Semantic Object Selection and Detection for Diminished Reality based on SLAM with Viewpoint Class

Yoshikatsu Nakajima* Keio University
Shohei Mori† Keio University
Hideo Saito‡ Keio University

ABSTRACT

We propose a novel diminished reality method which is able to (i) automatically recognize the region to be diminished, (ii) work with a single RGB-D sensor, and (iii) work without pre-processing to generate a 3D model of the target scene by utilizing SLAM, segmentation, and recognition framework. Especially, regarding the recognition of the area to be diminished, our method is able to maintain high accuracy no matter how the camera moves by distributing the viewpoints for each object uniformly and aggregating recognition results from each distributed viewpoint as the same weight. These advantages are demonstrated on the UW RGB-D Dataset and Scenes.

Keywords: Diminished Reality, Object Recognition, Convolutional Neural Network, SLAM, Segmentation

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Object recognition

1 INTRODUCTION

In contrast to augmented reality (AR) and mixed reality (MR) which superimpose virtual information on the real scene, diminished reality (DR) is a research field that visually erases real objects [15]. The research on DR originates from Mann *et al.* [14]. Mann *et al.* proposed a method of overwriting unnecessary objects with virtual objects to diminish them. On the other hand, in recent years, DR methods to make scenes without a specific object have become mainstream. Therefore, there are many challenging issues related to the goal (e.g., 6 degree of freedom (DoF) camera pose estimation in arbitrary scenes, detection of objects to be diminished, etc.).

Barnum *et al.* proposed a method [1] that is able to manage dynamic hidden areas by using three cameras, one shooting a hidden background and the other located at the user's viewpoint, assuming the object to be diminish is several plane to work in real-time. Yoshida *et al.* proposed a method [22] to visualize blind spots of the vehicle using multiple cameras installed in a vehicle. Although these methods are able to cope with dynamic hidden areas, as they use a plurality of sensors as inputs, the costs are high and the application destination is limited. Therefore, many methods for achieving DR with one camera have also been proposed. The method proposed by Mori *et al.* [16] generate a three-dimensional (3D) model of a target scene to synthesize a hidden background image, estimate the camera pose with respect to the 3D model, and project the 3D model to the user's viewpoint in real-time. To generate a hidden background image, [23] does not require a 3D model of the target scene as pre-processing, however, DR is achieved by assuming the shape of the hidden area as a plane, so the scenes to be managed are limited.

The identification of areas to be diminished and their tracking are also major technical issues in DR. As represented by the method of Kawai *et al.* [6], many DR methods prompt the user to border the region of interest in the initial frame, and track the region using a tracking method, such as SLAM and feature-based tracking in subsequent frames. Cosco *et al.* proposed a method for diminishing a haptic device so the user does not need to draw the hidden area [2]. However, since this method requires the preparation of the 3D model of the hidden object, it cannot manage unknown objects.

To solve the problems of the conventional method described above, we propose a novel DR method that has the following three advantages with respect to conventional methods:

- Category-based automatic object selection to be diminished rather than dragging-based user input,
- Progressive background reconstruction in arbitrary scenes without pre-learning of the scenes, and
- Working with a single RGB-D sensor as an input.

Hidden background image generation is performed by expanding a part of the SLAM framework by utilizing the 3D model, sequentially reconstructed by the SLAM framework, in which the object categories are labeled by the segmentation framework. The diminished result is generated by integrating the hidden background image and the input RGB image by specifying the hidden area based on the final recognition result obtained through the recognition framework.

Especially, to detect and track the area to be diminished accurately, we propose a novel recognition method that has the following advantages by distributing the viewpoints for each object uniformly and aggregating the recognition results from each distributed viewpoint as the same weight: **1.** working in real-time while processing SLAM, segmentation, and object recognition, **2.** managing smooth-surfaced objects and a large number of categories, and **3.** maintaining high accuracy regardless of the motion of the camera.

Our recognition method equally divide the viewpoint of each object (see Figure 2, upper right) while maintaining the computational complexity of $O(n^2)$ (i.e. the size of the input image). We call each divided viewpoint a Viewpoint Class (i.e., each small sphere distributed around each segmented object in Figure 3). Only when the object is observed from a new Viewpoint Class where the object is not yet recognized from, we crop the region of the object from a current frame to input the cropped images into a trained CNN for feature extraction. The aim of this procedure is to improve the final recognition accuracy by avoiding repeating the recognition computation when the camera stops at a poor view direction. As a secondary effect, the processing time is reduced by limiting the number of times to input the region to a CNN. Therefore, there is no trade-off between accuracy and real-time in this method. Furthermore, by utilizing a CNN as a tool for feature extraction, high scalability is achieved. In our method, any CNN structure that takes one input image and outputs its category can be used, so that the

*e-mail: nakajima@hvrl.ics.keio.ac.jp
†e-mail:mori@ieee.org
‡e-mail:hs@keio.jp

978-0-7695-6327-5/17 $31.00 © 2017 IEEE
DOI 10.1109/ISMAR-Adjunct.2017.98

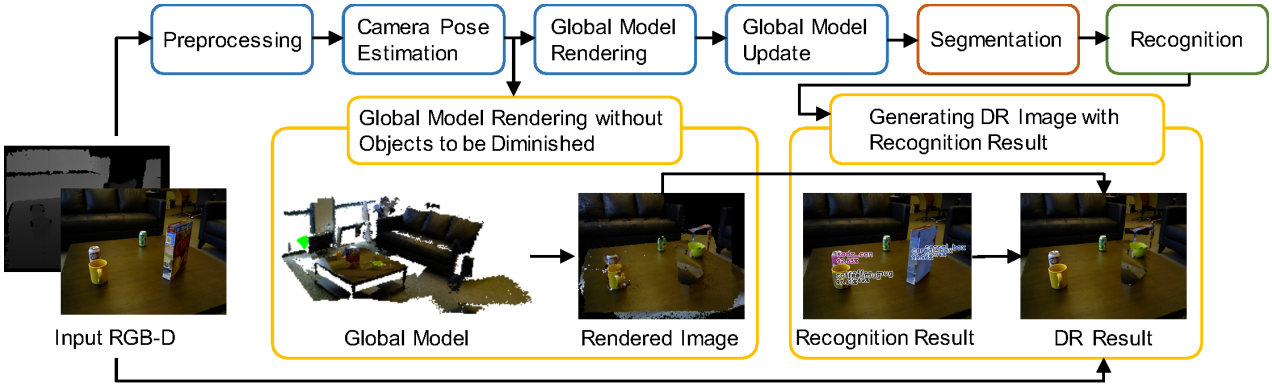

Figure 1: Flow of the proposed method (Depicted in blue: SLAM Part, red: segmentation part, green: recognition part, and yellow: DR part)

range of application of this method is widened to consider various kinds of datasets for a CNN and trained CNN models are provided recently [3, 8].

2 METHOD

In this section, we describe our proposed method, which simultaneously processes reconstruction, segmentation, object recognition, and DR. Figure 1 shows a flow diagram of the proposed method. Our method consists of four parts (depicted in blue: SLAM, red: segmentation, green: recognition, and yellow: DR). After an overview of the SLAM and segmentation parts, we describe in detail the recognition part, which is one of the main contribution of this work. Then, we explain in detail the DR part, which is the core of this work. The inputs are simply RGB and depth images obtained from a moving RGB-D sensor, which were processed individually.

2.1 SLAM Part

This section provides an overview of the SLAM part (see Figure 1, depicted in blue).

We employ the SLAM system proposed by Keller *et al.* [7] because a global model, which is a model reconstructed through the SLAM framework, consists only of point clouds. Thus, it can manage a wider environment compared to voxel-based methods, including KinectFusion [5]. Each point s_k of a global map $\mathcal{S}$ has information including a 3D position $\boldsymbol{v}_k \in \mathbb{R}^3$, a normal $\boldsymbol{n}_k \in \mathbb{R}^3$, a confidence $c_k \in \mathbb{R}$, and a time stamp $t_k \in \mathbb{N}$.

The Pre-processing Stage is for smoothing a depth image $\mathcal{D}_t$ at current frame t with a bilateral filter [21] and transforming $\mathcal{D}_t$ into a vertex map $\mathcal{V}_t(\boldsymbol{u}) = \boldsymbol{K}^{-1}\dot{\boldsymbol{u}}\mathcal{D}_t(\boldsymbol{u})$ using the camera intrinsic parameter $\boldsymbol{K}$, a depth map element $\boldsymbol{u} = (x, y)^{\mathrm{T}}$ in the image domain $\boldsymbol{u} \in \mathbb{R}^2$, and its homogeneous coordinate $\dot{\boldsymbol{u}}$. The normal map $\mathcal{N}_t$ is also generated in this stage by using a cross-product calculation to $\mathcal{V}_t$.

The Camera Pose Estimation Stage is for calculating the current camera pose $\boldsymbol{T}_t = [\boldsymbol{R}_t, \boldsymbol{t}_t] \in \mathbb{SE}(3)$, $\boldsymbol{R}_t \in \mathbb{SO}(3)$, and $\boldsymbol{t}_t \in \mathbb{R}^3$ by using $\mathcal{V}_t$, $\mathcal{V}^m_{t-1}$, and $\mathcal{N}^m_{t-1}$. At this time, we denote the rendered map of the global model with respect to a particular camera pose as m. The point-to-plane ICP algorithm proposed by Low [13] takes these three maps and outputs a rotation and translation between the current frame and the previous frame.

The Global Model Rendering Stage is for obtaining the correspondences between the point clouds generated by the current depth map $\mathcal{D}_t$ and the global model $\mathcal{S}$. The index map $\mathcal{L}$ is generated in this stage by projecting point clouds from the global map via the projection matrix $\boldsymbol{P}_t$, which consists of the current camera pose $\boldsymbol{T}_t$ and the intrinsic parameters $\boldsymbol{K}$. $\mathcal{V}^m_t$ and $\mathcal{N}^m_t$ are also generated at this stage for the "Camera Pose Estimation" stage in the next frame.

The Global Model Update Stage is for merging or adding the point clouds generated from the current depth map $\mathcal{D}_t$ to the global model. Only when specific geometric conditions are satisfied is the point $\mathcal{D}_t(\boldsymbol{u})$ merged to a point s_k already present on the global map $\mathcal{S}$, and the associated confidence c_k is incremented.

2.2 Segmentation Part

This section provides an overview of the segmentation part (see Figure 1, depicted in red), which determines the object targeted for object recognition. The segmentation part in the proposed method consists of four stages, and this section outlines each.

We employed the segmentation framework based on the method by Tateno *et al.* [19]. It takes the current depth map $\mathcal{D}_t$ to incrementally build up and to update a Global Segmented Map (GSM) $\mathcal{L}$ for each frame. The components of the GSM are the same as those for the global map $\mathcal{S}$, and each point on the GSM is labeled. The main advantage of this system, and our reason for employing it, is that the computational cost for updating a GSM never increases, as with other segmentation systems [4].

The Depth Map Segmentation Stage is for segmenting the inputted depth map $\mathcal{D}_t$ by conducting a normal edge analysis. The process takes the vertex map $\mathcal{V}_t$ and normal map $\mathcal{N}_t$ as inputs and a binary edge map $\mathcal{L}_t$ is outputted by comparing the nearby normal angles and vertex distances. Then, a connected component algorithm is applied to the binary map to obtain a label map $\mathcal{L}_t$ on which each element $\mathcal{L}_t(\boldsymbol{u})$ is associated with l_j.

The Segment Label Propagation Stage is for generating a propagated label map $\mathcal{L}^p_t$, where each element $\mathcal{L}^p_t(\boldsymbol{u})$ is associated with a label on the GSM. To achieve this goal, first, the rendered label map $\mathcal{L}^m_t$ is computed by projecting the GSM with $\boldsymbol{P}_t$, which was created in the "Camera Pose Estimation Stage". Next, the overlap percentage between $l_i \in \mathcal{L}^m_t$ and $l_j \in \mathcal{L}_t$ is computed and used to decide whether l_i is propagated to $\mathcal{L}^p_t$ or l_j to be used directly. Finally, a propagated label map $\mathcal{L}^p_t$ of a current frame t (see Figure 2, left bottom) is obtained.

The Segment Merging Stage is for merging segments that originally consisted of the same object. When the overlapped percentage of $l_a, l_b \in \mathcal{L}^m_t$, calculated in the "Segment Label Propagation Stage", is sufficiently larger than the threshold, the segment pair (l_a, l_b) is merged and replaced with l_a.

The Segment Update Stage is for updating the GSM with $\mathcal{L}^p_t$. The labels of each point in the GSM are updated only when their label confidence is over the threshold; otherwise, only label confidence changes.

Figure 2: Actual image of Viewpoint Class uniformly distributed around each segmented object in the Global Segment Map (GSM). Green pyramids represent the camera trajectory up to the current frame t. The Viewpoint Class colored in red is the one from which the object has already been recognized. Left side, top to bottom: input RGB image, normal map $\mathscr{N}_t$, propagated label map $\mathscr{L}_t^p$.

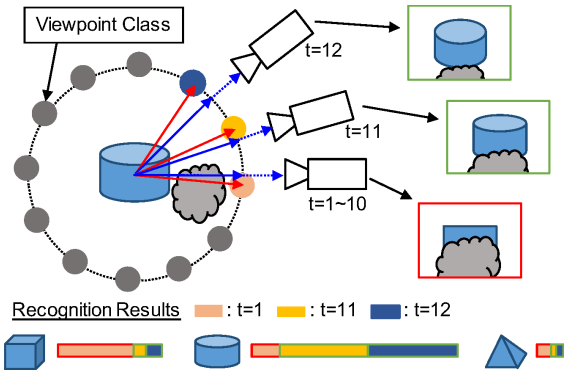

Figure 3: The concept of our Viewpoint Class based recognition system. Each circle uniformly distributed around the object indicates a Viewpoint Class. Since the recognition results from each Viewpoint Class (i.e., CNN outputs at $t = 1, 11, 12$) are aggregated as the same weight, even if the camera idles in a bad position ($t = 1 \sim 10$), the accuracy of the recognition result increases in the end.

2.3 Recognition Part

This section provides an overview of the recognition part (see Figure 1, depicted in green), which determines the object area to be diminished. The recognition part consists of three stages, and this section explains each stage in detail.

As shown in Fig. 3, one of the main contributions of this work is equally dividing the viewpoint around each object in the GSM and impartially merging the recognition results from each divided viewpoint (Viewpoint Class) with the same weight to detect and track the area to be diminished accurately. To achieve this goal, in contrast to [19], each segmented object $\mathscr{O}_j$ has information about its centroid $\boldsymbol{C}_j \in \mathbb{R}^3$ for placing the Viewpoint Class centered on the centroid $\boldsymbol{C}_j$. Viewpoint Class generation is performed only once before the initial frame as a pre-processing step. N points can be distributed uniformly over the surface of a sphere whose radius r is 1 with the following equations [18]. We store each coordinate $\boldsymbol{\psi}_\gamma \in \mathbb{R}^3$ that is generated by converting θ_γ and ϕ_γ into xyz coordinates.

$$\theta_\gamma = arccos(h_\gamma), h_\gamma = -1 + \frac{2(\gamma-1)}{(N-1)}, 1 \leq \gamma \leq N,$$
$$\phi_\gamma = (\phi_{\gamma-1} + \frac{3.6}{\sqrt{N}} \frac{1}{\sqrt{1-h_\gamma{}^2}})(mod 2\pi), 2 \leq \gamma \leq N-1, \phi_1 = \phi_N = 0 \quad (1)$$

2.3.1 Viewpoint Class Judgment

The objective of this stage is to determine whether the current camera pose belongs to a new Viewpoint Class for each object $\mathscr{O}_j$. We perform the following processing for each object appearing on the propagated label map $\mathscr{L}_t^p$.

First, we compute the vector $\boldsymbol{V}_j^{ct}$ starting at the centroid $\boldsymbol{C}_j$ and ending at the current camera position $\boldsymbol{t}_t$ in world coordinates with $\boldsymbol{V}_j^{ct} = \boldsymbol{t}_t - \boldsymbol{C}_j$ for each object $\mathscr{O}_j$. At this time, the current camera position $\boldsymbol{t}_t$ is already computed in the Camera Pose Estimation Stage. Next, the vector $\boldsymbol{V}_j^{ct}$ is normalized to length r. Considering that each prepared Viewpoint Class $\boldsymbol{\psi}_\gamma$ is distributed on a sphere whose center is the origin of the coordinate, we can determine the Viewpoint Class to which the current camera pose belongs by comparing vectors $\boldsymbol{\psi}_\gamma$ and $\boldsymbol{V}_j^{ct}$. Thus, the γ that minimizes the distance between $\boldsymbol{\psi}_\gamma$ and $\boldsymbol{V}_j^{ct}$ is the Viewpoint Class to which the current camera pose belongs.

$$\bar{\gamma}_j = \underset{1 \leq \gamma \leq N}{\operatorname{argmin}} \| \boldsymbol{\psi}_\gamma - \boldsymbol{V}_j^{ct} \| \quad (2)$$

We denote the Viewpoint Class as $\bar{\gamma}_j$. We denote the recognition result of an object $\mathscr{O}_j$ from Viewpoint Class γ as Ω_{γ_j}. If the recognition result $\Omega_{\bar{\gamma}_j}$ is empty, the object O_j is recognized in the next stage and its index is denoted as $\hat{j}$.

2.3.2 Recognition with CNN

After the objects recognized in this stage are determined, segments of each object $\mathscr{O}_{\hat{j}}$ in the RGB image of the current frame are cropped based on the propagated label map $\mathscr{L}_t^p$. Next, these images are input into the convolutional neural network (CNN) tuned by deep learning with a specific dataset (e.g. ImageNet [3, 8]). At this time, the softmax function is not applied to the output of the CNN because merging the outputs of the CNN from each Viewpoint Class and calculating the probability of what each object are performed in the next stage. Therefore, the raw output of the CNN is stored as $\Omega_{\bar{\gamma}_j}$.

2.3.3 Merging the Recognition Results

To recognize each object, the recognition results are merged and renewed for each object $\mathscr{O}_{\hat{j}}$ with the following equation, where ψ_j^r represents a subset of Viewpoint Classes from which the object $\mathscr{O}_j$ has already been recognized.

$$y_{\hat{j}}^{\lambda} = \frac{exp\left(\sum_{\gamma_j \in \psi_j^r} \Omega_{\gamma_j}(\lambda)\right)}{\sum_{i=1}^{i=\Lambda} exp\left(\sum_{\gamma_j \in \psi_j^r} \Omega_{\gamma_j}(i)\right)} \quad (3)$$

With this equation, the probability $y_{\hat{j}}^{\lambda}$ that the object $\mathscr{O}_{\hat{j}}$ categorized to λ is calculated with $\psi_{\hat{j}}^r$, the total number of categories Λ, and $\Omega_{\gamma_j}(i)$, which denotes the CNN output of category i from a Viewpoint Class γ_j of an object $\mathscr{O}_j$.

2.4 DR Part

In this section, we describe in detail the DR part, which is the main contribution of this work. In Figure 1, the part relating to DR is shown in yellow. In the "Global Model Rendering without Objects

Table 1: Precision/Recall rate using the UW RGB-D Scene Dataset [9, 10]

Method	View(s)	Input	Precision/Recall						
			Bowl	Cap	Cereal Box	Coffee Mug	Soda Can	Background	Overall
DetOnly [11]	Single	RGB	46.9/90.7	54.1/90.5	76.1/90.7	42.7/74.1	51.6/87.4	98.8/93.9	61.7/87.9
Det3DMRF [11]	Multiple	RGB-D	91.5/85.1	90.5/91.4	93.6/94.9	90.0/75.1	81.5/87.4	99.0/99.1	91.0/88.8
HMP2D+3D [9]	Multiple	RGB-D	97.0/89.1	82.7/99.0	96.2/99.3	81.0/92.6	97.7/98.0	95.8/95.0	90.9/95.6
BoVW+FLAIR [17]	Multiple	RGB	88.7/70.2	99.4/72.0	95.6/84.3	80.1/64.1	89.1/75.6	96.6/96.8	89.8/72.0
Ours	Multiple	RGB	96.2/91.8	92.2/95.9	98.4/96.1	91.9/87.1	91.7/89.3	94.0/100.0	94.1/93.4

to be Diminished" stage, we create the hidden background image necessary for generating the diminished result by extending part of the SLAM framework. In the "Generating DR Image with Recognition Result" stage, we identify the object area to be diminished in the input image using the recognition result obtained through the recognition framework and we superimpose the hidden background image generated at the "Global Model Rendering without Objects to be Diminished" stage on the object area. This section describes the details of these two stages.

2.4.1 Global Model Rendering without Objects to be Diminished

As shown in Figure 1, the purpose of this stage is to generate a hidden background image superimposed on the object area that is to be diminished and specified through the recognition result of the recognition framework. In this proposed method, a hidden background in the user view is recovered based on the global map $\mathscr{S}$. We project point clouds within a region labeled with a certain object to generate a corresponding hidden background image.

More specifically, when projecting each point s_k constituting the global map $\mathscr{S}$ during the "Global Model Rendering" stage in the SLAM framework, a hidden background image of the same size as the input image $\mathscr{B}$ is generated according to the following procedure. At this time, let $\boldsymbol{u}_{s_k}$ be the image coordinate at which the point s_k is projected.

1. When $\boldsymbol{u}_{s_k}$ is not inside the hidden background image $\mathscr{B}$ or when the label of the point s_k is an object category to be diminished, the processing is discarded and shifted to the next point s_{k+1} constituting the global map $\mathscr{S}$.

2. When $\mathscr{B}(\boldsymbol{u}_{s_k})$ is empty, fill $\mathscr{B}(\boldsymbol{u}_{s_k})$ with the pixel value of point s_k.

3. When $\mathscr{B}(\boldsymbol{u}_{s_k})$ is not empty, $\mathscr{B}(\boldsymbol{u}_{s_k})$ is filled with the pixel value of the point s_k only when one of the following three conditions is satisfied.

    - If the label of the point filling $\mathscr{B}(\boldsymbol{u}_{s_k})$ is 0 and the label of s_k is not 0.
    - If both the label of the point filling $\mathscr{B}(\boldsymbol{u}_{s_k})$ and the label of s_k are 0 and the Euclidean distance between the vertex coordinates of the point filling $\mathscr{B}(\boldsymbol{u}_{s_k})$ and the camera position $\boldsymbol{t}_t$ at current frame t is greater than the Euclidean distance between the vertex coordinates of the point s_k and the camera position $\boldsymbol{t}_t$.
    - If both the label of the point filling $\mathscr{B}(\boldsymbol{u}_{s_k})$ and the label of s_k are **NOT** 0 and the Euclidean distance between the vertex coordinates of the point filling $\mathscr{B}(\boldsymbol{u}_{s_k})$ and the camera position $\boldsymbol{t}_t$ at current frame t is greater than the Euclidean distance between the vertex coordinates of the point s_k and the camera position $\boldsymbol{t}_t$.

Here, the point where the label is 0 is a point on the edge. Therefore, there is a possibility that the point constituting the object to be diminished exists in points that are labeled 0. In the above procedure, filling in the hidden background image $\mathscr{B}$ is given the highest priority. Next, when the point assigned with a label other than 0 fills $\mathscr{B}$, that point is not replaced with the point assigned with label 0. Lastly, the above procedure gives lowest priority to filling $\mathscr{B}$ with a point close to camera position $\boldsymbol{t}_t$ in the current frame.

Although the SLAM framework of this proposed method employs dense SLAM, it does not guarantee that all pixels of a hidden background image $\mathscr{B}$ will be filled. Therefore, in this method, we apply the median filter to the hidden background image $\mathscr{B}$ generated by the above procedure and render it the final hidden background image $\tilde{\mathscr{B}}$.

2.4.2 Generating DR Image with Recognition Result

In this stage, as shown in Figure 1, the purpose is to generate diminished result $\mathscr{I}_{DR}$ by utilizing the hidden background image $\tilde{\mathscr{B}}$ generated through the "Global Model Rendering without Objects to be Diminished" stage, the propagated label map $\mathscr{L}_t^p$, and the final recognition result of each object $\mathscr{O}_j$ included in the propagated label map $\mathscr{L}_t^p$ obtained through the recognition framework. Here, seen in Figure 2, the propagated label map $\mathscr{L}_t^p$ contains the region labeled 0, which is shown in black. Therefore, we reduce the area whose label is 0 and generate the label map $\tilde{\mathscr{L}}_t^p$, in which the area to be diminished is dilated by performing the dilation processing of 8-neighborhood twice on a label whose recognition result category is the same as the category to be diminished. Next, each pixel $\mathscr{I}_{DR}(\boldsymbol{u})$ of the diminished result $\mathscr{I}_{DR}$ is filled by the following procedure:

1. When the recognition result category of the label $\tilde{\mathscr{L}}_t^p(\boldsymbol{u})$ is the category to be diminished, $\mathscr{I}_{DR}(\boldsymbol{u})$ is filled with $\tilde{\mathscr{B}}(\boldsymbol{u})$.

2. When the recognition result category of the label $\tilde{\mathscr{L}}_t^p(\boldsymbol{u})$ is **NOT** the category to be diminished, $\mathscr{I}_{DR}(\boldsymbol{u})$ is filled with $\mathscr{I}(\boldsymbol{u})$.

3 Experiments

In this section, we demonstrated experimentally the validity of our method. In our experiments, we evaluated our method using the popular UW RGB-D Dataset (v2) [9, 10]. The followings are the details of the evaluation environment: CPU: Intel Core i7-6950X 3.00 GHz, GPU: GeForce GTX 1080, and RAM: 125.8 GB. The deep learning framework used in this evaluation experiment was Chainer [20]. Throughout the experiment, the number of Viewpoint Classes was 700. The CNN model used in this experiment was Network In Network (NIN) [12]. Since the UW RGB-D Dataset provides mask images, we masked the region for each object on each training image. Next, we trained the CNN model by randomly rescaling and adding noise for robust predictions.

3.1 Results

Table 1 shows the mean-Average Precision (mAP) estimates of our method and the existing methods reported in [9, 11, 17]. As shown in Table 1, we were able to achieve a performance of 94.1 mAP as

Figure 4: DR results in several frames (object category to be diminished: "Cereal Box", upper stage, left to right: diminished result $\mathscr{I}_{DR}$, recognition result, lower stage, left to right: input RGB $\mathscr{I}$, hidden background $\tilde{\mathscr{B}}$, propagated label map $\tilde{\mathscr{L}}_t^p$, global map $\mathscr{S}$, GSM $\mathscr{L}$)

compared to the detector performance of 61.7 mAP and the SLAM-aware BoVW+FLAIR performance of 89.8 mAP. Therefore, we could detect and track objects to be diminished accurately through recognition based on Viewpoint Class and CNN.

Figure 4 shows diminished results in each frame of UW RGB-D Dataset when the category to be diminished is "Cereal Box". As can be seen from Figure 4, in the first half frame, since the 3D model of the hidden area was not reconstructed, a part of the hidden background image $\tilde{\mathscr{B}}$ and the diminished result $\mathscr{I}_{DR}$ were incomplete. However, as the frame progressed, the 3D model was generated densely and completely in the hidden area. Also, by accurately estimating the camera pose with the SLAM framework, the borders of the diminished area and the non-diminished area are geometrically indistinguishable with each other in the diminished result $\mathscr{I}_{DR}$. On the other hand, as described in section **2.4.1 Global Model Rendering without Objects to be Diminished**, the procedure was designed to preferentially eliminate the point with a label 0, which consists edge of each object from the hidden background image $\tilde{\mathscr{B}}$. However, the edge is still included in the hidden background image $\tilde{\mathscr{B}}$, and its removal is a future task.

Table 2: Average processing time for each stage of the recognition and DR parts

	(Unit: ms)
Viewpoint Class Judgment	1.0
Recognition with CNN	98.9
Recognition Result Merging	10.7
Global Model Rendering without Objects to be Diminished	32.5
Generating DR Image with Recognition Result	6.1
Total	149.2

Table 2 shows the processing time for each stage of the recognition and DR parts. The average processing time for the Recognition with CNN stage of the proposed method achieved real-time, because only the cropped image of the object whose recognition result from the current Viewpoint Class is empty is recognized in the Recognition with CNN stage. Thus, the number of images inputted to the CNN was decreased and we could reduce the processing time. Considering that the SLAM and segmentation part achieved 72 fps [19], our system would work in real-time.

In addition, the proposed method generates a hidden background based on the 3D model reconstructed by the SLAM framework. Therefore, since the 3D structure of the hidden background is known, the transparency of the object to be diminished can be adjusted. Figure 5 shows diminished results when the hidden background image $\tilde{\mathscr{B}}$ is generated by the following procedure instead of the procedure described in section **2.4.1 Global Model Rendering without Objects to be Diminished**.

1. When $\boldsymbol{u}_{s_k}$ is not inside the hidden background image $\mathscr{B}$ or when the label of the point s_k is an object category to be diminished, the processing is discarded and shifted to the next point s_{k+1} constituting the global map $\mathscr{S}$.

2. When $\mathscr{B}(\boldsymbol{u}_{s_k})$ is empty, fill $\mathscr{B}(\boldsymbol{u}_{s_k})$ with the pixel value of point s_k.

3. When $\mathscr{B}(\boldsymbol{u}_{s_k})$ is **NOT** empty and the Euclidean distance between the vertex coordinates of the point filling $\mathscr{B}(\boldsymbol{u}_{s_k})$ and the camera position $\boldsymbol{t}_t$ at current frame t is less than the Euclidean distance between the vertex coordinates of the point s_k and the camera position $\boldsymbol{t}_t$, fill $\mathscr{B}(\boldsymbol{u}_{s_k})$ with the pixel value of point s_k.

The above procedure gives top priority to filling in the hidden background image $\mathscr{B}$, then to satisfying $\mathscr{B}$ at a point further from the camera position in the current frame. As shown in Figure 5, especially in the latter frame, the area to be diminished and the desk behind it are diminished, and furthermore, based on the camera pose estimated by the SLAM framework, the couch and the floor were naturally superimposed.

Figure 5: DR results in several frames when the transparency is adjusted (object category to be diminished: "Cereal Box", upper stage, left to right: diminished result $\mathscr{I}_{DR}$, recognition result, lower stage, left to right: input RGB $\mathscr{I}$, hidden background $\tilde{\mathscr{B}}$, propagated label map $\tilde{\mathscr{L}}_t^p$, global map $\mathscr{S}$, GSM $\mathscr{L}$)

4 Conclusion

In this work, we proposed a SLAM, segmentation, and recognition-based DR method that would achieve real-time processing, auto-recognition of the region to be diminished, and work without pre-processing to generate a 3D model of the target scene. We leveraged a state-of-the-art SLAM-based segmentation method and utilized a CNN to determine the area to be diminished and to generate the hidden background image. Furthermore, by distributing Viewpoint Classes uniformly around each object and aggregating the recognition results from each Viewpoint Class, robustness for camera movement was achieved. These contributions of our method were demonstrated through experiments using the UW RGB-D Dataset and Scenes.

Acknowledgements

This research presentation is supported in part by a research assistantship of a Grant-in-Aid to the Program for Leading Graduate School for "Science for Development of Super Mature Society" from MEXT in Japan.

References

[1] P. Barnum, Y. Sheikh, A. Datta, and T. Kanade. Dynamic seethroughs: Synthesizing hidden views of moving objects. In *8th IEEE International Symposium on Mixed and Augmented Reality, 2009.*, pages 111–114. IEEE, 2009.

[2] F. I. Cosco, C. Garre, F. Bruno, M. Muzzupappa, and M. A. Otaduy. Augmented touch without visual obtrusion. In *8th IEEE International Symposium on Mixed and Augmented Reality, 2009*, pages 99–102. IEEE, 2009.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[4] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[5] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pages 559–568. ACM, 2011.

[6] N. Kawai, T. Sato, and N. Yokoya. Diminished reality based on image inpainting considering background geometry. *IEEE Transactions on Visualization and Computer Graphics*, 22(3):1236–1247, 2016.

[7] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *International Conference on 3DTV-Conference*, pages 1–8. IEEE, 2013.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[9] K. Lai, L. Bo, and D. Fox. Unsupervised feature learning for 3d scene labeling. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3050–3057, 2014.

[10] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1817–1824. IEEE, 2011.

[11] K. Lai, L. Bo, X. Ren, and D. Fox. Detection-based object labeling in 3d scenes. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1330–1337. IEEE, 2012.

[12] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[13] K.-L. Low. Linear least-squares optimization for point-to-plane icp surface registration. *Chapel Hill, University of North Carolina*, 4, 2004.

[14] S. Mann and J. Fung. Videoorbits on eye tap devices for deliberately diminished reality or altering the visual perception of rigid planar patches of a real world scene. *EYE*, 3:P3, 2001.

[15] S. Mori, S. Ikeda, and H. Saito. A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects. *IPSJ Transactions on Computer Vision and Applications*, 9(1):17, Jun 2017.

[16] S. Mori, F. Shibata, A. Kimura, and H. Tamura. Efficient use of textured 3d model for pre-observation-based diminished reality. In *IEEE International Symposium on Mixed and Augmented Reality Workshops (ISMARW)*, pages 32–39. IEEE, 2015.

[17] S. Pillai and J. Leonard. Monocular slam supported object recognition. *arXiv preprint arXiv:1506.01732*, 2015.

[18] E. B. Saff and A. B. Kuijlaars. Distributing many points on a sphere. *The mathematical intelligencer*, 19(1):5–11, 1997.

[19] K. Tateno, F. Tombari, and N. Navab. Real-time and scalable incremental segmentation on dense slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4465–4472. IEEE, 2015.

[20] S. Tokui, K. Oono, S. Hido, and J. Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

[21] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision*, pages 839–846. IEEE, 1998.

[22] T. Yoshida, K. Jo, K. Minamizawa, H. Nii, N. Kawakami, and S. Tachi. Transparent cockpit: Visual assistance system for vehicle using retro-reflective projection technology. In *IEEE Virtual Reality Conference*, pages 185–188. IEEE, 2008.

[23] S. Zokai, J. Esteve, Y. Genc, and N. Navab. Multiview paraperspective projection model for diminished reality. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 217–226. IEEE, 2003.