The International Journal of Robotics Research

http://ijr.sagepub.com

Vision-guided Robot System for Picking Objects by Casting Shadows

Amit Agrawal, Yu Sun, John Barnwell and Ramesh Raskar

The International Journal of Robotics Research 2010; 29; 155 originally published online Nov 13, 2009;
DOI: 10.1177/0278364909353955

The online version of this article can be found at:
http://ijr.sagepub.com/cgi/content/abstract/29/2-3/155

Published by:
SAGE
http://www.sagepublications.com

On behalf of:
ijrr

Multimedia Archives

Amit Agrawal
Mitsubishi Electric Research Labs,
Cambridge, MA 02139,
USA
agrawal@merl.com

Yu Sun
University of South Florida,
Tampa, FL 33620,
USA
yusun@cse.usf.edu

John Barnwell
Mitsubishi Electric Research Labs,
Cambridge, MA 02139,
USA
barnwell@merl.com

Ramesh Raskar
MIT Media Lab,
Cambridge, MA 02139,
USA
raskar@media.mit.edu

Vision-guided Robot System for Picking Objects by Casting Shadows

Abstract

We present a complete vision-guided robot system for model-based three-dimensional (3D) pose estimation and picking of singulated 3D objects. Our system employs a novel vision sensor consisting of a video camera surrounded by eight flashes (light emitting diodes). By capturing images under different flashes and observing the shadows, depth edges or silhouettes in the scene are obtained. The silhouettes are segmented into different objects and each silhouette is matched across a database of object silhouettes in different poses to find the coarse 3D pose. The database is pre-computed using a computer-aided design (CAD) model of the object. The pose is refined using a fully projective formulation of Lowe's model-based pose estimation algorithm. The estimated pose is transferred to a robot coordinate system utilizing the hand–eye and camera calibration parameters, which allows the robot to pick the object. Our system outperforms conventional systems using two-dimensional sensors with intensity-based features as well as 3D sensors. We handle complex ambient illumination conditions, challenging specular backgrounds, diffuse as well as specular objects, and texture-less objects, on which traditional systems usually fail. Our vision sensor is capable of computing depth edges in real time and is low cost. Our approach is simple and fast for practical implementation. We present real experimental results using our custom designed sensor mounted on a robot arm to demonstrate the effectiveness of our technique.

KEY WORDS—3D bin picking, active illumination, multi-flash camera, 3D pose estimation, cast shadows

The International Journal of Robotics Research
Vol. 29, No. 2–3, February/March 2010, pp. 155–173
DOI: 10.1177/0278364909353955

Figures 2–8, 10, 11, 13–18, 21, 23 appear in color online:
http://ijr.sagepub.com

1. Introduction

Humans are extremely good at identifying objects in the scene and picking them. However, developing robust and efficient vision-guided robotics systems for picking objects (Horn and Ikeuchi 1984) has proven to be a challenging task for last few decades. Typically, custom-designed mechanical and electro-mechanical systems are used to feed parts in a specific pose to the robot (Goemans et al. 2006). In some cases, manual labor is used to sort the parts from a pile or bin so that the robot can pick them up. The last decade has seen an increase in efforts towards automating the process of automatic part acquisition using vision systems. Vision sensors are increasingly being used in such robotics systems as their cost is reducing and computation is becoming cheaper and faster. They are successful in identifying, inspecting and locating parts on a conveyor belt in carefully engineered manufacturing settings. Still, current systems can only operate in very strict conditions and can handle geometrically simple objects.

Current systems for two-dimensional (2D)/three-dimensional (3D) pose estimation typically find simple features in intensity images such as lines, corners, ellipses or circles and try to infer the object pose based on the feature size and their relationships with each other. Thus, they are limited to geometrically simple shapes. Changing the part to a new object often requires developing new algorithms or extensive fine tuning. Changes in ambient illumination and complex non-uniform backgrounds lead to the failure of vision algorithms utilizing intensity-based features. It is desirable for the vision algorithms to be robust to illumination changes and to be capable of handling different object shapes. In addition, the appearance of the parts also plays an important role. Specular objects are difficult to handle due to their non-Lambertian reflectance. Feature matching is extremely difficult for specular surfaces and texture-less objects, leading to the failure of common 3D sensors to estimate reliable 3D geometry for such objects. A successful vision system should be able to handle *variations* in the operating conditions. These variations include (a) changing illumination conditions due to the ambient illumination, (b) non-uniform backgrounds and (c) objects of varying shape (3D/planar objects) as well as reflectance properties (diffuse/specular, textured/texture-less).

In this paper, we address several of these problems and present real experimental results to demonstrate the effectiveness of our system. Our system is based on using depth edges (silhouettes) of objects. We show that reliable depth edges can be obtained in real time by simply casting shadows from different directions *without* estimating 3D geometry. We show that by using depth edges as features, one can eliminate the need for accurate 3D reconstruction for 3D pose estimation, which is difficult for specular objects. Depth edges also enable our approach to be independent of scene texture and intensity edges, allowing texture-less objects as well as illumination changes to be handled easily. By using depth edges as features, we use significantly more information about the object shape than is provided by specific features such as lines, ellipses or circles. Our approach can thus handle complicated 3D shapes which may not have enough of these simple discriminating features. It also leads to a simple approach for pose estimation which typically has high computational complexity in matching set of specific features with the known model of the object. Another important distinction with the traditional 2D sensors is that the cast shadows provide occlusion information, which allows easy segmentation of the objects. Such information cannot be obtained from the intensity images. As shown in later sections, we use the occlusion information as a constraint to avoid incorrect and over-segmentation of parts. Our system currently handles *singulated* (non-stacked) objects, where objects are separated from each other but can have any possible position and orientation. We present real experimental results using a robot arm on several of the above scenarios. Extensions 1 and 2 show our system in operation for picking an object with a complex 3D shape (shown in Figure 1) and a specular object (shown in Figure 2), respectively.

1.1. Contributions

The contributions of our paper are as follows:

- We present a complete system for 3D pose estimation and picking of objects using a low-cost modified sensor.
- We show that depth edges are sufficient to estimate the precise 3D pose of the object for picking without requiring absolute depths.
- We show that cast shadows can be used to estimate depth edges as well as to simplify segmentation of objects. We use physical constraints based on depth edges and shadow boundaries that avoid incorrect and over-segmentation of objects.

1.2. Benefits and Limitations

Our approach has several benefits over related approaches as follows:

- Our system handles complex ambient illumination and non-uniform shiny backgrounds.
- Since depth edges are independent of scene reflectance edges, our approach works well with objects having different reflectance properties such as diffuse (Lambertian), specular as well as texture-less objects.
- Our approach can handle objects of different shapes and sizes since it does not depend on object-specific features such as lines, circles, etc. Thus, new objects can be handled without any change in the algorithm, except for the pre-computation of a database of the object silhouette features.
- Our system utilizes a low-cost modified 2D sensor and does not require an expensive 3D sensor, yet provides 3D position and orientation of objects.

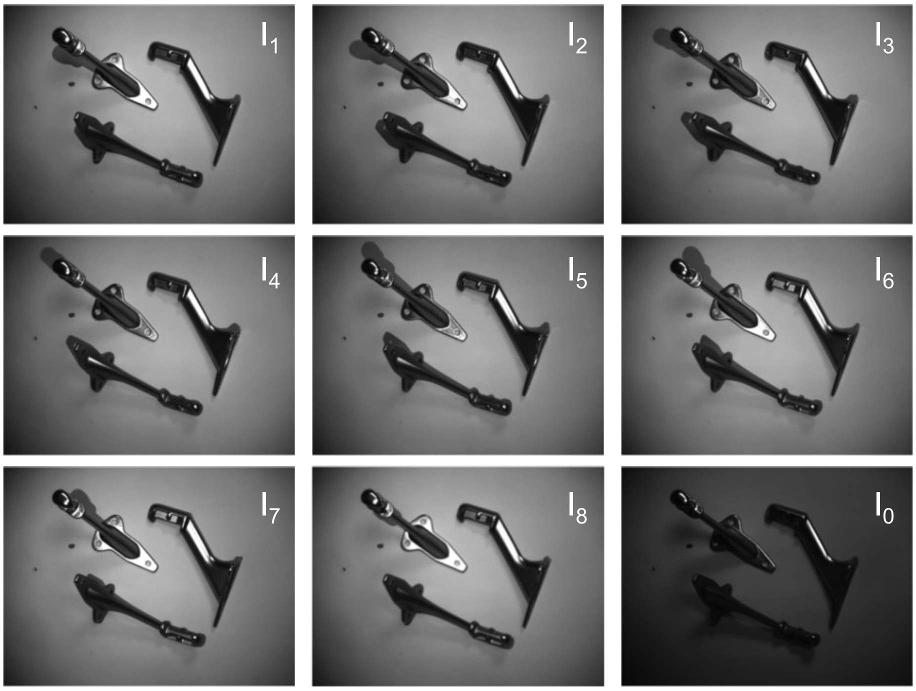


Fig. 1. Captured images of a scene with three brass hooks using our camera. Here I_1 to I_8 correspond to the images taken with different flashes (I_1 corresponds to the image taken with the flash on top of the camera); I_0 corresponds to the image taken without any flash. Note how the shadows move with the position of the flash.

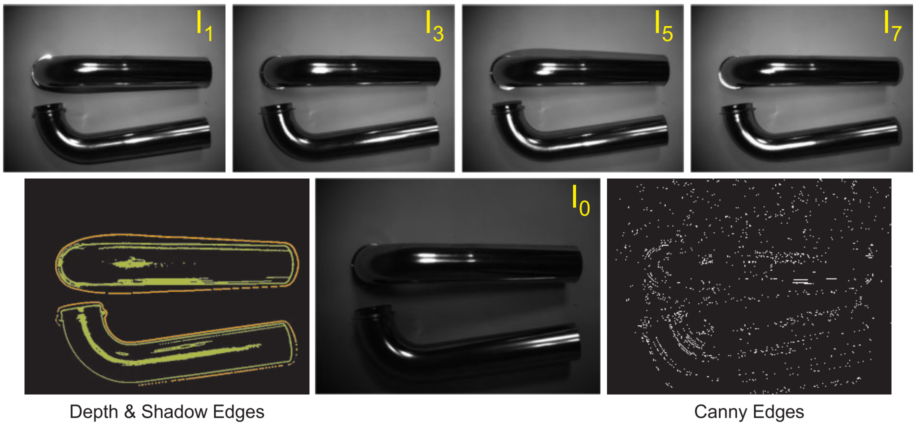


Fig. 2. Specular objects. 3D range scanners do not provide reliable geometry information on specular objects. In addition, reliable 2D intensity-based features are not also not obtained on specular objects due to inter-reflections. Our approach can easily handle specular objects and estimate reliable depth edges.

- Our approach is well suited for real applications as it is robust, simple and leads to fast implementation.

Some of the limitations of our approach are as follows:

- We currently cannot handle stacked specular objects, as the shadows cast *on* the specular objects are not estimated reliably.
- For thin objects, the shadows may become detached from the object leading to spurious depth edges.
- Although we can handle shiny backgrounds, dark (black) backgrounds reduce the contrast of shadows. Depth edges cannot be estimated reliably in that case.

- Transparent and translucent objects cannot be handled by our approach due to inter-reflections of light leading to unreliable depth edges.

Although the proposed system works in open loop, closed-loop error correction and visual servoing approaches can be added to the system. These approaches could use intensity-only features or depth edges for dynamic scenes as shown in Raskar et al. (2004).

1.3. Related Work

Vision-based robot systems have been the focus of significant research in both academia and industry. These systems typically employ single/multiple cameras and illumination devices to analyze the scene, locate the part and provide feedback to the robot arm for subsequent operations. To successfully grip and pick up parts, the vision system needs to recognize the position and the orientation of the objects. The vision sensor can be mounted on the end-effecter of the robot arm (Allen et al. 1993; Hutchinson et al. 1996; Chaumette 1998) or located at a position near the robot (Liu et al. 1999; Astolfi et al. 2002; Piepmeier et al. 2004).

Vision-based robot systems (Horn 1986) can be broadly classified into (a) 2D, (b) 2.5D and (c) 3D vision systems.

2D vision systems have been successfully employed in several industrial applications (see, e.g., http://www.cognex.com). Most of the current vision systems fall into this category. These systems have been used for several tasks such as inspection and limited part acquisition. Typically such systems can recognize the in-plane orientation and location of the part but cannot determine the out-of-plane rotation and the distance of the part precisely. They require parts to be non-overlapping and placed on a flat surface. A model-based approach can be used for 3D pose estimation. Edges are extracted in captured 2D images, and the contours of the object are detected by connecting the edges. The detected contours are then matched with a stored computer-aided design (CAD) model and the location and orientation of the object is estimated (Tuji and Nakamura 1975; Perkins 1977; Turney et al. 1985). However, these systems are highly susceptible to background color and illumination variations. In contrast, our approach based on depth edges can easily handle challenging backgrounds and non-uniform illumination.

2.5D vision systems augment the 2D vision system by also calculating the distance of the object from the change in the size of the image of the object or by finding depths at a few points. However, they cannot estimate the exact out-of-plane rotation and are often unreliable in their depth estimates. Such systems often misleadingly claim to estimate 3D pose but can only handle a few degrees of out-of-plane rotation for simple objects.

3D vision systems use sensors for estimating the 3D geometry of the scene. The object is recognized and localized by comparing the estimated range image with the standard orientated CAD models in a database (Besl and Jain 1985; Chin and Dyer 1986; Brady et al. 1988). 3D range data can either be obtained with shape-from-texture (Brady 1981), laser triangulation or edge-based binocular stereo (Pollard et al. 1985; Ayache and Lustman 1991). Some of the popular approaches are described below. Our system does not require a 3D sensor but it can estimate the 3D pose of the object using depth edges:

- *Stereo vision*. Stereo systems use two cameras looking at the object to estimate the object's depth. The corresponding features are localized in the images captured from the two cameras and the geometric relationship between the cameras can be used to identify the depth of the feature points. However, finding the corresponding features itself is a challenging problem, especially for parts which are specular, shiny and homogeneous (texture-less). In addition, stereo has a high degree of sensitivity of the depth estimates with the noise in feature localization. Another problem with stereo is that the depths are recovered only at the feature points and not on the entire object. The reduced accuracy can be tolerated for certain applications such as un-racking body panels in body shops, but is not sufficient for accurate picking of the object.

- *Laser triangulation*. These systems use structure light (see, e.g., http://www.sick.com/gus/products/new/s300/en.html.html) to create a pattern on the surface of the object, which is viewed from a camera. The laser triangulation can recover the 3D point cloud on the object surface. This technology has been used for applications involving edge tracking for welding, sealing, glue deposition, grinding, waterjet cutting and deburring of flexible and dimensionally unstable parts. Use of lasers for part pose determination requires registration and accounting for shadows/occlusions. Laser triangulation does not work well on specular shiny objects due to laser light being reflected from the object surface. In addition, the use of lasers also leads to safety issues when deployed in close proximity to human operators.

Active illumination: controlling illumination is important for vision algorithms. Back-light illumination is used to segment objects by illuminating them from behind. In bright-field illumination, the light comes in approximately perpendicularly to the object surface. The whole object appears bright, with features displayed as a continuum of gray levels. This sort of illumination is used for most general-vision applications. In dark-field illumination, the object is illuminated at a low angle from a point parallel to its surface. Texture and other angular features appear bright while most of the object appears dark. Dark-field illumination is useful for imaging surface contamination, scratches and other small raised features. In coaxial illumination, the object is illuminated from precisely the direction of the imaging lens using a beam-splitter. Coaxial illumination is used to inspect features on flat, specular surfaces, to image within deep features and to eliminate shadows. Shadows are usually considered a nuisance, but in our approach the illumination source is intentionally placed close to the camera and cast shadows are utilized to estimate depth edges.

Several vision approaches use active illumination to simplify the underlying problem. Nayar et al. (1995) recover the shape of textured and textureless surfaces by projecting an illumination pattern onto the scene. Shape from structured light (Scharstein and Szeliski 2003; Zhang et al. 2004) has been an active area of research for 3D capture. Raskar et al. (2004) proposed the multi-flash camera (MFC) by attaching four flashes to a conventional digital camera to capture depth edges in a scene. Crispell et al. (2006) exploited the depth discontinuity information captured by the MFC for a 3D scanning system which can reconstruct the position and orientation of points located deep inside concavities. The depth discontinuities obtained by the MFC have also been utilized for robust stereo matching (Feris et al. 2005), recognition of finger-spelling gestures (Feris et al. 2004b), automated particle size analysis with applications in mining and quarrying industry and for 3D segmentation (Koh et al. 2007). Our approach also uses a variant of MFC (with eight flashes) to extract depth discontinuities, which are then used to segment objects and estimate their 3D pose.

Model-based pose estimation has been a topic of significant research in computer vision. Initial work on using a CAD model and features in intensity images for 3D pose estimation was shown in Lowe (1987, 1991). Dementhon and Davis (1995) proposed an algorithm for pose estimation from given 2D/3D correspondences. Silhouettes have also been used for model-based human pose estimation (Gavrila and Davis 1996; Sminchisescu and Telea 2002; Agarwal and Triggs 2004) and object recognition/classification (Marr and Nishihara 1978; Gorelick et al. 2006). In these approaches, background segmentation to obtain the silhouettes is typically a difficult problem. We show that by using cast shadows, obtaining silhouettes is easy even for challenging environments. Several techniques on classifying objects based on silhouettes assume complete closed contours, but our approach can work with incomplete silhouettes and missing depth edges.

2. System Overview

Our system consists of a robot arm equipped with a gripper and a vision sensor as shown in Figure 3. The overview of our approach is shown in Figures 4 and 5. The robot arm is mounted with a camera surrounded by eight flashes. Eight images are captured, each by turning on a different flash. In addition, an image is also captured with all of the flashes turned off. The images are processed to obtain depth edges as described in Section 2.1. The silhouettes are then segmented into different objects (Section 2.2). Using the CAD model of the object, a database of silhouettes is generated for several poses. The obtained silhouettes are matched against the database to estimate the coarse pose which is further refined using the CAD model (Section 3). The final pose estimated is transferred to the robot coordinate system using the hand–eye and camera calibration parameters. The location of a pre-decided pick point and pick direction for the object is estimated in the robot coordinate system, using which the robot arm picks the object. Next we describe each of these steps in detail.

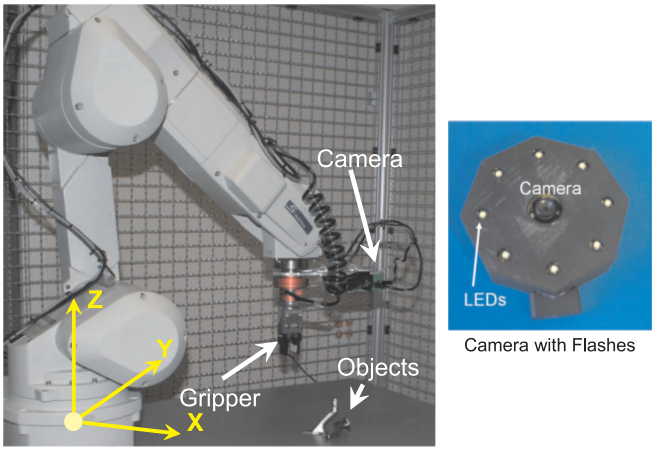


Fig. 3. Experimental setup. A six-degree-of-freedom robot arm equipped with a pneumatic gripper is used for experiments. The camera attached to the robot arm is housed inside a plastic case and is surrounded by eight light emitting diodes controlled by a micro-controller. Objects to be picked are placed on a table as shown. The world coordinate system is attached to the base of the robot with the axis directions as shown.

2.1. Estimating Depth Edges and Shadow Edges by Casting Shadows

The key difference in our system over other systems is our vision sensor, which consists of a video camera surrounded by eight flashes. This technique for finding depth edges was first proposed by Raskar et al. (2004). Illumination from flashes is often used in machine vision in the form of ring lights to avoid shadows, or colored LEDs for color-based analysis. In our approach, we turn on one flash at a time to cast shadows and capture an image. Since shadows will be cast due to object boundaries and not due to reflectance boundaries, shadows give information about the depth discontinues of the object as shown in Figure 1. To compute the depth edges, we use the technique described in Raskar et al. (2004) and Koh et al. (2007). Let $I_1 \ldots I_8$ denote the images taken with flashes turned on.

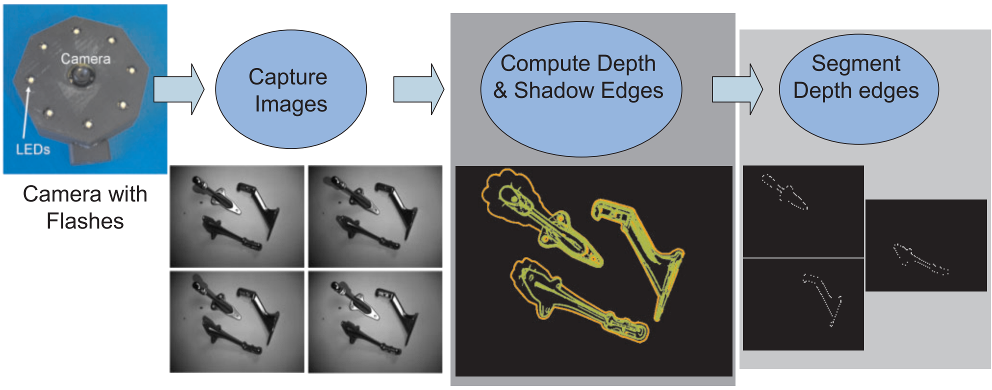


Fig. 4. Flowchart of our approach. Eight images are captured by turning on light emitting diodes surrounding the camera in succession (only four are shown for simplicity). Cast shadows are utilized to compute depth edges (green) and shadow edges (orange) in the scene. Depth edges are then segmented into silhouettes corresponding to different objects. Each silhouette is then used to estimate the 3D pose of the corresponding object as shown in Figure 5.

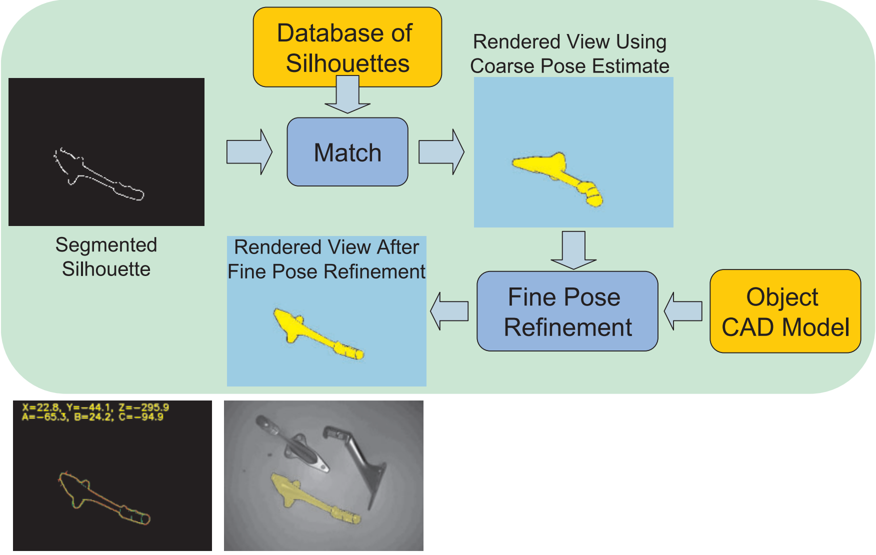


Fig. 5. Flowchart describing 3D pose estimation. The pose for each object is estimated using the corresponding segmented silhouette. A coarse pose estimate is achieved by feature matching with a pre-computed database of silhouette features to obtain approximate rotation angles. Fine pose refinement using the CAD model is then performed to estimate accurate rotation and translation of the object. Overlay of the rendered CAD model on I_0 and the rendered CAD model silhouette on the segmented object silhouette using the final estimated pose is shown at the bottom for one of the objects.

2.1.1. Canceling Ambient Illumination Effects

To cancel the effects of the ambient illumination, we capture an image without any flash (referred to as I_0). The effect of ambient illumination is removed by subtracting I_0 from $I_1 \ldots I_8$. This simple procedure provides our system with the robustness towards illumination changes in the scene. Let $D_1 \ldots D_8$ denote the images after subtracting I_0.

2.1.2. Finding Depth Edges

To compute depth edges, first a max composite image, $D_{\max}$, is obtained by taking the maximum of intensity value at every pixel from $D_1 \ldots D_8$:

$$D_{\max}(x, y) = \max_{i=1\ldots 8}(D_i(x, y)). \qquad (1)$$

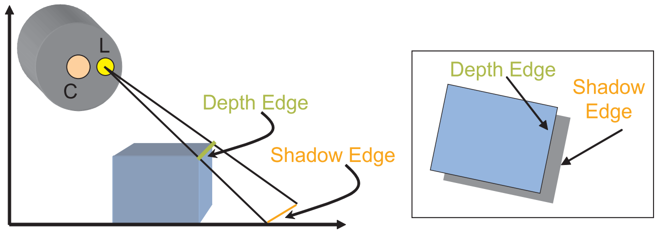


Fig. 6. Depth and shadow edges. Left: C denotes the camera and L denotes one of the light sources. Depth edges are attached to the object, while shadow edges are defined as the boundary between the shadow and the background. Right: Depth and shadow edges in the camera image.

Note that $D_{\max}$ will be a shadow-free image. Then, ratio images are calculated as

$$R_i(x, y) = D_i(x, y)/D_{\max}(x, y), \quad i = 1 \ldots 8. \tag{2}$$

Ideally, in the absence of noise with linear camera response, each ratio image R_i equals zero for shadowed and one for non-shadowed parts of the image. Depth edges are obtained by estimating the *foreground to shadow* transition in each ratio image (Figure 6) and combining all of the estimates. To handle noise and non-linearities, we run oriented Sobel filters on ratio images and add the filter responses to obtain a depth edge confidence map C_{depth}, which is thresholded to obtain binary depth edges.

$$C_{\text{depth}}(x, y) = \sum_{i=1\ldots8} h_i(x, y) * R_i(x, y), \tag{3}$$

where $*$ denotes convolution and $h_i(x, y)$ corresponds to the oriented Sobel filter according to the position of the flash with respect to the camera for that image. For example, for the image taken with the flash on the left side of the camera, the shadows will be casted on the right of the objects. Scene depth edges then correspond to those vertical edges, which are bright on the left and dark on the right and can be obtained by applying the filter given by $\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$. Similarly, for the image corresponding to the flash on the right of the camera, the corresponding filter is given by $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$. We use hysteresis thresholding to obtain binary depth edges. Note that the entire procedure for obtaining depth edges involve simple operations such as filtering and obtaining ratios and can run in real time.

2.1.3. Finding Shadow Edges

We define *shadow edges* as the *shadow to background* transition in captured images (Figure 6). The shadow edge confidence map, C_{shadow}, can be obtained similarly by simply flipping the sign of corresponding oriented Sobel filter:

$$C_{\text{shadow}}(x, y) = \sum_{i=1\ldots8} -h_i(x, y) * R_i(x, y). \tag{4}$$

Figure 7 shows the estimated depth and shadow edges for the scene containing three brass hooks as shown in Figure 1. Note that the region between the depth and shadow edges corresponds to the shadow region and thus shadow edges provide occlusion information. Another key idea of our approach is to use this information present in the shadow edges. In the next section, we show how shadow edges can be used to significantly simplify the segmentation of depth edges into object silhouettes. Note that the shadow edges are unique to our approach and cannot be obtained from traditional 2D intensity images.

2.2. Segmentation Using Depth Edges and Shadow Edges

Image and range segmentation (Hoover et al. 1996; Yim and Bovik 1998; Comaniciu and Meer 2002; Shapiro and Stockman 2001; Shi and Malik 2000; Christoudias et al. 2002) is a well-researched area in image processing and computer vision. Although 2D segmentation can segment an image into semantic regions, it cannot provide occlusion information due to the lack of shadows or depth information. Even when a depth map of the scene is available, we need to explicitly find occlusions using depth edges. In contrast, using depth and shadow edges together leads to a simple and effective segmentation algorithm for singulated objects.

The key issue in segmenting obtained silhouettes into different objects is missing depth edges and incomplete contours. Suppose that complete contours were obtained for depth edges and shadow edges. Since we assume that the objects are not stacked and are singulated, we can simply find connected components in depth edges. Each connected component then corresponds to a particular object silhouette. However, in practice, the silhouettes are incomplete and noisy due to image noise, specularities on specular objects, pixel saturation, low-contrast (soft) shadows and other non-linearities. Thus, one needs to complete depth edges to form closed contours for segmentation.

Edge completion is also an active area of research in image processing. To complete missing edges, Gestalt rules are applied to link perceptually similar edges (Kimchi et al. 2003). This involves several heuristics such as edge proximity, edge length, etc. These techniques, however, require several tuning parameters, are not robust and are highly susceptible to noise. To this end, we propose *physical* constraints which help

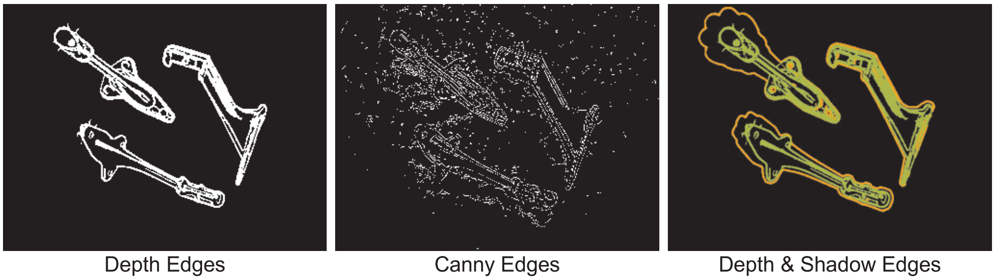


Fig. 7. Detected depth edges (green) and shadow edges (orange) using our approach on the scene shown in Figure 1. In comparison, Canny edges (Canny 1986) on ambient image I_0 are also shown in the middle. Note that the Canny intensity edges are noisy and do not provide reliable features due to the non-Lambertian reflectance properties of the brass hook.

in completing depth edges. In any scene, cast shadows lead to physical constraints between the depth and shadow edges. Cast shadows have a penumbra region and depth edges as defined above correspond to the discontinuity on one side of the penumbra, while the shadow edges correspond to the discontinuity on the other side of the penumbra. Thus, two physical constraints can be derived as follows:

1. for every depth edge pixel, there exists a shadow edge pixel;
2. a depth and shadow edge cannot exist simultaneously at the same pixel;

These two rules enable us to complete missing depth edges to form closed contours. We achieve this by fitting line segments to the depth edges and shadow edges and extending each depth edge line segment. A consequence of these rules is that (a) for every extended depth edge line segment, there should be a parallel shadow edge line segment, and (b) extended depth edges line segments cannot intersect any existing shadow edge. Every extended line segment is checked with respect to the above rules and is kept if it satisfies both of them. This significantly helps to remove spurious connections as shown in Figure 8. In practice, we discard line segments with length smaller than a threshold (15 pixels). Note that if a line segment is connected on both ends to other line segments, it is not extended. Thus, only those line segments with at least one open end-point (terminal points) are checked for extension. In addition, the process is non-recursive. Typically, a few tens of terminal points are obtained and the entire process take less than 0.5 s in C/C++.

At the end of this stage in the system, we obtain close contours for depth edges for which connected component analysis results in segmentation as shown in Figure 9. An important point to note here is that the extended depth edges are only used for segmentation and not for pose estimation. Once the silhouettes corresponding to an object are segmented, we only use the original depth edges for pose estimation. We also ignore the depth edges inside the object to avoid spurious depth edges due to specularities and only use the outermost silhouettes for pose estimation, but this may lose useful internal silhouettes.

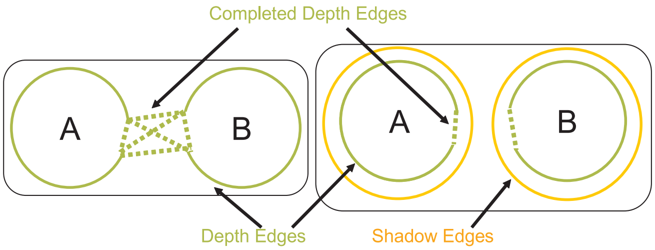


Fig. 8. Completing depth edges using information from shadow edges. Two objects A and B are shown with depth edges in green. Depth edges are missing for both A and B. Without using any constraints, edge completion could result in six new connections as shown on the left and heuristics need to be applied to avoid incorrect connections. Instead of using heuristics, we use the physical constraint that depth edges and shadow edges cannot intersect. This automatically removes the incorrect connections as shown on the right.

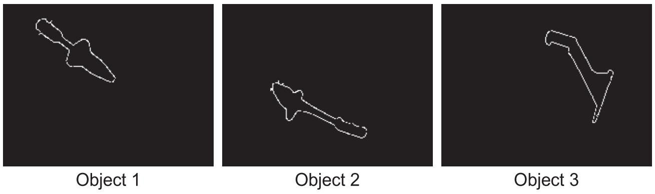


Fig. 9. Segmentation of depth edges into different objects. Depth edges inside the object (internal silhouettes) are ignored to avoid spurious edges due to specular reflections and only external silhouettes are kept. Each of these three silhouettes is used to estimate the pose of the corresponding object.

3. 3D Pose Estimation

In the last section, we described how depth and shadow edges are estimated and used for object segmentation. In this section, we describe how the segmented silhouettes are used for estimating the 3D position and orientation of the object. For multiple objects in the scene, the process is repeated for every object. We assume that a CAD model of the object is known in advance and thus our approach is model based. The pose estimation involves obtaining the rotation and translation parameters of the object. The pose estimation is performed in two steps for faster processing. In the first step, a coarse pose is estimated to obtain the approximate rotation angles. In the second step, all six rotation and translation parameters are optimized.

3.1. Coarse Pose Estimation

In coarse pose estimation, the optimization is performed over the rotation angles only. Several techniques based on moments have been proposed for estimating the pose based on silhouettes and we propose to use Zernike moments. The Zernike moment formulation outperforms the alternatives in terms of noise resilience, information redundancy and reconstruction capability (Teh and Chin 1988). The pseudo-Zernike formulation proposed by Bhatia and Wolf (1954) further improves these characteristics.

Let $s(x, y)$ be the binary image corresponding to the estimated silhouette of an object. The complex Zernike moments, $M(x, y)$, of $s(x, y)$ are obtained by taking the projection of $s(x, y)$ on complex Zernike polynomials V_{mn}

$$\begin{aligned} M(x, y) &= \frac{m+1}{\pi} \int_x \int_y s(x, y) V_{mn}(x, y)^* \, dx \, dy, \\ x^2 + y^2 &\leq 1, \end{aligned} \tag{5}$$

where m defines the order of the moments and $*$ denote complex conjugate. The integer n is such that

$$m - |n| = \text{even}, \quad |n| \leq m. \tag{6}$$

The Zernike polynomials V_{mn} are expressed in polar coordinates as

$$V_{r,\theta} = R_{mn}(r) \exp(\sqrt{(}-1)n\theta), \tag{7}$$

where r, θ are defined over the unit disk and

$$R_{mn}(r) = \sum_{s=0}^{(m-|n|)/2} (-1)^s \frac{(m-s)!}{s!((m+|n|)/2-s)!((m-|n|)/2-s)!} r^{m-2s} \tag{8}$$

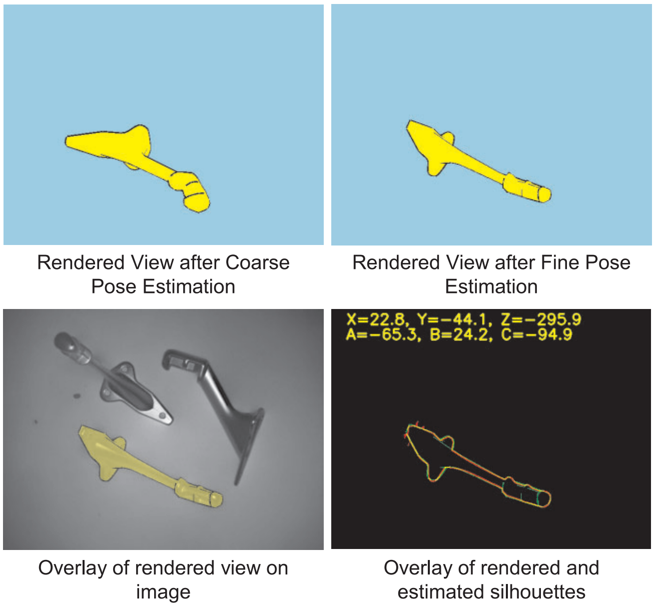
if (6) is satisfied, otherwise $R_{mn}(r) = 0$.

To calculate the Zernike moments, the image (or region of interest) needs to be mapped to the unit disk using polar coordinates. We first find the bounding box of the segmented object silhouette in the image and resample the bounding box to a square of size 129 × 129. We use $m = 6$, giving rise to 36 Zernike basis polynomials of size 129 × 129. Thus, for each silhouette, a 36-dimensional feature vector is obtained.

3.1.1. Building Pose Database

Given the CAD model of the object, we find silhouettes in different poses and store their Zernike moments in a database. We sample the pose space uniformly, at an equal interval of 9°, leading to $360/9 = 40$ rotations along each axis. This results in a database of $40^3 = 64{,}000$ poses. We use a fast silhouette rendering algorithm described in Raskar and Cohen (1999) to compute silhouettes using a CAD model. The entire database generation takes 10–15 minutes on a desktop PC. Note that this database generation needs to be done only once for an object.

The coarse pose is obtained by finding the L_2 norm of the Zernike moments of the query silhouette with the moments stored in the database and choosing the pose corresponding to the minimum L_2 norm. Figure 10 shows the result of the coarse pose estimation for object 2 in Figure 9.


Rendered View after Coarse Pose Estimation

Rendered View after Fine Pose Estimation

Overlay of rendered view on image

Overlay of rendered and estimated silhouettes

Fig. 10. Pose estimation results on object 2. Top left: Rendered CAD model using rotation angles obtained after coarse pose estimation. Top right: Rendered CAD model after fine pose refinement, which updates all six rotation and translation parameters. Bottom left: Overlay of the rendered CAD model on I_0 according to the final estimated pose. Bottom right: Overlay of the rendered CAD model silhouette (green) on the segmented object silhouette (red) shows the success of the fine pose refinement. Here X, Y, Z denote translation and A, B, C denote rotation angles after fine pose refinement.

3.2. Fine Pose Refinement

Note that since the Zernike moments are normalized with respect to the scale and the translation, the obtained coarse pose is close to the correct pose only in terms of rotation angles. The fine pose refinement procedure then updates all six parameters. The goal in pose refinement is to find that rotation R and translation T, for which the projection of the CAD model silhouette matches the segmented silhouette of the object in the scene. We use OpenGL to compute the 3D silhouette of the CAD model for the given pose.

We refine the pose starting with the rotation angles given by the coarse pose estimate. The initial translation and scale is obtained by matching the scale and image translation of the projected CAD model silhouette with the segmented object silhouette. Note that given a set of 3D/2D correspondences, one could use existing algorithms for model-based pose estimation (e.g. the algorithm of Lowe (1987)). However, the 3D silhouettes depend on the pose itself and it is computationally expensive to update them at each iteration.

For fine pose refinement, we use an outer minimization loop. At each iteration of outer minimization, the CAD model is rendered and the 3D coordinates of the CAD model silhouette are obtained using the rendered silhouette and the OpenGL depth buffer. Then, correspondences between the 3D CAD model silhouette and the segmented 2D object silhouette are obtained. Given this set of 3D–2D correspondences, rotation and translation are updated using a fully projective formulation described in Araujo et al. (1998), which is an improvement of the original algorithm of Lowe (1987). The updated pose is again used to obtain new 3D silhouettes and correspondences for the next iteration of outer minimization. The error at each iteration of outer minimization is calculated using the mismatch between the projected CAD model silhouette and the segmented object silhouette using the distance transform (Fabbri et al. 2007). Let $s(x, y)$ denote the segmented object silhouette and $p(x, y)$ denote the projected CAD model silhouette. Then the error between them is defined as

$$e = \frac{\sum_{x,y} s(x, y)d(p(x, y))}{n_s} + \frac{\sum_{x,y} p(x, y)d(s(x, y))}{n_p}, \quad (9)$$

where $d(\cdot)$ denote the distance transform operator, n_s denote the number of silhouette pixels in $s(x, y)$ and n_p denote the number of silhouette pixels in $p(x, y)$. If both silhouettes match, e will be equal to zero. Outer minimization is performed until the error goes below some pre-defined threshold (0.05). Usually 10–15 iterations are sufficient. After fine pose refinement, the rotation and translation of the object in the camera coordinate system is known as shown in Figure 10.

3.3. Picking the Object

Let R and T denote the estimated rotation and translation of the object in the camera coordinate system. Let

$$\mathbf{M}_{\text{camera}}^{\text{object}} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$$

denote the 4 × 4 estimated pose matrix. The robot arm is equipped with a gripper for picking the object. In order to pick the object, it needs to be located in the world coordinate system. First, the transformation between the camera and the robot gripper ($\mathbf{M}_{\text{gripper}}^{\text{camera}}$) is obtained by the hand–eye calibration (Shiu and Ahmad 1989; Zhuang and Roth 1996). The transformation between the robot gripper and the world coordinate system $\mathbf{M}_{\text{world}}^{\text{gripper}}$ can be computed with forward kinematic and encoder readings. The transformation matrix of the object in the world coordinate system is then given by

$$M_{\text{world}}^{\text{object}} = M_{\text{world}}^{\text{gripper}} M_{\text{gripper}}^{\text{camera}} M_{\text{camera}}^{\text{object}}. \quad (10)$$

In order to pick the object, a pick point and a pick direction on the CAD model need to be specified. If $p_{\text{object}} = \{p_x, p_y, p_z\}$ denotes the pick point on the CAD model, then the location of the pick point in the world coordinate system, p_{world}, is given by

$$p_{\text{world}} = M_{\text{world}}^{\text{object}} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}. \quad (11)$$

Then p_{world} is sent to robot controller. The gripper is also rotated according to the final pose angles to align with the pick direction (e.g. vertical) with respect to the object. Note that the pick point and the pick direction can be different for different poses of the object. A trajectory is computed and the robot controller moves the grippers accordingly. After the gripper reaches the gripping pose, it closes to pick up the object and then moves it to a predefined location with a new trajectory. If there are multiple objects in the scene to pick, pose estimation for the next object is performed while the robot is picking the current object to reduce operational delay.

4. Results

In this section, we demonstrate the effectiveness of our system using several examples on objects with complex 3D shapes, texture-less objects, shiny backgrounds and specular objects. Extensions 1 and 2 show our system in operation for picking an object having complex 3D shape (shown in Figure 1) and a specular object (shown in Figure 2), respectively.

4.1. Implementation

Our system consists of a Mitsubishi MELFA RV-6S six-axis industrial robot equipped with a pneumatic gripper as shown

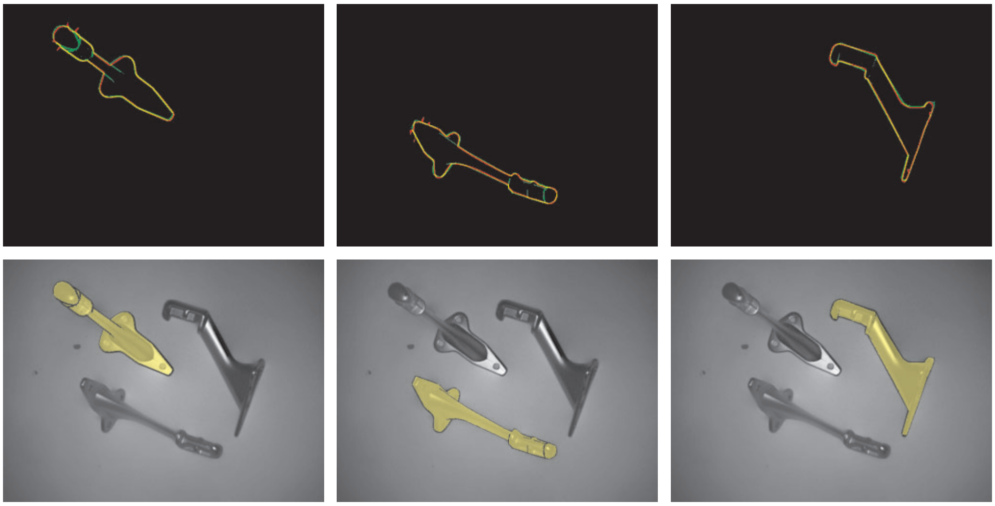


Fig. 11. 3D pose estimation results for all three objects shown in Figure 1. The top row shows the overlay of the rendered CAD model silhouette (green) according to the final estimated pose on the segmented object silhouette (red). The bottom row shows the overlay of the rendered CAD model according to the final estimated pose on I_0.

in Figure 3. The robot is directly controlled by a Mitsubishi MELFA CR2B controller. The robot has 0.02 mm repeatability and has been fully calibrated by Mitsubishi. The vision sensor is composed of a *Dragonfly* VGA camera from *Point-Grey* (http://www.ptgrey.com) surrounded by eight *Lumiled* light emitting diodes (LEDs) and is housed in a plastic box. A micro-controller inside the camera box (Figure 3) triggers the camera and flash synchronously. Extension 3 shows the flashes triggering in succession around the camera.

The camera is rigidly mounted onto the robot hand immediately after the wrist roll joint. The camera is calibrated using the Matlab Camera Calibration Toolbox available at http://www.vision.caltech.edu/bouguetj/calib_doc/. Hand–eye calibration is performed using the software available at http://www.vision.ee.ethz.ch/~cwengert/calibration_toolbox.php. The center of the camera is 128.3 mm away from the center of the gripper in the vertical direction and 150.4 mm off the center of the robot wrist roll joint, as estimated by the hand–eye calibration. The camera is placed ≈ 375 mm above the table for capturing images. Nine images are captured, eight with individual flashes turned on and the last with all flashes turned off to capture the contribution of the ambient illumination. Software is written in C/C++ and takes ≈ 1 second for image capture, segmentation and coarse pose estimation and 5–10 seconds for fine pose refinement depending on the complexity of the CAD model. Note that for multiple objects, fine pose refinement for the next object is done while picking the current object to reduce the operational delay.

4.2. Objects With Complex 3D Shape

The brass hook example shown in Figure 1 is an example of an object with complex 3D shape. In addition, the brass hook does not have diffuse Lambertian reflectance properties. The 3D scanner fails in obtaining a reliable geometry for this object. Our approach can easily find silhouettes of this object having complex 3D shape and non-Lambertian reflectance properties. Figure 11 shows the pose estimation result for all three objects shown in Figure 1. Extension 4 shows a video of fine pose refinement starting from the initial coarse pose for one of the brass hooks. Extension 1 shows a video of the robot picking two brass hooks from the table and placing them in a pre-determined pose on the side of the table.

4.3. Non-uniform Shiny Background

Shiny reflective backgrounds create significant problems for 3D scanners and 2D vision systems. In contrast, our system can work well even in such harsh environments. A example is shown in Figure 12 where we place a shiny metallic plate as the background. First, note that the ambient illumination image (I_0) has non-uniform illumination due to the metallic background and thus leads to a significant amount of noise for Canny edges. Second, flashes result in strong specularities and saturation in images as shown in Figure 12. Note that the specularities on the background change their spatial location in the image as the flashes go around the camera. This fact is used to remove the specular highlights. We use the gradient domain method described in Raskar et al. (2004) and Feris

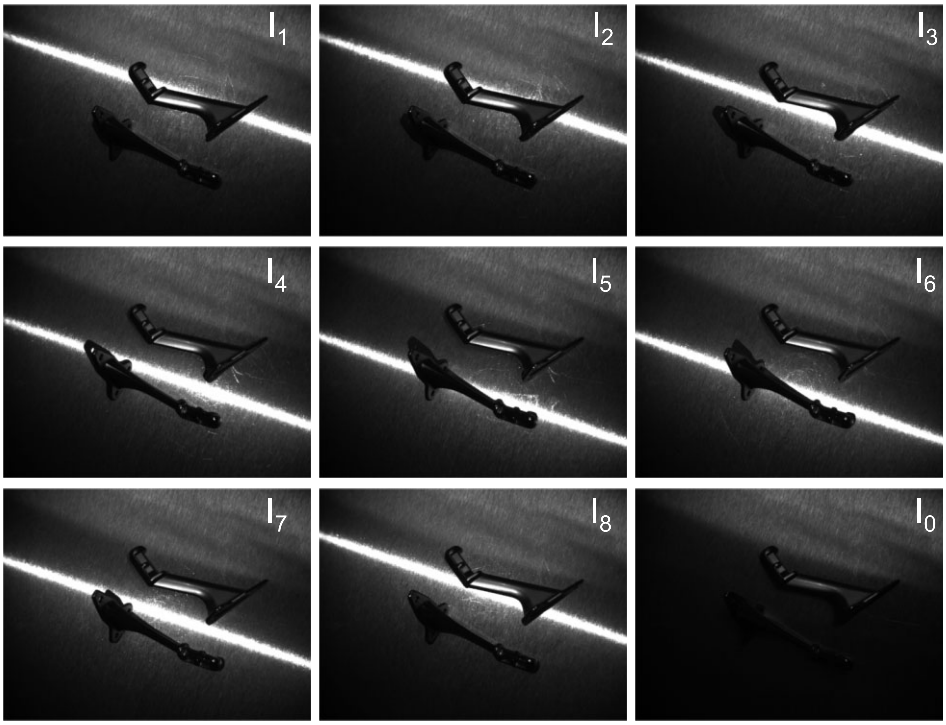
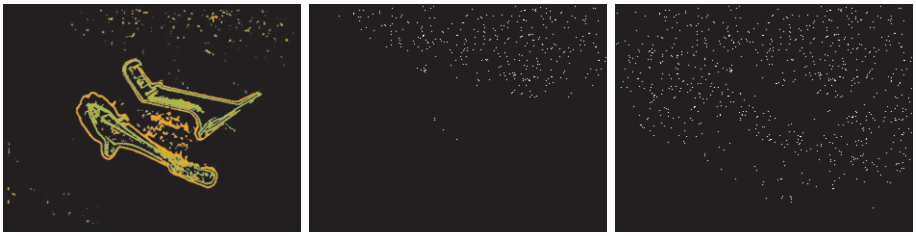


Fig. 12. Non-uniform shiny background. A metallic plate is placed as the background. The ambient illumination image I_0 shows non-uniform illumination on the background. Images captured with a flash show specularities and highlights due to the metallic plate. These can be removed if their location in the image changes.



Depth & Shadow Edges

Canny Edges, High Th = 0.05

Canny Edges, High Th = 0.01

Fig. 13. Non-uniform shiny background. Our technique results in reliable depth edges. In comparison, Canny intensity edges (on ambient illumination image) result in significant noise due to non-uniform illumination and metallic background. Shown are the Canny edge detection results using two different thresholds. Increasing the threshold reduces noise but also loses important edges of the objects.

et al. (2004a) to reduce the specular highlights on the background. Figure 13 shows the depth edges estimated using our technique compared with Canny edges on I_0. Our approach is robust against the effects of strong highlights and non-uniform background illumination. The 3D pose estimation result on both of the objects are shown in Figure 14.

4.4. Texture-less Objects

Our approach can also handle texture-less objects with no change in algorithm. A challenging example of white objects on a white background is shown in Figure 15, on which intensity edge detection does not give reliable features. Stereo-based algorithms will also fail on texture-less objects. Note that the depth edge estimation using our technique is noiseless. Figure 16 shows the overlay of the rendered CAD model silhouette on the segmented object silhouette and the rendered CAD model on I_0 after final pose estimation.

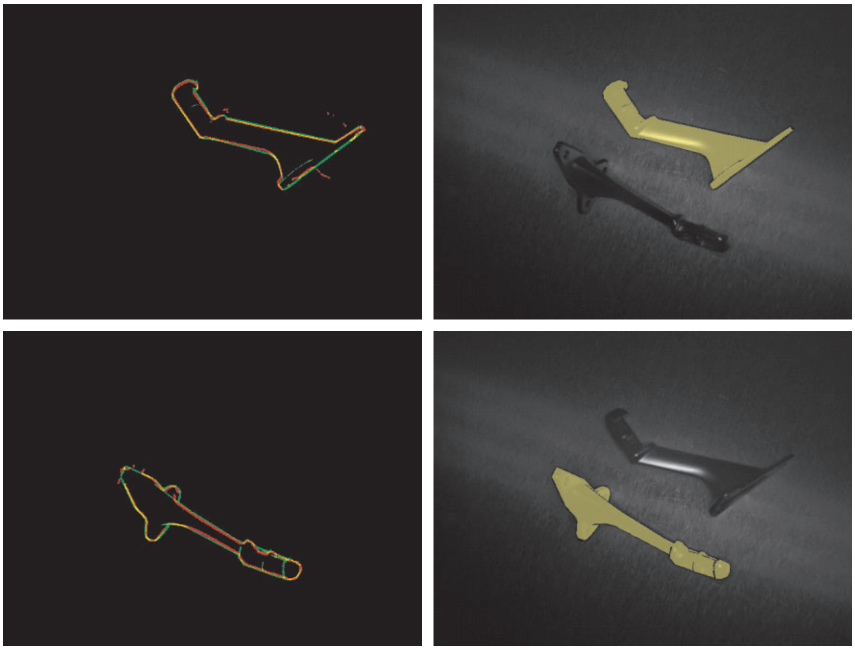


Fig. 14. Non-uniform shiny background. Overlay of the rendered CAD model silhouette on the segmented object silhouette and the rendered CAD model on I_0 for both objects after fine pose estimation. Note that although depth edges have noise, silhouettes used for pose estimation are clean as only the outermost object boundary is utilized.

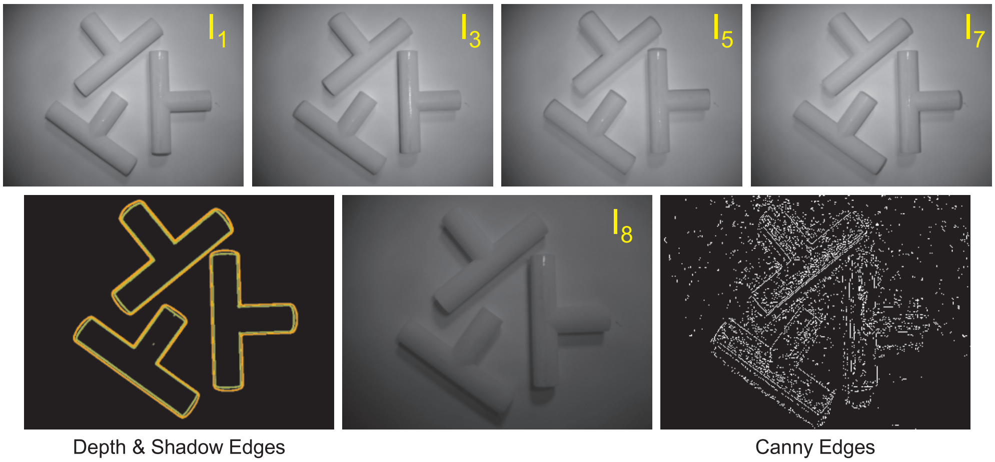


Fig. 15. White objects on a white background. Texture-less and colorless objects are difficult to handle for stereo-based 3D reconstruction algorithms. Useful intensity edges are also not obtained on such scenes. In contrast, our technique works well on such scenes since depth edges are obtained by casting shadows. The extracted depth edges are significantly better compared with Canny edges on I_0.

4.5. Specular Objects

Our approach also works well on specular objects on which 3D scanning fails to give reliable depth estimates. Figure 2 shows an example on two specular pipes. Note the specularities within the object and inter-reflections between the object and the background. This creates problems for intensity edges, so that clear object boundaries cannot be obtained using a traditional 2D camera. However, as shown in Figure 2, reliable depth edges can be easily obtained using our approach. Fig-

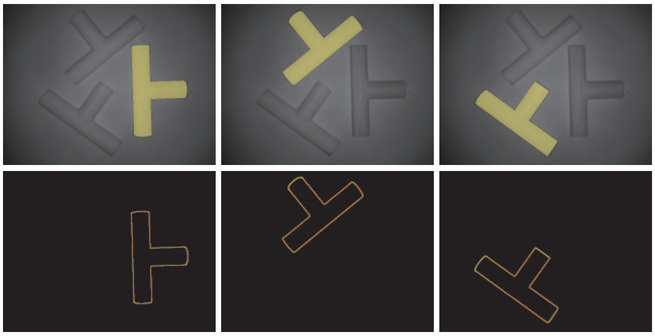


Fig. 16. Pose estimation results on objects shown in Figure 15. The top row shows the overlay of the rendered CAD model on I_0 and the bottom row shows the overlay of the rendered CAD model silhouette on the segmented object silhouette according to the final estimated pose for all three objects.

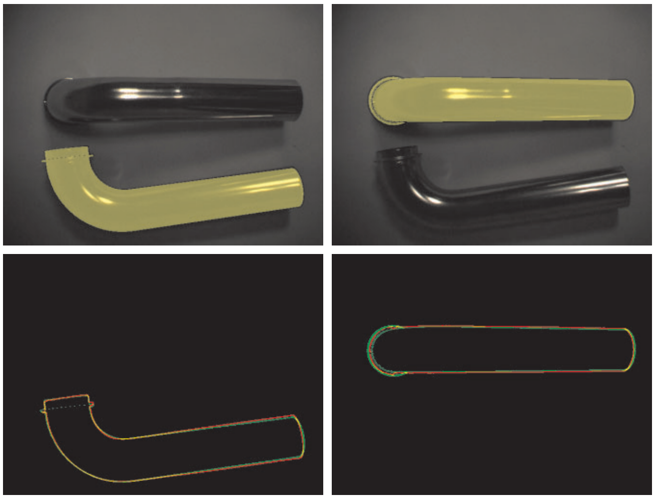


Fig. 17. Pose estimation results on objects shown in Figure 2. The top row shows the overlay of the rendered CAD model on I_0 and the bottom row shows the overlay of the rendered CAD model silhouette on the segmented object silhouette according to the final estimated pose for both objects.

ure 17 shows the pose estimation results on both objects. Extension 2 shows a video of the robot picking two specular pipes from the table and placing them in a pre-determined pose on the side of the table. Extension 5 shows a video of fine pose refinement starting from the initial coarse pose for one of the specular pipes.

4.6. Camera Non-parallel to Background

Our approach can handle general camera orientation that is not necessarily parallel to the background. Figure 18 shows an example, where the camera position is not parallel to the background for capturing the images. The captured images and rendered 3D model silhouettes on one of the images are also shown. The estimated pose allows gripping of the object as shown. Note that the object has concavities and holes. Extension 6 shows the video demonstrating picking for this object.

In summary, we have shown that our technique works well on objects with different shapes and reflectance properties, as well as non-uniform background. In handling a new object, our system only requires the CAD model of the object.

5. Analysis

We now analyze the accuracy of our system. There are several sources of error that could lead to the failure of the picking process. These sources include:

- image noise
- camera calibration and hand–eye calibration errors;
- camera lens distortions (radial distortion, barrel distortion, vignetting, etc.);
- errors in the CAD model of the object;
- missing and spurious depth edges;
- errors in 3D pose estimation.

In the following, we analyze these errors in several ways.

5.1. Calibration Errors

The camera calibration and hand–eye calibration was performed using a standard checkerboard. The checkerboard was placed on the table and images were captured by moving the robot arm to different positions. Figure 19 shows 4 out of 11 checkerboard images used for camera and hand–eye calibration. First, the intrinsic camera calibration parameters including the focal length, principal point and radial distortion parameters and the extrinsic parameters (rotation and translation of camera for each position) were obtained. The average reprojection pixel error of the checkerboard corners on to the captured images was 0.14 and 0.13 pixels in x and y directions, respectively.

Next, hand–eye calibration was performed and the location of the checkerboard in the world coordinate system was determined for each of the 11 views. Since the checkerboard was not moved, its location in the world coordinate system should remain the same for all of the views. However, the estimated location would differ in each view due to image noise and calibration errors. Figure 20 shows the plots of the estimated X, Y, and Z coordinates of one of the checkerboard corners for all the views. The maximum average absolute error in the estimated coordinates is 1.64 mm and the maximum variance of the estimated coordinates is 4.69 mm^2.

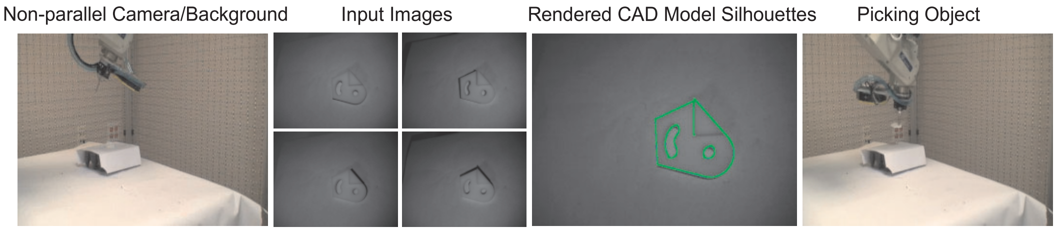

Non-parallel Camera/Background
Input Images
Rendered CAD Model Silhouettes
Picking Object

Fig. 18. Generality of the proposed method. The image plane of the camera does not have to be parallel to the background, and objects can have holes.

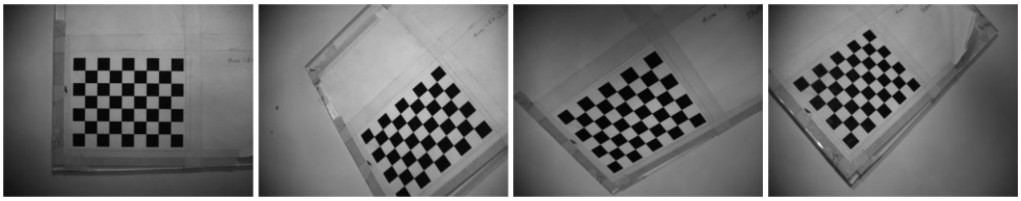


Fig. 19. Four out of 11 checkerboard images used for camera and hand–eye calibration.

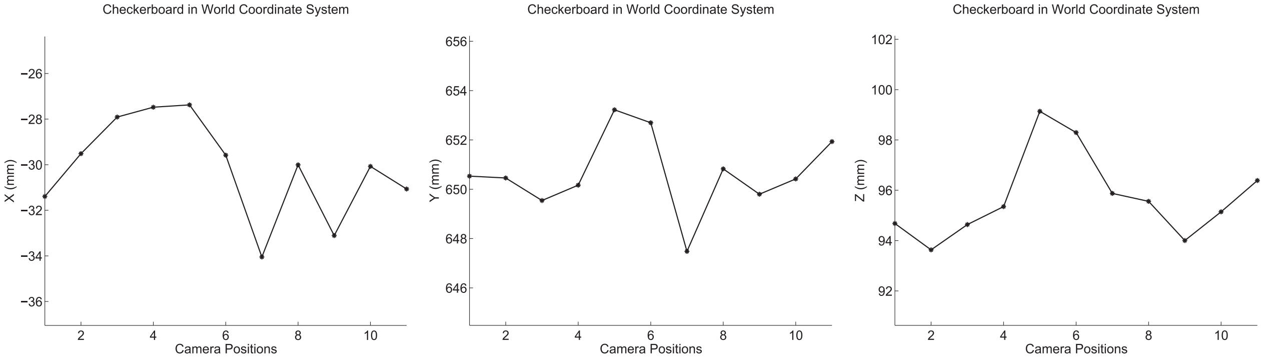


Fig. 20. Location of one of the corners of the checkerboard in the world coordinate system for all 11 views.

5.2. Repeatability Analysis for Pose Estimation

In repeatability analysis, we fix the position of the robot arm (and camera), repeat image capture, segmentation and pose analysis for an object and locate the position of the pick point in the world coordinate system. Ideally, the location of the pick point should remain the same and the variance in the location should be zero. However, note that even if the experiment is run again from the *same* camera position, due to image noise and hysteresis thresholds, the estimated depth edges will not be *exactly* same. Thus, the goal is to measure the variance in the location of pick point due to image noise and pose estimation errors.

Figure 21 shows one of the images of brass hook from a particular camera position. We repeat the pose estimation 20 times for this camera position. The pick point is set to the top of the brass hook. Figure 21 shows the plots of the estimated translation and rotation angles. Note that the estimated pose is very close to the true pose. The maximum variance in the estimated translation is 0.59 mm^2 and in the estimated rotation is 0.04°.

5.3. Pose Estimation Accuracy with Silhouette Size

The accuracy of the pose estimation also depends on the size of the silhouette or the number of pixels in the object silhouette. If the camera is too far from the object, the resolution of the silhouettes will be low and the pose estimation could have ambiguity between Z translation and out-of-plane rotation. To evaluate this, we repeat the pose estimation by placing the camera at different heights (Z-axis), while keeping the object fixed. Figure 22 shows the plots of the estimated location

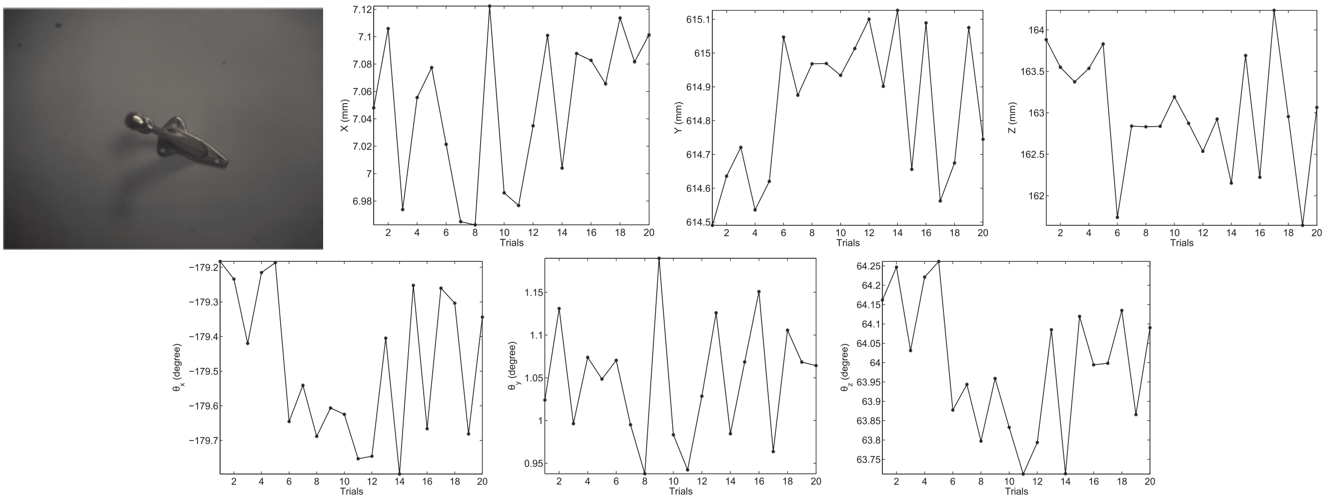


Fig. 21. Repeatability analysis for pose estimation: the pose of the brass hook as shown was estimated 20 times from the same camera position. Shown are the plots for the estimated translation and rotation angles of the pick point. The true location and pick angles was determined manually using teach-box to be $X = 7.9$ mm, $Y = 612.62$ mm, $Z = 164.68$ mm, $\theta_x = -180°$, $\theta_y = 0°$ and $\theta_z = 65.33°$.

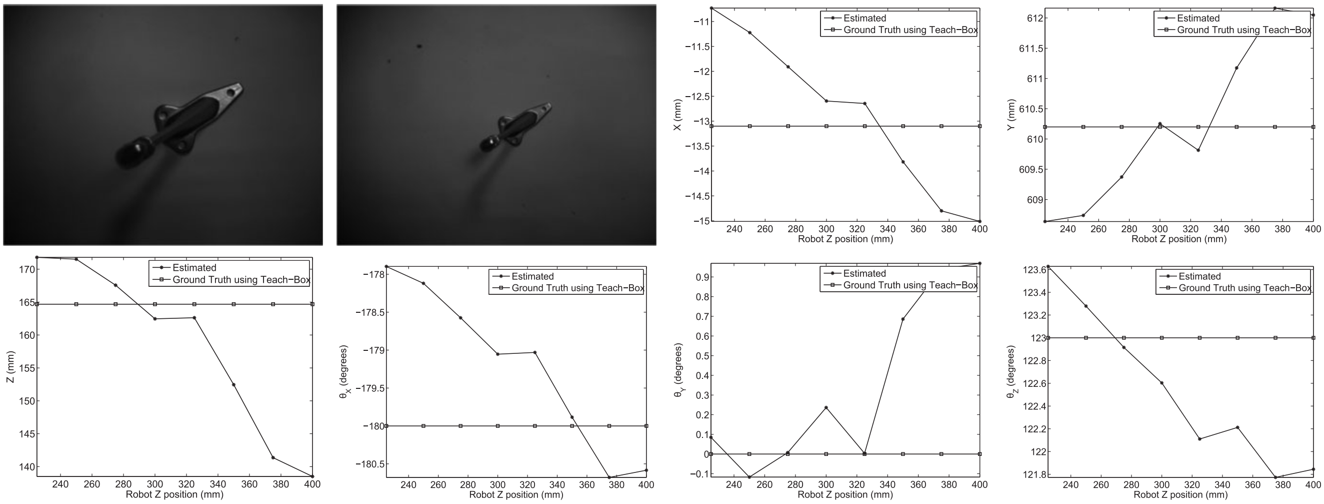


Fig. 22. Pose estimation accuracy with silhouette size: the pose of the object was estimated at nine different camera positions by moving the robot arm from 225 mm to 400 mm (in the Z direction) in steps of 25 mm. The input images corresponding to the first and last camera location shows the difference in the object size. The corresponding object silhouettes will also differ in size. Note that as the camera moves up, the resolution of the silhouettes decreases and the Z estimate and out-of-plane rotation angles (θ_X and θ_Y) of the pick point worsens. The in-plane rotation and X–Y translation estimates are more robust to the silhouette size. The maximum error in Y translation is only 3.5 mm, while the maximum error in Z translation is 26.1 mm.

of the pick point and the pick angles with respect to the changing distance of the camera from the object. Note that as the camera moves up, the size of the object (and its silhouette) decreases. The estimates of in-plane rotation and X–Y translation are more robust to silhouette size, compared with the estimates of Z translation and out-of-plane rotation.

5.4. Pose Estimation Accuracy with Varying Camera Position

Similar to the situation above, we estimated the accuracy and success/failure rate of the system by capturing images from different viewpoints over a sphere. We use 13 azimuth and 13 elevation angles leading to 169 camera viewpoints. Since the object is not moved, the variance of the estimated pose in the world coordinate system should be zero. Out of 169 trials, pose estimation failed in 17 trials leading to large errors in pose estimation. For remaining 152 trials, the variance of the estimate in location were less than 5 mm^2. To calculate the success/failure rate, we obtain ground truth location manually using the teach-box. We declare a trial as a success if the estimated location differs from the ground truth location less than the tolerance provided by the gripper (4 mm). The estimated success rate was 83%. Note that this could be improved by combining information from multiple viewpoints for better pose estimation.

5.5. Ambiguities in Pose Estimation

Since we only use the external object silhouettes, the coarse pose estimation could have ambiguities if the external silhouettes of the object are approximately same in different poses. This is highly dependent on the shape of the object. To handle such ambiguities, one needs to identify poses which can give rise to similar silhouettes and test for all of them in fine pose refinement. In some cases, we can identify a certain "axis", rotation along which could result in similar silhouettes. For the brass hooks shown in Figure 1, such an axis connects the top of the hook with the end of "V" shape bottom. A 180° rotation along this axis could result in similar external silhouettes as shown in Figure 23. Since the CAD model is known, we pre-determine poses which could lead to ambiguity. If the estimated final pose is close to being ambiguous, we record the error between the estimated and projected silhouettes (Equation (9)) and rotate the brass hook by 180° along this axis using the coarse pose estimate and repeat fine pose estimation. The new error between the estimated and projected silhouettes is compared with the previous error. If it has decreased, the new pose estimate is used, else it is discarded. Extension 7 shows a video of fine pose refinement starting from an incorrect initial coarse pose, followed by the rotation of the CAD model by 180° along the pre-defined axis and further fine pose refinement to obtain the correct pose.

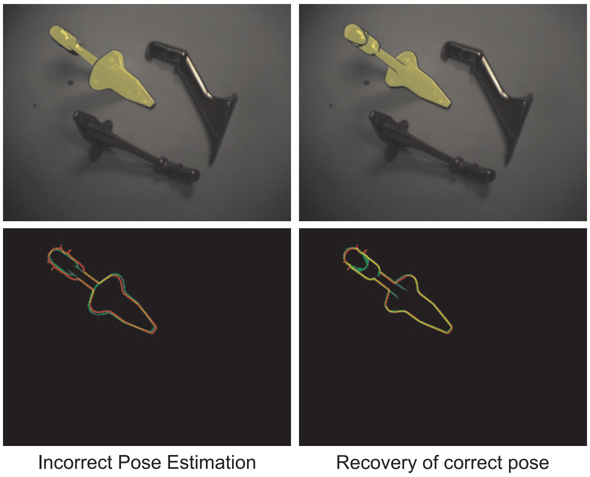


Fig. 23. Ambiguities in pose estimation arise if the external silhouettes are similar for different poses as shown above. The coarse pose estimate followed by fine pose refinement results in an incorrect pose as shown on the left. The object is rotated along the pre-defined axis by 180° and fine pose refinement is performed again to check whether it matches better, resulting in the correct pose.

6. Discussions and Future Work

Several improvements can be made to our system. Our system currently handles singulated objects. *Stacked objects* lead to a more difficult segmentation problem but occlusions and shadow information can be used (Koh et al. 2007) for 3D segmentation of diffuse objects. Stacked specular objects, however, are more challenging as shadows are not cast *on* the specular surfaces properly, leading to a significant amount of missing depth and shadow edges. One possible solution could be to combine segmentation and pose estimation instead of first doing segmentation and then obtaining 3D pose from segmented silhouettes.

Our system runs in open loop and the estimated pose is used to control the robot. *Visual servoing* and pose verification will improve the robustness of the system. Currently we sample the pose space uniformly, but for a given object certain poses are more likely than others. Thus, *adaptive pose sampling* could reduce the size of the database and reduce ambiguities in pose estimation for symmetric objects. In addition, combining information from multiple views of the object could improve pose estimation accuracy.

Our approach could also be combined with 3D sensors such as stereo vision and laser triangulation systems that employ a camera by augmenting the camera with LEDs around it to build a *hybrid sensor*. This would complement our approach which provides excellent depth discontinuities but not absolute depths, with 3D sensors that provide absolute depths but often have difficulty in estimating precise 3D geometry at depth discontinuities.

7. Conclusions

We have presented a vision-based robotic system for model-based 3D pose estimation and picking of objects. Our system utilizes a low-cost novel sensor consisting of a camera surrounded by flashes. Cast shadows are used to estimate depth edges (silhouettes) of objects, which are then used for segmentation and 3D pose estimation using a CAD model of the object. We show that instead of absolute 3D estimates, depth discontinues are sufficient to precisely estimate the 3D pose of the object. Our approach outperforms similar vision systems based on 2D intensity-based features and 3D sensors in terms of robustness, ability to handle objects of different shapes/size and reflectance properties including specular, diffuse and texture-less objects, as demonstrated by several real examples using our sensor mounted on a robot arm. Our system can also handle harsh environmental conditions such as non-uniform backgrounds and complex ambient illumination. Our technique is simple, low-cost, fast and generic enough to accommodate variations in industrial automation applications.

Acknowledgements

The authors would like to thank several members of Mitsubishi Electric Research Labs for technical discussions. We also thank Keisuke Kojima, Jay Thornton and Joseph Katz, along with Haruhisa Okuda and Kazuhiko Sumi, Mitsubishi Electric, Japan for their help and support.

Appendix: Index to Multimedia Extensions

The multimedia extension page is found at http://www.ijrr.org

Table of Multimedia Extensions

Extension	Type	Description
1	Video	Robot picking two brass hooks
2	Video	Robot picking two specular pipes
3	Video	Camera with Flashes
4	Video	Fine pose refinement for brass hook
5	Video	Fine pose refinement for specular pipe
6	Video	Camera viewpoint non-parallel to background and object with holes
7	Video	Correct pose recovery in case of ambiguity

References

Agarwal, A. and Triggs, B. (2004). 3d human pose from silhouettes by relevance vector regression. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 882–888.

Allen, P. K., Timcenko, A., Yoshimi, B. and Michelman, P. (1993). Automated tracking and grasping of a moving object with a robotic hand–eye system. *IEEE Transactions on Robotics and Automation*, **9**(2): 152–165.

Araujo, H., Carceroni, R. and Brown, C. (1998). A fully projective formulation to improve the accuracy of Lowe's pose-estimation algorithm. *Computer Vision and Image Understanding*, **70**(2): 227–238.

Astolfi, A., Hsu, L., Netto, M. and Ortega, R. (2002). Two solutions to the adaptive visual servoing problem. *IEEE Transactions on Robotics and Automation*, **18**(3): 387–392.

Ayache, N. and Lustman, F. (1991). Trinocular stereo vision for robotics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**(1): 73–85.

Besl, P. J. and Jain, R. C. (1985). Three-dimensional object recognition. *ACM Computing Surveys*, **17**(1): 75–145.

Bhatia, A. B. and Wolf, E. (1954). On the circle polynomials of zernike and related orthogonal sets. *Proceedings of the Cambridge Philosophical Society*, **50**: 40–48.

Brady, J., Nandhakumar, N., and Aggarwal, J. (1988). Recent progress in the recognition of objects from range data. *Proceedings of the 9th International Conference on Pattern Recognition*, pp. 85–92.

Brady, J. M. (1981). *Computer vision*. Amsterdam, North-Holland.

Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**(6): 679–698.

Chaumette, F. (1998). *Potential Problems of Stability and Convergence in Image Based and Position Based Visual Servoing*. Berlin, Springer.

Chin, R. and Dyer, C. (1986). Model-based recognition in robot vision. *ACM Computing Surveys*, 18(1): 67–108.

Christoudias, C., Georgescu, B. and Meer, P. (2002). Synergism in low level vision. *Proceedings of the International Conference on Pattern Recognition*, Vol. IV, pp. 150–155.

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(5): 603–619.

Crispell, D., Lanman, D., Sibley, P., Zhao, Y. and Taubin, G. (2006). Beyond silhouettes: Surface reconstruction using multi-flash photography. *Third International Symposium on 3D Data Processing, Visualization, and Transmission*, pp. 405–412.

Dementhon, D. and Davis, L. (1995). Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, **15**: 123–141.

Fabbri, R., Bruno, O. M., Torelli, J. C. and da F. Costa, L. (2007). 2d euclidean distance transform algorithms: A comparative survey. *ACM Computing Surveys*, **40**(1): 2:1–2:44.

Feris, R., Raskar, R., Chen, L., Tan, K.-H. and Turk, M. (2005). Discontinuity preserving stereo with small baseline multi-flash illumination. *Proceedings of the International Conference on Computer Vision*, Vol. 1, pp. 412–419.

Feris, R., Raskar, R., Tan, K. and Turk, M. (2004a). Specular reflection reduction with multi-flash imaging. *Proceedings of SIBGRAPI*, pp. 316–321.

Feris, R., Turk, M., Raskar, R., Tan, K. and Ohashi, G. (2004b). Exploiting depth discontinuities for vision-based fingerspelling recognition. *IEEE Workshop on Real-Time Vision for Human-Computer Interaction*.

Gavrila, D. and Davis, L. (1996). 3-d model based tracking of humans in action:a multiview approach. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 73–80.

Goemans, O., Goldberg, K. and van der Stappen, A. (2006). Blades: a new class of geometric primitives for feeding 3d parts on vibratory tracks. *IEEE International Conference on Robotics and Automation*, pp. 1730–1736.

Gorelick, L., Galun, M., Sharon, E., Basri, R. and Brandt, A. (2006). Shape representation and classification using the poisson equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**: 1991–2005.

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P., Bunke, H., Goldgof, D., Bowyer, K., Eggert, D., Fitzgibbon, A. and Fisher, R. (1996). An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**(7): 673–689.

Horn, B. (1986). *Robot Vision*. Cambridge, MA, MIT Press.

Horn, B. and Ikeuchi, K. (1984). The mechanical manipulation of randomly oriented parts. *Scientific American*, **251**(2): 100–109.

Hutchinson, S., Hager, G. D. and Corke, P. I. (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, **12**(5): 651–670.

Kimchi, R., Behrmann, M., and Olson, C., eds (2003). *Perceptual Organization in Vision: Behavioral and Neural Perspectives*. Lawrence Erlbaum Associates.

Koh, T., Agrawal, A., Raskar, R., Miles, N., Morgan, S. and Hayes-Gill, B. (2007). Detecting and segmenting unoccluded items by actively casting shadows. *Asian Conference on Computer Vision*, pp. 945–955.

Liu, Y. H., Kitagaki, K., Ogasawara, T. and Arimoto, S. (1999). Model-based adaptive hybrid control for manipulators under multiple geometric constraints. *IEEE Transactions on Control Systems Technology*, **7**(1): 97–109.

Lowe, D. (1987). Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, **31**: 355–395.

Lowe, D. (1991). Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**(3): 441–450.

Marr, D. and Nishihara, H. (1978). Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society, London*, **200**: 269–294.

Nayar, S., Watanabe, M. and Noguchi, M. (1995). Real-time focus range sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**: 1186–1198.

Perkins, W. (1977). A model-based vision system for scenes containing multiple parts. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pp. 678–684.

Piepmeier, J. A., McMurray, G. V. and Lipkin, H. (2004). Uncalibrated dynamic visual servoing. *IEEE Transactions on Robotics and Automation*, **20**(1): 143–147.

Pollard, S. B., Mayhew, J. E. W. and Frisby, J. P. (1985). Pmf: A stereo correspondence algorithm using a disparity gradient constraint. *Perception*, **14**: 449–470.

Raskar, R. and Cohen, M. (1999). Image Precision Silhouette Edges. *Proceedings of the Conference on the 1999 Symposium on Interactive 3D Graphics*, Spencer, S. N. (ed.). New York, ACM Press, pp. 135–140.

Raskar, R., Tan, K., Feris, R., Yu, J. and Turk, M. (2004). Nonphotorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. *ACM Transactions on Graphics*, **23**(3): 679–688.

Scharstein, D. and Szeliski, R. (2003). High-accuracy stereo depth maps using structured light. *Proceedings of the Conference on Computer Vision and Pattern Recognition*.

Shapiro, L. and Stockman, G. (2001). *Computer Vision*. Englewood CLiffs, NJ, Prentice-Hall.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(8): 888–905.

Shiu, Y. and Ahmad, S. (1989). Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $ax = xb$. *IEEE Transactions on Robotics and Automation*, **5**(1): 16–27.

Sminchisescu, C. and Telea, A. (2002). Human pose estimation from silhouettes. a consistent approach using distance level sets. *Level Sets—WSCG International Conference on Computer Graphics, Visualization and Computer Vision*, pp. 413–420.

Teh, C. and Chin, R. T. (1988). On image analysis by the method of moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **10**(4): 496–513.

Tuji, S. and Nakamura, A. (1975). Recognition of an object in a stack of industrial parts. *Proceedings of the 4th IJCAI*, pp. 811–818.

Turney, T., Mudge, T. N. and Volz, R. A. (1985). Recognizing partially hidden objects. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 48–54.

Yim, C. and Bovik, A. (1998). Multiresolution 3-D range segmentation using focus cues. *IEEE Transactions on Image Processing*, **7**: 1283–1299.

Zhang, L., Snavely, N., Curless, B. and Seitz, S. M. (2004). Spacetime faces: high resolution capture for modeling and animation. *ACM Transactions on Graphics*, **23**: 548–558.

Zhuang, H. and Roth, Z. S. (1996). *Camera-aided Robot Calibration*. New York, CRC Press.