1—2

Pose Estimation for Insertion of Orbital Replacement Units Against Cluttered Background Using a Non-calibrated Camera

Chiun-Hong Chien
Hernandez Engineering Inc.
cchien@ems.jsc.nasa.gov

Abstract

In this paper, we present work from our on-going project on vision-guided retrieval and insertion of ORUs. Guidance is to be provided through estimated relative poses between an ORU (to be retrieved/inserted), a robotic arm and the related worksite. The major challenges of this work include objects with highly reflective or mirror surfaces moving with cluttered background, along with unreliable or unavailable camera calibration. Moving edge detection and model-based feature matching and tracking are proposed to deal with those challenges. The relationship between image and model features is used to estimate projective matrices, which are then used to predict feature locations in later images. The effectiveness of the proposed techniques is illustrated by encouraging results.

1 Introduction

The International Space Station (ISS), currently advancing through various stages of assembly, has been designed to be operational for up to 30 years. It is expected that maintenance of equipment, such as orbital replaceable units (ORUs), will be a major task, in addition to space scientific research, to be performed on ISS in the foreseeable future. In order to more effectively utilize scarce resources provided by astronauts and to minimize potential dangers to which they are exposed, it is desirable to off-load routine maintenance jobs to intelligent space robots with supervision from astronauts. In this paper, we present work from our on-going project on vision-guided retrieval and insertion of ORUs. ORU insertion and retrieval is currently being carried out by trained operators [1]. However, lack of favorable views, due to constraints in camera placement on ISS, cause operations to be tedious and time consuming. The main objective of our work is to develop a machine vision based method to provide guidance either to operators or directly to manipulators. Guidance is to be provided through estimated relative poses between an ORU (to be retrieved/inserted), a robotic arm and the related worksite. The major challenges of this work include (1) the cluttered background against which insertion and retrieval of ORUs are performed; (2) the complicated ORU structure coupling with highly reflective (or even mirror) surfaces (as shown in Figure 1); (3) all existing ORUs, installed or yet to be installed, have been designed to make them tele-operation friendly, and is impossible or highly unlikely to be modified to be favorable to machine vision, so is the configuration of cameras (and pan-tilt units); and (4) camera parameters (e.g. focal lengths) may be changed through non-calibrated zooming mechanisms and accurate calibration may not be maintained.

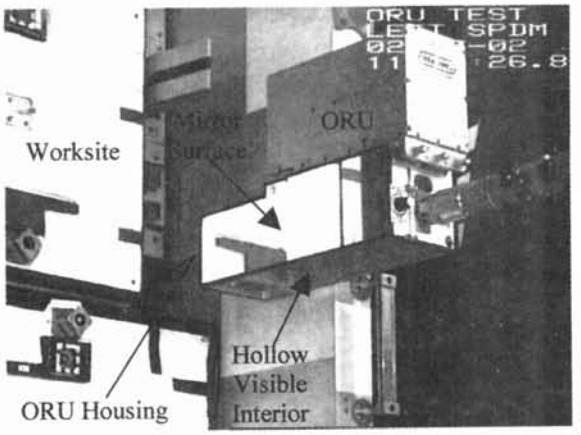


Figure 1: A snap shot of an ORU with mirror surface moving through cluttered background

Pose estimation, along with object recognition, has been extensively studied in the past decades [2,3,4]. However, to our knowledge, very little work has been conducted to deal with all or most of the issues mentioned earlier. In this work, algorithms are proposed to detect and track image features (against highly reflective surfaces and cluttered background), to estimate poses based directly on the relationship between locations of model features in 3-space and corresponding features in an image coordinate system. The algorithms sidestep camera parameters when they are not available or when the parameters changed. While there are usually multiple cameras set up for ORU insertion and removal, all of them may not be in good working condition or in a configuration favorable to integration of information when needed. In this work, a basic framework will be established for estimating pose from a single camera in an uncooperative environment. The framework will be extended in the near future to integrate information from multiple cameras. Also, since initial pose estimates are usually available from other sensors (such as joint angles of manipulators), this paper will focus on updating of pose estimate as new images are acquired.

The remaining of the paper is as follows. The projective relationship between a 3D point and its 2D projection is described in section 2. Algorithms for moving edge detection and model based feature detection and tracking are presented in section 3 and section 4 respectively. Experimental results are given in section 5, followed by concluding remarks in section 6.

$$x_{i=}\frac{p_{00}X_i + p_{01}Y_i + p_{02}Z_i + p_{03}}{p_{20}X_i + p_{21}Y_i + p_{22}Z_i + p_{23}}$$

$$y_{i=}\frac{p_{10}X_i + p_{11}Y_i + p_{12}Z_i + p_{13}}{p_{20}X_i + p_{21}Y_i + p_{22}Z_i + p_{23}}$$

The above equations could be reformulated in term of $\mathbf{p}$, where $\mathbf{p}$ is a column vector.

$$\mathbf{p} = [p_{00} \quad p_{01} \quad p_{02} \quad p_{03} \quad p_{10} \quad p_{11} \quad p_{12} \quad p_{13} \quad p_{20} \quad p_{21} \quad p_{22} \quad p_{23}]^T$$

$$\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -x_iX_i & -x_iY_i & -x_iZ_i & -x_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -y_iX_i & -y_iY_i & -y_iZ_i & -y_i \end{bmatrix}\mathbf{p} = \mathbf{0}$$

2 Computation of Projective Matrix

The mapping between a 3D point and its projection through a camera is dictated by camera internal and external parameters. Internal camera parameters include the focal length, camera center, as well as lens distortion coefficients, while external camera parameters describe the transformation between the (3D) camera coordinate system and the reference coordinate system. The mapping between a 3D point and its projection could be formulated by using a pinhole camera model along with proper modeling of lens distortion (primarily in the radial direction). As lens distortion can be handled separately, it will be left out in the following discussion. For a world point $M(X,Y,Z)$, and corresponding (undistorted) image point $m(x,y)$, their relation in projective spaces [5,6] (expressed in homogeneous representation), is described by

$$m(x,y) = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \approx$$

$$\begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \\ 0_3 & 1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \mathbf{K}\begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \end{bmatrix}M = \mathbf{P}M$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ s \end{bmatrix} = \begin{bmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \\ p_{20} & p_{21} & p_{22} & p_{23} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Where $\mathbf{K}$ is an upper triangular matrix containing internal camera parameters, $\mathbf{R}$ and $\mathbf{t}$ are rotation and translation matrices, and $\mathbf{P} = \mathbf{K}\begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \end{bmatrix}$ is a 3×4 matrix, known as the camera projection matrix. It can be seen that the coordinates of a 3D point $M(X_i,Y_i,Z_i)$ and its 2D image point $m(x_i,y_i)$ are related by

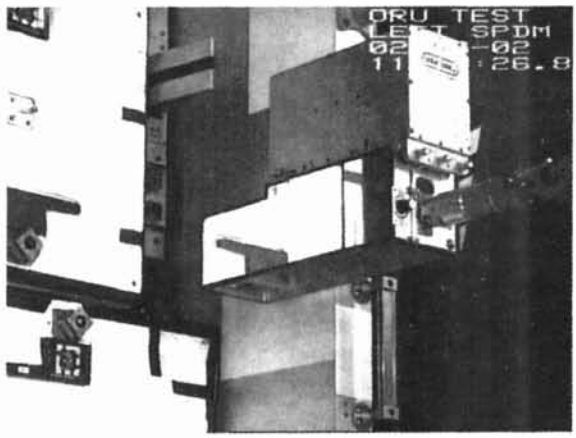
Given six correspondences between 3D points and their associated image projections, a system of twelve linear equations (formed by stacking six of the above equations) is obtained with twelve unknowns p_{ij}. Thus the twelve elements of $\mathbf{P}$ can be estimated, up to a scaling factor, by solving a system of linear equations. The initial estimate can then be refined through an optimization process such as Levenberg-Marquardt optimization. It has also been shown that $\mathbf{K}$, $\mathbf{R}$, $\mathbf{t}$ may be extracted from $\mathbf{P}$ as follows. Let $\mathbf{P} = \begin{bmatrix} \mathbf{P}_{3\times3} & \mathbf{P}_c \end{bmatrix}$ (where $\mathbf{P}_{3\times3}$ is the first 3×3 sub-matrix of $\mathbf{P}$, and $\mathbf{P}_c$ the last column of $\mathbf{P}$), then $\mathbf{P}_{3\times3} = \mathbf{KR}$, and $\mathbf{R}$ is a rotation matrix. The characteristics of $\mathbf{K}$ and $\mathbf{R}$ make it possible to have them (roughly) computed through $\mathbf{QR}$ factorization or using vector geometry, (though more image frames acquired with fixed internal camera parameters are needed in order to obtain better estimates of $\mathbf{K}$ and $\mathbf{R}$). The associated translation matrix $\mathbf{T}$ can also be obtained since $\mathbf{T} = \mathbf{K}^{-1}\mathbf{P}_3$.

3 Moving Edge Detection

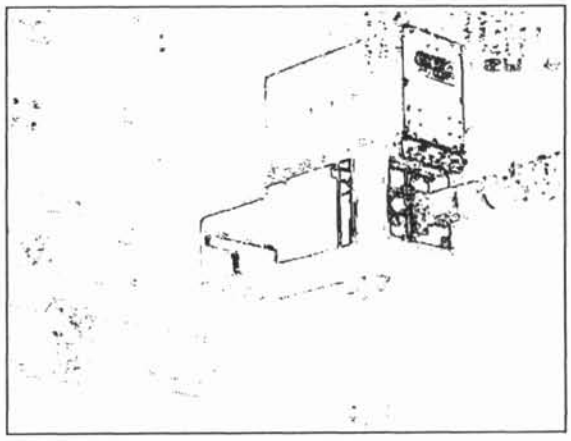
However, $\mathbf{P},\mathbf{K},\mathbf{R},\mathbf{T}$ cannot be obtained without related image and model locations being estimated with sufficient accuracy. One major thrust of this work is to develop a novel technique to accurately detect and track image features of ORUs with highly reflective surfaces moving against a cluttered background. Physical corners (well defined by adjacent edges) are selected as primary model features, and surface points with distinct textures are considered as secondary. Locations of corner features are determined by intersection of two adjacent edges as discussed in the next section. If a sufficient number of corner features are not detected, a search for "textured" points will be conducted using template matching by taking small rotations into consideration.

Extracting objects from cluttered background is by itself a highly challenging task. This is especially true in cases where objects of interest have similar intensity profiles as background, and even more so if object surfaces are highly reflective. The human visual system has a great capability for determining figure ground separation using various cues. One of these cues is motion. For ORU insertion (or removal), relative motion between the worksite and the ORU in question (or the tool to remove the ORU) provides an important cue for separating the ORU (or tool) from the background (including the worksite). While the motion cue alone is not sufficient to facilitate the separation of moving objects from stationary background, it does suggest where edges (or high intensity gradients) of the moving objects have been (except those parallel to the motion of the objects). One method to "extract" the motion cue is through differencing of consecutive image containing the moving objects. However, differencing two consecutive images does not indicate the direction of motion. This ambiguity could be easily resolved by using three consecutive images, instead of two, for extracting motion cue (and "moving" edges). Based on this observation, an algorithm is proposed to detect moving edges. The algorithm is described as follows. Let I_0, I_1, I_2 be three consecutive images. Compute two difference images $DI_{10} = Diff(I_1, I_0)$ and $DI_{12} = Diff(I_1, I_2)$, where $Diff(I_i, I_j)$ is defined such that a pixel in $DI_{ij} = Diff(I_i, I_j)$ is set to the pixel value of the corresponding pixel in source image I_i if the difference between values of its corresponding pixels in the two source images exceeds a preset threshold (e.g. 16 out of 256). Otherwise, the pixel is reset (to 0). Next compute a composite image $CI_1 = And(DI_{10}, DI_{12})$, where $And(DI_i, DI_j)$ is defined such that a pixel in $CI_1 = And(DI_i, DI_j)$ is set to the average pixel values of the corresponding pixels in the two source images, if both pixels are set. Otherwise the pixel is reset. At this stage, the composite image CI_1 indicates where moving edges may be. Finally, the (detectable) moving edges can be located through bit-wise And of the image CI_1 and an edge map EI_1 (computed by applying any proper edge detector to image I_i).

Shown in Figure 2 is the center image (a) of three consecutive images and the image in (b) containing extracted moving edges. It can be seen that the resulting image gives a clear indication where the moving edges are located.



(a)



(b)

Figure 2: (a) An image of an ORU and (b) its edges extracted through moving edge detection

4 Model-based Feature Tracking

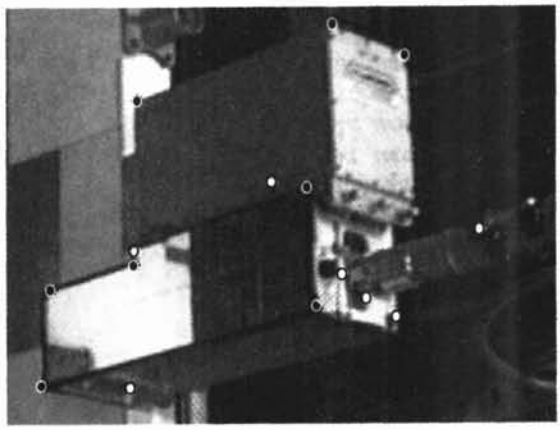


Figure 3: An example illustrating physical corners in black dots and texture features in white dots

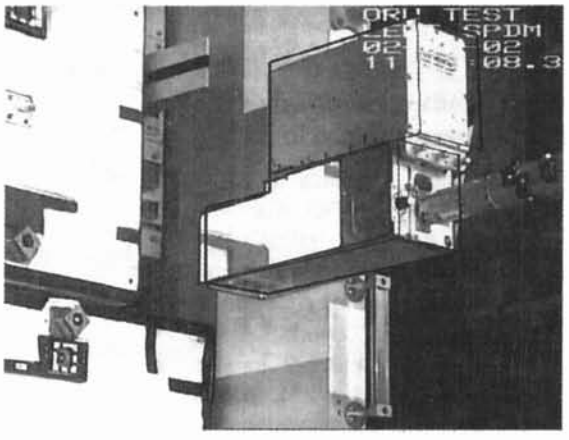
Features used in this work include physical corners and adjacent straight edges (if available), as well as points of high contrast and patches with distinct texture on the surfaces of target objects (i.e. ORUs) as shown in Figure 3. Physical corners and straight

edges have been widely used as features for facilitating geometric computation. However, those on occluding contours of target objects may be sensitive to interference from spurious features detected from background. This problem may be alleviated by incorporating texture features.

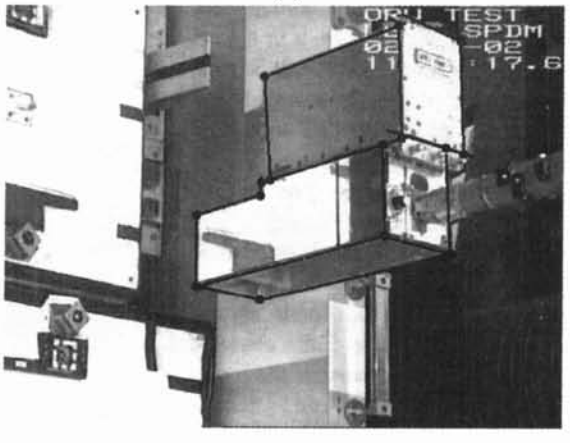
As the main focus of this work is to deal with objects with highly reflective surfaces moving against cluttered background, feature recognition (matching) is a very challenge task. In this work, moving edge detection and model-based feature tracking are proposed to alleviate this difficulty. Moving edge detection is used to filter out most edges associated with the background. Model-based feature tracking provides constraints to pinpoint where features of interest are located. It is carried out as follows. A wire-frame of the object is generated based on the predicted pose. Each edge of the wire-frame is labeled as an occluding edge, visible internal edge, or invisible edge. For each point on each visible edge, a "peak" pixel with the highest intensity gradient is searched for in a small neighborhood along the direction orthogonal to the predicted edge direction. The location of the actual edge is then determined by fitting a line over all "peak" pixels associated with each visible edge. A RANSAC like procedure is needed to exclude outliers from line fitting. After the actual locations of all the visible edges are determined, "corners" can then be located through intersection of adjacent edges. It should be pointed out that moving edge detection could not detect edges parallel to the motion of the observed object. In this case, the edge map of the processed image is used to locate missing edges.

A data structure for organizing model features is set up to facilitate model-based tracking. The root of the data structure contains all parts of the associated model. Each of the parts could be polyhedral, cylindrical, spherical or other types. For brevity, only the data structure associated with a polyhedral part will be described. Each part consists of all the faces of the part. Each face consists of enclosing edges and internal edges, as well as surface orientation and surface type. (No feature would be detected on a face with a mirror surface). Each face may also have texture features with associated "texture" description. Each edge has two end points. All geometric primitive (i.e. face, edge, and point) are encoded with related geometric information. Additional fields are attached to each geometric primitive to maintain geometric and topological information of its projection on the (current) processed image. This information includes visibility, tracking status (e.g. acquisition, tracking, loss-of-tracking), predicted image location and image edge orientation. The information greatly simplifies tracking and feature correspondences (between image features and model features).

Shown in Figure 4 is an example illustrating model-based feature detection and tracking. Figure 4(a) is an image overlaid with external edges (in thick lines) and internal edges (in thin lines) at predicated locations. Locations in the previous image frames are used as predicted locations for now, and will be computed using a Kalman filter for more robust detection in the near future. Figure 4(b) is the same image overlaid with detected edges and corners (intersections of the edges).



(a)



(b)

Figure 4: An image of an ORU overlaid with (a) its predicted wire frame, (b) physical edges detected through model-based edge detection

5 Experimental Results

The proposed algorithms for moving edge detection, and model-based feature detection and tracking have been implemented and applied to stored image sequences of ORU insertions carried out in the Robotic Systems Evaluation Laboratory at the NASA Johnson Space Center. Formulation for computing projective matrices from correspondences between image features and model features has also been implemented. Shown in Figure 5 is an image overlaid with a wire-frame generated with the estimated pose (projective

matrix). It can be seen that the wire frame aligns fairly well with the ORU in the image, though accuracy may be further improved. Results from another two test runs (on two different ORUs) are shown in Figure 6. All results seem to validate our proposed approach.

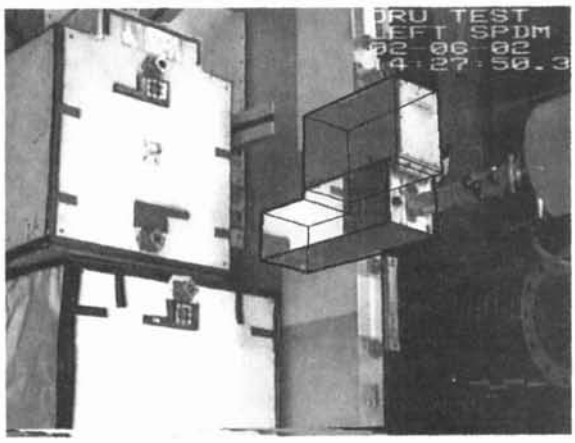
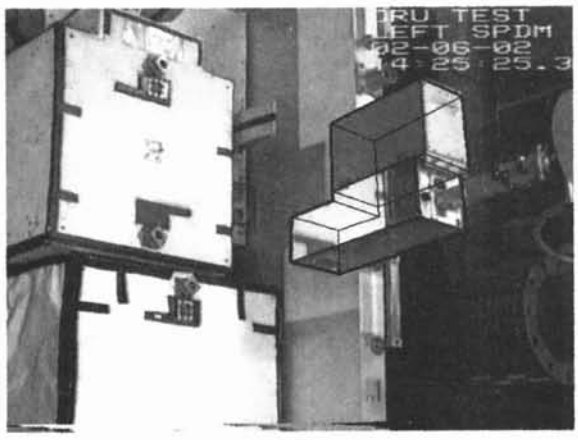
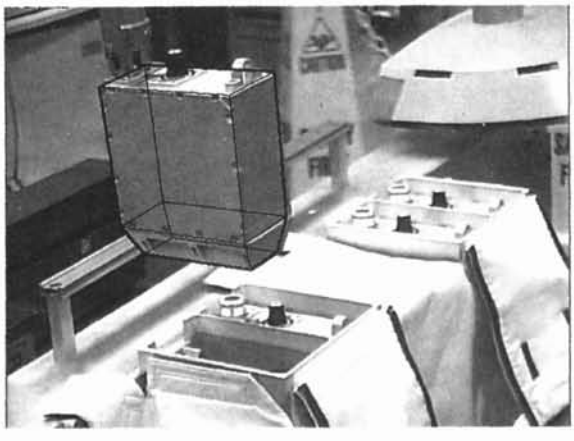


Figure 5: An image overlaid with its wire-frame generated with the estimated projective matrix



(a)



(b)

Figure 6: Images overlaid with wire-frames generated with estimated projective matrices

6 Concluding Remarks

We have presented in this paper results from our on-going work of use of machine vision techniques to support vision-guided insertion and removal of ORUs (with highly reflective or mirror surfaces) against a cluttered background, as typically found on the ISS Algorithms such as moving edge detection and model-based feature matching and tracking have been proposed to deal with uncooperative environment with encouraging results. Kalman filter formulation with image feature locations as measurements and estimated pose of ORUs as states will be developed in the near future. It is expected that the proposed algorithms will be made even more robust when integrated with Kalman filters.

Work reported in this paper has been conducted with an image sequence from a single camera. It will be extended to integration of images from multiple cameras as would be available on the ISS. In other words, the techniques proposed here could be either applied to cases where only a single camera is available, or extended to the case where fusing the information from multiple cameras would make pose estimation more efficient and more robust

Acknowledgement

The author would like to thank Kenneth Baker of the Automation, Robotics and Simulation Division at the NASA Johnson Space Center for his support and fruitful discussion.

References

[1] *Photovoltaic Control Unit (PVCU) Verification Test Plan*, Technical document JSC-47553, Jan. 2002.

[2] A. R. Pope, *Model-Based Object Recognition: A Survey of Recent Research*, TR 94-04, Jan. 19954, Dept. of Computer Sciences, University of British Columbia.

[3] T. Drummond, and R. Cipolla, "Real-Time Visual Tracking of Complex Structures," *IEEE Transaction On PAMI, Vol. 24, No. 7,* July 2002, pp. 932-946.

[4] C. H. Chien, "A Computer Vision System for Extravehicular Activity Helper & Retriever," *Intern. Journal on Applied Intelligence, vol. 5, no. 3,* 1995, pp. 251-268.

[5] Coxeter, H. S. M., Projective Geometry, second edition, University of Toronto, 1974.

[6] Faugeras O. D., Three-dimensional Computer Vision, second edition, The MIT Press, Cambridge Massachusetts, 1996