8th IEEE International Conference on Automation Science and Engineering
August 20-24, 2012, Seoul, Korea

3D Visual Perception System for Bin Picking in Automotive Sub-Assembly Automation

Sukhan Lee, Jaewoong Kim, Moonju Lee, Kyeongdae Yoo, Leandro G. Barajas and Roland Menassa

Abstract—**Industries are being swept up by the tide of market innovations originated from pervasive application of Robotics and Automation (RA). Given the critical role of RA in industry, it has become relevant for RA to have human-like decision making capabilities. Such enablers intrinsically require the use of flexible and robust 3D perception and control systems. In the process of automating complex automotive sub-assemblies, 3D vision-based recognition as well as grasping of complex objects is required not only for detection and categorization but also for pose estimation and robotic pick-and-place operations. In this paper, we propose a novel 3D visual perception system for sub-assembly automation based on a structured light 3D vision system. We use a novel geometric surface primitive patch segmentation approach based on Hough transforms to obtain accurate surface normal estimations from 3D point clouds for the identification of patch primitives. The most relevant primitives for our application include planar, cylindrical, conic and spherical surface patches. We extract primitive surface patches from automotive CAD models in DXF format. The models are then decomposed to simple entities such as planar polygons, vertexes and lines. Our resulting models based on the 3D point clouds are composed only by simple planes and cylinders. Our system takes advantage of the available CAD data for both object recognition and for pose estimation. Our experimental results demonstrate that we can achieve, in only a few seconds, a highly accurate pose and object class estimation.**

I. Introduction

THEstate-of-the-art commercial robotic systems used in manufacturing today rely mostly on highly calibrated, task-specific end-effectors and part fixtures to implement assembly processes. In general, setting up such a system is time and effort consuming, often requiring extensive offline calibration and robot programming. Recently, 2D and 3D machine vision technologies have been brought into assembly processes at a reasonable cost and shown to be reliable to locate specific parts or to confirm assembly tasks, adding flexibility to conventional robotic manufacturing systems. However, the resulting assembly systems cannot yet overcome challenging issues as commissioning, calibration, and support. This is due not only to the fact that they tend to be limited in applications to one set of tasks under specific conditions but also to the unsolved and undiminished issues of non-value-added engineering, installation, and conversion expense and lead-time from the requirements for fixturing, tooling and material handling. Therefore, a next generation of robotic assembly systems that can significantly reduce the reliance on fixturing, tooling, and other pre-set, sunk cost, manufacturing infrastructures are very much anticipated. Ultimately, smart or intelligent assembly systems are expected to emerge with little or no more infrastructure requirements than those for manual assembly. This new generation of smart assembly systems may take advantage of the recent advancement on robotic vision and cognition technologies to build powerful vision/cognition components for flexible robotic assembly. Such powerful vision/cognition components will play an integral part for the successful introduction of the next generation of smart assembly systems, together with the advancement of highly dexterous robots and end effectors as well as of the capability of robots to assist or collaborate with people in shared workspaces and tasks. The advancement of vision/cognition components would also enable the automated planning and execution of assembly tasks via assistance and collaboration with human operators to perform tasks not yet appropriate or cost effective for automation. This capability will enable a smart assembly process which will be highly flexible and self-contained and that can be used for various and changing assembly tasks as required by the variable production mix and business needs of today.

Automobile manufacturing General Assembly (GA) lines are yet to be widespread automated via robotics given that they require highly dexterous, complex, and labor-intensive operations. Especially, sub-assembly lines that support the main lines with the preparation of parts, sub-assemblies and tools are the least structured and thus rely heavily on manual operations. Relevant examples related to this paper of sub-assembly operations include palletizing alternators and batteries in a rack in the right pose and sequence for use in primary assembly lines.

In robotic automation, bin-picking systems[3] have been used to perform pick-and-place tasks for randomly oriented parts from bins or boxes[4, 5].Bin-picking systems as well as laser-vision-based bin-picking systems were proposed by Ban[6]. To address the problem of pose estimation in robotic assembly automation systems, many groups have researched both pre-computed2D views-based approaches [7], and

This work was mainly funded by the General Motors Global R&D Center, Manufacturing Systems Research Laboratory. This work is also supported by the Intelligent Robotics Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea. This work is also supported by the Brain Korea 21 Program of Ministry of Education & Human Resources Development and the Science and Technology Program of Gyeonggi Province as well as in part by the Sungkyunkwan University.

S. Lee, J. Kim, M. Lee, and K. Yoo are with the School of Information and Communication Engineering of Sungkyunkwan University, Suwon, Korea, (phone: 82-31-299-7150; fax: 82-31-290-6479; e-mail: lsh@ece.skku.ac.kr)

L. G. Barajas and R. Menassa are with the Advanced Robotics Group, Manufacturing Systems Research Laboratory at the General Motors Global R&D Center, Warren, MI 48090, USA, (e-mail: L.G.Barajas@ieee.org)

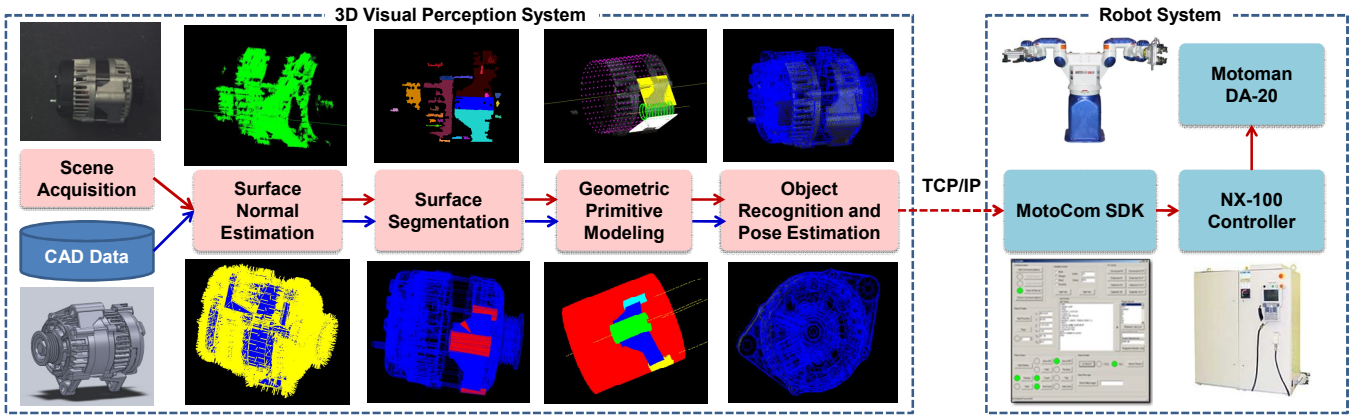

Fig. 1. Overview of proposed system for 3D vision based automotive sub-assembly automation.

feature-based approaches [8,9]. Other similar view-based approaches tried to reduce the search space[10] and assumed that the object is placed on a conveyor belt or a table and that the distance from camera to the object was known. Other groups have developed pose estimation using also CAD data. Recently, Choi [11] used 3D CAD data with 2D key points for object recognition and camera pose estimation for a fixed object with a moving camera. Harris [12] used RAPiD to implement pose estimation comparing CAD model edges with grey level images. Finally, Ulrich's group [13] evaluated an algorithm for textureless 3D object recognition by using CAD data.

This paper is organized as follows. Section II presents an overview of the proposed 3D Visual Perception System; Section III explains the proposed surface normal estimation and surface segmentation approach. In Section IV, the surface segmentation and object modeling are explained. Sections V describes the overall object recognition and pose estimation needed for grasping. Section VI presents experimental results. Finally, Section VII summarizes results, offers conclusions & proposes future work.

II. Overview

The proposed automotive sub-assembly automation consists of two main systems: i) The 3D visual perception system, and ii) the robotic system. The 3D visual perception system uses a 3D structured light camera, and consists of five modules:

- Surface normal estimation from 3D point clouds by structured light camera;
- Surface segmentation from surface normal of 3D point clouds by predefined geometric primitives;
- Surface segmentation from surface normal of CAD data by predefined geometric primitives;
- Planar and cylindrical object modeling using joint parametric surface segmentation;
- Object recognition and pose estimation using a matching strategy to enable robust object grasping.

The robotic system consists of: i)a DA-20 MOTOMAN robot, and ii)a Yaskawa NX-100 robot controller.

There are also some previous research results using MOTOMAN robots. In [1] a probabilistic motion planning technique is illustrated via a simulated MOTOMAN SV3X Manipulator, and in [2] several control and communication methods with a MOTOMAN UP6 are analyzed. The NX-100 controller can communicate to PC or laptop via the MotoCom SDK which serves as a C++/C# API for the user to directly implement the robot control. We illustrate the entire system overview in Fig 1.

III. Surface Primitive Patch Segmentation By Transformation in Three Space

We define three spaces for efficient surface segmentation of surface primitives such as plane, cylinder and cone from raw data: 1) IJK Space, 2) Parameter Space and 3) 3D Space.

- IJK Space: The Gaussian Image is called IJK Space in which we can find plane candidates by extreme density of points in the space.
- Parameter Space: Parameter Space consists of theta and phi which are axis direction of cylinder/cone and d which is Euclidean distance from origin of IJK Space to a normal vector. In the IJK Space cylindrical/conical group of normal vectors are spread as circular. So we transform all points in IJK Space into Parameter Space and then we can get cylinder/cone candidates by extreme density of points in the space.
- 3D Space: High density area in IJK Space sometimes has a lot of separate planes. So, all 3D point clouds in that area should be separated in the 3D Space additionally. Same reason high density area in Parameter Space also should be separated to different location and radius cylinders/cones in 3D Space.

Using input 3D point clouds from above three spaces and surface normal vectors, we can obtain dominant surface primitives are segmented effectively for object recognition and pose estimation. We explain in more detail about algorithms and mathematic formula in the next.

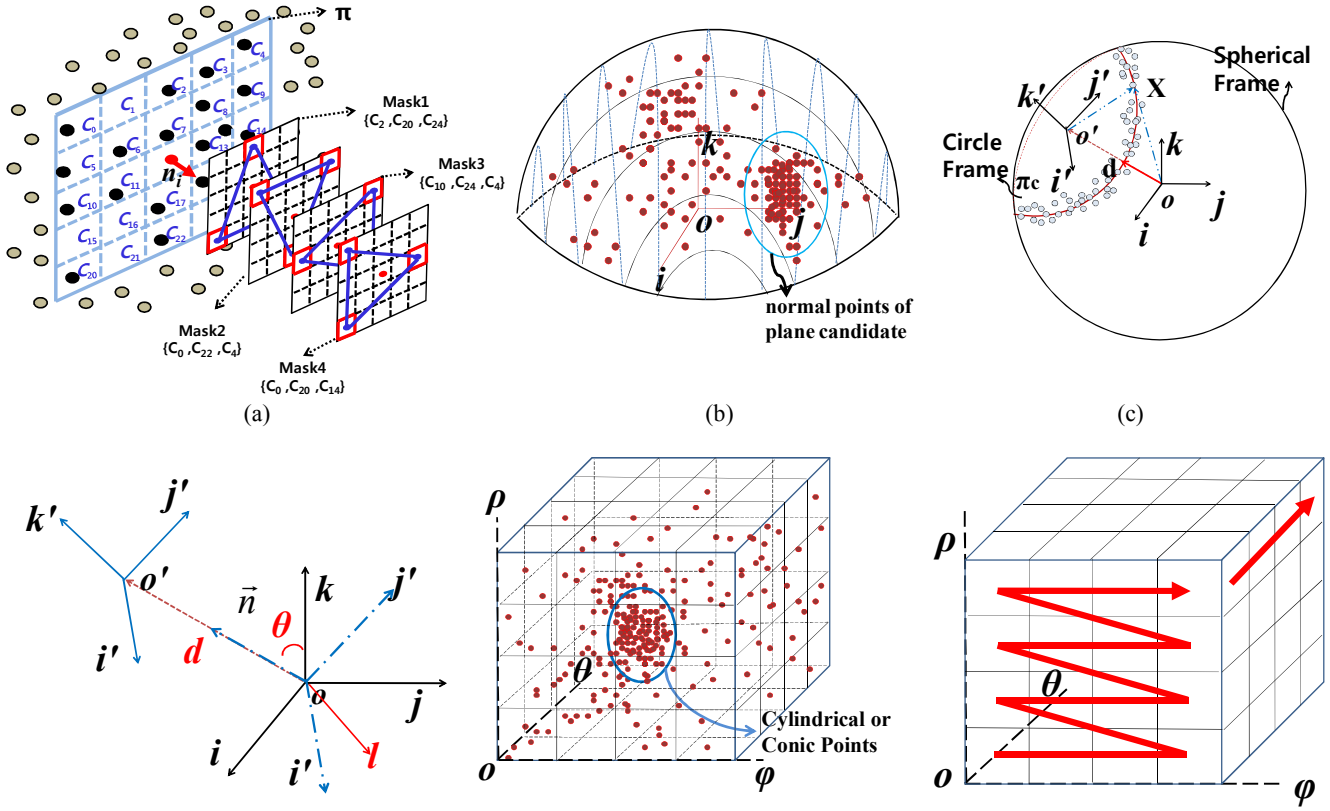

Fig. 2. Surface normal estimation and surface segmentation method. (a) Tangent plane mask; all points in real 3D space transformed to x-y plane, mask1-2 and mark3-4 are designed symmetric pairs for normal estimation. (b) All normal values are moved to IJK space and represented on a half of sphere in IJK space. If there is a dense area with many points markedly, points in the dense area can compose a plane. (c) Spherical Frame is shown all (i, j, k) points on a half surface of sphere. If points are gathered in a circular shape, those can be a point of cylindrical or conic object. Moreover, those points depict a circle in a new circle Frame, (i′, j′, k′) called parametric space. (d) We get a transformation from sphere frame to circle frame. First, Making a rotation around *l* vector with θ, we calculate a transformation matrix (rotation) and a translation, d. (e) Candidate points of cylinder or conic objects in parametric space are gathered around one point. (f) The parameter space changes into histogram space for finding out meaningful surfaces. The most dominant cylinder/cone surface has the largest amount of points in a cell.

A. Surface Normal Estimation

Many surface normal estimation methods that use 3D point clouds, and various surface normal estimation strategies have been compared in [15-17]. To reduce a noise that is satisfied to erroneous results in all 3D point clouds initially we perform the *Gaussian Smoothing* in the local image coordinate frame(Z axis pointing forwards in measurement direction). Then all surface normal vectors, $N = \{n_1, n_2, \cdots, n_n\}$, are obtained to perform normal vectors estimation from triangles formed with pairs of neighbors are averaged of N by N mask of image coordinate[15]. For getting more robust surface normal from noisy point clouds instead of the commonly used nearest N-neighbors method, N by N mask of tangential plane in camera coordinates frame is used. By equation (1) we calculate the rotation matrix $R_i(k,\theta)$of each mask by transforming all neighbor 3D points those are able to project on each tangential plane mask into local coordinates of the tangential plane where $\theta = acos\,(n_r \cdot n_i)$, $c\theta = cos\theta$, $s\theta = sin\theta$ and $v\theta = 1 - cos\theta$.

$$R_i(k,\theta) = \begin{bmatrix} k_x k_x v\theta + c\theta & k_x k_y v\theta - k_z s\theta & k_x k_z v\theta + k_y s\theta \\ k_x k_y v\theta + k_z s\theta & k_y k_y v\theta + c\theta & k_y k_x v\theta - k_z s\theta \\ k_x k_z v\theta - k_y s\theta & k_y k_z v\theta + k_x s\theta & k_z k_z v\theta + c\theta \end{bmatrix} \quad (1)$$

The sign of θ is determined by the right-hand rule. If θ is satisfies $180 \le \theta < 360$, then $R_i(-k,-\theta)$ is a feasible solution. The vector n_z is a unit vector which is usually defined by the z axis component of Cartesian coordinate frame. All transformed points in each cell of tangential plane mask are averaged, then we get represented N^2 3D points for estimating surface normal vector from triangles formed with pairs of neighbors are averaged of N by N tangential plane mask once again.

B. Surface Segmentation

Object and shape reconstruction have been widely studied in literature through approaches like Hough transform(with geometric primitives: plane, cylinder, cone and sphere) [18-20] which may become inefficient as the dimensionality of the parameter space increases. We assume that a specific

group of points X_s on a Gaussian image can be transformed into a group of points X_c on the Circle Frame by transformation matrix T_s^c.

We use a set of unit surface normal vectors, $\{(i,j,k)_n\ , n = 1, \dots, k\}$ defined at individual 3D surface points to specify the input data. The variables we need to find include: $\{\boldsymbol{n}\}$ which captures the circle equation with the parameters of vector $\boldsymbol{n}$, $\boldsymbol{n}$=[*x, y, z*], which is the vector normal to the plane where the circle is defined, and *d,* the distance from the origin of the (i,j,k) space to the plane where the circle is defined.As shown in Fig.2. (b), we make local bins in (i,j,k) space and find the ones that represent the local maxima with a count of which is larger than a specified threshold. Those points in the local maxima might be a part of a plane. As shown in Fig. 2. (c), we obtain a transformation matrix, ${}^{S}T_{C}$, from a IJK frame to the parametric frame, (i',j',k'). The l vector is represented as $\boldsymbol{n} \times \boldsymbol{k}$, ${}^{S}l = [-y \quad x \quad 0]^T$. We conclude by rotating the spherical frame around l throughθ. The position of one point in IJK space is ${}^{S}X = {}^{S}R_C\,{}^{C}X + {}^{S}P_C$, where ${}^{S}P_C$ is a translation.

$$\Theta(l,\theta) = T_C^S = \begin{bmatrix} y^2(1-z)+z & -xy(1-z) & x\sqrt{1-z^2} & 0 \\ -xy(1-z) & x^2(1-z)+z & y\sqrt{1-z^2} & 0 \\ -x\sqrt{1-z^2} & -y\sqrt{1-z^2} & z & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

In order to move each point from the Sphere frame to the Circle frame, we use the inverse matrix of T_C^S. We finally can transfer all points to the Circle frame using $T_S^C = (T_C^S)^{-1}$.

$$\mathrm{X_c} = T_s^c \mathrm{X_s} \quad (4)$$

Transformed points in the Circle Frame could be located on one small area by (4). If a group of points are in a Gaussian image, which implies that a cylindrical property was successfully transformed into a small area in the Circle Frame, then we can calculate α by (5) and it should approach 1.

$$\mathrm{X_c}^T A \mathrm{X_c} = 1, \qquad A = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$
$$\left(X_s{}^T T_s^c\right)^T A (T_s^c X_s) = \alpha$$

There are 4 variables, *i*, *j*, *k* and *d,* that we can map from 4 dimensions to 3 dimensions by the use a spherical coordinate mapping, $\rho(=d), \theta\ and\ \varphi$. However, there is one constraint, which is that z values of all transferred points in the Circle frame should be zero. To search for cylindrical and conic surface patches, a set of bins are generated in the parameter space of the circle equation as shown in Fig. 2. (e). For each bin representing a particular parameter value, all the (i, j, k)s are collected such that they satisfy the circle equation (6). The complete surface patch segmentation is outlined by algorithm of surface patch segmentation. For each bin representing a particular set of parameter values, all the (i, j, k)s that satisfy the circle equation (6) are selected.

$$x = \sin\theta * \cos\varphi \; y = \sin\theta * \sin\varphi \qquad z = \cos\theta \; r = \sqrt{1-d^2} \quad (6)$$

ALGORITHM
SURFACE PATCH SEGMENTATION

Input:	A Set of Unit Surface Normal Vectors, $\{(i,j,k)_n\ , n = 1, \dots, k\}$, defined at individually captured 3D surface points, the circle equation with the parameters of vector *n*, the vector normal to the plane, where the circle is defined, and *d,* the distance from the origin of (i,j,k) space to the plane where the circle is defined.
Output:	A set of plane and cylinder surface patches$\{\pi_i\}$ and $\{\varphi_i\}$ containing parameters and 3D point clouds
Step 1:	In the IJK Space, generate a set of local bins, $\{B_i\ , i = 1, \dots, n\}$, representing the local region around uniformly distributed reference points. Then count the number of (i,j,k)s that fall into this bin.
Step 2:	Find the bins that represent the extreme density of the count of (i,j,k)s which are greater than a threshold. Those (i,j,k)s listed in these bins are labeled as plane candidates.
Step 3:	To search for cylindrical and conic surface patches, generate a set of bins in the Parameter Space of the circle equation in Circle Frame. For each bin representing particular parameter values, collect all the (i,j,k)s that satisfy the circle equation (We may use multi-step computation for making use of necessary conditions for efficiency.) Step 3 performs the Hough transformation for the identification of cylindrical and conic surface patches.
Step 4:	Apply the same procedure defined in Step 2 for the bins defined in the Parameter Space. Those (i,j,k)s listed in these bins are labeled as cylindrical and conic surface candidates.
Step 5:	Based on the plane candidate (i,j,k)s, segment them based on their 3D Space for spatial homogeneity.
Step 6:	Based on the cylindrical and conic candidate (i,j,k)s, the unit sphere that represents the collection of unit surface normal vectors, find the circles.

After transferring all points into a Circle Frame, we create uniform cells. Because of the use of spherical coordinates, it is not desirable to separate a constant interval of theta and phi. We use a *Golden Section Spiral* for generating points with uniform distribution on a sphere. Counting how many points there are in all cells, we choose the cell with the maximum amount of points. As shown in Fig. 2. (f), the parameter space changes into a histogram space for finding out the greatest or greater than a threshold cell. All points in the cell make a circle on a sphere in IJK space and a cylinder or a cone in real 3D space. Finally we find a cylinder or a cone. In IJK space, we make points with uniform distribution and then count the points around the point we selected. We choose some of points that have many IJK points. These most likely will be part of a plane.

We can also segment parts in 3D and 2D image spaces using a distance based classifier. Overall, we have three spaces, 3D/2D space, IJK space and Circle space; if we cannot segment the part in one space, we move all points among those spaces until the segmentation is successful.

IV. OBJECT RECOGNITION AIDED BY CAD DATA

A. Feature Extraction in CAD Data

A vehicle alternator is composed of many small geometric surface primitive patches and complex parts to accurately represent the entire object model. So for system speed and

efficiency a few dominant features as like surface patch primitives, namely plane and cylinder, were used for object categorization and recognition. For getting meaningful surface patch segmentation from CAD data, we try to extract geometric primitive patches from DXF models automatically which is widely used not only for historical reasons but also for adopting simple entities like planes, vertexes and lines for model representation. For surface patch classification aided by CAD data, first, we extract all triangle meshes from DXF data by open source converting program. Then we can classify the surface patch with the same method mentioned in our proposed surface patch segmentation algorithm. In this case triangle mesh does not have any 3D point clouds for finding specific pattern in IJK Space and Parameter Space. So we give a weight factor ω of mesh area instead of counting 3D point clouds in each triangle mesh. The area, $A(\Delta_i)$, of given triangle mesh $\Delta_i = V_0V_1V_2$ is given by half the magnitude of the cross product of two of its edge vectors; namely, $\frac{1}{2}|V_0V_1 \times V_0V_2|$. The ω is normalized by $A(\Delta_i) / [A_{max} - A_{min}]$, then maximum weight group of meshes in each spaces are used.

After performing the segmentation from the CAD data, we get a set of plane and cylinder, and especially we also extract various cylinder parameters by modeling method in[3][14].Some characteristics: 1) Amount of plane and cylinder, 2)Cylinder and plane parameters: radius and height and 3) Intersection lines and points between plane to plane, cylinder to plane and cylinder to cylinder are used for object recognition and pose estimation.

B. *Object Recognition*

To recognize the target object, an alternator, initially we assume the target object is consisted of two key cylinders were connected to planes or other cylinders. Ideally, the set of plane$\{\pi_i\}$and cylinder $\{\varphi_i\}$ patches from raw data are always included in the set of plane $\{\pi'_i\}$and cylinder $\{\varphi'_i\}$ from CAD data. For successful object recognition, we consider four possible cases:

- Find the largest radius cylinder in the surface patch sets of $\{\varphi_i\}$ and $\{\varphi'_i\}$;
- Find sub-cylinder smaller radius than the largest radius cylinder and very close to the largest radius cylinder in surface patch sets of $\{\varphi_i\}$ and $\{\varphi'_i\}$;
- Find plane, intersected with the largest radius cylinder or sub-cylinder, in surface patch sets $\{\pi_i\}$and $\{\pi'_i\}$;
- Find intersection relationship, plane-to-cylinder, from base sub-cylinder and large plane.

We define a set of "models" that consists of all cylinders and planes in $\{\varphi'_i\}$ and $\{\pi'_i\}$, and we also define a set of all cylinders and planes in $\{\varphi_i\}$and $\{\pi_i\}$ are "observation". Different radiuses *r* and heights *h* of cylinder parameters are prepared from all cylinders in model. For finding a suitable cylinder between the model and the observation, *r* and *h* of cylinder of observations and models are used to compare with each other for matching. If one cylinder is selected in model, all *r*and *h* values of the cylinders in observation are calculated

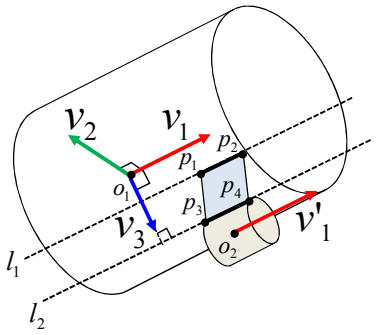

Fig. 3. Geometric relation of cylinders and plane in alternator

with the *r* and *h* of the model, if the difference between the value of r and h is smaller than a threshold, an observed cylinder is selected from set as a matching candidate. The best fitting cylinder from many candidates to the 3D point clouds from observation is fitted to the ideal cylinder which is generated by the superquadric implicit equation with constant parameters $e_1 \approx 0$ and $e_2 = 1$ and by the estimated radius parameter $r$of observation. Then, the fitness calculation (7) is iterated and if the 3D point clouds fitting constraint to the cylinder are satisfied, the iteration is terminated and the cylinder observation is finally selected for both main cylinder and sub cylinder.

$$f(x,y,z) = \left(\left(\frac{x}{r}\right)^{\frac{2}{e_1}} + \left(\frac{y}{r}\right)^{\frac{2}{e_1}}\right)^{\frac{e_1}{e_2}} + \left(\frac{z}{r}\right)^{\frac{2}{e_2}} \approx 1 \qquad (7)$$

After a *M* to *N* matching by cylinder between the two sets, we can define the key cylinder that has at least one or more planes. For reducing invalid candidates and incorrect matching, the intersection relation between planes and cylinders was successfully used in the recognition process. For this we consider the following indicators:

- Number of intersected planes with key cylinder;
- Area similarity measure between intersected planes and key cylinder;
- Existence of shared planes between two key cylinders.

We regard this consideration as a key step in our object categorization. The main parts of the alternator mostly consist of independent plane and cylinder or have intersection relationships with each other. However, other metrics can also be used to achieve a more reliable categorization process. Such metrics include the angle of intersection between two planes and the distance between two cylinders. If we find correct CAD data set from $\{\varphi'_i\}$ and $\{\pi'_i\}$, we can also try to estimate pose from key cylinders and planes.

C. *Pose Estimation*

To get the pose of recognized object, we estimate the rotation matrix and the translation vector independently. For example two cylinders or one intersected plane are matched from object recognition as shown in Fig. 3.Initially, we can easily get two normal vectors, $\boldsymbol{v_1}$ and $\boldsymbol{v'_1}$,of two cylinders. Ideally, these two vectors should be parallel each other. It is possible to estimate the normal vector $\boldsymbol{v_3} = \frac{\boldsymbol{o_2 - o_1}}{|\boldsymbol{o_2 - o_1}|}$. The

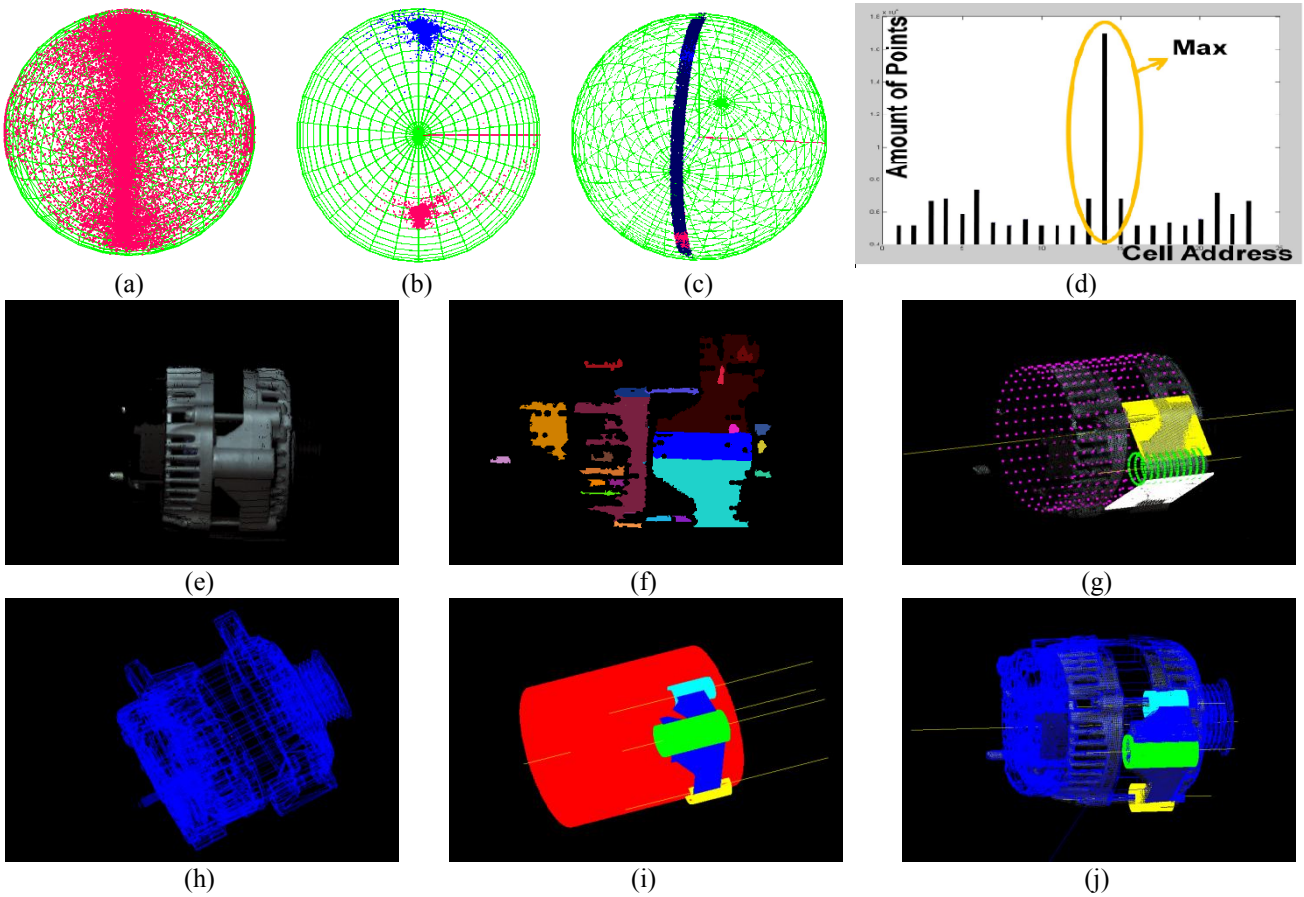

Fig. 4. Experiment results of "Alternator A": (a) (i,j,k) space representation, (b) dense area for plane candidates, (c) dense area for cylinder/cone candidates, (d) histogram space from parameter space for dominant cylinder/cone candidates, (e) 3D point clouds, (f) segmentation results, (g) object modeling from 3D point clouds; observation, (h) CAD data represented by DXF format, (i) object modeling from CAD data; model, (j) object recognition and pose estimation results

normal vector $\boldsymbol{v_2}$ can be obtained by the cross product between $\boldsymbol{v_1}$ and $\boldsymbol{v_3}$, $\boldsymbol{v_2} = \boldsymbol{v_3} \times \boldsymbol{v_1}$.The three normal vectors can be used to generate for rotation matrix $R_{3\times3} = [v_1^T \quad v_2^T \quad v_3^T]$. We can now have two rotation matrixes from the model and the observation.

In order to obtain the translation vector firstly we transform all cylinders and planes in the model to sampled observation by using the two rotation matrixes found before. An initial position can be found by estimating the centroid of all 3D points from the observations. To get the translation vector we are looking for it is useful to know that we can obtain four corresponding points in model and observation: four intersection points between cylinder and plane (p_1, p_2, p_3 and p_4), which intersect the line l_1 and l_2 between cylinder and the plane. Having a large number of corresponding pairs between the model and the observations will increase the accuracy of pose. Afterwards, we can use the Iterative Closest Points (ICP) algorithm to position the origin in measurements coordinates.

$$d = argmin \sum_{i=1}^{n} \sqrt{(q_i - p_i)^2} \qquad (8)$$

If we denote $\{q_i\}$ is model and $\{p_i\}$is observations points, then for all corresponding pairs, the sum of distance d would need to be minimized. Finally, we can get a transformation from the two rotation matrix and one translation vector.

V. Experiment Results

In this section, we validate our system performance via several experiments. We performed test runs with two different objects. The 3D visual perception system was composed of a desktop computer and 3D structured light camera. The desktop computer was an Intel core 2 duo E 7500 2.93 GHz CPU and 2GByte RAM. We evaluated our system performance by measuring the computation time at each step of the process as shown in Table I. At this time, no real-time conversion of CAD models was done between different formats (IGES to DXF). Such computation was done offline and it was not included in the results presented in Table I.

We have applied our algorithm to the problem of alternators pose estimation. We are suggesting a new Hough transformation method to find cylindrical or conic surface patches. A complication lies in that cylindrical and planar parts in the alternator are oriented in same direction. Moreover, the actual object in reality has a large number of features that are perceived as noise and therefore it makes the

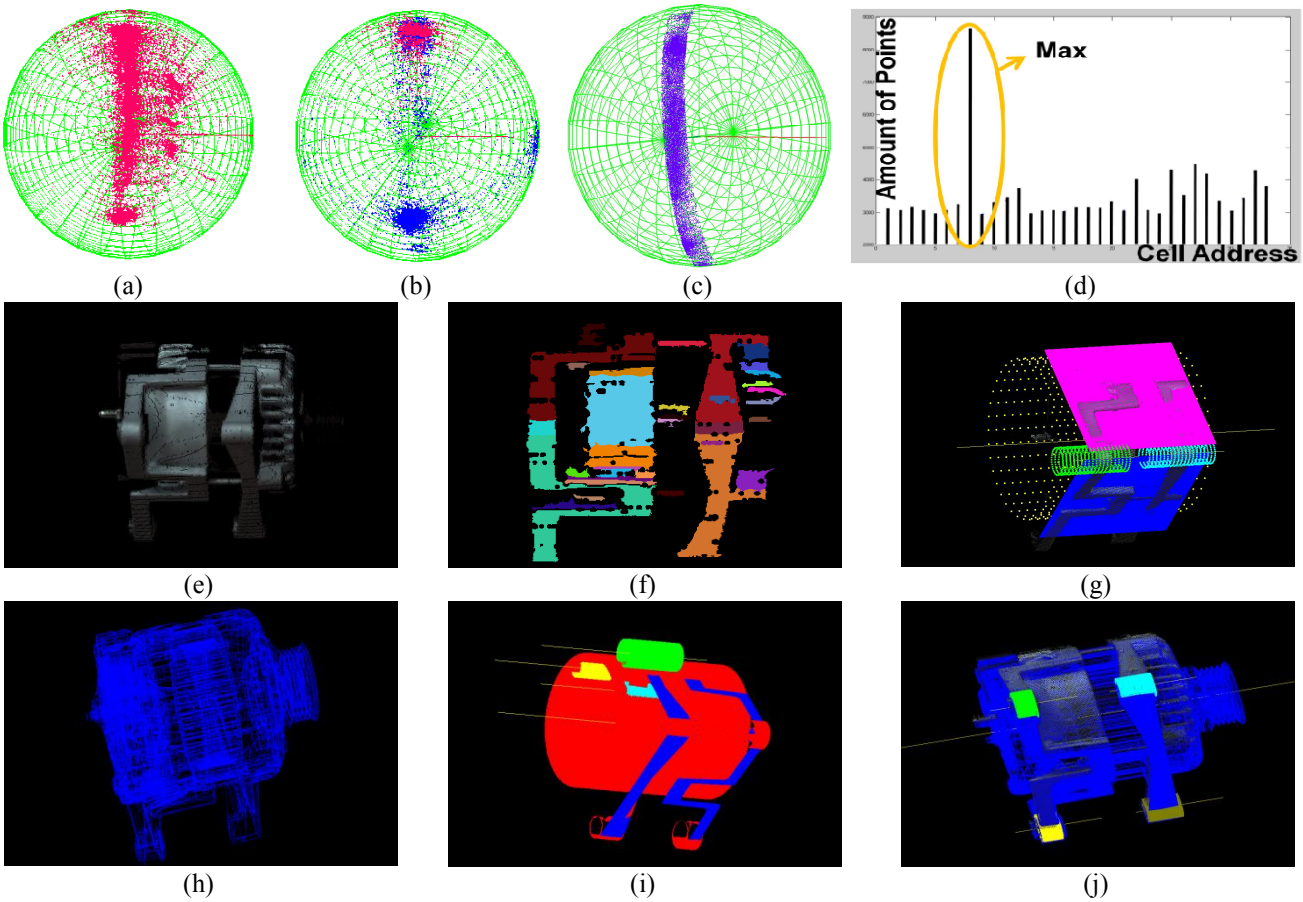

Fig. 5. Experiment results of "Alternator B": (a) (i, j, k) space representation, (b) dense area for plane candidates, (c) dense area for cylinder/cone candidates, (d) histogram space from parameter space for dominant cylinder/cone candidates, (e) 3D point clouds, (f) segmentation results, (g) object modeling from 3D point clouds; observation, (h) CAD data represented by DXF format, (i) object modeling from CAD data; model, (j) object recognition and pose estimation results

segmentation quite challenging. In order to address this issue, we present two different methods for segmentation.

First, we find plane parts in IJK space that represent the normal value for each point. Those points form a circle on a half sphere in IJK space. Given that there are both, cylindrical and planar parts on the alternator, they need to be correctly separated. To this end, we create uniform size cells and count the points which are near each cell. Then we find the local maxima of each cell that has most points count and that gives us a candidate plane primitive. Next, we use a Hough transformation over all point in the alternator to transfer them to a parametric space. We assume that the points forming a circle on a sphere in IJK spaces are gathered around one point in parametric space. Then we find a cell count number of neighboring cell points and get the local maxima just as before. Those points that make a circle on a unit sphere in IJK space are good candidates for cylindrical or conic parts.

We can find the cylindrical parts on alternators using the procedure outlined in Fig. 4 and Fig. 5 where the object recognition process for alternators "A" and "B" are shown.

The pose errors for both cases are presented in Table II. In our robot configuration, the z axis rotation error directly affects the end-effector rotation. For our grasping application, this is the most sensitive factor. We found that such orientation errors are sufficiently small for safe and reliable grasping of both alternator types.

TABLE I
SYSTEM PERFORMANCE

Process	Computation Time
Data Acquisition	2,431.95 msec
Surface Normal Estimation	5,421.80 msec
Surface Segmentation	6,926.70 msec
Object Modeling	8,458.12 msec
Object Recognition& Pose Estimation	1,127.30 msec
Total	**24,367.87 msec**

TABLE II
6-DOF ERROR OF TARGET OBJECTS RECOGNITION

Object	X	Y	Z	X theta	Y theta	Z theta
Alternator A	15.3mm	5.2mm	18.7mm	2.10°	0.50°	1.17°
Alternator B	13.7mm	4.75mm	16.7mm	1.94°	1.70°	1.42°

VI. CONCLUSION& FUTURE WORK

In this paper, we presented an automotive sub-assembly system that uses active 3D vision for object recognition and

pose estimation. Our approach uses surface normal estimation and a Hough-transform-based surface segmentation technique. We presented recognition results for different types of alternators in an integrated task of recognizing, locating and picking parts using a dual arm robot and a 3D vision system. Results demonstrate that both accuracy and runtime were within acceptable operational limits for the specific sub-assembly case under consideration. Future improvements to our approach will include more robust pose estimation under borderline image quality conditions, reduced runtime to match main assembly line throughput requirements, and extending the number and type of recognized objects.

ACKNOWLEDGMENT

This work is mainly supported by the Intelligent Robotics Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea(F0005000-2010-32). This work is also funded by the General Motors R&D, Manufacturing Systems Research Laboratory. This work is also supported by WCU program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (R31-2010-000-10062-0), by PRCP through NRF of Korea funded by MEST (2011-0018397) and This work is also part by the Gyeonggi International Collaboration Research Program (Code I09200) sponsored by the Gyeonggi Province of Korea. The Han Shin Power Tech Laboratory also provided funding and expertise for cooperation.

REFERENCES

[1] R. Guernane and M. Belhocine, "A smoothing strategy for PRM paths application to six-axes MOTOMAN SV3X manipulator," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, 2005, pp. 4155-4160.

[2] Y. Zhang and Y. Jia, "The Control of Archwire Bending Robot Based on MOTOMAN UP6," in *Biomedical Engineering and Informatics, 2009. BMEI '09. 2nd International Conference on*, 2009, pp. 1-5.

[3] J.K.Oh, K. Baek, D. Kim, S. Lee "Development of Structured Light based Bin Picking System Using Primitive Models," 2009 IEEE Int. Symposium on Assembly and Manufacturing, pp. 46-52, 2009.

[4] Iversen: Vision-guided Robotics: In Search of the Holy Grail. In: Automation World, Feb. 2006, pp. 28-31.

[5] Hardin, W.: Vision enables freestyle bin picking. Vision System Design 12(6), June, 2007.

[6] Ban, K., Warashina, F., Kanno, I., Kumiya, H.: Industrial Intelligent Robot. FANUC Tech. Rev. 16(2), 2003, pp.29-34.

[7] H. Borotschnig, L. Paletta, M. Prantl, and A. Prinz, "Appearance based active object recognition," Image and Vision Computing, vol. 18, no. 9, pp. 715-727, June 2000.

[8] P. David and D. DeMenthon, "Simultaneous pose and correspondence determination using line features," in Computer Vision and Pattern Recognition, 2003, pp. 424-431.

[9] P. David and D. DeMenthon, "Object recognition in high clutter images using line features," in 10th International Conference on Computer Vision, 2005, pp. 1581-1588.

[10] C. Von Bank, D. M. Gavrila, and C. Wohler, "A visual quality inspection system based on a hierarchical 3d pose estimation algorithm," in Pattern Recognition, ser. Lecture Notes in Computer Science, vol. 2781. Springer-Verlag, 2003, pp. 179-186.

[11] Changhyun Choi and Henrik I. Christensen, "Real-time 3D Model-based Tracking Using Edge & Keypoint Features for Robotic Manipulation," IEEE Int. Conf. on Robotics & Automation, May, 2010.

[12] C. Harris, Tracking with Rigid Objects. MIT Press, 1992.

[13] Markus Ulrich, Christe Wiedemann, and Carsten Steger, "CAD-Based Recognition of 3D Objects in Monocular Images," IEEE International Conf. on Robotics and Automation, May 2009.

[14] Joon-Young Park, Kyeong-Keun Baek, Yeon-Chool Park and Sukhan Lee, "Generic Model Based Axis Sysmetric 3D Object Modeling," International Conference on Intelligent Systems, Information Processing, and Network, vol 23, 2007.

[15] K. Klasing, D. Althoff, D. Wollherr, and M. Buss, "Comparison of surface normal estimation methods for range sensing applications," in Proceedings IEEE ICRA, 2009.

[16] S. Jin, R. Lewis, and D. West, "A comparison of algorithms for vertex normal computation," The Visual Computer, vol. 21, no. 1-2, pp. 71–82, 2005.

[17] T. K. Dey and et al., "Normal estimation for point clouds: a comparison study for a voronoi based method," in Eurographics/IEEE VGTC Symposium Proceedings on Point-Based Graphics, pp. 39–46, 2005.

[18] A. Sarti and S. Tubaro, "Detection and characterization of planar fractures using a 3-D Hough transform," Signal Process., vol. 82, no. 5, pp. 1269–1282, 2002.

[19] T. Rabbani and F. Heuvel, "Efficient Hough Transform for Automatic Detection of Cylinder in Point Clouds," in ISPRS WG III/3, III/4, V/3 Workshop, Laser scanning 2005, Enschede, The Netherlands, 2005.

[20] G. Vosselman and S. Dijkman, "3d building model reconstruction from point clouds and ground plans," in International Archives of Photogrammetry and Remote Sensing, Volume XXXIV-3/W4 pages 37- 43,Annapolis, MD, 22-24 Oct. 2001, 2005.