified as a 2-D projected seed feature with possible occurrence of 3-D rotations. In our experiment, where seed features are circular regions, we verify candidates by the following Ellipse Verification Test.

Ellipse Verification Test (EVT)

1. From the moment of inertia computation, the lengths of the principal axes of a candidate regions should be within some specific range.
2. The standard deviation σ_d in Eq. (2) of the distances from the centroid $\mathbf{q}_0$ of a seed feature candidate to its boundary points must be less than some threshold $\sigma_{threshold}$, as shown in Fig. 6(a).

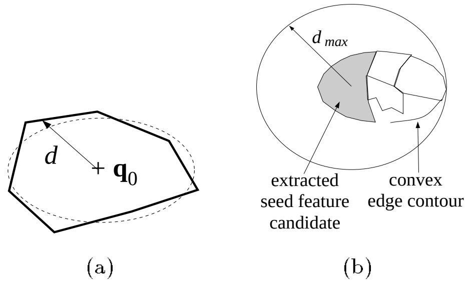


Figure 6: *(a) Feature verification by EVT; (b) Region growing assisted by the edge contour that surrounds the seed feature candidate.*

A seed feature candidate is accepted if it satisfies EVT.

Otherwise, with the help of OMPT module, FV module examines the proximity of the candidate and looks for possible region patches that might transform the appearance of the candidate to be closer to a 2-D projection of a seed feature when they are fused. Then by the abovementioned verification test, the transformed candidate is either accepted or rejected. With the alternator covers, due to the nature of holes/openings on each object, a seed feature candidate might not satisfy EVT caused by exposed background regions that may correspond to some sectional parts of other objects. The process to possibly transform a seed feature candidate is a region growing process that is described as the following steps and Fig. 6(b).

1. Given a seed feature candidate, the reduced search space for region patches is defined as a circular area with radius d_{max} (3) from the centroid of the candidate.
2. Within the search space, find a convex enclosure of edge-pixel traces that are connected to the seed feature candidate, to grow/transform it into an approximate ellipse. Note that this edge-pixel traces can be connected as a closed contour, or broken into several contours when edge-detection failed.
3. Accept/reject the transformed candidate by EVT.

This process is iterated for all seed feature candidates.

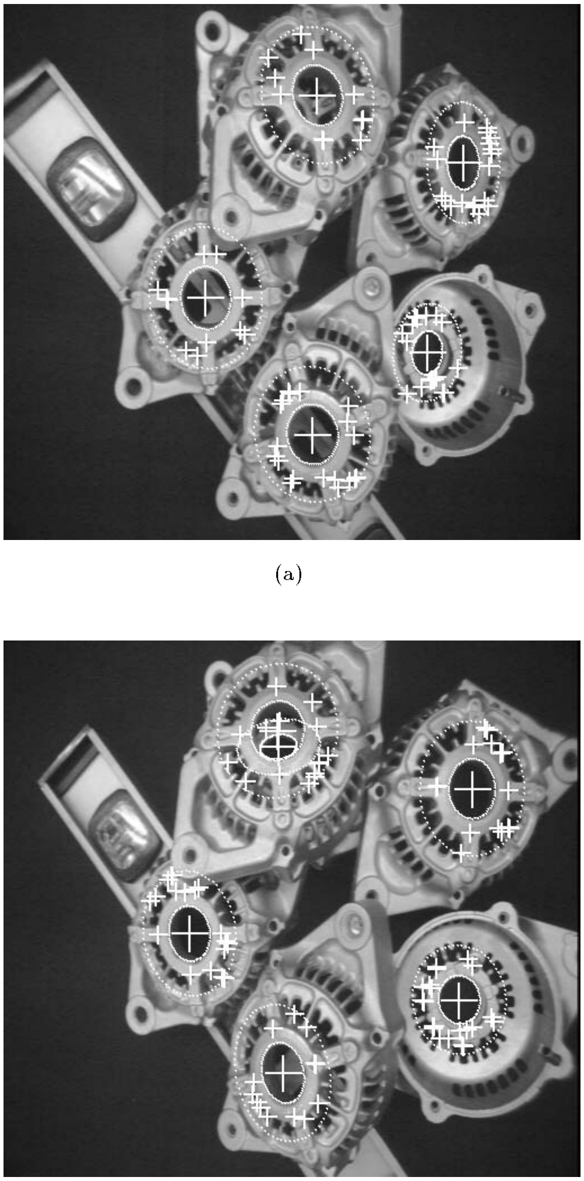
Potential *supporting features* are also selected for each accepted seed feature candidate. Selection is done by examining the proximity of the seed feature where supporting features should be located. The determination of this examination area is also precompiled by the simulation analysis done by OMPT module as described in subsection 3.2. The result of feature verification on the images of Fig. 1 is shown in Fig. 7, where dotted ellipses are the accepted seed features (both extracted by FDP and transformed candidates). Supporting feature candidates that are in the valid vicinity of each seed feature are also highlighted with *cross*es.

6 Stereo Correspondence Search and Pose Estimation (SCPE) Module

The 3-D pose of a given object can be estimated from its 3-D model. However the estimation is not reliable when the workspace is relatively far from the camera such as the setting shown in Fig. 1. To generate better estimates of the 3-D pose of the objects, we incorporated a stereo vision tool. This SCPE module is responsible for generating pose-estimates of all recognizable targets from stereo correspondences of seed and possibly supporting features.

This SCPE module first looks for correct correspondences between the available seed features from left and right images using the epipolar and geometric/locational constraints. That is for each pair of left and right seed features, say $\mathbf{s}^{left}$ and $\mathbf{s}^{right}$, the epipolar line constraint and the geometric/locational constraint are examined as follows:

1. Examine the epipolar line constraint for the each centroid of $\mathbf{s}^{left}$ and $\mathbf{s}^{right}$. If the constraint is



(a)

(b)

Figure 7: *The result of FV module for the left viewpoint (a) and right viewpoint (b). Dotted ellipses are the accepted seed features. Potential supporting features for each associated seed feature candidates are highlighted with the crosses.*

consistent, go to step 2. If not, start with a new pair.

2. Compute the 3-D position of the centroid $\mathbf{p}$ from $\mathbf{s}^{left}$ and $\mathbf{s}^{right}$.
3. If the 3-D location of $\mathbf{p}$ is within the workspace, accept this pair of seed features as a correct correspondence.

Then for each accepted pair of seed features, SCPE module needs to find the associated supporting feature.

As shown in Fig. 7, each chosen seed feature has a large number of supporting feature candidates; thus establishing one-to-one correspondence through epipolar and geometric constraint is difficult. Rather than using the same method as the seed features, the correspondence of supporting features is found by optimizing an objective function that is a measure of best-fit based on the 3-D geometry of the model. The definition of the objective function is as follows:
For each set A of four pairs of supporting features, say,

$$\begin{aligned} A &= \{(\mathbf{q_1}^{left}, \mathbf{q_1}^{right}), (\mathbf{q_2}^{left}, \mathbf{q_2}^{right}), \\ &\quad (\mathbf{q_3}^{left}, \mathbf{q_3}^{right}), (\mathbf{q_4}^{left}, \mathbf{q_4}^{right})\} \end{aligned}$$

compute the 3-D positions of the centroids of these supporting features. Let $\mathbf{p}_i$ be the 3-D positions of the centroid of the supporting feature estimated from $\mathbf{q}_i{}^{left}$ and $\mathbf{q}_i{}^{right}$. The objective function to be minimized for a set A is

$$F_A = \sum_{i,j=1,2,3,4} \Big(||\mathbf{p_i} - \mathbf{p_j}|| \ - \ C_{ij} \Big)^2 \qquad (4)$$

where C_{ij} is the 3-D distance between the centroids of two supporting features i and j which is obtained from the 3-D model of the object (e.g. in Fig. 4(a), C_{ij} is either $\sqrt{2} r_{support}$ or $2r_{support}$), and F_A is bounded by some value F_{max}. This bound is derived empirically and is highly specific to the physical system. To find an optimal correspondence, the algorithm performs an exhaustive search over all sets of possible correspondences.

Finally, after the optimal correspondence is established, the algorithm estimates the pose of the object with respect to seed features and supporting features. If the supporting features are optimized with the objective function case of $F_A < F_{max}$ over both image frames, then the 3-D position of the supporting features are estimated. The 3-D object center $\mathbf{p}$ and the normal vector $\mathbf{n}$ with respect to the world frame are also estimated. If no supported features are matched (i.e. $F_A > F_{max}$), substitutes for supporting features are generated by appropriately choosing four evenly-spaced points along the seed feature's ellipse boundary. This substitute is then used to compute the pose of the object (i.e. vectors $\mathbf{p}$ and $\mathbf{n}$). The result of this module is shown in Fig. 2, where the accuracy of the pose estimation is shown by reprojecting the seed and supporting features onto the original images.

The result of the 3-D pose estimation is then passed to the Manipulator Interface Module (*MI*) as in Fig. 3 where motion-path-plans are generated for grasping.

7 Experimental Results

In our experiments, several types of alternator covers that share the common model shown in Fig. 4 were used as target objects. The objects were randomly cluttered, with possibility of being upside down. The radius of the bearing hole is $15.0mm$ and the distance between the origin of the bearing hole to each neighboring screw hole is $22.5mm$ (i.e. r and $r_{support}$ as in Fig. 4(a)).

As a testbed we used a gripper-mounted camera on a PUMA 700 manipulator. Two gripper/camera positions were chosen as the left and right viewpoints with a distance of $243.5mm$ and vergence angle of 20^o. The approximate distance from the camera locations to the objects is $350.0mm$.

For each trial, stereo-pair images of 512×480 pixels are digitized, and processed individually up to the SCPE module as shown in Fig. 3. On our SUN Sparc 1000 server, the average processing time (i.e. until completion of pose-estimations) is approximately 1.5 minutes. Of course we realized that a lot of improvements still can be done.

To verify the accuracy of pose-estimation, instead of actually grasping the recognized objects, we drive the gripper-mounted camera along the estimated normal vector $\mathbf{n}$ on all recognized targets. Figs. 8 and 9 show the results of the pose-estimation where three images were taken from the approaching camera at estimated distances of (a) $300.0mm$, (b) $150.0mm$ and (c) $100.0mm$ away from the center of the bearing hole $\mathbf{p}$. In these figures, the white cross hair indicates the estimated location of the image where the object center $\mathbf{p}$ should be. The white dotted circles depict the estimated perimeter where the screw holes should be. The discrepancy in these figures, shows the error of the pose estimation. Note that the images are normalized by correcting the aspect ratio so that white dots form regular circles. These sets of images depict a typical pose-estimation error in our system.

We performed 35 experiments with an average of six objects in the workspace. The typical 3-D positional and orientational errors are less than 7 mm and less

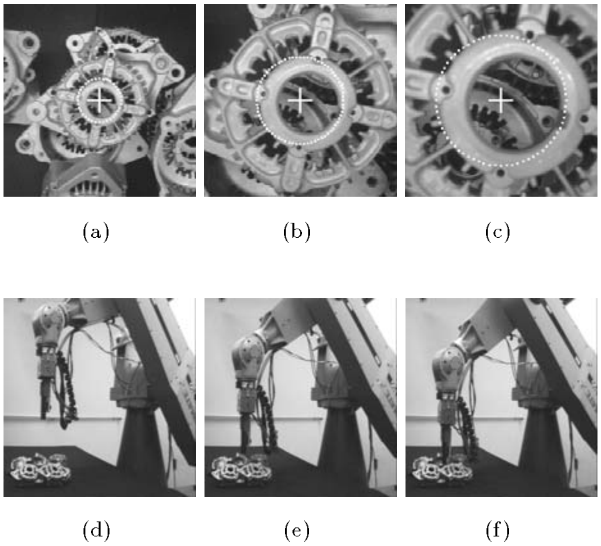


Figure 8: *Typical error of the pose estimation. These images were taken at the estimated distances of (a,d)* 300.0*mm, (b,e)* 150.0*mm and (c,f)* 100.0*mm away from the estimated centroid* $\mathbf{p}$ *of the object, as the camera was driven along the estimated normal vector* $\mathbf{n}$ *of the seed feature of the object.*

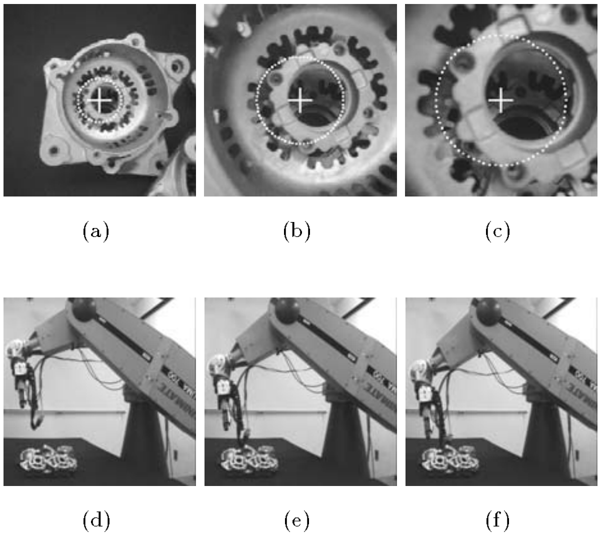


Figure 9: *Worst case of the pose estimation. As opposed to the previous figures, the estimation error is relatively large. Our algorithm, however, estimates the 3-D object pose accurately enough for the manipulator to grasp the object.*

than 10^o respectively. We classify the success rates of our system into the following four categories:

- *successful estimation*, where recognition was successful and the cross hair is within the bearing hole.
  Class rate: 91.3 %
- *unsuccessful estimation*, where recognition was successful and the cross hair fell outside the bearing hole.
  Class rate: 0.0 %
- *incomplete identification*, where identification was unsuccessful even though the seed feature was fully visible. This occurred mostly when the bearing hole appearance is distorted by other exposed background features.
  Class rate: 8.7 %
- *false identification*, where the algorithm incorrectly identified some arbitrary object as a target object.
  Class rate: 0.0 %

Another fact that we should state is that, by varying the window length of Canny detector (i.e. its *standard deviation* parameter), we can approximately select the desired types of edge features that should be extracted. For example, if a target object bears a textured surfaces, we can eliminate textural edges by using a larger standard deviation parameter. On the alternator covers, we use the default value of Canny's σ parameter, i.e. $\sigma = 1.0$.

8 Conclusion

We presented a vision-based bin-picking algorithm for complex objects, where the computational complexity can be greatly reduced by using simple object features. Our experimental results have demonstrated the robustness of this approach with real industrial objects. At the moment, we have not experimented with actual grasping of objects, but the conducted experiments has proven that the method of recognition and pose-estimation of target objects is sufficient to be extended for actual grasping. However we understand that in order to make our system a feasible solution for a given task, it requires a great deal of human assistance to build the plausible choice of object model.

References

[1] P. J. Besl, *Surfaces in Range Image Understanding*, Springer-Verlag, 1988.

[2] R. C. Bolles and R. A. Cain, "Recognizing and locating partially visible objects: the local-feature-focus method," *International Journal of Robotics Research*, Vol. 1, No. 3, pp.57-82, 1982.

[3] R. C. Bolles and P. Horaud, "3DPO: A three-dimensional part orientation system," *International Journal of Robotics Research*, Vol. 5, No. 3, pp. 3-26, 1986.

[4] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679-698, Nov. 1986.

[5] C.-H. Chen and P. G. Mulgaonkar, "Automatic vision programming," *Computer Vision, Graphics and Image Processing*, Vol. 55, No. 2, pp. 170-183, 1992.

[6] O. D. Faugeras, *Three-Dimensional Computer Vision*, MIT Press, Cambridge, MA, 1993.

[7] D. Geman and S. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6 , no. 6, pp. 609-628, Jul. 1990.

[8] W. E. L. Grimson, *Object Recognition by Computer*, MIT Press, 1990.

[9] S.L. Horowitz and T. Pavlidis, "Picture Segmentation by a Tree Traversal Algorithm," *Journal of ACM*, vol. 23, no. 2, pp. 368-388, Apr. 1976.

[10] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," International Journal of Computer Vision, vol. 1, pp. 321-331, 1988.

[11] T. Onda, H. Igura, and M. Niwakawa, "A Handling System for Randomly Placed Casting Parts Using Plane Fitting Technique," *Proc. of IEEE/RSJ IROS 1995*, vol. 3, pp. 435-440.

[12] J.M.S. Prewitt, "Object Enhancement and Extraction," pp. 75-149, *Picture Processing and Psychopictorics*, B.S. Lipkin and A. Rosenfeld, Eds., Academic Press, New York, 1970.

[13] K. Rahardja and A. Kosaka, in preparation for publication, 1996.

[14] A. Rosenfeld and A. Kak, *Digital Picture Processing*, Academic Press, 1982.

[15] S. H. Wang, R. L. Cromwell, A. C. Kak, I. Kimura, and M. Osada, "Model-based vision for robotic manipulation of twisted tubular parts: Using affine transforms and heuristic search," in *Proceedings of IEEE International Conference on Robotics and Automation*, 1994.