MODEL BASED 3D OBJECT RECOGNITION USING LINE FEATURES[1]

Samuel H. Chang, Sukhan Lee, DongJu Moon, WoongMyung Kim, YeungHak Lee
Intelligent System Research Center,
Sungkyunkwan University,
Suwon, KOREA
chang@sisinc.org, lsh@ece.skku.ac.kr, dj_moon@ece.skku.ac.kr, wmkim89@empal.com, annaturu@skku.edu

Abstract – **This paper presents an effective and robust model-based 3D object recognition algorithm using line feature correspondences. The algorithm identifies an object and estimates the pose of the object and provides the object recognition information to a home service robot system. The identification of the object and estimation of the poses is done by a three-stage processing: In the first stage, a list of corresponding image line sets that share same properties of mode line sets is generated. In the second stage, the correspondences in the list are evaluated and ranked. Only a small number of correspondences from the list are selected. Each correspondence contributes a pose hypothesis consisting of a similarity transformation. In the last stage, the approximate model pose hypotheses are estimated based on the selected correspondences. Local adjustment is used to ensure the matching and to improve pose accuracies.**

1. INTRODUCTION

Most approaches to model-based partial object recognition problem use feature-based techniques and rely on corresponding features: correspondences between model features and image features are first created. Then a hypothesized model pose is estimated based on the correspondence. And finally additional image features are searched to support the pose. In approach using line feature correspondences, a list of correspondences between model features and image features need to be generated. Each pair of a model line to its corresponding image line contributes a pose hypothesis. However, each image line may correspond to a number of model lines and may contribute multiple poses to the hypothesis list. The list can be too long when the number of image lines and model lines are large. In a real time system, large number of correspondences and hypothesized model poses make it difficult even impossible to implement because of the slow processing speed. Practically, when the number of model and image features becomes very large, most of the pose hypotheses are inaccurate because the presents of supporting correspondences become intractable.

In this paper, we introduce a new method that only considers the most tractable correspondences, and reduces most of inaccurate and intractable correspondences and skips the estimate of their related pose hypotheses. In our algorithm, we assume that there is more than one line features detected. Therefore, the correspondences between two sets of line features but not between a pair of single line features can be found. Practically, with a single feature, it is almost impossible either to recognize an object or to estimate the pose of the object. We should note that, even though the fact that a corresponding set contains larger number of lines may provide more tractable correspondences, we still have to take the object surfaces the lines represent into the consideration.

In our algorithm, the correspondences between a set of image feature and a set of model features are first created. Then the correspondences are ranked based on the numbers of line feature in the sets and the number of the surfaces of the model the line set covers. At the end, only a small number of correspondences are selected to estimate the poses of the hypothesized model. The hypothesized model poses are calculated using an optimal estimation that minimizes the measurement distances between the corresponding features. In an application for a home service robot system shown in the experiment, the algorithm outputs a model id and a small number of the best model pose hypotheses.

2. RELATED WORK

The most recent work closest to ours is object recognition in cluttered environment using line features by David and DeMenthon [1]. A large number of approximate pose hypotheses is generated and then ranked based on local information. A robust pose is found from the list at the end.

[1] This research was supported by MIC (the Ministry of Information and Communication, Korea), under the Foreign Professor Invitation Program of the IITA (Institute for Information Technology Advancement).

But the performance depends on how reliably the algorithm can move to the top of the sorted hypothesis list and camera orientations. The line matching algorithm of Kamgar-Parsi [2] relies on the repeated use of matching sets of equal-length line segments. While the algorithm works for all basic line matching cases, however, in case of finite-finite line matching, partially overlapping line segments is not considered. Zhang and Faugeras [3] were the first to present line matching problems. There is, however, an implicit assumption that the corresponding points are the midpoints of the corresponding line segment pairs that in fact is difficult to find. They also use line lengths to reject corresponding line hypotheses but don't use the line lengths in matching [4]. Guerra and Pascucci [5] present a different matching method between two sets of 3D line segments with unknown line correspondences. Combination of lines and points are also matched with a random start local search [6]. Features of 2D model line [7], 3D model lines [8] and 2D image lines are also matched to find pose. A geometric hashing method introduced by [9] and has been proved very efficient for object recognition by line feature indexing [10]. However, in the case that a large number of models are considered, it suffers from the large size of voting table.

David and DeMenthon's approach is similar to ours in two ways, first, both approaches generate and rank a number of pose hypotheses. Second, there is only a small number of the best pose hypotheses selected. Significant differences between the two approaches are follows: first, ours uses line set correspondences instead of line correspondences. Therefore, a much smaller number of pose hypotheses is generated. Second, while they rank the pose hypotheses based on local information, we do it based on coverage between the corresponding image features over the model lines.

3. GENERATING CORRESPONDING LINE SETS

We wish first to find a set of image line features that corresponds to a set of model lines and then to match the two sets of corresponding lines. The two sets of lines may be extracted from 3D images or measured from the models. The 3D images can be acquired directly from stereo camera systems or other scanner systems. Practically, we refer to one set of lines extracted from 3D images and the other set from a given model.

In order to find the corresponding line features between a model and an image, the properties of the line features and measurement of the similarities between the line features have to be defined. The properties of line features include positions, orientations, lengths, and relationships among a set of lines, such as intersect and parallel. Practically the length of a line segment is not accurate because the end points of a line segment can be noisy. In most of applications, a line can be broken and its end points can be missing. In our algorithm, positions and orientations are used to establish the relationship among lines.

3.1 Line Set

A line set is defined of a set of line segments related to each other based on their orientations and their locations. For instance, a set of parallel lines forms a plane of a model. A set of intersect lines forms a corner of a model. A line set can contain a parallel line set and a intersect line set if two line set are geometrically connected.

A parallel line set is defined as a set of line segments with similar orientations. The difference of the orientations among line segment is smaller than a threshold value. Let L_b be a base line and L_i be a line other than the base ling from the image. Let O_b be the orientation of the base line and O_i be the orientations of L_i. We want to find all line features from image that satisfy

$$\psi_d(O_b, O_i) \le \delta_\varphi \qquad (1)$$

and

$$D_d(L_b, L_i) \cong \lambda_i \qquad (2)$$

and

$$D_{\max}(L_b, L_i) \le \rho_i \qquad (3)$$

Where $\psi_d(.)$ denotes the absolute different orientation value between two orientations, and $D_d(.)$ denotes the distance between the base line and each other lines, and $D_{\max}(.)$ denotes the longest distance between end points of a line pair. δ_φ is a threshold value, and λ_i is the distant between two parallel lines, and ρ_i is the longest distance between end points of two parallel line segments.

An intersect line set is defined as a set of line segments that intersects at a point either lies on one of the line segments. We want to find all line segments from the image that satisfy

$$\psi_d(O_b, O_i) \cong \varphi_i \qquad (4)$$

and

$$D_{\min}(L_b, L_i) \le \delta_d \qquad (5)$$

and

$$D_{\max}(L_b, L_i) \le \vartheta_i \qquad (6)$$

Where $D_{\min}(.)$ denotes the shortest distance between end points of a line pair. φ_i is an intersect angle in radians between any two model lines. δ_d is a threshold value. ϑ_i is a distance between two end points of a line pair. A "L" junction is formed if $\varphi = 0.5\pi$ and $\delta_d = 0$. Two line set are combined if they share at least one line member.

3.2 Corresponding Line Sets

In order to establish the correspondences between two line sets, the properties of a line set need to be defined. If we define V as a parameter vector describes the line features of a model by giving a set of parameters of δ_φ, λ_i and ρ_i in Equations (1), (2) and (3), and φ_i, δ_d and ρ_i in Equations (4), (5) and (6).

$$V=[\delta_\varphi \quad \lambda \quad \rho \quad \varphi \quad \delta_d \quad \vartheta]' \tag{7}$$

Then the correspondence can be found if the parameter vectors of two line sets are close enough. To establish the correspondence between the image line features and the model line segments is to generate the line set parameter vectors, and then, to search for image line sets that share the same parameters. The corresponding line set can be generated by following steps:

Step 1: Each line segment of a model is given a line id. All model line segments are analyzed and divided into two types of line sets: parallel line sets and intersect line sets. Parameters of δ_φ, λ_i and ρ_i in Equations (1), (2) and (3) are calculated to form parallel line sets. For intersect line sets, parameters φ_i, δ_d and ϑ_i in Equations (4), (5) and (6) are estimated. Together with their properties such as positions, orientations, lengths, line ids and parameter vector V, all model line sets are recorded in a database.

Step 2: A line id is assigned to each image line feature. For all image line features, target lines to form a parallel line set or an intersect line set are searched from other image line features.

Step 3: An image line set is promoted to a corresponding line set if it shares same parameter vector with a model line set from the database.

Step 4: The corresponding model line set is assigned to the image corresponding line set.

Step 5: An image corresponding line set is promoted if it shares a same member with another image corresponding line set.

Step 6: Image corresponding line set is ranked based on their numbers of line members and the surface the cover.

Step 7: A small number of higher rank image corresponding line sets is selected to calculate hypothesis poses.

When the correspondences are found, a similarity transformation can be estimate. However, the similarity transformation can be trivial because there is no knowledge of which image line of a parallel lines corresponds to which model line of the corresponding parallel lines. We consider all possibilities and generate a similarity transformation for each. As a result, hidden line features may be selected. In our algorithm, a processing is designed to remove the hidden features. When a hypothesis pose is computed based on a correspondence, the model is then transformed to the scene. The faces of the model consist of the corresponding lines can be identified. Faces connected to the closest points can be viewed. By checking the orientation of the transformed model, and finding distances between the model corners and the viewpoint may get knowledge of the number of faces are hidden. Therefore, features connected to the hidden faces can be removed.

4. ESTMATE HYPOTHESIS POSES

To estimate the hypothesis from two correspondences is to find a rigid transformation so that the two come to a best match with a minimum distance measurement error. In our algorithm, this is done by two steps: first, an initial transformation is estimated using matching between corresponding line sets. Second, the local adjustment is done by matching between a subset of image features and their corresponding model features. Only accurate image features are used in the matching.

Let $\Re$ and $\aleph$ be two sets features with the same object between two coordinates. Let P be a set of 3-D points representing model feature $\Re$ and Q a set of 3-D points presenting image features $\aleph$. Then P can be mapped into Q by a rigid body transformation of the form

$$Q = RP + T \tag{8}$$

where R is a 3×3 rotation matrix and T is a 3×1 translation vector. The mapping is represented by a rotation followed by a translation. Or

$$Q = A(R,T)P \tag{9}$$

where $A(R,T)$ is a 4×4 transformation matrix with the components of R and T from equation (8). The optimal estimate of the transformation is the mapping that transforms all points in P to all points in Q such that the SSD is minimized. In other words, the problem of minimizing SSD is to find the transformation $A(R,T)$ such that the mean SSD of the distance between mappings of all points in P to all corresponding points in Q achieves the minimum. That is

$$SSD(R,T) = \frac{1}{N_p}\sum_{i=1}^{N_p} \|q_i - Rp_i - T\|^2 \tag{10}$$

This problem has been solved using the ICP algorithm. The ICP algorithm was proposed first by Besl and McKay in [5] and Chen and Medioni in [6] and improved later by Zhang [7]. It is the most common method for aligning views of 3-D objects given a rough estimate of the initial alignment. It has been shown to be a robust method for registering multiple such views. In order to solve the problem of equation (10), a distance metric d between an individual point p_i in P and Q is denoted

$$d(p_i, Q) = \min_{q \in Q} \| q - p_i \| \qquad (11)$$

Equation (4) produces a point q in Q with the shortest distance from point p_i in P. Therefore, q is the closest point to point p_i in P. By using the equation, corresponding points in Q can be found for each point in P. It is then possible to find a transformation of $A(R,T)$ that transforms P to a new point set P'. In its most basic form, the ICP algorithm consists of repeating the update of the transformation of $A(R,T)$ until convergence is achieved.

The advantages of ICP include stable and converge matching results, flexible matching criteria, and local optimal matching. However, the result of recognition using ICP algorithm depends on the initial positions. If a wrong initial pose is given, the algorithm may diverge. Therefore, such an initial pose would be provided. The initial alignment can be done by either aligning a few characteristic points, or in a controlled measurement setup such as calculating the sensor motion between the two views. It should be emphasized that the initial alignment may not be accurate. This allows us to include some important but very noisy line features in initial alignment. However, in local judgment, noisy line features may cause unstable matching. To increase the accuracy of local adjustment, noises of features are measured and only qualify features are used in local matching. Qualify features are consider longer features with smaller noise.

5. EXPERIEMENTS AND CONCLUSIONS

To validate our approach, we recognize partially occluded 3D objects in a home kitchen environment under variety of viewpoints and variant of simple models such as box, books and complex model such as a box shown in Figure 1 and a refrigerator and juice dispenser as shown in Figure 3. All images were acquired at a resolution of 640 × 480 pixels. The numbers of 2D line features detected from the images and 3D line features extracted from the point cloud are variance based on the model size and complexity of the robot working environment. For a complex model such as the juice dispenser in a kitchen as shown in Part (b) of figure 4, there are 97 2D line features detected, and 84 3D line features extracted as shown in part (c) of Figure 3. The performance of the algorithm depends on configured computer system. In the experiment, a computer system with an Intel Pentium-M processor of 1.86 GHz and 1.5 GB RAM is used. The 2D and 3D image capture speed is 30 frames per second. It takes 30 ms to detect 2D lines and 30 ms to extract 3D line features and 300 ms to generate the corresponding line sets and to estimate the pose.

The approach was applied in two experiments. The first, a simple object of a box and the second a complex one. Figure 1 shows the view of a box on a table and the line model of the box. The selected line sets were plotted in Figure 2. By using the line matching, one of the poses transfers the mode to the scene can be viewed from a 3D viewer. One of the view was plotted in Figure 2.

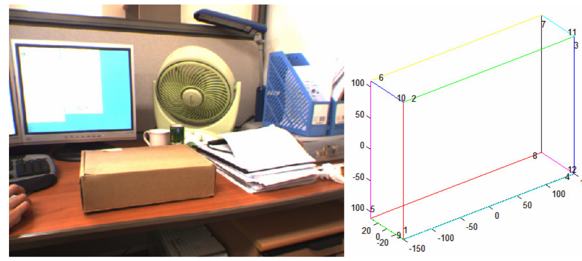

Figure 1: A view of a box object (left) and its line model (right) is used in the experiment.

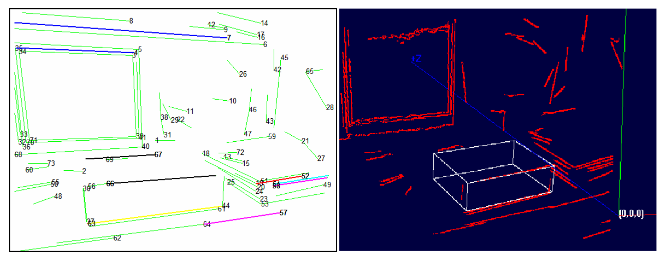

Figure 2: The line sets are detected and plotted in different colors (left) and a 3D view of matching results (right) shows right position of the box.

Figure 3 shows a stereo camera was attached to a robot and one of the view of the camera is shown in Figure 3. A more complex object of a juice dispenser is on the view and the line model of the dispenser is plotted in Figure 4. The selected line sets were plotted in Figure 5 from where both parallel line sets and intersect line sets are found. The line sets then were used to match line sets of the line model. One of the poses transfers the model to the scene as shown in part (d) of the figure 5.

Figure 3: A home service robot with a stereo camera system searching for a juice dispenser object in a kitchen.

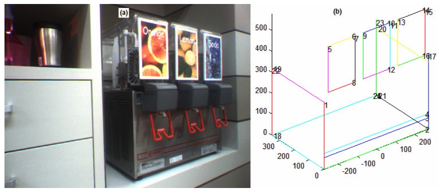

Figure 4: The left image (a) shows the view from the stereo camera system containing an object of a juice dispenser. A designed line feature model is shown in (b).

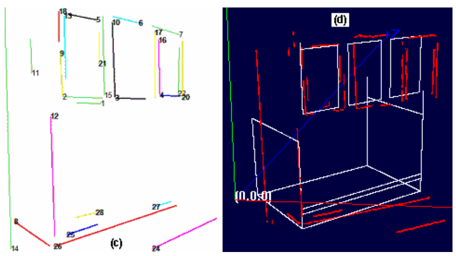

Figure 5: Line set correspondences has been found (c) and one of the poses transforms the dispenser model to the scene shown from a 3D viewer.

We have presented an efficient approach to recognize partially occluded object in a kitchen environment using line features. Our approach improves previous approaches by promoting matching between line features to line set features. Therefore the number of correspondences is largely reduced from a single line correspondence. By searching the maximum number of corresponding features, we are able to greatly reduce the number of poses that need to be examined and still find a correct precise pose in the end and are able to increase the accuracy of the initial matching in pose estimation.

ACKNOWLEDGEMENT

The authors wish to thank Mr. Jaewoong Kim, Mr. Seongsoo Lee, Mr. Changhyun Choi, Mr. Jeihun Lee and Mr. Jangwon Lee for their assistants in experimental data captures and model measurement.

6. REFERENCES

[1] P. David and D. DeMenthon, "Object recognition in high clutter images using line features," Proc. of the 10th IEEE ICCV, 2005.

[2] B. Kamgar-Parsi, and B. Kamgar-Parsi, "Algorithms for matching 3D line sets," IEEE trans. on PAMI, 26(5): 582-593, 2004.

[3] Z. Zhang and O. Faugeras, "Determining motion from 3D line segment matches: A comparative study," Image and Vision Computing, vol. 9, no. 1, pp. 10-19, 1991.

[4] Z. Zhang and O. Faugeras, 3D Dynamic Scene Analysis. Springger-Verlag, 1992.

[5] C. Guerra and V. Pascucci, "On matching sets of 3D segments," Proc. Conf. Vision Geometry, vol. 3811, pp. 157-167, 1999.

[6] J. Beveridge and E. Riseman, "Optimal geometric model matching under full 3D perspective," Computer Vision and Image Understanding, 61(3), pp.351-364, 1995.

[7] N. Ayache and O. Faugeras, "HYPER: "A new approach for the recognition and positioning of two-dimensional objects," IEEE Trans. on PAMI, 8(1):44-54, 1986.

[8] P. David, D. DeMenthon, R. Duraiswami, and H. Samet, "Simultaneous pose and correspondence determination using line features," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. II-424-II-431, 2003.

[9] Y. Lamdan, "Object recognition by affine invariant matching," In Proc. Of the IEEE Conf. on Computer vision and pattern recognition, pp.335-334, 1988.

[10] F. Tsai, "Geometric hashing with line feature," Pattern Recognition, vol. 27, no. 3, pp. 377 – 389, 1994.