2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Pose Estimation in Heavy Clutter using a Multi-Flash Camera

Ming-Yu Liu[†], Oncel Tuzel[‡], Ashok Veeraraghavan[‡], Rama Chellappa[†], Amit Agrawal[‡], Haruhisa Okuda[§]
[†]University of Maryland [‡]Mitsubishi Electric Research Labs [§]Mitsubishi Electric Corporation
{mingyliu,rama}@umiacs.umd.edu {oncel,veerarag,agrawal}@merl.com {Okuda.Haruhisa}@ct.MitsubishiElectric.co.jp

***Abstract*— We propose a novel solution to object detection, localization and pose estimation with applications in robot vision. The proposed method is especially applicable when the objects of interest may not be richly textured and are immersed in heavy clutter. We show that a multi-flash camera (MFC) provides accurate separation of depth edges and texture edges in such scenes. Then, we reformulate the problem, as one of finding matches between the depth edges obtained in one or more MFC images to the rendered depth edges that are computed offline using 3D CAD model of the objects. In order to facilitate accurate matching of these binary depth edge maps, we introduce a novel cost function that respects both the position and the local orientation of each edge pixel. This cost function is significantly superior to traditional Chamfer cost and leads to accurate matching even in heavily cluttered scenes where traditional methods are unreliable. We present a sub-linear time algorithm to compute the cost function using techniques from 3D distance transforms and integral images. Finally, we also propose a multi-view based pose-refinement algorithm to improve the estimated pose. We implemented the algorithm on an industrial robot arm and obtained location and angular estimation accuracy of the order of 1 mm and 2° respectively for a variety of parts with minimal texture.**

I. Introduction

Machine vision systems that robustly identify and locate objects have a multitude of applications ranging from intelligent homes to automatic manufacturing. Although the population of robots has been growing fast, most robots work in restricted and constrained environments. For example, parts in assembly lines have to be placed with a fixed orientation and position for robots to grasp and manipulate them. It remains a great challenge to handle non-structured scenes containing multiple objects. Here, we propose a machine vision system that provides (a) robust feature extraction (b) accurate pose estimation and (c) fast operation with minimal lag. A multi-flash camera [13](MFC) provides accurate separation of depth edges and texture edges in heavily cluttered environments. We then reformulate the problem of object detection and localization, as one of finding matches between the depth edges obtained in one or more MFC images to the rendered (from CAD model) depth edges. We introduce a novel cost function that respects both the position and the local orientation of each edge pixel. This cost function is superior to Chamfer matching cost and leads to accurate matching even in cluttered scenes. Finally, we also perform continuous optimization to refine the estimated pose.

A. Related Work

Model-based pose estimation using 3D model to 2D image correspondences can be found in [11][12][5]. Unfortunately the 3D-2D point correspondences are hard to obtain for industrial parts due to their textureless surfaces. The situation is particularly severe when multiple identical objects are placed together and overlap each other.

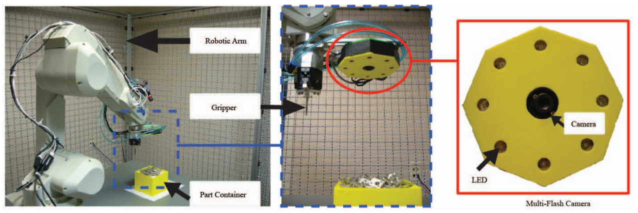

Fig. 1: The robotic grasping platform. *The MFC is mounted on the robotic arm and the objects are placed in a container.*

Object contours provide rich information about object identities and their poses. One fundamental problem in exploiting this cue is in matching an exemplar contour to a newly observed contour, especially when the observed contour is partially occluded or located in a cluttered background. Besides Chamfer matching [3], various contour matching algorithms have been proposed in [7][2][10]. Although these methods perform better than Chamfer matching for well segmented objects[3], their performance in cluttered scenes tends to be inferior (see [17]). Edge orientation can be used in order to improve the Chamfer matching in cluttered background as shown in [9][15][16].

Active illumination patterns can greatly assist vision algorithms by extracting robust features in such challenging environments. Examples of such techniques include depth estimation by projecting a structured illumination pattern [14]. In this paper, we exploit the MFC which uses active illumination in order to extract depth edges thereby removing the clutter from texture and shadow edges. In environments where texture is indeed a rich source of information, we could still use our algorithm with canny edges instead of depth edges obtained via MFC. Since the focus of this paper is on industrial environments where texture is not an important visual information source, we restrict our attention to depth edges from MFC for the rest of this paper.

Contributions: The technical contributions are

- We show that a MFC provides accurate extraction of depth edges even in heavy clutter.
- We introduce a novel cost function that respects both the position and the local orientation of each edge pixel and show its superiority over traditional edge based costs. We also develop a sub-linear time algorithm to compute the cost function using techniques from 3D distance

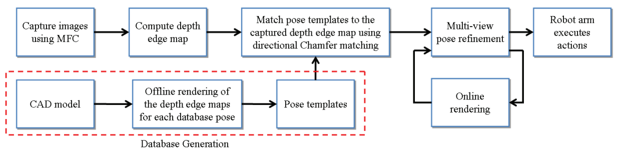
Capture images using MFC
Compute depth edge map
Match pose templates to the captured depth edge map using directional Chamfer matching
Multi-view pose refinement
Robot arm executes actions
CAD model
Offline rendering of the depth edge maps for each database pose
Pose templates
Online rendering
Database Generation

Fig. 2: System Overview.

transforms and integral images.
- We propose a multi-view pose-refinement algorithm in order to improve the pose estimate.
- We implemented the algorithm on an industrial robot and obtained location and angular estimation accuracies of about 1 mm and 2° respectively.

II. System Overview

Physical Layout and Calibration: System design is shown in Figure 1 where MFC is mounted to the robot arm. The MFC is calibrated (both internal and external) using a calibration grid. Also, we performed the gripper-camera(hand-eye) calibration so that the gripper can interact and grasp objects. This gives us a complete calibrated framework.

Algorithmic Layout: The algorithmic layout of the system is described below and shown in Figure 2.

1) Offline Rendering of Database: For each object (a) render the 3D CAD model for every hypothesized pose (b) compute the depth-edges and (c) fit lines to the rendered MFC depth edges.
2) Feature Extraction: Capture 8 images using the 8 different flashes of the MFC. These images are then used to compute the depth edges in the scene.
3) Object Detection and Pose Estimation: Compute the matching cost between each rendered pose of each object in the database and the edge map obtained using the MFC to perform coarse pose estimation. Refine this estimate further using a multi-view based pose-refinement algorithm.
4) Grasping and Assembly: Once the 3D poses of the objects are accurately estimated, grasp the objects in order using the gripper at the end of the robot arm and perform the required assembly task.

III. Multi-Flash Camera

An MFC is an active illumination based camera that contains 8 point light sources (LED's) arranged in a circle around the camera as shown in Figure 3. The MFC exploits the change in the shadows caused by changes in illumination source positions in order to extract depth edges even for challenging objects such as textureless objects and mildly specular objects. Consider the MFC shown in Figure 3. As the different LED's around the camera flash, the position of the shadows cast by the object change. While points on the object (such as P1) do not change intensity as the flash moves around, points that are in the shadow of one of the flashes (such as P2) change intensity significantly. This change in intensity of the shadow pixels can be used to detect and extract view dependent depth edges[13].

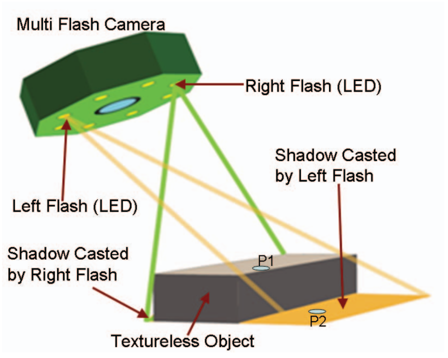

Fig. 3: Principle of operation of MFC. *As the different LED's around the camera flash, the shadows cast by the object change. While points on the object (such as P1) do not change intensity as the flashes move around, points in the shadow of one of the flashes (such as P2) change intensity. This is used to detect depth edges.*

Let us denote the image captured (after ambient subtraction) during the flashing time of the i_{th} LED as I_i. The maximum intensity value at each pixel location among these ambient subtracted images are found and used to construct the maximum illumination image $I_{max}(x,y) = \max_i I_i(x,y)$. Next, we compute the ratio images of the ambient subtracted images to the maximum illumination image, $RI_i = \frac{I_i}{I_{max}}$. Ideally, the ratio value in the shadow region (eg., point P2) should be zero since the contribution of the illumination from the ambient source has been removed. In contrast the ratio values in other regions (eg., point P1) should be close to one since these regions are illuminated by all the flashes. Notice that the point of transition between the pixels in the shadow region and those not in the shadow region is always a depth edge. For each ratio image, we apply a Sobel filter designed to detect this transition from shadow to non-shadow (0 to 1).

IV. Object Detection and Localization

In this section, we present our algorithm for detection and localization of objects in cluttered scenes using depth edges acquired through MFC. Without loss of generality, we describe the method as applied to a single object. However, this assumption is only for ease of presentation, while in reality the algorithm locates and estimates the pose of multiple objects simultaneously.

A. Database Generation

Given the CAD model of the object, we generate a database of depth edge templates by simulating MFC in software. In the simulation, a virtual camera having the internal parameters of the real MFC is placed at the origin and its optical axis is aligned with the z-axis of the world coordinate system. Eight virtual flashes are evenly placed on a circle on x-y plane having center at origin and radius equal to the actual baseline between the camera and flashes. The CAD model of the object is then placed on the z-axis at a distance t_z from the virtual camera. The virtual flashes are switched on one at a time and eight renderings of the object (including cast shadows) are acquired. The depth edges in the scene are detected using the procedure described in Section III.

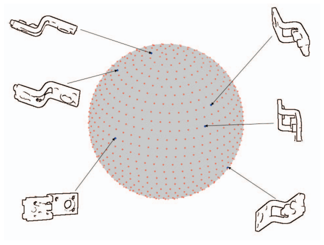

Fig. 4: Database generation. *We uniformly sample the rotation angles (θ_x and θ_y) on the 2-sphere. The template database is generated by rendering the CAD model of the object with respect to the sampled rotations.*

An arbitrary 3D rotation can be decomposed into a sequence of three elemental rotations about three orthogonal axes. We align the first of these axes to be the camera optical axis and call the rotation about this axis as the in-plane rotation (θ_z). The other two axes are on a plane perpendicular to the camera optical axes and the rotation about these two axes are called the out-of-plane rotation(θ_x and θ_y). Note that an in-plane rotation results in an in-plane rotation of the observed images, whereas the effect of an out-of-plane rotation depends on the 3D structure of the object. Due to this distinction, we only include out-of-plane rotations of the object into the database. We sample k out-of-plane rotations (θ_x and θ_y) uniformly on the 2-sphere, S^2, as shown in Figure 4 and generate the depth edge templates for each of these rotations.

B. Directional Chamfer Matching

During matching, we search for the database template together with its optimal 2D Euclidean transformation, $\mathbf{s} \in SE(2)$, which aligns the depth edges of the template to the query image edges. A 2D Euclidean transformation is represented with three parameters, $\mathbf{s} = (\theta_z, \bar{t}_x, \bar{t}_y)$, where $\bar{t}_x$ and $\bar{t}_y$ are the image plane translations along the x and y axes respectively and θ_z is the in-plane rotation angle. Its action on an image pixel is given as

$$\mathbf{W}(\mathbf{x};\mathbf{s}) = \begin{pmatrix} \cos(\theta_z) & -\sin(\theta_z) \\ \sin(\theta_z) & \cos(\theta_z) \end{pmatrix} \mathbf{x} + \begin{pmatrix} \bar{t}_x \\ \bar{t}_y \end{pmatrix}. \quad (1)$$

Chamfer matching [1] is a popular technique to find the best alignment between two edge maps. Let $U = \{\mathbf{u}_i\}$ and $V = \{\mathbf{v}_j\}$ be the sets of template and query image edge maps respectively. The Chamfer distance between U and V is given by the average of distances between each point $\mathbf{u}_i \in U$ and its nearest edge in V

$$d_{CM}(U,V) = \frac{1}{n} \sum_{\mathbf{u}_i \in U} \min_{\mathbf{v}_j \in V} |\mathbf{u}_i - \mathbf{v}_j|. \quad (2)$$

where $n = |U|$. The best alignment parameter $\hat{\mathbf{s}} \in SE(2)$ between the two edge maps is then given by

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s} \in SE(2)} d_{CM}(\mathbf{W}(U;\mathbf{s}), V). \quad (3)$$

Chamfer matching becomes less reliable in the presence of background clutter. To improve robustness, several variants of Chamfer matching were introduced by incorporating edge

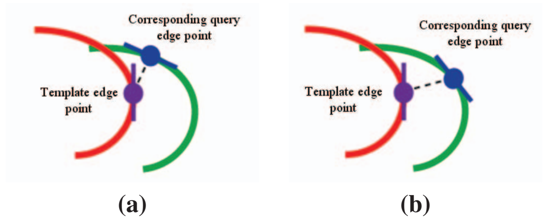
Corresponding query edge point
Template edge point
(a)
Corresponding query edge point
Template edge point
(b)

Fig. 5: Matching costs per edge point. *(a) Shotten et al. [16]; (b) Directional Chamfer matching. DCM jointly minimizes location and orientation errors whereas in [16] the location error is augmented with the orientation error of the nearest edge point.*

orientation information into the matching cost. In [9], the template and query image edges are quantized into discrete orientation channels and individual matching scores across channels are summed. Although this method alleviates the problem of cluttered scenes, the cost function is very sensitive to the number of orientation channels and becomes discontinuous in channel boundaries. In [16], the Chamfer distance is augmented with an additional cost for orientation mismatch which is given by the average difference in orientations between template edges and their nearest edge points in the query image.

Instead of an explicit formulation of orientation mismatch, we generalize the Chamfer distance to points in $\mathbb{R}^3$ for matching directional edge pixels. Each edge point $\mathbf{x}$ is augmented with a direction term $\phi(\mathbf{x})$ and the directional Chamfer matching (DCM) score is given by

$$d_{DCM}(U,V) = \frac{1}{n} \sum_{\mathbf{u}_i \in U} \min_{\mathbf{v}_j \in V} |\mathbf{u}_i - \mathbf{v}_j| + \lambda |\phi(\mathbf{u}_i) - \phi(\mathbf{v}_j)| \quad (4)$$

where λ is a weighting factor between location and orientation terms. Note that the directions $\phi(\mathbf{x})$ are computed modulo π, and the orientation error gives the minimum circular difference between the two directions

$$min\{|\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)|, ||\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)| - \pi|\}. \quad (5)$$

In Figure 5, we present a comparison of the proposed cost function with [16]. In [16], the nearest point in V is initially located for a given template point $\mathbf{u}$ and the cost function is augmented with the difference between their orientations, whereas the cost function proposed in the paper jointly minimizes the sum of location and orientation error terms. It can be easily verified that the proposed matching cost is a piecewise smooth function of both translation $\bar{t}_x$, $\bar{t}_y$ and rotation θ_z of the template edges. Therefore the matching algorithm is more robust against clutter, missing edges and small misalignments. To our knowledge, the best computational complexity for the existing Chamfer matching algorithms is linear in the number of template edge points, even without the directional term. In the following section, we present a sub-linear time algorithm for exact computation of the 3D Chamfer matching score (4).

C. Search Optimization

The search problem given in (3), requires optimization over three parameters of planar Euclidean transformation

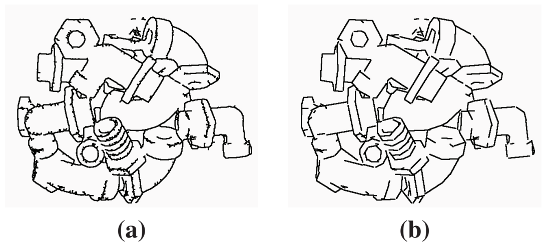

Fig. 6: Linear representation. *(a) Edge image. The image contains 11542 edge points. (b) Linear representation of the edge image. The image contains 300 line segments.*

$(\theta_z,\ \bar{t}_x,\ \bar{t}_y)$ for each of the k templates stored in the database. Given a 640x480 query image and a database of $k = 300$ edge templates, the brute-force search requires more than 10^{10} evaluations of the cost function in (4). We perform search optimization in two stages: (1) We present a sublinear time algorithm for computing the matching score; (2) We reduce the three-dimensional search problem to one-dimensional queries by aligning the major lines of template images to the query image.

1) Linear Representation: The edge map of a scene does not follow an unstructured binary pattern. Instead, the object contours comply with certain continuity constraints which can be retained by concatenating line segments of various lengths, orientations and translations. Here, we represent an edge image with a collection of m-line segments. Compared with a set of points which has cardinality n, its linear representation is more concise. It requires only $O(m)$ memory to store an edge map where $m << n$.

We use a variant of RANSAC [8] algorithm to compute the linear representation of an edge map. The algorithm initially hypothesizes a variety of lines by selecting a small subset of points and their directions. The support of a line is given by the set of points which satisfy the line equation within a small residual and form a continuous structure. The line segment with the largest support is retained and the procedure is iterated with the reduced set until the support becomes smaller than a few points.

The algorithm only retains points with certain structure and support, therefore the noise is filtered. In addition, the directions recovered using the line fitting procedure are more precise compared with local operators such as image gradients. An example of linear representation is given in Figure 6 where a set of 11542 points are modeled with 300 line segments.

2) Three-Dimensional Distance Transform: The matching score given in (4) requires finding the minimum cost match over location and orientation terms for each template edge point. Therefore the computational complexity of the brute-force algorithm is quadratic in the number of template and query image edge points. Here we present a three-dimensional distance transform representation ($DT3$) to compute the matching cost in linear time.

This representation is a three dimensional image tensor where the first two dimensions are the locations on the image plane and the third dimension is the quantized edge orientation. The orientations are quantized into q discrete channels $\hat{\Phi} = \{\hat{\phi}_i\}$ evenly in $[0 \quad \pi)$ range. Each element of the tensor encodes the minimum distance to an edge point in joint location and orientation space:

$$DT3_V(\mathbf{x}, \phi(\mathbf{x})) = \min_{\mathbf{v}_j \in V} |\mathbf{x} - \mathbf{v}_j| + \lambda |\hat{\phi}(\mathbf{x}) - \hat{\phi}(\mathbf{v}_j)|. \quad (6)$$

where $\hat{\phi}(\mathbf{x})$ is the nearest quantization level in orientation space to $\phi(\mathbf{x})$ in $\hat{\Phi}$.

The $DT3$ tensor can be computed in $O(q)$ passes over the image. Equation (6) can be rewritten as

$$DT3_V(\mathbf{x}, \phi(\mathbf{x})) = \min_{\hat{\phi}_i \in \hat{\Phi}} \left(DT_{V\{\hat{\phi}_i\}} + \lambda |\hat{\phi}(\mathbf{x}) - \hat{\phi}_i| \right) \quad (7)$$

where $DT_{V\{\hat{\phi}_i\}}$ is the two-dimensional distance transform of the edge points in V having orientation $\hat{\phi}_i$. Initially we compute q two dimensional distance transforms $DT_{V\{\hat{\phi}_i\}}$ using the standard algorithm [6]. Subsequently, the $DT3_V$ tensor (7) is computed by solving a second dynamic program over the orientation costs, for each location separately.

Using the 3D distance transform representation $DT3_V$ the directional Chamfer matching score of any template U can be computed in linear time via

$$d_{DCM}(U,V) = \frac{1}{n} \sum_{\mathbf{u}_i \in U} DT3_V(\mathbf{u}_i, \hat{\phi}(\mathbf{u}_i)). \quad (8)$$

3) Distance Transform Integral: Let $L_U = \{l_{[\mathbf{s}_j,\mathbf{e}_j]}\}_{j=1...m}$ be the linear representation of template edge points U where $\mathbf{s}_j$ and $\mathbf{e}_j$ are the start and end locations of the j-th line respectively. For ease of notation, we sometimes refer to a line with only its index $l_j = l_{[\mathbf{s}_j,\mathbf{e}_j]}$. We assume that the line segments only have directions among the q discrete channels $\hat{\Phi}$, which is enforced while computing the linear representation. All the points on a line segment are associated with the same orientation which is the direction of the line $\hat{\phi}(l_j)$. Hence the directional Chamfer matching score (11) can be rearranged as

$$d_{DCM}(U,V) = \frac{1}{n} \sum_{l_j \in L_U} \sum_{\mathbf{u}_i \in l_j} DT3_V(\mathbf{u}_i, \hat{\phi}(l_j)). \quad (9)$$

In this formulation, the i-th orientation channel of the $DT3_V$ tensor, $DT3_V(\mathbf{x}, \hat{\phi}_i)$, is only evaluated for summing over the points of line segments having direction $\hat{\phi}_i$.

Integral images are intermediate image representations used for fast calculation of region sums [18]. Here we present a tensor of integral distance transform representation ($IDT3_V$) to evaluate the summation of costs over any line segment in $O(1)$ operations. For each orientation channel i, we compute the one-directional integration along $\hat{\phi}_i$ (Figure 7).

Let $\mathbf{x}_0$ be the intersection of an image boundary with the line passing through $\mathbf{x}$ and having direction $\hat{\phi}_i$. Each entry

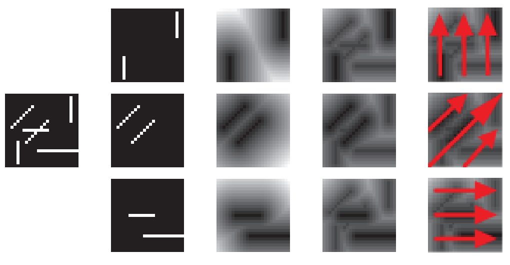

Fig. 7: Computation of the integral distance transform tensor. *(a) The input edge map. (b) Edges are quantized into discrete orientation channels. (c) Two-dimensional distance transform of each orientation channel. (d) The three dimensional distance transform $DT3$ is updated based on the orientation cost. (e) $DT3$ tensor is integrated along the discrete edge orientations and integral distance transform tensor, $IDT3$, is computed.*

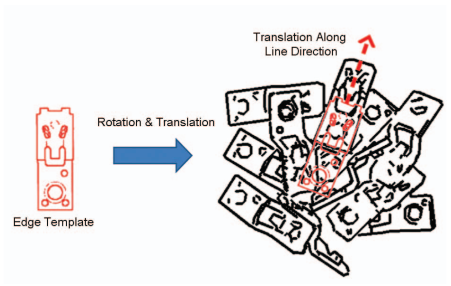

Fig. 8: One-dimensional search. *A template is rotated and translated such that one template line segment is aligned with one line segment in the query image. The template is translated along the query line segment and the directional Chamfer matching cost is evaluated.*

of $IDT3_V$ tensor is given by

$$IDT3_V(\mathbf{x}, \hat{\phi}_i) = \sum_{\mathbf{x}_j \in l_{[\mathbf{x}_0, \mathbf{x}]}} DT3_V(\mathbf{x}_j, \hat{\phi}_i). \quad (10)$$

The $IDT3_V$ tensor can be in one pass over the $DT3_V$ tensor. Using this representation, the directional Chamfer matching score of any template U can be computed in $O(m)$ operations via

$$d_{DCM}(U,V) = \frac{1}{n} \sum_{l_{[\mathbf{s}_j,\mathbf{e}_j]} \in L_U} [IDT3_V(\mathbf{e}_j, \hat{\phi}(l_{[\mathbf{s}_j,\mathbf{e}_j]})) - IDT3_V(\mathbf{s}_j, \hat{\phi}(l_{[\mathbf{s}_j,\mathbf{e}_j]}))]. \quad (11)$$

Since $m << n$, the computational complexity of the matching is **sub-linear** in the number of template points n.

The $O(m)$ complexity is an upper bound on the number of computations. For pose estimation we would like to retain only the best hypothesis. We order the template lines with respect to their support and start the summation from the lines with the largest support. The hypothesis is eliminated during the summation if the cost is larger than the current best hypothesis. The supports of the line segments show exponential decay, therefore for majority of the hypotheses only a few arithmetic operations are performed.

4) One-dimensional Search: The search for the optimal pose over three parameters of planar Euclidean transformation is computationally intensive to be practical for real-time applications. The linear representation provides an efficient method to reduce the size of the search space. The observation is that, the template and query image line segments are near perfectly aligned with the true estimate of the template pose. In addition, the major lines of the template and query images are very reliably detected during the line-fitting since the procedure favors segments with larger support.

We order template and query line segments based on their support and retain only a few major lines to guide the search. The template is initially rotated and translated such that the template line segment is aligned with the direction of the query image line segment and its end point matches the start point of the query segment as illustrated in the Figure 8. The template is then translated along the query segment direction and the cost function is evaluated only at locations where there is an overlap between the two segments. This procedure reduces the three-dimensional search to one-dimensional searches along only a few directions. The search time is invariant to the size of the image and is only a function of number of template and query image lines, and their lengths. In almost all our experiments, search along 5 template and 50 query image lines produces near identical results to brute-force search and the optimal pose can be found under a few seconds for a database size of $k = 300$ templates.

V. Pose Refinement

The minimum cost template together with its in-plane transformation parameters $(\theta_z, \bar{t}_x, \bar{t}_y)$ provide a coarse estimate of the 3D object pose. Let θ_x, θ_y be the out-of-plane rotation angles and t_z be the distance from the camera which are used to render the template. We back project the in-plane translation parameters to 3D using the camera calibration matrix $\mathbf{K}$, and the initial 3D pose of the object, $\mathbf{p}^0$, is given by the three Euler angles $(\theta_x, \theta_y, \theta_z)$ and a 3D translation vector $(t_x, t_y, t_z)^T$. The 3D pose $\mathbf{p}$ can also be written in matrix form

$$\mathbf{M_p} = \begin{pmatrix} \mathbf{R_p} & \mathbf{t_p} \\ \mathbf{0} & 1 \end{pmatrix} \in SE(3) \quad (12)$$

where $\mathbf{R_p}$ is the 3x3 orthogonal matrix computed by a sequence of three rotations around $x-y-z$ axes $\mathbf{R}_{\theta_z}\mathbf{R}_{\theta_y}\mathbf{R}_{\theta_x}$, and $\mathbf{t_p}$ is the three-dimensional translation vector.

The precision of the initial pose estimation is limited by the discrete set of out-of-plane rotations included into the database. In this section, we present a continuous optimization method to refine the pose estimation. The proposed method is a combination of iterative closest point (ICP) [19] and Gauss-Newton [4, pp.520] optimization algorithms.

Three-dimensional pose estimation from a single view is an ill-posed problem. To minimize the uncertainty in pose estimation, we use a two-view approach, where the robot arm is moved to a second location and the scene imaged with MFC. The edge points detected in the two views are given by the sets $V^{(j)} = \{\mathbf{v}_i^{(j)}\}$, $j \in \{1,2\}$.

Let $\mathbf{M}^{(j)} \in SE(3)$, $j \in \{1,2\}$ be the 3D rigid motion matrices determining the location of the two cameras in world coordinate system and $\mathbf{P} = (\mathbf{K} \;\; \mathbf{0})$ be the 3x4 projection matrix. The optimization algorithm minimizes the sum of squared projection error between the detected edge points $\mathbf{v}_i^{(j)}$, and the corresponding 3D points $\tilde{\mathbf{u}}_i^{(j)}$ in the 3D CAD model, simultaneously in both views

$$\varepsilon(\mathbf{p}) = \sum_j \sum_{\tilde{\mathbf{u}}_i^{(j)}} \|\mathbf{P}\mathbf{M}^{(j)}\mathbf{M}_\mathbf{p}\mathbf{M}^{(j)^{-1}}\tilde{\mathbf{u}}_i^{(j)} - \mathbf{v}_i^{(j)}\|^2. \quad (13)$$

Note that, the projections of 3D points $\tilde{\mathbf{u}}_i^{(j)}$ are expressed in homogeneous coordinates and in this formulation we assume that they have been converted to 2D coordinates.

We find the 3D-2D point correspondences via closest point assignment on the image plane. We simulate the 2 camera setup and render the 3D CAD model with respect to the current pose estimate $\mathbf{p}$. Let $U^{(j)} = \{\mathbf{u}_i^{(j)}\}$, $j \in \{1,2\}$ be the sets of detected edge points in two synthetic views and $\tilde{U}^{(j)} = \{\tilde{\mathbf{u}}_i^{(j)}\}$ be the corresponding point sets in the 3D CAD model. For each point in $U^{(j)}$ we search for the nearest point in $V^{(j)}$ with respect to the directional matching score

$$arg \min_{\mathbf{v}_j \in V} \|\mathbf{u}_i - \mathbf{v}_j| + \lambda|\phi(\mathbf{u}_i) - \phi(\mathbf{v}_j)\|. \quad (14)$$

and establish point correspondences $(\tilde{\mathbf{u}}_i^{(j)}, \mathbf{v}_i^{(j)})$.

The non-linear least squares error function given in (13) is minimized using the Gauss-Newton algorithm. Starting with the initial pose estimate $\mathbf{p}^0$, we improve the estimation via the iterations $\mathbf{p}^{t+1} = \mathbf{p}^t + \Delta\mathbf{p}$. The update vector $\Delta\mathbf{p}$ is given by the solution of the normal equations $(\mathbf{J}_\varepsilon^T \mathbf{J}_\varepsilon)\Delta\mathbf{p} = \mathbf{J}_\varepsilon^T \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is the N dimensional vector of each of the summed error terms in (13), and $\mathbf{J}_\varepsilon$ is the Nx6 Jacobian matrix of $\boldsymbol{\varepsilon}$ with respect to $\mathbf{p}$, evaluated at $\mathbf{p}^t$. The correspondence and minimization problems are solved repeatedly until convergence. The initial pose estimate given by the matching algorithm is usually close to the true solution, therefore in general $5-10$ iterations suffice for convergence.

VI. Experiment

We performed an extensive evaluation of the proposed system using synthetic and real experiments with a robot arm.

A. Experiments on Synthetic Examples

We quantitatively evaluated the accuracy of the proposed system to detect and localize objects in highly cluttered scenes on an extensive synthetic dataset. The synthetic dataset consisted of 6 objects of varying complexity in their 3D shape placed randomly one over the other to generate several cluttered scenes (Figure 9). There were a total of 606 such synthetic images that were rendered. The average occlusion in the dataset was 15% while the maximum occlusion was 25%. Moreover, in order to simulate missing depth edges and imperfections in the MFC, a small fraction (about $10-15\%$) of the depth edges was also removed.

Detection and Localiztion: We compared the performance of the proposed cost function to those of the Chamfer cost function and the oriented Chamfer cost function [15]. The detection failure rate is shown in the Table I. The proposed matching cost formulation reduced the detection failure rate of Chamfer matching from 0.24 to 0.05. It also reduced the error rate of competing state of art matching formulation of oriented Chamfer matching by half. We also observe that objects with discriminative shapes were easier for detection such as the diamond toy and the circuit breaker. On the contrary, the T-Nut object, which has a simple shape, is relatively hard for detection since false edges from clutter and other objects frequently confuse the optimization algorithm. Several examples of successful detections for various objects in challenging scenarios are shown in Figure 9.

Det. Rate	Circuit Breaker	Diamond Toy	Ellipse Toy	T-Nut	Knob	Wheel	Avg.
Ours	0.03	0.01	0.05	0.11	0.04	0.08	0.05
[15]	0.05	0.05	0.14	0.17	0.04	0.17	0.10
Chamfer	0.11	0.22	0.26	0.34	0.26	0.22	0.24

TABLE I: Detection Failure Rate comparison in highly cluttered scene with multiple objects.

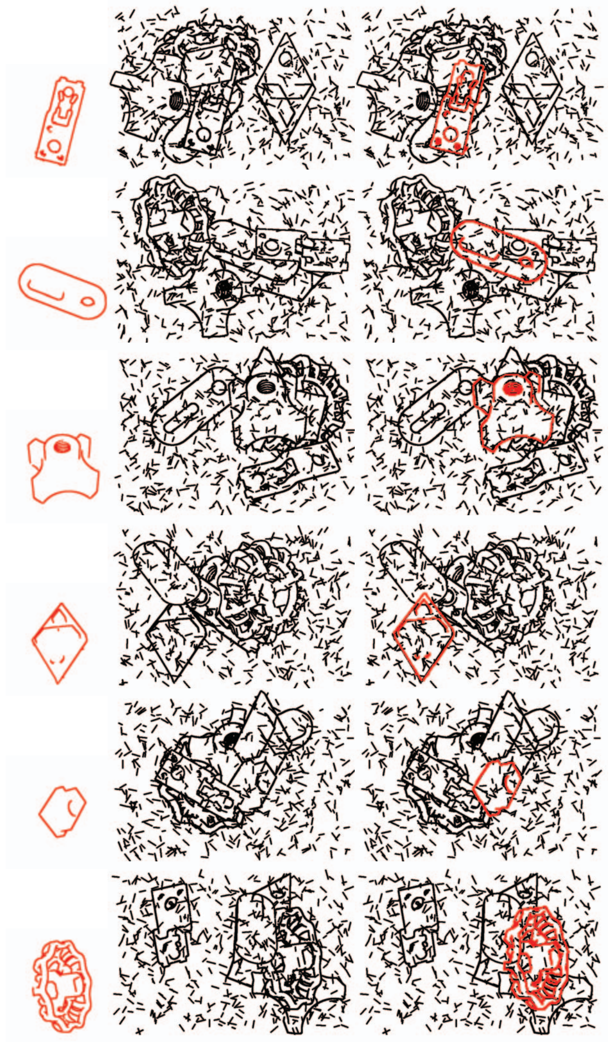

Fig. 9: Examples of successful localization on the synthetic dataset. *First column represents the six different target objects.*

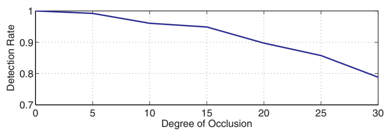

Fig. 10: Detection rate versus percentage of occlusion.

Avg. abs err.	t_X mm	t_Y mm	t_Z mm	θ_X degree	θ_Y degree	θ_Z degree
1 View	0.127	0.165	1.156	0.674	0.999	0.349
2 View	0.094	0.096	0.400	0.601	0.529	0.238

TABLE II: Comparison of the average absolute pose estimation error between the one-view and two-view approaches.

Robustness to Occlusions: We further quantitatively evaluated the robustness of the proposed cost function against varying degrees of occlusion from no occlusion to an average occlusion of 0.30. The results are presented in Figure 10. We achieved greater than 99% detection upto 5% occlusion and about 85% detection rate when one-fourth of the object is occluded.

Two View Pose Estimation We evaluated the pose estimation algorithm using a similar synthetic setting. We randomly rendered a set of poses of various objects. After a coarse pose estimate was computed, both the refinement scheme using one view and that of using two views were applied independently to further refine the estimate. The final estimates were compared to the ground truth and the results show that the two-view approach outperformed the one-view approach (Table II). $1mm$ corresponded to about 6.56 pixels on the image plane indicating that the two-view estimate was sub pixel accurate.

B. Experiments on Real Examples:

Object Detection and Pose Estimation in Clutter: To quantitatively evaluate the performance, we created several real test examples. Seven different objects were laid one on top of another in a cluttered manner as shown in Figure 11. We then performed object detection, localization and pose estimation on these real examples. The experiment was repeated for several hundred trials and found that the detection rate was about 95%. Shown in Figure 11 are some typical example trials of the real experiment. In each image, we render the silhouettes of the top detector outputs for three different objects. We render the estimated depth edges of the objects over the actual captured images in order to show the accuracy of the algorithm. Notice, that some of the parts have no texture while others are mildly specular. Traditional image edge (eg., canny) based methods usually fail in such challenging scenarios. The use of MFC allows us to robustly extract depth edges even in such challenging scenarios. Also notice, that since the MFC feature is texture independent, the method works robustly for parts that have artificial texture painted on them. This indicates that the method can work in the presence of oil, grime and dirt (which are all common in industrial environments) all of which add artificial texture to the surface of objects.

Statistical Evaluation: In order to statistically evaluate the accuracy of the proposed system, we need a method for

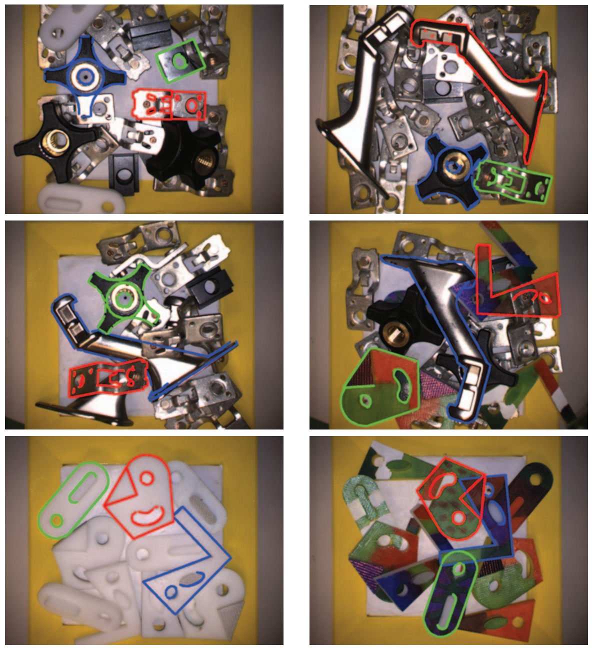

Fig. 11: Performance on real examples. *The system detected and accurately estimated the pose for specular as well as textureless objects. Shown in these images are the top detector outputs for three different objects overlaid on top of the original images.*

independently obtaining the $3D$ ground truth pose of the object. Since there was no simple way of obtaining this (especially for cluttered scenes), instead we devised a method to evaluate the consistency of pose estimate irrespective of the viewpoint of the camera. We placed an object in the scene. At each time, the robot arm was commanded to perform a different rotation and translation. From each of these views, the MFC image was obtained and the pose estimate was obtained in the camera coordinate frame. Since the object is static, the estimated pose of the object in the world coordinate system should be identical irrespective of viewpoint of the MFC. For each view, the estimated pose of the object was transformed to the world coordinate frame using knowledge of the position and orientation of the robot arm. The same experiment was repeated for 7 different objects with 25 trials for each object with the object in a different pose in each trial. During each of these independent trials the robot arm was moved to 40 different viewpoints in order to evaluate the consistency of the pose estimate. The histogram of the deviations (from the median) of the pose estimate is shown in Figure 12. The results demonstrate that the algorithm results in consistent estimates with standard deviation of less than $0.5mm$ in the in-plane directions (X,Y) and about 3 degrees in all three orientation estimates. The standard deviation in estimate of Z (Z-axis coincides with the optical axis of camera) is slightly larger (about 1.2mm).

C. Real Experiments on Robot Arm

We evaluated the algorithm on an industrial robot arm as shown in Figure 1. Several parts were thrown together in a bin to create cluttered scenes just as shown in Figure

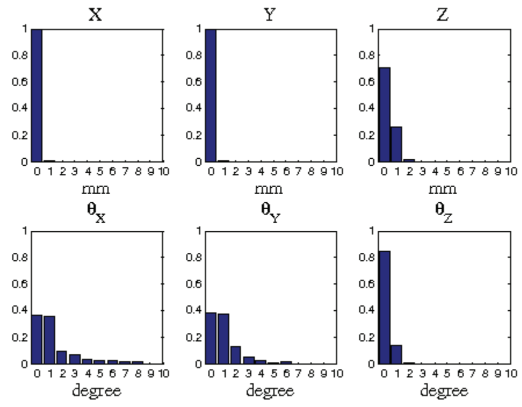

Fig. 12: Histograms of deviations from the pose estimates to their medians in the real examples.

11. The gripper was made up of three vertical steel pins each of 1*mm* diameter. The gripper of the robot arm was designed to pick each of the objects by first inserting the three vertical pins through a hole in the objects. Then the gripper opens the 3 pins thereby exerting horizontal force on the inside edges of the hole. The hole in the objects was about 5mm to 8mm in diameter. Therefore, in order to successfully insert the gripper inside the hole (before lifting the object) the error in pose estimate should be less than about 1.5mm. When the pose estimate error is greater than about 1.5mm the pins are not inserted into the hole and this results in a failure to pick up the object. The proposed system is able to successfully guide the robot arm in the grasping task. We achieved a 0.95 grasping rate over several hundred trials. Among the 5% grasp failures a significant majority (about 3%) were acually successful pose estimations. But in these cases, the hole for grasping the target object was occluded by other objects. Hence, while attempting to pick these objects the gripper hit other objects and failed. We refer the readers to the supplemental material for videos of the robot arm accomplishing this task. In all these cases, the object detection, localization and pose estimation took an average of 6 seconds for an object in extremely cluttered environments (on an Intel 2.66Ghz CPU with 3GB memory). In environments with minimal clutter the algorithm runs almost twice as fast since there are much fewer edges on an average. On average the matching task requires 2 seconds and pose refinement requires 2 seconds where the rest of the computation time is shared among depth edge extraction, thinning and line fitting.

VII.Conclusion and Future Work

We presented a system for object detection, localization and pose estimation using MFC. We formulated the problem as one of finding matches between the depth edges obtained in one or more MFC images to the rendered depth edges that are computed offline using 3D CAD models of the objects. We introduced a novel cost function that is significantly superior to traditional chamfer cost and developed multi-view based pose estimation and refinement algorithms. We implemented the system on a robot arm and achieved location and angular estimation accuracies of the order of 1 mm and 2° respectively.

References

[1] H. Barrow, J. Tenenbaum, R. Bolles, and H. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *International Joint Conference of Artificial Intelligence*, pages 659–663, 1977.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.

[3] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865, November 1988.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

[5] D. DeMenthon and L. S. Davis. Model-based object pose in 25 lines of code. In *European Conference on Computer Vision*, pages 335–343. Springer-Verlag, 1992.

[6] P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions, 2004.

[7] P. F. Felzenszwalb and J. D. Schwartz. Hierarchical matching of deformable shapes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

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

[9] D. M. Gavrila. Multi-feature hierarchical template matching using distance transforms. In *International Conference on Pattern Recognition*, pages 439–444, 1998.

[10] H. Ling and D. W. Jacobs. Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):286–299, February 2007.

[11] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, March 1987.

[12] D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[13] R. Raskar, K.-H. Tan, R. Feris, J. Yu, and M. Turk. Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. *ACM Transactions on Graphics*, 23(3), 2004.

[14] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:195–202, June 2003.

[15] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *IEEE International Conference on Computer Vision*, volume 1, pages 503–510. IEEE Computer Society, October 2005.

[16] J. Shotton, A. Blake, and R. Cipolla. Multi-scale categorical object recognition using contour fragments. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 30(7):1270–1281, July 2008.

[17] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 127–133, June 2003.

[18] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[19] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.