

# Adaptive Voxel Mapping Based on 3D Occupancy Grid Maps Using Object Detection

Yeong Eun Choi\*, Soo Young Shin<sup>o</sup>

## ABSTRACT

This paper proposes a voxel-based adaptive resolution mapping method to enable autonomous robots to efficiently perceive and map 3D spaces in various environments. Using the YOLO real-time object detection model, the system selectively switches the required areas and forms a consistent map by excluding overlapping points between high-resolution and low-resolution maps on an octomap. The study measures the number of point clouds in single-resolution and multi-resolution scenarios and analyzes map capacity by varying the ratio of high-resolution to low-resolution areas in adaptive resolution mapping. Experimental results demonstrate that the proposed mapping method maintains the accuracy of environmental perception while demonstrating efficiency in data capacity and computational performance, confirming its potential for resource optimization in real-time autonomous robot operations.

**Key Words** : Mapping, Occupancy Grid Map, Adaptive Resolution, Octree

## I. Introduction

With advancements in 3D sensing technology, autonomous agents have significantly improved their ability to accurately perceive their surroundings and perform real-time interactions. These technologies are essential in various applications, such as autonomous navigation, path planning, and exploration. However, challenges remain in generating efficient 3D maps. Traditional single-resolution mapping methods represent all areas with the same resolution, which leads to unnecessary high-resolution storage in noncritical areas, wasting memory resources and increasing computational overhead.

To address these issues, an adaptive resolution

switching approach is required, which enables the precise mapping of specific regions from the moment a particular object is detected. For example, during disaster scenarios or building collapse situations, it is crucial to map areas in greater detail once a person is identified, ensuring an accurate assessment of the situation. Systems that dynamically adjust resolution based on object detection can be highly effective in such contexts.

This paper proposes an efficient adaptive mapping system for detailed mapping of required regions after object detection in the aforementioned scenarios. The primary contribution of this system lies in optimizing memory usage through automatic resolution switching based on object detection. To achieve this, techniques for resolution switching and overlapping point filter-

※ “This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2024-RS-2024-00437190, 50%) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation)”

※ “This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICAN(ICT Challenge and Advanced Network of HRD) program(IITP-2024-RS-2022-00156394, 50%) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation)”

◆ First Author : Kumoh National Institute of Technology, choiye1122@naver.com, 학생회원

◦ Corresponding Author : Kumoh National Institute of Technology ,wdragon@kumoh.ac.kr, 종신회원

논문번호 : 202412-304-E-RU, Received November 29, 2024; Revised December 17, 2024; Accepted December 17, 2024

ing are employed, maximizing memory efficiency while ensuring real-time operation.

The structure of this paper is as follows. Section 2 introduces existing studies related to this research. Section 3 describes the overall structure of the proposed adaptive voxel mapping system, detailing the resolution switching and overlapping point filtering algorithms. Section 4 validates the performance and efficiency of the proposed system through experiments. Finally, Section 5 presents the conclusions of this study and discusses future research directions.

## II. Related Work

### 2.1 3D representation methods

Various methods for representing 3D scenes possess unique characteristics and play a crucial role in enabling autonomous agents to perceive their environment and perform tasks such as navigation and exploration. This paper compares several 3D representation methods and discusses their suitability for robotic applications.

A point cloud<sup>[1]</sup> represents a 3D scene using individual points in space, often generated directly from LiDAR or depth cameras, providing a relatively straightforward way to construct 3D shapes. Each point represents a specific coordinate in space, allowing rapid scene perception due to minimal initial data processing. However, the lack of connectivity between points makes it challenging to represent surface continuity and handling large-scale point data requires significant memory and computational resources.

A mesh<sup>[2]</sup>, composed of triangles or polygons, directly represents the surfaces of objects and is widely used in 3D graphics and virtual reality. Mesh-based methods clearly define the shapes and structures of surfaces, enabling a precise representation of complex geometries. However, generating and maintaining meshes involves high computational and memory costs. Additionally, since each face is treated as an independent entity, this method has limitations in representing occupancy or spatial states.

Surfel<sup>[3]</sup> uses disk-shaped textures to represent surface information, where each surfel encodes local sur-

face properties such as position, normal vector, and color. Surfels are useful for quickly scanning and updating the environment in real time, making them advantageous for exploratory tasks. However, as they only store local surface information, surfels have limitations in representing the interiors of objects or occupancy states.

The 3D Gaussian method<sup>[4]</sup> probabilistically represents the positions and shapes of objects using Gaussian distributions, allowing for the handling of uncertainties in the environment. This approach effectively models simple structures and is suitable for inference tasks based on probability distributions. However, it is better suited for simple geometries and may not adequately capture spatial occupancy or structural details.

The voxel-based representation<sup>[5][6]</sup> divides the 3D space into a grid, storing occupancy or probabilistic information in each cell. This method enables precise spatial occupancy management, making it suitable for robot navigation, path planning, and collision avoidance. Each voxel can store occupancy status or probabilities, allowing robots to quickly assess the state of specific locations in the environment. However, representing high-resolution details requires substantial memory and computational resources, necessitating efficient management strategies.

This study adopts OctoMap<sup>[7]</sup>, a voxel-based occupancy representation managed with an octree structure. OctoMap reduces memory usage and increases data access speed, making it suitable for tasks such as real-time environment mapping, path planning, and collision avoidance in robotic systems.

Other 3D representation methods, such as point clouds, are advantageous for simple structure recognition, but lack connectivity and consume significant memory, limiting their effectiveness for real-time navigation. Meshes are suitable for high-quality visualization, but they are computationally intensive and misaligned with the objectives of the study. Similarly, surfels and 3D Gaussian methods are limited in their ability to represent occupancy states, with 3D Gaussians being particularly resource-intensive for real-time tasks. For these reasons, this study uses OctoMap to efficiently store voxel-based occupancy

information, ensuring suitability for real-time navigation and path planning.

## 2.2 Criteria for applying adaptive resolution

Adaptive resolution is applied on the basis of various criteria in different studies. [8] adjusts the resolution according to the semantic labels and geometric complexity of objects, allowing detailed representation of areas with large or complex structures, while applying lower resolution to relatively simple regions to maintain efficiency.

In [9], the resolution is adjusted based on the distance from the camera, with closer areas represented at higher resolution. This approach focuses resource-intensive computations on nearby regions, making it suitable for tasks that require high levels of detail in real-time operations.

Studies such as [10] and [11] refine the resolution of cells when new sensor measurements deviate from current map estimates. On the contrary, cells that maintain similar states are merged to reduce memory usage. This approach offers flexibility by adjusting the resolution based on measurement consistency and geometric complexity.

This paper proposes an adaptive resolution application method that is triggered from the moment a specific object is detected in a region. Existing adaptive mapping methods based on distance or geometric complexity are effective in various environments but are not optimized for applications requiring precise information in specific regions after detecting critical objects, such as in disaster scenarios or collapsed buildings. The proposed method enhances resolution upon object detection, efficiently providing detailed information for critical areas, while reducing unnecessary computations to avoid resource wastage. This ensures both efficiency and accuracy in real-time mapping, offering high practicality in specialized applications.

## III. System

### 3.1 System Overview

Fig. 1 illustrates the system model, which utilizes three types of data—RGB-D images, grayscale im-

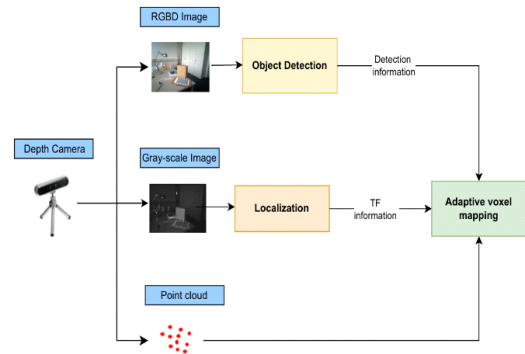


Fig. 1. System model

ages, and point clouds—captured by a depth camera. RGB-D images are used for object detection, while localization<sup>[1]</sup> is performed based on feature points extracted from the grayscale images. Finally, adaptive voxel mapping is conducted using point-cloud data. In this process, adaptive voxel mapping switches the resolution for mapping from the moment a specific object is detected. Additionally, voxel positions are updated accurately using positional information(TF) provided by localization, ensuring an error-free cumu-

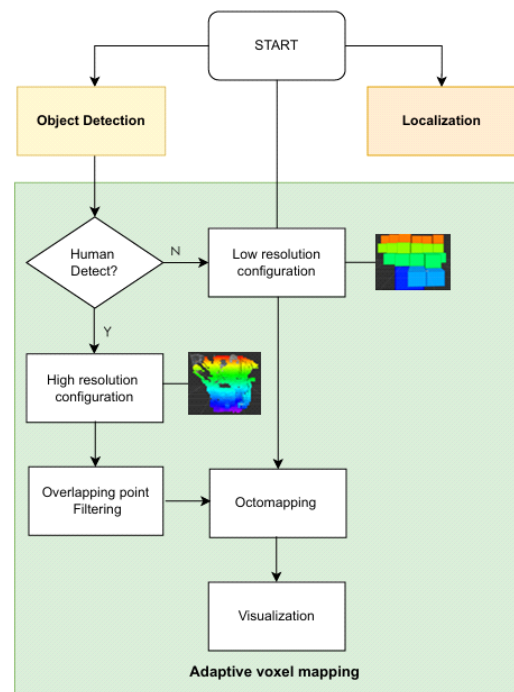


Fig. 2. Adaptive voxel mapping flow chart

lative mapping of the environment.

Fig. 2 illustrates the flow chart of the entire system, explaining the process of adaptive voxel mapping utilizing object detection and camera-based localization. When the system starts, 2D object detection, localization, and low-resolution octomap-based mapping (Octomapping) are performed in parallel. During the object detection process, the detected area is switched to high resolution as soon as a human is detected. Subsequently, the points overlapping with the existing low-resolution map are removed and finally, a unified map integrating both low and high resolutions is visualized.

### 3.2 Adaptive voxel mapping system

#### 3.2.1 Object detection

In this paper, the moment when an object is detected is defined as the criterion for adaptive resolution switching, and the YOLO (You Only Look Once) technique is employed to achieve fast and accurate real-time object detection. YOLO enables efficient detection by simultaneously predicting the location and class of objects in images or videos. Since this study requires the simultaneous execution of localization, mapping and object detection, the lightweight YOLOv8n model<sup>[12]</sup>, which has low computational requirements and high speed, is adopted. Additionally, a pre-trained model based on an open dataset is utilized, and during the inference process, the system is configured to detect only objects of the "person" class. Fig. 3 shows the inference results of object detection using the YOLOv8n model.

#### 3.2.2 Octomap with localization

Although OctoMap effectively manages the occu-



Fig. 3. Object detection result

pancy state of 3D space, it lacks built-in localization functionality, which limits its ability to accurately estimate the robot's position. To address this limitation, this study incorporates camera-based localization. By utilizing image information acquired from the camera, the robot's position and orientation are estimated.

Fig. 4 illustrates the node graph showing the interaction between OctoMap and camera-based localization nodes using a depth camera. In this graph, the topic `/camera/depth/color/points` is subscribed to by the `/octomap_server` node, which publishes the topic `/occupied_cells_vis_array` to visualize the occupied cells in the 3D space. Simultaneously, the `/image_topics` topic from the depth camera is subscribed to by the `/localization` node, where image tracking and position estimation are performed. The measured position information is then transmitted to the `/octomap_server` node via the `/tf` topic, enabling map updates and error correction based on these data.

#### 3.2.3 Resolution switching

Fig. 5 illustrates the resolution switching algorithm and provides a detailed explanation of the transition process from low resolution to high resolution. When

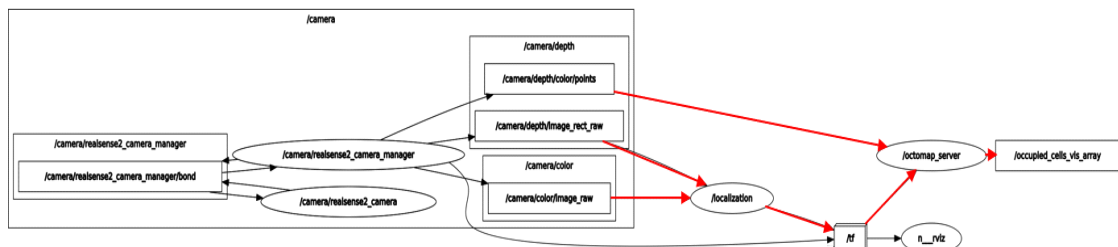


Fig. 4. Octomap with localization

**Algorithm 1** Resolution Switching Algorithm

---

```

1: Input : Point cloud data
2: Output : 3D Occupancy Grid Map

3: Initialize:
4:   resolutionSet  $\leftarrow$  false
5:   lowResOctree and highResOctree  $\leftarrow$   $\emptyset$ 

6: while Point cloud data cloud_msg is received do
7:   Preprocess cloud_msg
8:   if resolutionSet = true then
9:     Update highResOctree with cloud_msg
10:  else
11:    Update lowResOctree with cloud_msg
12:  end if
13:  Publish Octree Visualization
14: end while

15: Upon Object Detection Event:
16: function SWITCHRESOLUTION(resolution)
17:   Reconfigure Octree Parameters:
18:     Update octreeResolution  $\leftarrow$  resolution
19:     Set hit/miss probabilities
20:     Set clamping thresholds
21:     Set tree depth
22:   Initialize highResOctree with Octree Parameters
23:   Set resolutionSet  $\leftarrow$  true
24: end function

```

---

Fig. 5. Resolution switching algorithm

the initial point-cloud data are received, they undergo preprocessing and are then inserted into the low-resolution octree (lowResOctree). During this process, each point is classified as either free space or occupied space, and the state of the octree is visualized in real-time. When an object detection event occurs, the SwitchResolution function is triggered, reconfiguring parameters such as resolution, hit/miss probabilities, clamping thresholds, and tree depth, and initializing a high-resolution octree (highResOctree) based on these updated parameters. After the resolution switch, subsequent point-cloud data are also preprocessed and updated into the high-resolution octree.

## 3.2.4 Overlapping point filtering

The overlapping point filtering in OctoMap is the process of eliminating redundant points between two octrees with different resolutions to ensure non-overlapping adjustments. Fig. 6 illustrates the overlapping point filtering algorithm, where the filtering procedure begins by iterating through each point in the new point cloud data. For each point, a corresponding key is generated in both the high-resolution and low-resolution octrees. The algorithm then searches for a node

**Algorithm 2** Overlap point Filtering Algorithm

---

```

1: Input: processedCloud (preprocessed point cloud data)
2: Output: highResOctree

3: Initialize:
4:   highResKey  $\leftarrow$   $\emptyset$ 
5:   lowResKey  $\leftarrow$   $\emptyset$ 

6: for each point in processedCloud do
7:   Create a key highResKey in highResOctree using point
8:   Create a key lowResKey in lowResOctree using point
9:   Search for a node in lowResOctree using lowResKey
10:  if node is null then
11:    Insert highResKey to highResOctree
12:  end if
13: end for

```

---

Fig. 6. Overlapping Point Filtering Algorithm

in the low-resolution octree that matches the generated key. If the pointer of the retrieved node does not reference a value, this area is considered unoccupied by the low-resolution octree and the current high-resolution key is inserted into the high-resolution octree.

## IV. Experiment

## 4.1 Experimental setup and configuration

In this study, a voxel mapping system based on a 3D occupancy grid map is used to create a low-resolution map in an indoor environment, which switches to high resolution for mapping when a search for a victim is performed. The hardware configuration includes an unmanned ground vehicle (Scout Mini Robot), a depth camera (Realsense 435), and a laptop equipped with a Nvidia Geforce RTX 3060. Fig. 7 shows the Scout Mini Robot platform with all the hardware components used in the experiment



Fig. 7. Hard ware configuration for mapping

attached.

4.2 Experiment result

4.2.1 System operation result

Fig. 8 presents the system operation results, where (a), (b), (c) and (d) depict object detection, robot driving, adaptive mapping, and movement trajectory, respectively. The figure demonstrates that when an object is detected, the mapping transitions from low resolution to high resolution, confirming the proper functioning of the system.

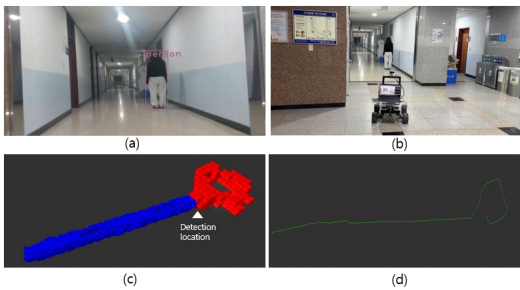


Fig. 8. System operation

4.2.2 Analysis and evaluation of the efficiency of the system

This experiment evaluates the efficiency of the overall map generated through resolution switching by comparing it with the traditional single-resolution mapping method. Using low resolution at 1.5 m and high resolution at 0.5 m as benchmarks, the adaptive resolution is assumed to be a map combining the two resolutions. By comparing the total point cloud counts for the low-resolution, high-resolution, and adaptive resolution mapping methods, data representation efficiency was analyzed. Table 1 and Fig. 9 below present the point cloud counts and the overall maps for each resolution.

The point cloud count increased in the order of low resolution, adaptive resolution, and high resolution, directly affecting the computational load. As the number of point clouds increases, the amount of data to be

Table 1. Comparison of poit cloud number

| Low resolution(1.5m) | Hight resolution(0.5m) | Adaptive resolution |
|----------------------|------------------------|---------------------|
| 641                  | 6415                   | 4225                |

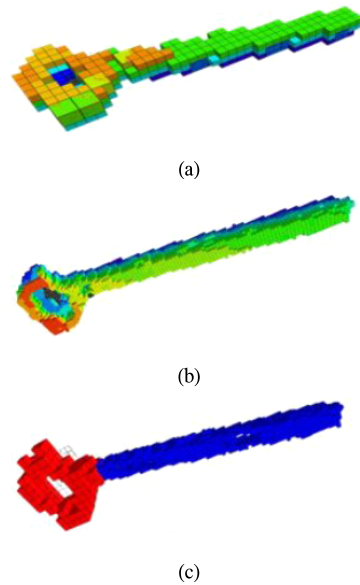


Fig. 9. (a) low resolution map, (b) High resolution map, (c) Adaptive resolution map.

processed also increases, leading to higher computational demands. Therefore, adaptive resolution is efficient as it requires less computation than high resolution while maintaining the necessary level of detail. This result demonstrates that adaptive mapping effectively manages computational resources.

Additionally, during the adaptive voxel mapping process, the ratio of high resolution to low resolution areas was adjusted to various values, as presented in Table 2, to analyze changes in the capacity of the map. This evaluation assessed the ability of adaptive mapping to dynamically allocate resources according to the situation. The analysis revealed a tendency for the map capacity to increase as the proportion of high-resolution areas grew. Fig. 10 visually represents the results of the overall map for each ratio set in Table 2.

Table 2. Comparison of map size based on low and high resolutoin ratios

| Number | Low resolution(1.5m) | Hight resolution(0.5m) | Size(Kb) |
|--------|----------------------|------------------------|----------|
| 1      | 100                  | 0                      | 0.454    |
| 2      | 70                   | 30                     | 1.1      |
| 3      | 50                   | 50                     | 1.4      |
| 4      | 30                   | 70                     | 1.6      |
| 5      | 0                    | 100                    | 1.7      |

## V. Conclusion

This study proposed a voxel mapping system based on adaptive resolution switching to enable autonomous robots to efficiently perceive and map 3D spaces in various environments. The proposed system combines object detection and resolution switching to map only the required regions at high resolution, while filtering overlapping areas to optimize memory usage and computational load.

Experimental results demonstrated that adaptive resolution mapping effectively reduces the total number of point clouds compared to single resolution mapping, thus contributing to reduced memory usage and CPU computation. Notably, the mixed-resolution maps of low and high resolution confirmed the system's ability to dynamically allocate resources, highlighting its flexibility and efficiency.

The proposed system shows great potential for application in various fields that require efficient real-time resource management, such as disaster rescue, logistics robotics, and indoor exploration. Future research will focus on verifying the system's performance in more complex environments and advancing its capabilities for real-time exploration and collaborative performance in autonomous systems.

## References

- [1] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Trans. Robotics*, vol. 33, no. 5, pp. 1255-1262, Dec. 2017.
- [2] A. Romanoni, D. Fiorenti, and M. Matteucci, "Mesh-based 3d textured urban mapping," in *2017 IEEE/RSJ Int. Conf. IROS*, pp. 3460-3466, Vancouver, BC, Canada, Dec. 2017.
- [3] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in *2017 IEEE ICRA*, pp. 4628-4635, Singapore, Jul. 2017.
- [4] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Trans. Graphics*, vol. 42, no. 4, Jul. 2023.
- [5] J. Ryde and H. Hu, "3d mapping with multi resolution occupied voxel lists," *Autonomous Robots*, vol. 28, pp. 169-185, Feb. 2010.
- [6] Tae-Hyun Eom and Woojin Paik, "Computer Vision-Based Object Detection and Depth Estimation to Generate Point Clouds and Estimate Distances" *The transactions of The Korean Institute of Electrical Engineers*, vol. 74, no. 1, pp. 164-169, Jan, 2025, 10.5370/KIEE.2025.74.1.164
- [7] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, pp. 189-206, Feb. 2013.
- [8] J. Zheng, D. Barath, M. Pollefeys, and I. Armeni, "Map-adapt: Real-time quality adaptive semantic 3d maps," in *Eur. Conf. Computer Vision*, 2024.
- [9] E. Vespa, N. Funk, P. H. Kelly, and S. Leutenegger, "Adaptive resolution octree based volumetric slam," in *2019 Int. Conf. 3DV*, pp. 654-662, Quebec City, QC, Canada, Sep. 2019.
- [10] C. Yuan, W. Xu, X. Liu, X. Hong, and F. Zhang, "Efficient and probabilistic adaptive voxel mapping for accurate online lidar odometry," *IEEE Robotics and Automation Lett.*, vol. 7, no. 3, pp. 8518-8525, Jul. 2022.
- [11] E. Einhorn, C. Schröter, and H.-M. Gross, "Building 2d and 3d adaptiveresolution occupancy maps using nd-trees," in *Proc. 55th Int. Sci. Colloquium, Ilmenau, Germany. Verlag ISLE*, vol. 55, pp. 306-311, Nov. 2010.
- [12] G. Jocher, A. Chaurasia, and J. Qiu, *Ultralytics yolov8*, version 8.0.0, 2023, [Online] Available: <https://github.com/ultralytics/ultralytics>

**Yeong Eun Choi**



Feb. 2023 : B.S. degree, Kumoh  
National Institute of Tech-  
nology  
Mar. 2023~Current : M.S. stu-  
dent, Kumoh National  
Institute of Technology

<Research Interests> Autonomous driving, SLAM  
[ORCID:0009-0009-7380-3684]

**Soo Young Shin**



Feb. 1999 : B.S. degree, Seoul  
University  
Feb. 2001 : M.S. degree, Seoul  
University  
Mar. 2010~Current : Professor  
Kumoh National Institute of  
Technology, Gumi, Gyeong-  
sangbuk-do, South Korea

<Research Interests> Wireless communications, Deep  
learning, Machine learning, Autonomous driving  
[ORCID:0000-0002-2526-2395]