Octree-Based Obstacle Representation and Registration for Real-Time

Jaewoong Kim, Daesik Kim, Junghyun Seo, Sukhan Lee and Yeonchool Park*
Intelligent System Research Center (ISRC) & Nano and Intelligent System Lab, 83654, 2nd Research building, 300 Cheoncheon-dong, Jangan-gu, Suwon, Gyeonggi- do, 440 - 746, Korea

ABSTRACT

This paper presents a novel approach for obstacle representation method when robotic manipulation task system understand real indoor environment for 3D workspace modeling. When many objects scattering on the table in workspace, if the robot want to grasp only one object, the robot system should has a path planning to approach the object. In this case, the obstacle representation in cluttered-environment is an important role for robot system. The research area of 3D workspace modeling, the research of step by step accumulating environment observation method is more difficult than the entire environment observing method and the research is not reported sufficiently. In this paper, we can contribute the two issues for real-time 3D workspace modeling to using the sequential input stereopsis scenes information. First, we can suggest a method to estimate the transformation matrix from using SIFT feature and Epipolar Geometry Constraint characteristics in the continuous process of accumulating sequential input stereopsis scenes for more fast and accurate than recently research. This method guarantees a feasible transformation matrix result better than the using traditional ICP and general SIFT method. Second, we can suggest a method for octree-based obstacle representation and we can also suggest an octree update method for real-time. It is faster than entire workspace observation method, and if the robot doesn't know about the objects and obstacles information in workspace, we can help the robot to understand environment himself from practical information. Taking the obstacle information from above method can help the robot system possible to do path planning for robotic manipulation task in 3D workspace. Through the experimental result, we can show that our method is well-performing and well-modeling the obstacle in 3D real environment workspace modeling in real-time.

Keywords: SIFT, octree, epipolar, transformation matrix, workspace modeling

1. INTRODUCTION

In robot vision part, the 3D modeling has been important part. 2D and 3D information from camera is used to make modeling that to represent symbol and data. The model located everywhere in our environment not only small objects but also buildings, nature and large objects. These models use to robot for navigation, understanding environment and objects and robot arm manipulation. In robotic manipulation task area, the result of 3D workspace modeling is very useful information for robot how to understand environment and perform theirs mission. The workspace model is an information donor to robots. When robot received workspace model information, firstly robot can understand the spatial environment information that the wall, floor, table and etc. And then find out to be able to manipulate object to grasp, finally robot do path planning for it. These two processes are important issues for robotic manipulation. Moreover all of them must do in real-time. The computer vision based real-time 3D workspace modeling for manipulation task researches have been reported by [1][2][3]. They suggested "3 Principle workspace modeling" method for workspace modeling. It is a good approach to make model in 3D workspace for robotic manipulation task. In indoor environment, firstly they use one stereo camera in indoor environment. For performing manipulation task, they found and understood global geometry feature, and specific object recognition using SIFT features and model databases. For two scene registration, they used SIFT based transformation matrix estimation. Finally they generated the volume-based obstacle representation for robotic manipulation path planning. They considered the real-time processing sufficiently. But, they performed only straight forward moving-based experiments and results. Moreover, the number of octree cells is too much existed in one scene. Also they had distance-based octree generation rule. It depends on the distance between robot and object. That means short distance object is represented small size octree cell, oppositely long distance object is represented large size octree cell. But, it didn't have reasonable distance measure rule. It is not guaranteed robustness in

*parkpd@skku.edu; phone +82-31-299-6473; fax +82-31-299-6466

ICMIT 2007: Mechatronics, MEMS, and Smart Materials, edited by Minoru Sasaki, Gisang Choi Sang, Zushu Li, Ryojun Ikeura, Hyungki Kim, Fangzheng Xue, Proc. of SPIE Vol. 6794, 67945E, (2007) · 0277-786X/07/$18 · doi: 10.1117/12.784184

real-time processing when robot met very close distance between robot and workspace and cluttered environment, and it is also unstable when robot moves around workspace.

2. TRANSFORMATION MATRIX ESTIMATION

Iterative Closest Point (ICP) method is well-known and traditional method to estimate the transformation matrix in two correspondence scenes [4][5]. Using ICP method is very weak a various camera angle position change. Moreover, through the iteration to find the closest point, it spends much time to perform the algorithm. So, other various methods for fill a gap has been reported during the recently time [6][7][8]. But, this algorithm strongly depended on the 3D point noise error and initial position, and it need to enough overlap area in each of two scenes. So it has still some problem for robustness. It brings that the robot available moving area decrease in the long term. When occur some occlusion or missing calculate the 3D points, it was also unstable.

Instead of that method, to guarantee the real-time, Sukhan Lee et al [1] and H. Jang et al [2][3] were focused on using SIFT feature[9] to estimate transformation matrix. This method expects a good system performance evaluation from reducing amount of 3D points better than ICP. But, when getting the correspondence from overlapping area in two sequential scenes should be occurred mismatch. There is a solution of the problem from removing outlier for saving performance time, but its result is not sufficient. So, we can analyze epipolar geometry constraint through the stereo camera characteristic between 2D scenes, and we can overcome the nature-born mismatching problem of SIFT features.

2.1 Epipolar Geometry Constraint

Consider two image planes with the distinct viewpoints as in Fig. 1. Let $\mathbf{x} = (x, y, 1)$ and $\mathbf{x}' = (x', y', 1)$ be the corresponding points in the two image planes, i.e., the perspective projection points through $\mathbf{o}^{c1}$ and $\mathbf{o}^{c2}$ of the point $\mathbf{x}^w$, in both image planes, respectively. The epipolar geometry defines the geometric relationship between these corresponding points. The plane passing through the optical center $\mathbf{o}^{c1}$, $\mathbf{o}^{c2}$ and the scene point $\mathbf{x}^w$ is called an epipolar plane. The projection $\mathbf{e}$ ($\mathbf{e}'$) of one camera center on to the image plane of the other camera frame is called an epipole. Note that the projection may occur outside the physical boundary of the imaging sensor. An epipolar line $\mathbf{l}$ ($\mathbf{l}'$) is the intersection of an epipolar plane for $\mathbf{x}^w$ with the image plane. All epipolar lines pass through the epipole.

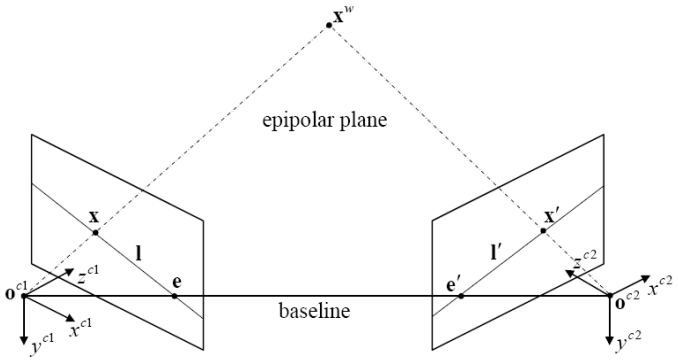


Fig. 1. Epipolar Geometry Constraint

The fundamental matrix represents the epipolar geometry algebraically and contains most of the information about the relative position and orientation between the two views. Suppose we have two images acquired by cameras with distinct view points, then the fundamental matrix $\mathbf{F}$ is the unique 3×3 rank 2 homogeneous matrix which satisfies

$$x'^{T} Fx = 0 \tag{1}$$

for all corresponding points $\mathbf{x}$ and $\mathbf{x}'$ in two image planes. For more details, see [10].

2.2 To Estimate Transformation Matrix

For a rigid 3D transformation, at least of three non-collinear point correspondences are required for a unique solution. With more correspondences, the accuracy of the transformation can be increased, however that does not be always guaranteed due to the erroneous data. Thus we estimate the transformation matrix with RANSAC. Given four non-collinear 3D point correspondences, then we can form the following equation:

$$m_1 = Tm_2 \tag{2}$$

then T can be computed simply by

$$T = m_1 m_2^{-1} \tag{3}$$

The matrix T is a transformation matrix, and that is consist of 3X3 rotation matrix R and 3X1 translation matrix T. Each of ideal rotation matrix three rows that is notated $\bar{r}_1, \bar{r}_2$ and $\bar{r}_3$ have a special characteristic which must orthogonal relation each other, and it must sufficient bellow equation:

$$(\bar{r}_1 \times \bar{r}_3) \bullet (\bar{r}_2 \times \bar{r}_3) < \lambda \tag{4}$$

where λ is user-threshold. This value has been zero when it is an ideal condition. But this value can't have zero, because previous explained two error conditions. So we must select almost zero value, that is consist of matched 3D SIFT points which approximate zero, then it is a best transformation matrix. After then, we can perform scene registration from the transformation matrix, and then it is possible real-time registration sequential two scenes.

3. OBSTACLE REPRESENTATION

Octree based obstacle representation is a good way to present the obstacle and to make the path for planning to grasp specific object. It must need real-time and accuracy. So we suggest our obstacle representation method to guarantee there two issues. We can make a reasonable octree cell in 3D point clouds by octree Fill Rate. That is camera parameter to generate 3D point, and then we do the octree update process. For the first, we can set up the world coordinate origin from initial camera position. And we can generate and store the initial octree cells of initial position from input 3D scene in there. Then the camera position must change the arbitrary position that must different from before position to understand unknown obstacle, then we can generate other octree cells. The method of firstly mentioned, we can have transformation matrix between before and current input scene, and through the transformation matrix, we can move current octree cells to initial position, then we can add initial octree cells to current octree cells. Under this method, we can represent the obstacle more and more accurate through the continuous simple octree update method from sequential input scenes.

3.1 Octree Fill Rate

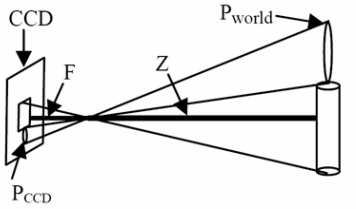


Fig. 2. Characteristic of pinhole camera model

In Fig. 2, P_{ccd} is a point size in the camera CCD, F is focal length, Z is the distance between camera to 3D point clouds. We want to measure of real size from observing point with P_{world} and P_{ccd}, and we can derivate simple equation bellow:

$$P_{CCD} : P_{world} = F : Z \tag{5}$$

And we can also derivate fill-rate equation bellow:

$$Fill\ Rate\ = \frac{Octs^2}{P_{CCD}{}^2 * Z / F} \tag{6}$$

where $Octs$ is octree cell size in real environment, $Fill\ Rate$ is a decision threshold of fill or not in the cell. Real 3D shape has volume, but stereopsis data that bring to take scene from camera has only 2.5D. So we define only surface area information instead of volume. In Equation (5), denominator means real 3D point size and numerator means arbitrary one octree cell size. For that reason, each fill-rate that brings to the real point indicates a ratio of each point which is a

proportion of point in each octree cells. So the computation point that multiplied fill-rate and point amount can measure quantity of points in octree cell that like a real environment.

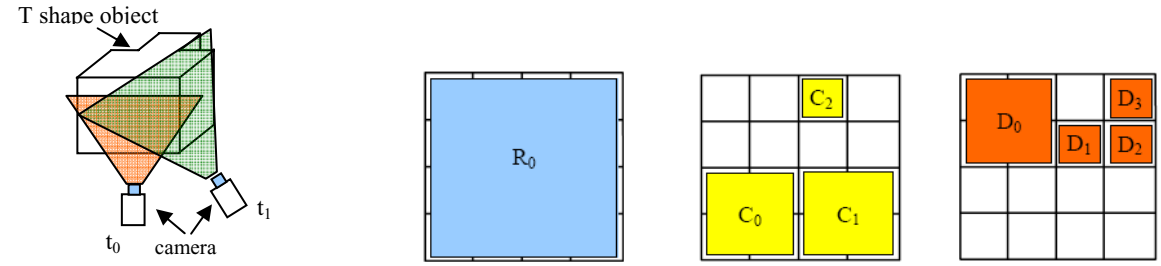
3.2 Simple Octree Update

In Fig. 3 (a), we show the example of our simple octree update method. There is two cases camera direction at time t_0 and t_1. Fig. 3's (b), (c) and (d) is a top view results of octree generation. To perform octree update has two issues. First, we must set up initial camera position and camera view direction. It brings initial size of the largest octree cell in that scene and initial camera view angle direction. At that time, if we generate octree, we define it is the *"Reference Octree(RO)"*. When second scene arrives, we must change all 3D points from the second observed position to initial position, then we can generate octree cell to the same octree fill rate rule. At that time, if we also generate octree cell, we define that is the *"Current Octree(CO)"*. And then, if we have both RO and DO, we can generate virtual octree cells from to find out intersection between RO and CO. Then we can remove intersection in RO, that means RO – CO, finally we can get the difference RO and CO. In that case, we can define the "*Difference Octree(DO)*". After all, we can do all of entire process iteratively.

Table. 1. The Pseudo code of simple octree update method for real-time

Algorithm : Simple Octree Update

1. **WHILE** all octree cell search end and generate octree cell
2. Generating Initial octree cells and call it "*Reference Octree(RO)*"
3. **IF** camera position changed **THEN** take new scene, and generating octree cell and call it *"Current Octree(CO)"*
4. Change current 3D point position to initial position with transformation matrix
5. *DO* = *CO* – *RO* and call it "*Difference Octree (DO)*".
6. **IF** there exist overlap in *DO* **THEN** remove overlap cell in *DO*
7. Generating new *RO* = *CO* + *DO*



(a) example of simple octree update (b) reference octree (c) current octree (d) difference octree

Fig. 3. Characteristic of pinhole camera model

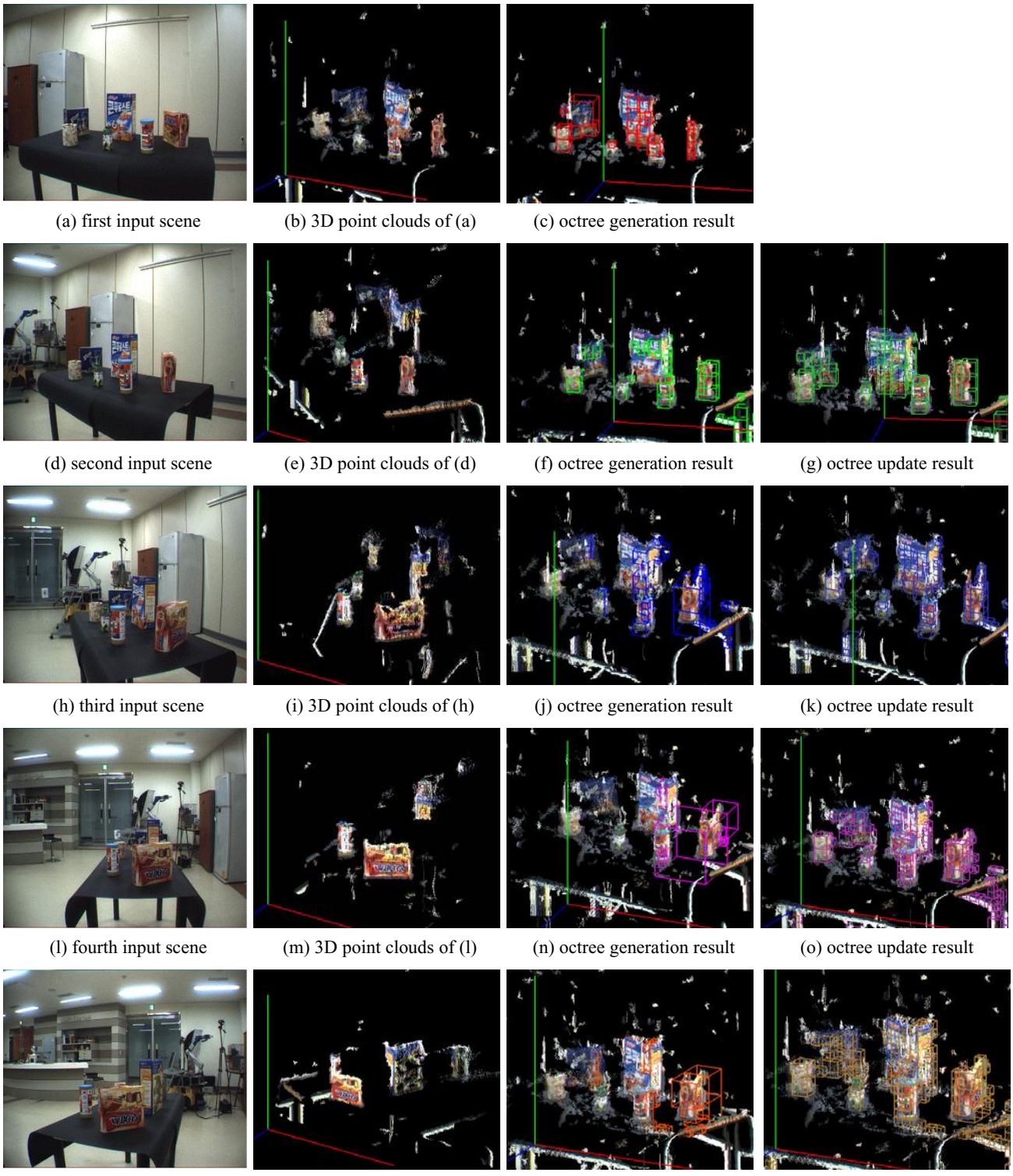
Difference Octree, in general, must have unknown and invisible area information, because of that made only check out simple relation of difference and intersection information both Reference Octree and Current Octree. So we will remain octree cell that is absolutely certain cells and to regard the cells as certain to use generating history information.

4. EXPERIMENTS AND RESULTS

In the experiments, we use one IEEE 1394 stereo camera that has 640X480 resolution of image. That camera is named "Bumble Bee" that produced by PointGray. This camera serves 30Hz 3D scenes, and it has 1/3'' CCD. Finally the

Table. 2. System Performance

SIFT Extraction	284 ms
Registration	39 ms
Octree Generation	103 ms
Octree Update	101 ms
Total	527 ms



(a) first input scene (b) 3D point clouds of (a) (c) octree generation result

(d) second input scene (e) 3D point clouds of (d) (f) octree generation result (g) octree update result

(h) third input scene (i) 3D point clouds of (h) (j) octree generation result (k) octree update result

(l) fourth input scene (m) 3D point clouds of (l) (n) octree generation result (o) octree update result

(p) fifth input scene (q) 3D point clouds (p) (r) octree generation result (s) octree update result

Fig. 5. Experiment results by input 5 images

camera focal length is 4mm. And we use the Pentium 4, 2.4GHz CPU PC and 1Gbyte RAM, we implemented algorithm code on the VC++ compiler. For measuring the performance time, we experienced 100 times of entire process that

consider the various angle of camera view. And we can show that statistical performance result in table 2. We can show that our system to guarantee the performance time at most in 600ms, it is sufficient considering of real-time. We can show the results of scene registration, octree generation and simple octree update from five sequential input scenes, and we also show the results in Fig. 5. In Fig 5, (a),(d),(h),(l) and (p) is presented original 2D input scenes. (b),(e),(i),(m) and (q) is showed 3D point clouds that was generated from original input 2D scenes, and (c),(f),(j),(n) and (r) is the results of octree generation with octree fill rate. Finally, we can show the final integrated OpenGL rendering results that (g),(k),(o) and (s) was generated by the simple octree update method.

5. CONCLUSION AND FUTURE WORK

In this paper, we can show the real-time obstacle representation method for robot manipulation task. In the first see sight, the result is not sufficient for representing obstacle. But, scene was registered more and more, we can show the more accurate and reasonable results than before. It can help the robot can do more various path planning and well understand the spatial area information for grasping specific object. And we can show to guarantee the system performance time for real-time processing. In the future, we will research more robust and faster modeling for obstacle than now. We also evaluate octree generation performance everywhere. The other plan includes the more precise octree update strategy and octree data expression of object, and we will serve workspace information and integration within robotic motion planners.

ACKNOWLEDGEMENT

This paper was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Science and Technology of Korea. This work is partly supported by the Science and Technology Program of Gyeonggi province.

REFERENCES

1. Sukhan Lee, Daesik Jang, Eunyoung Kim, Suyeon Hong, and JungHyun Han, “A real-time 3D workspace modeling with stereo camera,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2140 – 2147, 2005.
2. Han-Young Jang, Moradi. H, Lee. S, and JungHyun Han, “A visibility-based accessibility analysis of the grasp points for real-time manipulation,“*IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 3111 – 3116, 2005
3. Han-Young Jang, Hadi Moradi, Suyeon Hong, Sukhan Lee, and JungHyun Han, “Spatial Reasoning for Real-time Robotic Manipulation,” *IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp.2632 – 2637, 2006
4. P.Besl and N. McKay, “A method for registration of 3D shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligent*, vol. 14, pp. 239-256,1992.
5. Y.Chen and G. Medioni, “Object modeling by registration of multiple range images,” *In Proceedings of the IEEE Conference on Robotics and Automation*, 1991.
6. S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP," *The 3rd International Conference on 3D Digital Imaging and Modeling*, 2001.
7. A. Fitzgibbon, "Robust registration of 2D and 3D point sets," *In Proc. British Machine Vision Conference*, volume II, pp. 411−420, Manchester, UK, 2001.
8. G. C. Sharp, S. W. Lee and D. K. Wehe, “ICP Registration using invariant features,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(1), 90-102, 2002.
9. D. Lowe, “Object recognition from local scale invariant features,” *In Proc. 7th International Conf. Computer Vision (ICCV’99)*, pp. 1150–1157, Kerkyra, Greece, 1999.
10. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, *Second Edition*, Cambridge University Press, 2003.