virtual worlds

MDPI

Article

Real-Time Diminished Reality Application Specifying Target Based on 3D Region

Kaito Kobayashi * and Masanobu Takahashi

Graduate School of Science and Engineering, Omiya Campus, Shibaura Institute of Technology, Saitama 337-8570, Japan
* Correspondence: mf22055@shibaura-it.ac.jp

Abstract: Diminished reality (DR) is a technology in which a background image is overwritten on a real object to make it appear as if the object has been removed from real space. This paper presents a real-time DR application that employs deep learning. A DR application can remove objects inside a 3D region defined by a user in images captured using a smartphone. By specifying the 3D region containing the target object to be removed, DR can be realized for targets with various shapes and sizes, and the specified target can be removed even if the viewpoint changes. To achieve fast and accurate DR, a suitable network was employed based on the experimental results. Additionally, the loss function during the training process was improved to enhance completion accuracy. Then, the operation of the DR application at 10 fps was verified using a smartphone and a laptop computer.

Keywords: diminished reality; AR; AI; deep learning; generative adversarial network

1. Introduction

Diminished reality (DR) is a technology that creates the illusion of objects being removed from real space. In contrast to augmented reality (AR), which overlays virtual objects onto real objects to make them appear as if they exist in real space, DR displays a background image over the real object to make it appear as if the object has been removed.

DR is expected to have various applications, such as removing the current scene to recreate a past or future scene and removing obstacles like walls and car bodies to enable safety confirmation from previously impossible viewpoints [1,2].

Moreover, by removing such obstacles, DR has the potential to address a challenge in AR where virtual objects cannot be placed due to real-world obstacles. For this purpose, our research aimed to realize a DR application that displays a scene without a target object in the image in real time using the camera of a smartphone. The real-time DR application is also expected to have various uses, such as assisting in interior simulation and live streaming.

Focusing on the method of obtaining the background to hide the target object, DR methods can be classified into two types: observation-type methods that observe the real background and completion-type methods that estimate the background from the region around the target object [1]. The observation-type method is a DR method that has been used for a long time. This method obtains real background images from multiple viewpoints and past images and uses these images to perform DR. On the other hand, the completion-type method performs DR by guessing scenes in which the object does not exist. In many cases of this method, the object is filled in once and a scene is created in which the object is not present by guessing the background image of the filled-in region. The completion-type method performs DR by removing the object from the scene as a result of completing a filled-in region in the scene.

An example of an observation-type method is the removal of obstructive objects, such as pipes, in a video stream [3]. Another example is using the difference in viewpoints between a surveillance camera and a camera on a mobile device to enable a perspective

check for updates

Citation: Kobayashi, K.; Takahashi, M. Real-Time Diminished Reality Application Specifying Target Based on 3D Region. *Virtual Worlds* **2024**, *3*, 115–134. https://doi.org/10.3390/virtualworlds3010006

Academic Editor: Anton Nijholt

Received: 1 December 2023
Revised: 6 February 2024
Accepted: 29 February 2024
Published: 4 March 2024

view of a building [4]. Furthermore, a study of a system to reduce car accidents has been reported [5]. This system transmits camera information from the car in front to the car behind it and removes the car in front from the scene that is viewed from the car behind. In addition, an example has been reported that realizes fast and accurate DR using a 3D model for DR created by observing the background and surrounding environment without the target objects in advance [6]. Observation-type methods have many requirements, such as advanced preparation and special equipment. Also, they cannot cope with cases where the real background is unobservable. Therefore, this method cannot perform DR in real time. For this reason, observation-type methods are not suitable for DR applications, which are the focus of this research.

Studies using both observation-type and completion-type methods have also been reported [7–9]. In these studies, the observation-type method used builds a 3D model of a landscape without objects that need to be removed by moving the camera. The traditional completion-type method used compensates for unobservable regions using PixMix [10] or other methods that enable inpainting by filling in the regions with patches and pixels from the rest of the image. The observation-type method still has the disadvantage of performing incomplete DR until the construction of the 3D model is completed. Therefore, methods that only use the observation-type method are not suitable for our DR application. A method that combines image processing with the real background information observed in previous frames has also been reported [11]. However, the contribution of this method, in which real background information is combined with the inpainting results through image processing, requires observation of the hidden area by changing the camera's position and angle. Also, this method requires an RGB-D camera, unlike our application, which is realized using a smartphone camera. This is because the inpainting results and real background information are created as a 3D model.

The completion-type method guesses the background to be used for DR from the surrounding background information. Therefore, it does not require advance preparation, unlike the observation-type method, and it can be used for areas that cannot be observed by the camera. For these reasons, only the completion-type method was used for our DR application.

Examples of DR using the completion-type method include object removal in panoramic images [12], the removal of unwanted people and objects in photographs [13,14], and the removal of AR markers [15–17]. A study of object removal from panoramic images [12] aimed to remove furniture from a spherical panorama of an indoor scene. The method used in this study fills in the objects in the image once with white. Then, it uses a deep learning network to restore the filled-in area to a scene without objects in it, matching it to the surrounding background. The function of removing unwanted people and objects in a photo [13,14] is actually implemented in image editing applications as removing tools. They can remove objects from an image by specifying the objects to be removed. In the examples [15–17], AR markers are removed by applying traditional completion-type methods, such as generating backgrounds by calculating information from the surrounding background, and using backgrounds with similar patterns clipped from previous frames.

These completion-type methods using machine learning are more accurate than traditional completion-type methods [15–17] in terms of the background image used for DR and can be used for complex backgrounds. However, most of them are used for still images that have already been captured [12–14]. Therefore, there is no guarantee that they are suitable for real-time processing, and it is difficult to use these machine learning models under the conditions that were used in the examples for our DR application. Accordingly, in this study, the machine learning model used for the DR application was also examined in order to achieve real-time processing.

There are very few examples of real-time DR using machine learning. A study using real-time DR for buildings has been reported [18]. This involves recognizing buildings from images of landscapes that include buildings and performing DR in the region that contains the buildings. This method recognizes and removes all architectural objects in a

landscape image using semantic segmentation and does not remove arbitrary objects that are specified. It also requires the recognition of a 2D region for DR at each frame. DeepDR, an image inpainting network for DR, has also been reported [19]. In addition to the RGB images, the depth information is also required as an input for this network. Therefore, this network is not suitable for our application, which is realized using a smartphone camera, because a DR system using this network requires an RGB-D camera.

In the field of mixed reality (MR), which merges the virtual world with the real world, a study using DR based on machine learning has also been reported [20]. In this study, cars and people were removed in real space, and virtual objects were displayed in their place. However, the types of objects to be removed were limited. In addition, since client servers are used for DR and AR processing, an Internet environment is essential.

In this study, the deletion target is specified by a 3D region. By specifying a completion region that contains the object to be removed instead of the target, the process of detecting the target to be removed can be omitted, and DR for targets with various shapes and sizes can be realized. In addition, it is possible to keep the scene without the specified object even if the viewpoint position or angle changes. Moreover, 3D regions can be specified without relying on background information. A series of application processes can be executed on a laptop connected to a smartphone, eliminating the need for a network environment.

A study evaluating DR for 3D regions has already been reported [21]. However, the method used for DR is based on a patch search, which can only be used on backgrounds with a regular pattern, making it difficult to use in general environments. The background of the frame specifying the 3D region must be planar in order to project the 3D region from the camera position onto the estimated background plane.

A deep learning network is used for the completion function. To determine which network to use, we compared multiple networks experimentally and selected the network that could provide fast and accurate completion. In addition, by improving the loss function during training, the completion accuracy was significantly improved compared to previous methods.

Our DR application does not require an RGB-D camera. The DR application was run on a smartphone and a laptop computer, and operation at 10 fps was confirmed. We believe that a real-time DR system that can be realized using only a non-top-tier laptop computer and a smartphone can have a variety of applications.

2. Methods

2.1. Overview

We propose a DR application that visually removes specified objects in real time in a video captured using a smartphone camera. The application is used with a smartphone connected to a PC via a cable. DR is achieved using a background image to overwrite the target object in the video. The background image is created by guessing the scene behind the target object based on the background information around the object. The background image is generated using a small-scale deep learning network that is capable of high-speed processing. This high-speed processing enables real-time processing. Furthermore, this application does not require prior observation of the background directly behind the object, nor is there any restriction that the background must have a regular pattern. In addition, since the target object is specified manually using a 3D region, it is possible to accurately capture the 3D region that includes the target object and continue DR even if the camera viewpoint or angle changes. This application is intended for use with stationary objects.

Figure 1 shows the processing steps of the DR application.

(1) The plane in which the deletion target exists in the image is detected and captured using the smartphone camera. A virtual plane is constructed in the virtual space so that it overlaps the detected plane.
(2) The virtual cylinder is placed so that the target to be removed fits inside. The user specifies the placement position and the size of the virtual cylinder. The 3D completion region is inside the virtual cylinder.

(3) The 3D coordinates of eight points that could be the top, bottom, left, or right edge on the contour of the virtual cylinder are calculated. The coordinates of each point in the camera image are calculated from the camera parameters and the 3D coordinates, and their bounding rectangles are used as the 2D completion region.
(4) The 2D completion region of the camera image is filled to create the input image for the completion network.
(5) The completion network is used to generate the image in which the completion region is completed (completion images).
(6) The camera image and the completion image are combined to create and display a camera image in which DR is performed.
(7) Steps (3) to (6) are repeated per frame.

In this process, the iterative process of (7) after manual placement of the virtual cylinder is fully automated. In this study, ARCore, an SDK for smartphone AR, was used to perform basic AR processing such as camera self-localization and plane detection [22]. DR was performed using ARCore's Instant Preview, a function that enables AR on a PC connected to an Android device via USB.

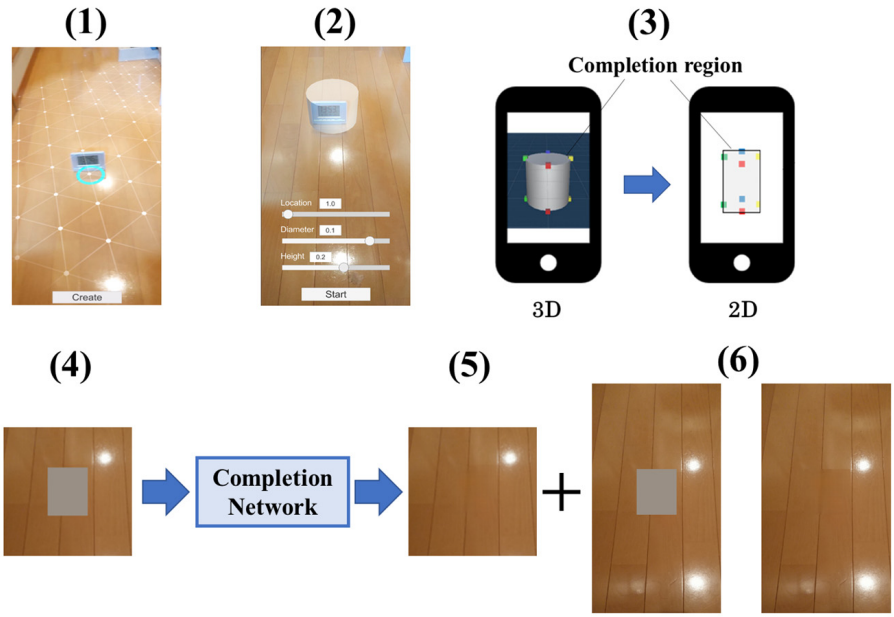

Figure 1. Overview of DR application process.

2.2. Self-Location Estimation of a Camera

DR and AR are realized by setting the origin of the virtual space in real space and rendering the virtual space superimposed on the real space. This requires technology to fix the origin of the virtual space in real space and to estimate the self-location of the camera in real and virtual spaces.

In this study, ARCore was employed to enable DR and AR. ARCore is an SDK for smartphone AR that utilizes concurrent odometry and mapping (COM) to establish the origin of the virtual space and estimate the self-location of the camera [22]. COM estimates the camera's position and rotation by utilizing the angular velocity and acceleration measured by calculating the characteristic points in the camera image and the inertial measurement unit in the smartphone together with past characteristic points and inertia. The origin of the virtual space is determined based on the camera position at the time of application startup, and it is fixed by correcting any misalignment observed by COM using the characteristic points and inertia. These processes allow for the calculation of the relative relationship between the camera's position and rotation in the virtual space and objects in the virtual space, enabling their rendering.

2.3. Detection of the Real Plane and Construction of the Virtual Plane

In step (1), a reference plane is detected, and a virtual plane is constructed in the virtual space to align with the detected plane. When specifying a 3D completion region that covers a real object placed on a plane, it can be challenging to perceive the depth of the virtual object, leading to potential difficulties in specifying the 3D completion region accurately. Therefore, the virtual plane is displayed with a pattern that makes it easy to recognize the depth, as shown in Figure 2. This method prevents the misrecognition of depth when specifying 3D completion regions.

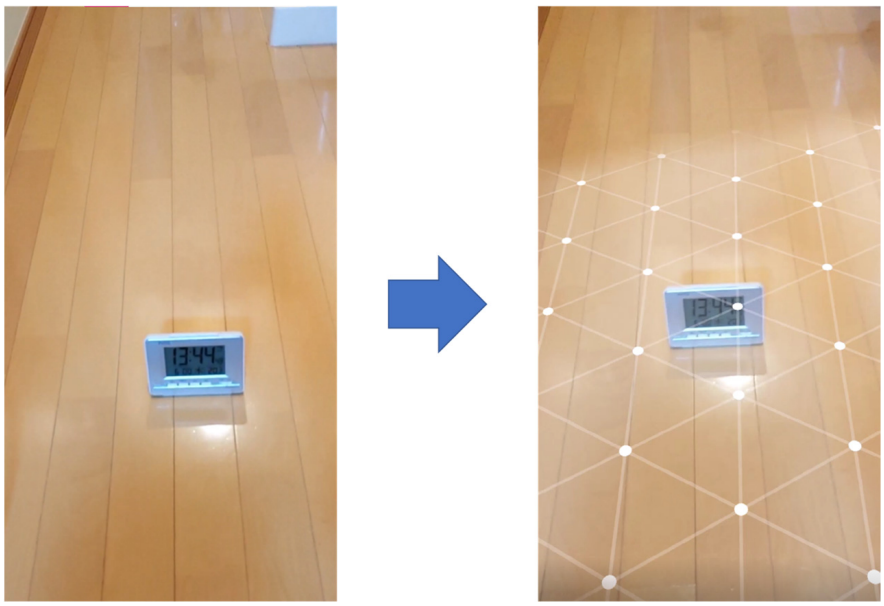

Figure 2. Construction of virtual planes.

ARCore can detect planes, such as floors and walls, from characteristic points detected by COM. The constructed virtual planes are displayed as triangular meshes, as shown in Figure 2.

2.4. Determination of 3D Completion Region

In step (2), a 3D completion region is defined by placing a virtual cylinder around the object to be removed and setting its size and the position, as shown in Figure 3. By defining a completion region that contains the object to be removed, rather than focusing on the specific details of the object, the need for detecting and identifying the object can be eliminated, thereby allowing DR to be performed on a wider range of objects with various shapes and sizes.

As shown in Figure 3a, the intersection between the virtual plane and the optical axis of the camera is calculated and visualized using a pointer. The pointer is displayed as a blue circle as shown in Figure 3a,b. Then, as shown in Figure 3b, a user moves the pointer to the position right under the object to be removed, and the create button is pressed to place a virtual cylinder with the 3D coordinates of the pointer as the center of the bottom surface, as shown in Figure 3c.

The user can adjust the size and the height of the completion region to fit the target to be removed by setting the diameter and the heights of the bottom and top surfaces of the virtual cylinder and its position in the height direction. The user can also check if the target fits within the completion region by moving to its side or back. This function can help to minimize the completion region. Furthermore, since DR is performed on the 3D region that includes the object, there is no need to recognize the object's contour.

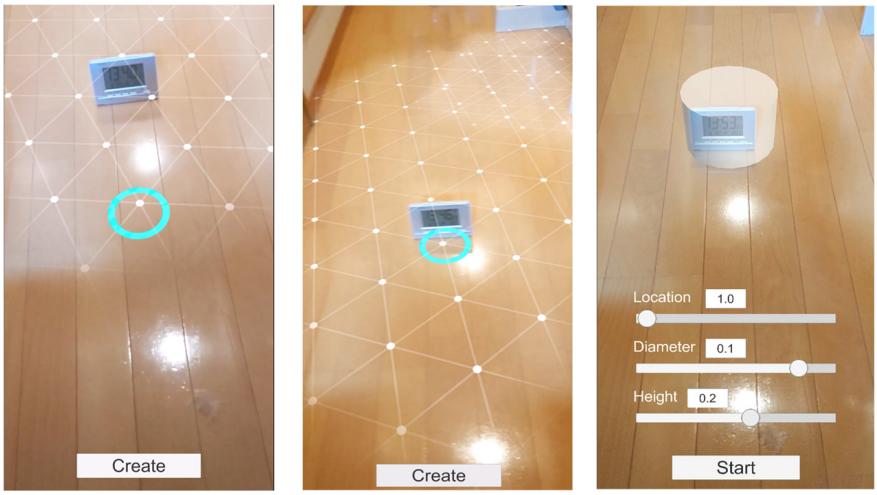

Figure 3. (**a**) Visualizations of the virtual plane and the pointer; (**b**) setting the pointer right under the target object; (**c**) creating a virtual cylinder with an appropriate diameter and a height to include the target object.

2.5. Calculation of 2D Completion Region

It is necessary to calculate the 2D completion region in the camera image since a trained completion network is used to complete the completion region. The 2D completion region is the bounding rectangle of the virtual cylinder that is the 3D completion region. The 2D completion region is calculated using the following procedure:

1. The 3D coordinates of the eight points on the contours of the virtual cylinder are obtained, including the top, bottom, left, or right edges, as shown in Figure 4.
2. The camera image coordinates for each point are calculated from their 3D coordinates and the camera parameters. The points on the contours of the virtual cylinder and the corresponding points in the camera image are marked with the same color in Figure 4.
3. The bounding rectangle of the eight camera image coordinates are calculated to obtain the 2D completion region.

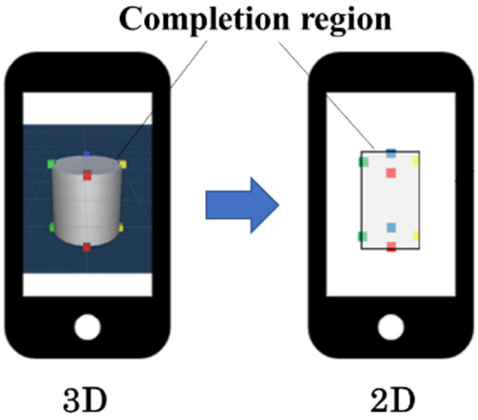

Figure 4. Calculation of 2D completion region.

2.6. Creation of Input Images

In step (4), the 2D completion region in the camera image is filled with the mean pixel values of the dataset for training to create the input image for the completion network. Inputting the camera image directly into the completion network would require a significant processing time. Therefore, to reduce the processing time, the camera image is first scaled down and then cropped to create a square image containing the completion

region, as shown in Figure 5. This cropped image is used as the input image for the completion network.

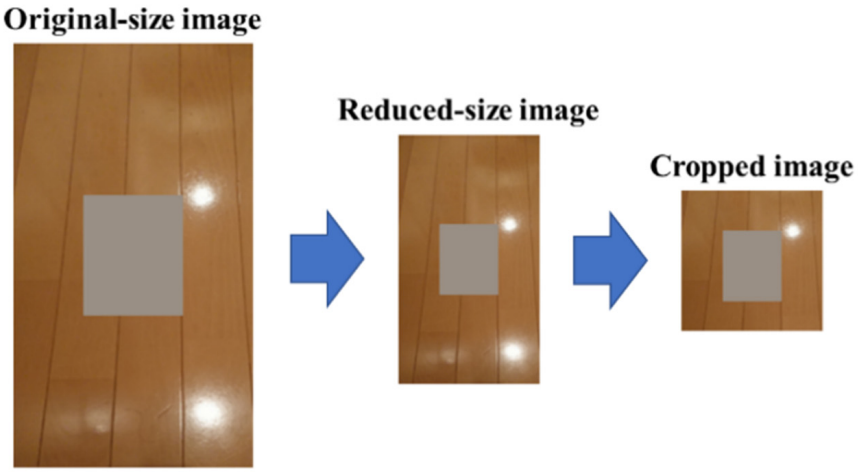

Figure 5. Process for creating input images.

2.7. Generation of Completion Images

In step (5), a completion image is generated using the completion network from the input image created in step (4). In this study, a trained deep learning network was utilized for the completion network. The specific details of the completion network are explained in Section 3.

2.8. Composition of the Camera Image and the Completion Image

In step (6), the completion region of the camera image is overwritten with the completion region of the completion image. It is necessary to upscale the completion image to fit the camera image before performing the composition process since the completion image is a downscaled version of the camera image.

3. Image Completion Function

3.1. Image Completion Network

A deep learning network is used for the completion function of the DR application. The deep learning network used in our application must be able to generate completion images in real time. Therefore, we used globally and locally consistent image completion (GLCIC), which has a small-scale structure, as the base network [23].

Furthermore, we attempted to introduce the channel attention mechanism [24] and the contextual attention mechanism [25] into GLCIC to generate completion images with the highest possible completion accuracy without deteriorating real-time performance. The channel attention mechanism is also used in image generation and image recognition. The contextual attention mechanism is used in image inpainting. We developed a network that incorporates channel attention into GLCIC (Ch-GLCIC), a network that incorporates contextual attention (Co-GLCIC), and a network that incorporates both attention mechanisms (CC-GLCIC). However, the extent to which the introduction of these mechanisms will reduce the processing speed is not clear. Therefore, we compared not only the completion accuracy of those networks but also their processing speeds, and we determined which networks are suitable for DR application.

In addition, as a measure to improve completion accuracy without affecting the processing speed at all, we also attempted to improve the cost function to train the network. Details of the training are described in Section 4.4.

3.2. Globally and Locally Consistent Image Completion (GLCIC)

GLCIC is a GAN-based method that is used to train image completion. As shown in Figure 6, GLCIC differs from a typical GAN because it consists of three networks. These networks are called the completion network, global discriminator, and local discriminator.

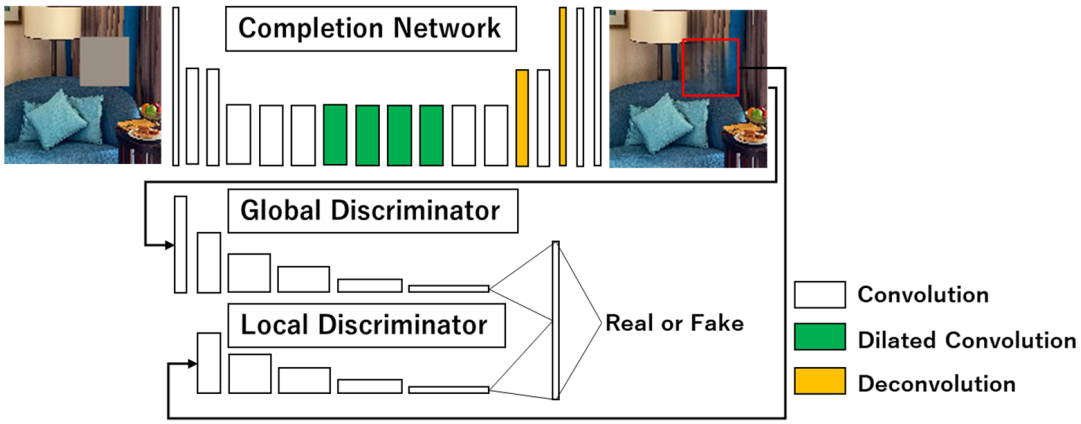

Figure 6. Globally and locally consistent image completion network.

The completion network is capable of generating an image containing completion information for arbitrary shapes and sizes of completion regions present in the input image. The generated image has the same size as the input image, with the completion region seamlessly integrated into the overall image. The target image and mask data that indicate the completion region are inputted to the completion network. The completion network then generates and outputs an image in which the completion region is completed.

The global discriminator uses the entire given image to determine whether the input image is the original or the completion image. The local discriminator uses a small region containing the completion region to determine whether the input image is the original or the completion image. The global and local discriminators take an original or a completion image of a predefined size as the input, respectively, and output the probability that it is the original image. In this paper, we refer to the combination of the local and global discriminators as a single discriminator.

The completion network is trained to generate completion images that can deceive the discriminator, while the discriminator is trained to determine images more accurately.

Only the completion region is used in the completion image output by the GLCIC. The final completion result is a composite image of the completion region of the image generated by the network and the non-completion region of the original image.

3.3. Channel Attention (Ch-GLCIC)

Channel attention has been newly incorporated into GLCIC to improve the completion accuracy of the generated completion image. Channel attention is a technique that weighs the channels of a convolutional neural network (CNN) feature map, highlighting important features and improving the quality of the network representation [24]. The layers incorporating channel attention were determined experimentally. In our implementation, the channel attention was added to the completion network at layers 2, 3, and 4; to the global discriminator at layers 2, 3, and 4; and to the local discriminator at layers 1, 2, and 3, as shown in Figure 7.

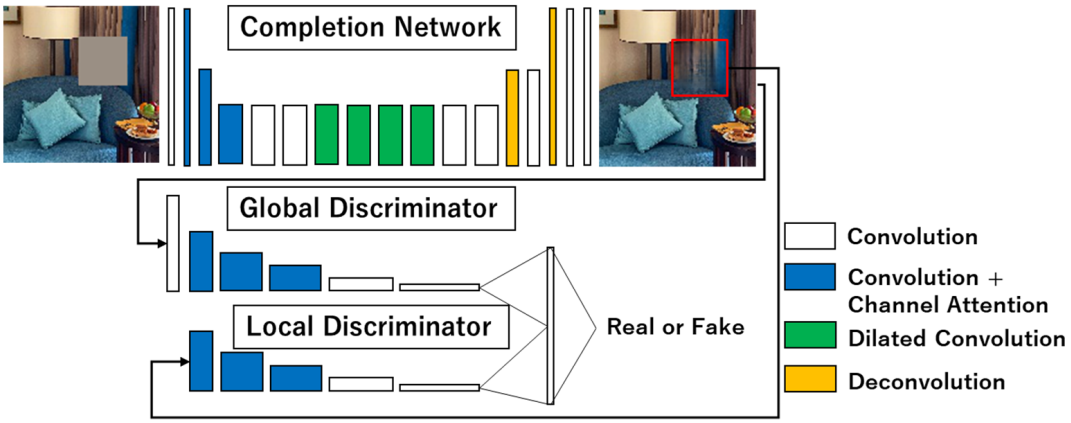

Figure 7. Ch-GLCIC network.

3.4. Contextual Attention (Co-GLCIC)

To improve the quality of the generated completion images, we incorporated contextual attention into the network. Contextual attention is a technique that uses the surrounding texture and structure correlated with the target region for image reconstruction [25]. However, since contextual attention cannot be used for completely uncorrelated objects, the coarse completion network roughly completes the region once, as illustrated in Figure 8. Subsequently, the refinement completion network, which includes the contextual attention layer, is used for more precise completion.

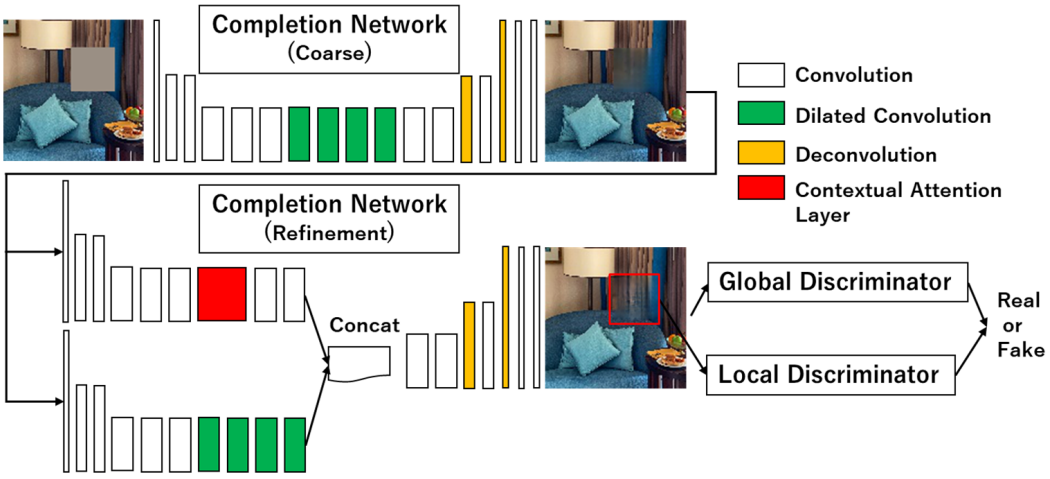

Figure 8. Co-GLCIC network.

The refinement completion network is composed of a concatenation of the coarse completion network and the contextual attention completion network [25].

The structures of the coarse generator, global discriminator, and local discriminator are similar to that of GLCIC.

3.5. Channel Attention and Contextual Attention (CC-GLCIC)

As a new attempt, a network combining both channel and contextual attention was implemented to further improve completion accuracy. As shown in Figure 9, CC-GLCIC is based on the Co-GLCIC network, with channel attention incorporated in three layers for each of the two completion networks and the two discriminators. The layers incorporating channel attention were determined experimentally.

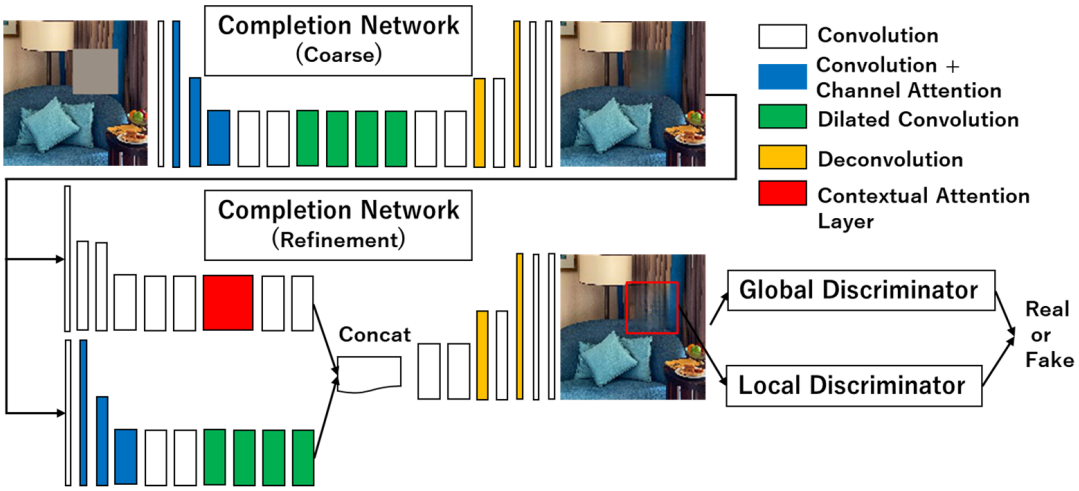

Figure 9. CC-GLCIC network.

4. Experiment

4.1. Training

A dataset was created using indoor images [26–29]. The image size of the dataset was determined as 128 × 128 pixels. Since the collected indoor images were larger than 128 × 128 pixels and were not square, they were first downscaled to 256 pixels on the short side. Then, multiple 128 × 128-pixel images were cropped from the downscaled images to create the dataset. Additionally, left–right inverted images of the cropped images were included in the dataset. A total of 16,094 images were used for training, while 1884 images were used for evaluation. The original indoor images were divided into separate sets for each purpose to ensure that the training and evaluation sets were distinct.

One simple method to train a completion network for DR involves extracting the contour of an object, creating a filled-in image using a single color within the contour, and training it to complete the filled-in region. However, this method has the following disadvantages:

- It requires a dataset of combinations of images with the target object to be removed and the same image without the object. Therefore, it is difficult to increase the number of variations in the dataset.
- The surrounding pixel values are altered by the light reflected by the object and shadows of the object. As a result, the pixel values of the generated completion images change from the background image without a target object. Therefore, it is difficult to evaluate the completion images accurately.
- It is difficult to evaluate completion images where objects in various positions and of various sizes are removed.

To solve the above problems, this study treated the training data images as background images after DR. Then, an object was assumed to be placed on this background image, and a completion region was created to include the object. In this way, an image with a completion region was combined with a background image that was the correct image.

The network is trained to recover the original background from the completion region image. The size of the completion region was determined as w × h (pixels), where w and h were randomly determined to be in the range of 16 to 64. Figure 10 shows examples of the original images (correct answers) and the images with completion regions (input images).

This approach has the following advantages:

- This method can easily increase the number of variations in the dataset rather than having combinations of images with the target object to be removed and the same image without the object.
- The effect of the object's placement on the surrounding pixel values can be suppressed. Therefore, the evaluation of the completion images can also be performed accurately.

- Completion images with various positions and various sizes of target objects can be evaluated.
- The data with variations may have a positive impact on the training of the network.

In addition, network training was conducted in three phases. The first phase involved training only the completion network. Then, in the second phase, the weights of the completion network were fixed, and only the discriminators were trained. Finally, in the third phase, the completion network and the discriminator were trained in parallel, as if they were competing against each other. The loss function and optimization method were the same as in GLCIC [23]. The input for the local discriminator was a 64 × 64-pixel image centered on the completion region of the 128 × 128-pixel completion images [23].

Figure 10. Examples of datasets (**top**) and input images (**bottom**).

4.2. Evaluation Method

The accuracies of the completion regions of the generated completion images were evaluated using the evaluation images. Since the accuracies of the completion regions depend on the sizes of the completion regions, the accuracies were evaluated for seven different sizes of completion regions (16×16, 24×24, . . ., 56×56, 64×64). The completion accuracies were calculated by averaging the completion accuracies of the 1884 evaluation images for each size of completion region. Furthermore, the average completion accuracy was evaluated for completion regions of various aspect ratios and sizes (VARAS). The completion regions of VARAS were determined randomly in the range of 16 to 64 in length and width. The evaluation indices used were PSNR (peak signal-to-noise ratio), SSIM (structural similarity), and LPIPS (learned perceptual image patch similarity). PSNR and SSIM only evaluate the completion regions, while LPIPS evaluates the entire images. Here, the completion regions of the original images are completed by the output images of the network because the evaluation of perceptual similarity should include not only the completion regions, but also the relationships with the surroundings of the completion regions. The speed of generating the completion images was also measured and compared.

4.3. Comparison of Completion Networks

Figures 11–13 show the completion accuracies of each network for each size of completion region and the completion regions of VARAS. Larger values for PSNR and SSIM represent better evaluations, while smaller values for LPIPS represent better evaluations. Ch-GLCIC, Co-GLCIC, and CC-GLCIC were more accurate than GLCIC in all completion region sizes for all measures. A comparison of Ch-GLCIC, Co-GLCIC, and CC-GLCIC shows that Ch-GLCIC and CC-GLCIC have higher completion accuracies than Co-GLCIC in all measures. In PSNR and SSIM, Ch-GLCIC showed the most improved completion accuracy compared to GLCIC at a 16×16-pixel completion region, with PSNR and SSIM being improved by more than 1 dB and 0.08. In LPIPS, Ch-GLCIC showed the most improved completion accuracy compared to GLCIC at a 64×64-pixel completion region, with LPIPS improving by approximately 0.024.

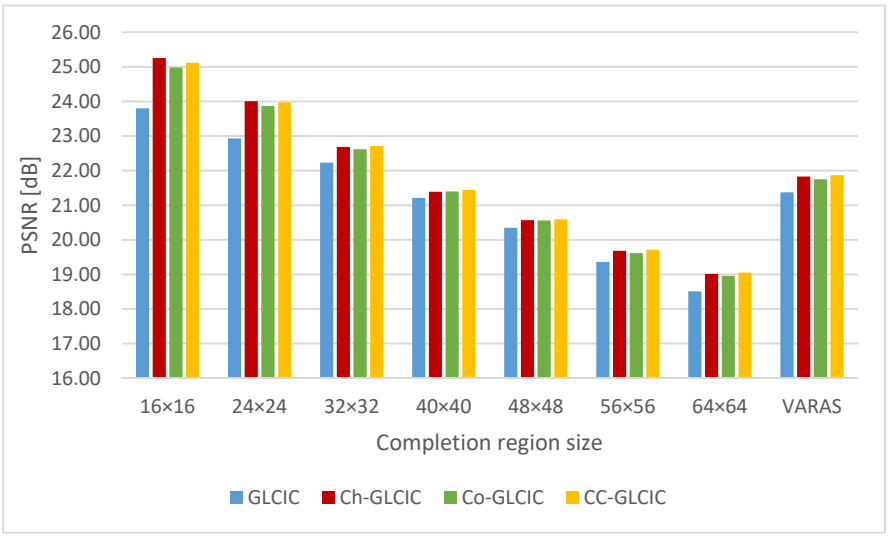

Figure 11. Completion accuracies (PSNR) for four networks.

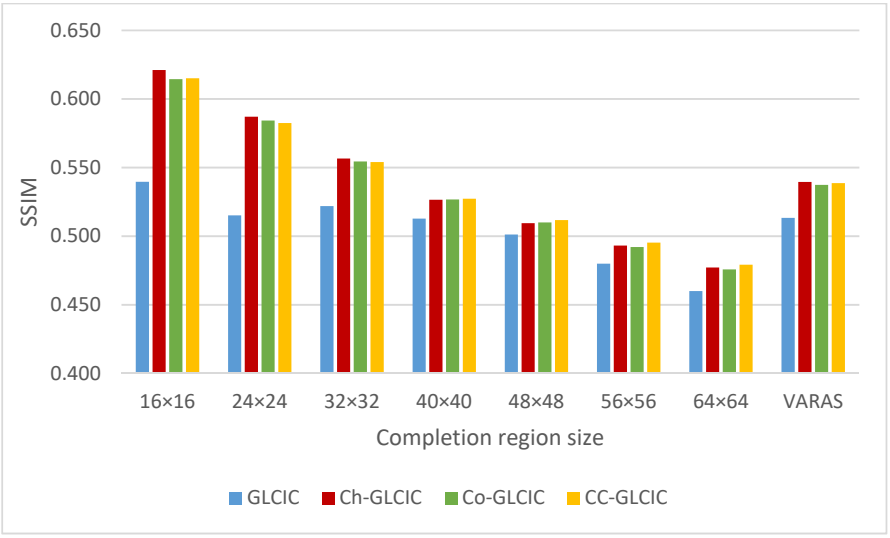

Figure 12. Completion accuracies (SSIM) for four networks.

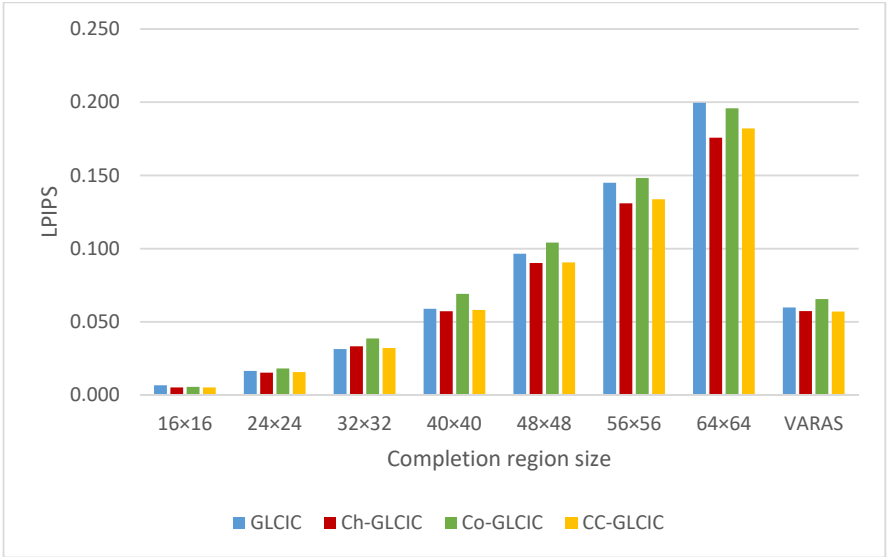

Figure 13. Completion accuracies (LPIPS) for four networks.

Figure 14 shows the sample completion results for each network. These images correspond to the average completion accuracy of Ch-GLCIC at each completion region size. It can be seen that Ch-GLCIC, Co-GLCIC, and CC-GLCIC produce images with fewer differences in terms of color and pattern than GLCIC.

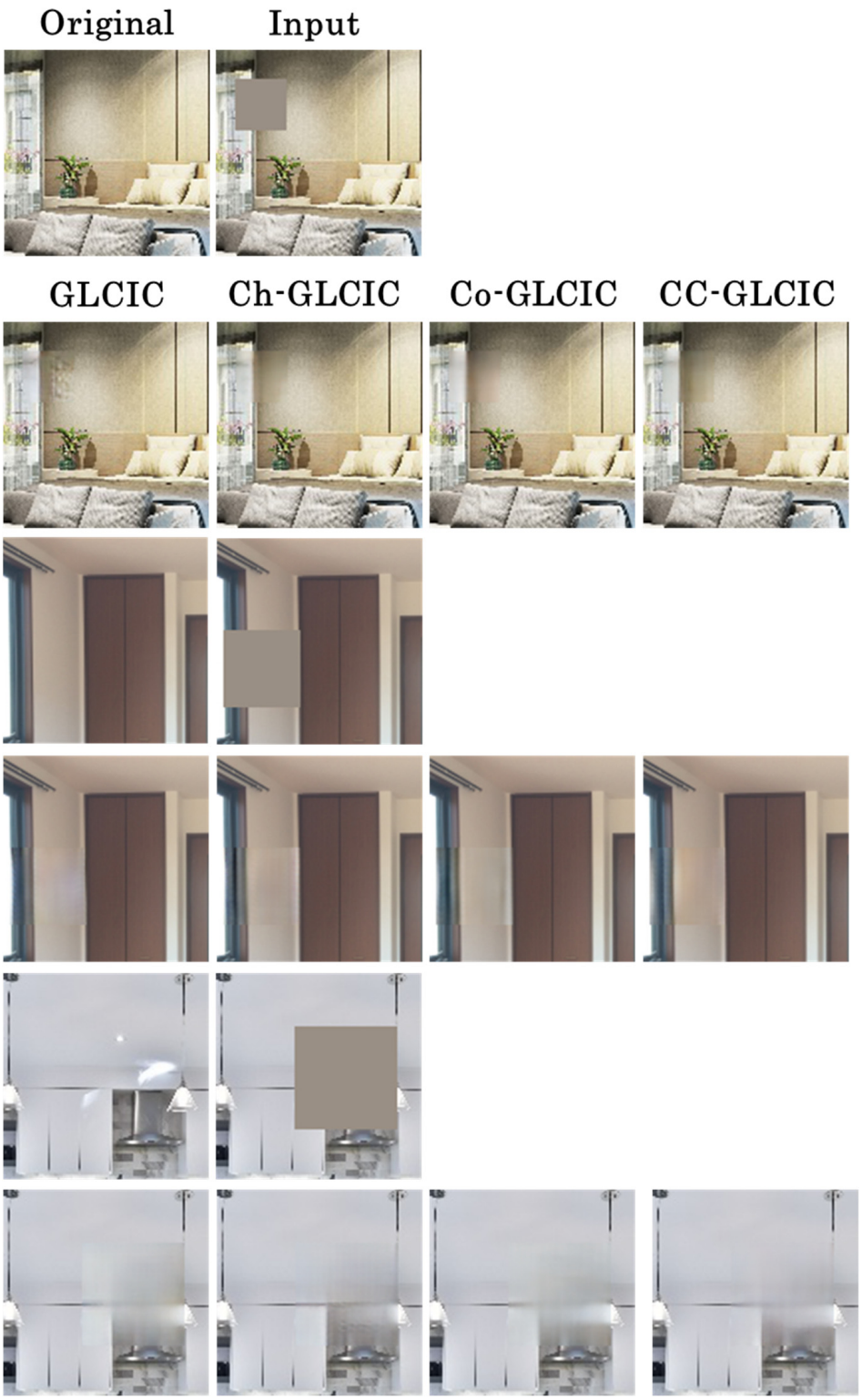

Figure 14. Examples of completion results (completion region size (pixels): top, 16 × 16; middle, 32 × 32; bottom, 64 × 64).

The processing speed of generating the completion images was measured using a notebook PC (Mouse Computer Co., Ltd., Tokyo, Japan) with the following specifications:

OS: Windows10 Pro;

CPU: Core i7-8750H (2.2 Hz);

GPU: GeForce GTX1050 (2 GB).

Table 1 shows the results. Among the four networks, Ch-GLCIC and GLCIC exhibited the fastest generation speeds, which can be attributed to their smaller network sizes compared to Co-GLCIC and CC-GLCIC. The generation speeds of GLCIC and Ch-GLCIC were almost the same, which shows that channel attention does not have a great effect on the generation speed.

Table 1. Generation speeds of completion images [fps].

GLCIC	Ch-GLCIC	Co-GLCIC	CC-GLCIC
80.4	79.3	37.7	36.6

Overall, Ch-GLCIC is the only one of the four completion networks that has both a high completion accuracy and high processing speed when generating completion images. Therefore, Ch-GLCIC was chosen as the completion function for our DR application.

4.4. Improvement in Network Training

The completion images generated using the completion network are combined with the original image only in the completion region, and the other regions are not used. However, completion regions are completed by the surrounding information. Therefore, if the accuracies of the non-completion regions are low, the accuracies of the completion regions are also expected to be low. On the other hand, the completion network of GLCIC focuses only on the completion region and trains using only the MSE loss in the completion region [23]. Figure 15 shows an example of the completion network's output image. Since the MSE loss out of the completion region is not taken into account, the generated image out of the completion region is degraded compared to the original image.

Figure 15. Completion image (before composite).

Therefore, we attempted to improve this problem by adding the MSE loss from the completion region to the loss function of the completion network. The new loss function is shown in Equation (1).

$$Loss = Loss_{in} + \beta \times Loss_{out} \quad (1)$$

In Equation (1), $Loss_{in}$ is the MSE loss within the completion region and $Loss_{out}$ is the MSE loss outside of the completion region. The training process outside of the completion region is faster than that within the completion region because of the information available in the input image. This may negatively impact training within the completion region due to overtraining outside of the completion region being carried out before training within the completion region reaches its maximum potential. To balance the learning rates inside

and outside of the completion region, a weight β is applied outside of the completion region. A β value of 0 corresponds to the conventional loss function.

The completion accuracies were evaluated by varying the value of β to determine its optimal value. The values of PSNR and SSIM for different values of β are presented in Figure 16, where they represent the average values obtained for seven different completion region sizes. As can be observed from Figure 17, the network trained with the new loss function ($\beta > 0$) performs better than the network trained with the conventional loss function ($\beta = 0$). The highest completion accuracy was achieved when β was set to 0.5.

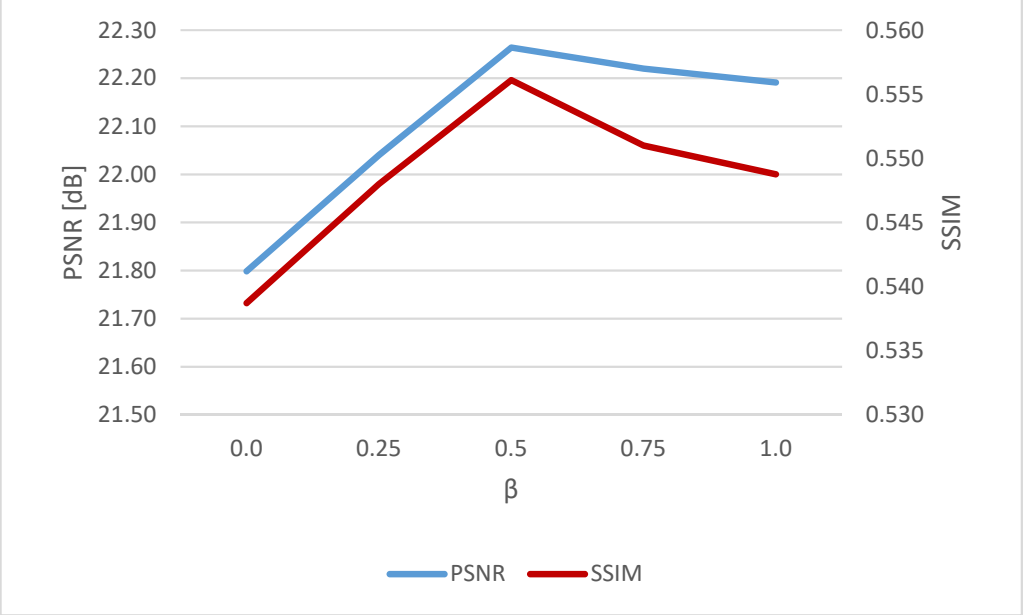

Figure 16. Completion accuracies when changing β.

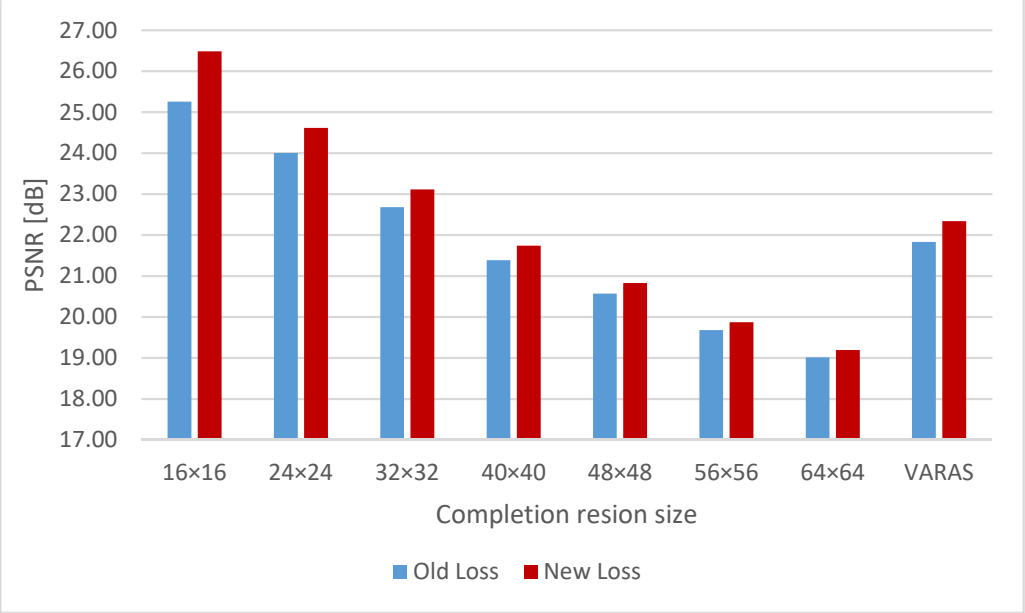

Figure 17. Completion accuracies (PSNR) of both trained networks.

Figures 17–19 show the completion accuracies of the conventional loss function and the new loss function ($\beta = 0.5$) for seven different completion region sizes and completion regions of VARAS. For all three measures, the completion accuracy was improved by the

new loss function. In PSNR and SSIM, the improvement in completion accuracy was most significant at the 16 × 16-pixel completion region, where PSNR and SSIM increased by approximately 1.2 dB and 0.03, respectively. In LPIPS, the largest improvement was seen at the 56 × 56-pixel completion region, with an improvement of approximately 0.004.

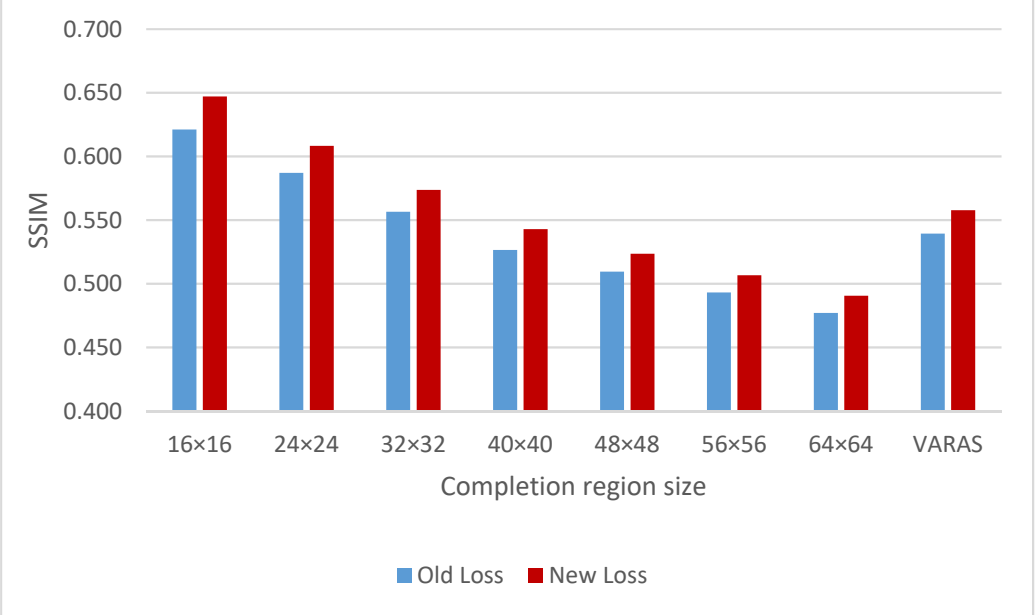

Figure 18. Completion accuracies (SSIM) of both trained networks.

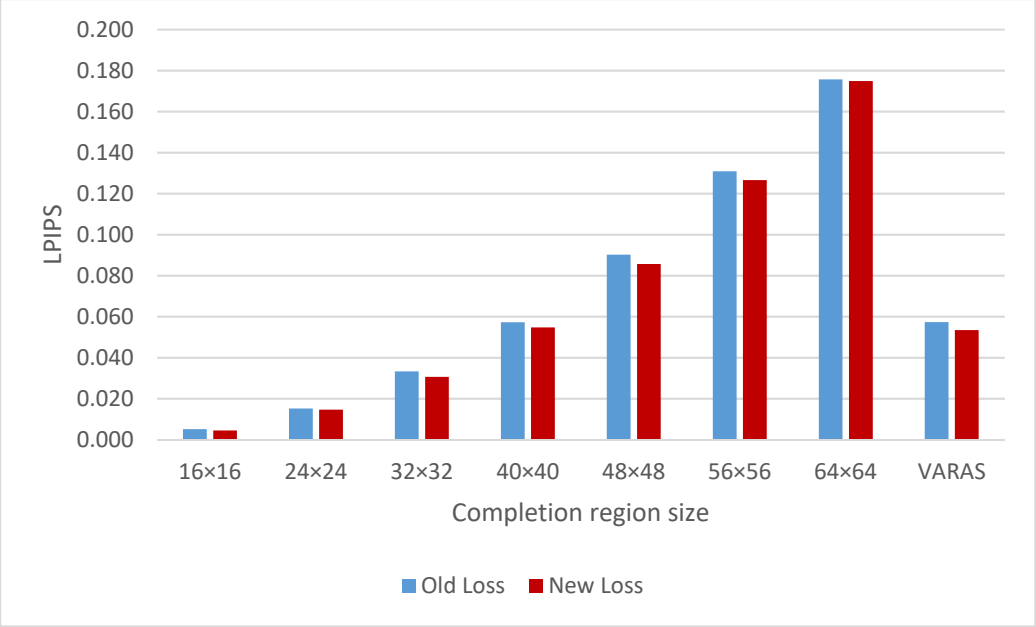

Figure 19. Completion accuracies (LPIPS) of both trained networks.

Figure 20 displays examples of completion results for networks trained with both loss functions. The completion images generated by the network using the new loss function have a clearer continuity of edges than the completion images generated by the network using the old loss function. This is because the network using the new loss function generates the entire images with high accuracy, and not only within the completion regions of the completion images. As a result, the final composite images become clearer.

Figure 20. Examples of completion results (completion region size (pixels): top, 16 × 16; middle, 32 × 32; bottom, 64 × 64).

4.5. Operational Experiments of the DR Application

Operational experiments were conducted for the DR application using Ch-GLCIC, which was trained with the improved loss function described in Section 4.4. The experiments were conducted using a Google Pixel 4 smartphone (Google Inc., Mountain View, CA, USA) connected to a laptop computer via USB.

Figure 21 shows examples of the execution results of the DR application. All operational experiments were conducted in environments that were not used to train the network. The results show that by setting up the completion region as the 3D region, the real object can be removed even if the viewpoint changes. Furthermore, as shown in the bottom example in Figure 21, the object located higher than the plane can be removed since a virtual cylinder can be placed higher than the virtual plane.

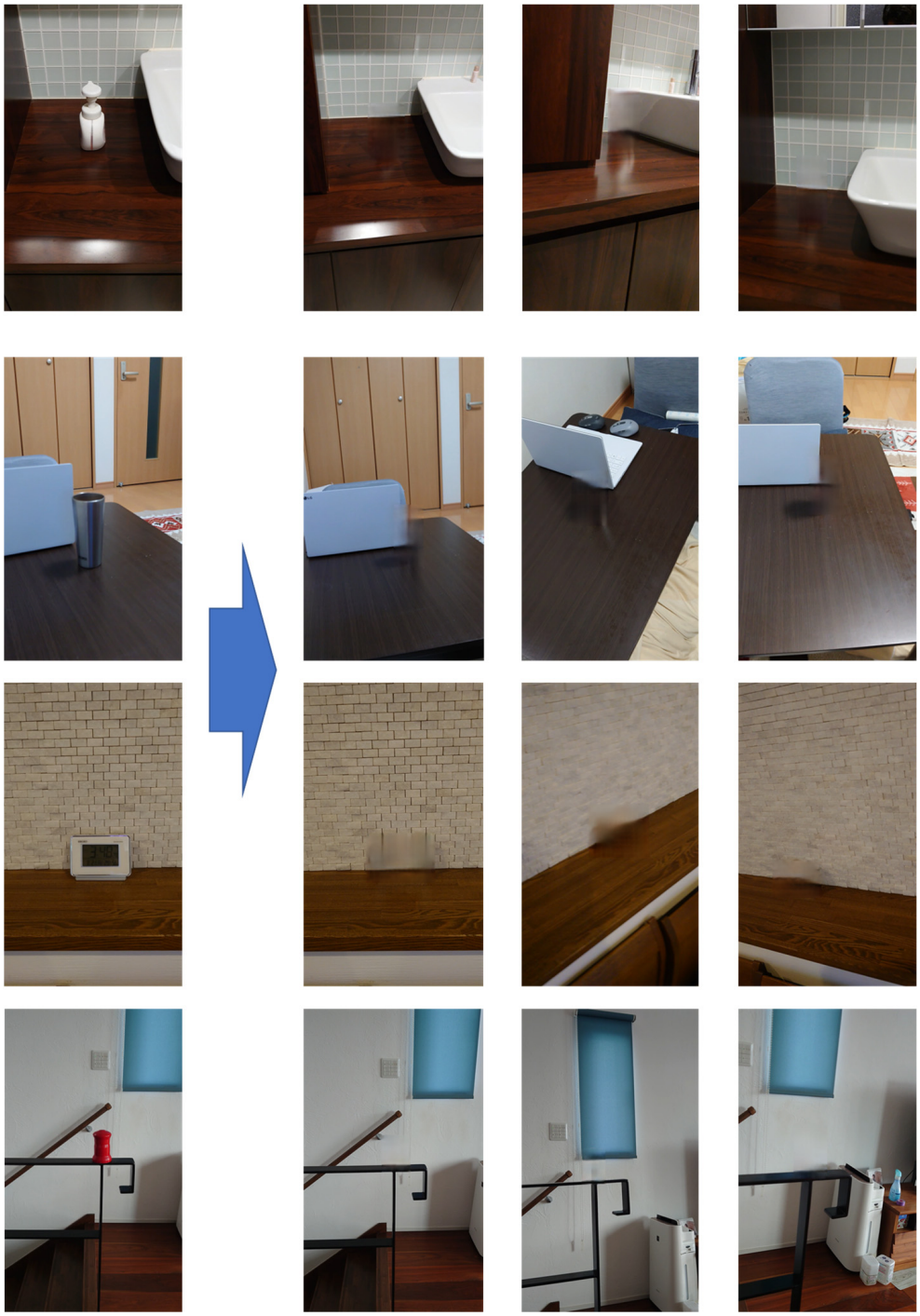

Figure 21. Execution examples of the DR application.

The processing speed of the DR application after manually setting the 3D regions (2.5 to 2.8) was 10 fps (please see the Supplementary Materials Video S1 for more details). Therefore, we consider that the DR application achieved real-time DR.

5. Conclusions

We developed a real-time diminished reality (DR) application using a smartphone camera. The DR application is capable of removing objects inside a 3D region specified by the user in images captured using a smartphone camera. By specifying a 3D region that contains the target object to be removed instead of the target, the process for detecting the deletion target can be omitted, and DR for targets with various shapes and sizes can be realized. In addition, the specified target can be removed even if the viewpoint changes. To achieve fast and accurate DR completion, a suitable deep learning network was employed based on the results of the performance evaluation experiments. As a result, Ch-GLCIC, a network in which channel attention is incorporated into GLCIC, was employed for the DR application. Additionally, the loss function during the training process was improved to enhance the completion accuracy. In the operational experiments, the application was used in rooms that were not included in the training data, and it was found that DR was performed with little sense of discomfort. These results indicate that if the training data are sufficient, it might be possible to use DR applications in environments that are different from those used for the training data. The application can run in real time (10 [fps]) using a non-top-tier notebook PC.

Future research includes further improvements in completion accuracy and processing speed. To improve the completion accuracy, we are considering incorporating additional information into the network input and further refining the network architecture and training approach. Also, since this study prioritized the development of an application that can run in real time on a non-top-tier laptop computer, the completion function of the application did not use the methods for the temporal coherence of the DR results, such as using previous frames, which required additional processing time. However, temporal coherence is an important factor in DR. Therefore, we will consider incorporating a method for temporal coherence, which does not significantly affect the processing time, into our application in the future. The 3D region is specified by the user using the GUI, so the evaluation of the UI would also be suitable for a future study.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/virtualworlds3010006/s1, Video S1: Video Results.

Author Contributions: Conceptualization, K.K. and M.T.; methodology, K.K. and M.T.; software, K.K.; validation, K.K.; formal analysis, K.K.; investigation, K.K.; resources, K.K.; data curation, K.K.; writing—original draft preparation, K.K.; writing—review and editing, M.T.; visualization, K.K.; supervision, M.T.; project administration, M.T.; funding acquisition, M.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The images of the dataset used to train and evaluate the networks are prohibited for redistribution by the data distributors. Therefore, the data presented in this study other than those are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mori, S.; Ikeda, S.; Saito, H. A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects. *IPSJ Trans. Comput. Vis. Appl.* **2017**, *9*, 17. [CrossRef]
2. Bamum, P.; Sheikh, B.; Datta, A.; Kanade, T. Dynamic Seethroughs: Synthesizing Hidden Views of Moving Objects. In Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality, Orlando, FL, USA, 19–22 October 2009; pp. 111–114. [CrossRef]
3. Zokai, S.; Esteve, J.; Genc, Y.; Navab, N. Multiview paraperspective projection model for diminished reality. In Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality, Tokyo, Japan, 10 October 2003; pp. 217–226. [CrossRef]
4. Kameda, Y.; Takemasa, T.; Ohta, Y. Outdoor see-through vision utilizing surveillance cameras. In Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality, Arlington, VA, USA, 5 November 2004; pp. 151–160. [CrossRef]

5. Rameau, F.; Ha, H.; Joo, K.; Choi, J.; Park, K.; Kweon, I.S. A real-time augmented reality system to see-through car. *IEEE Trans. Vis. Comput. Graph.* **2016**, *22*, 2395–2404. [CrossRef] [PubMed]
6. Queguiner, G.; Fradet, M.; Rouhani, M. Towards Mobile Diminished Reality. In Proceedings of the 2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct, Munich, Germany, 16–20 October 2018; pp. 226–231. [CrossRef]
7. Kunert, C.; Schwandt, T.; Broll, W. An efficient diminished reality approach using real-time surface reconstruction. In Proceedings of the 2019 International Conference on Cyberworlds, Kyoto, Japan, 2–4 October 2019; pp. 9–16. [CrossRef]
8. Kato, T.; Isoyama, N.; Kawai, N.; Uchiyama, H.; Sakata, N.; Kiyokawa, K. Online Adaptive Integration of Observation and Inpainting for Diminished Reality with Online Surface Reconstruction. In Proceedings of the 2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct, Singapore, 17–21 October 2022; pp. 308–314. [CrossRef]
9. Nakajima, Y.; Mori, S.; Saito, H. Semantic object selection and detection for diminished reality based on slam with viewpoint class. In Proceedings of the 2017 IEEE International Symposium on Mixed and Augmented Reality, Nantes, France, 9–13 October 2017; pp. 338–343. [CrossRef]
10. Herling, J.; Broll, W. PixMix: A Real-Time Approach to High-Quality Diminished Reality. In Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality, Atlanta, GA, USA, 5–8 November 2012; pp. 141–150. Available online: https://ieeexplore.ieee.org/abstract/document/6402551 (accessed on 27 August 2023).
11. Mori, S.; Erat, O.; Saito, H.; Schmalstieg, D.; Kalkofen, D. InpaintFusion: Incremental RGB-D Inpainting for 3D Scenes. *IEEE Trans. Vis. Comput. Graph.* **2020**, *26*, 2994–3007. [CrossRef]
12. Gkitsas, V.; Sterzentsenko, V.; Zioulis, N.; Albanis, G.; Zarpalas, D. Panodr: Spherical panorama diminished reality for indoor scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Montreal, QC, Canada, 11 October 2021. [CrossRef]
13. Adobe. Available online: https://helpx.adobe.com/jp/photoshop/how-to/remove-person-from-photo.html (accessed on 27 August 2023).
14. Google Photo. Available online: https://www.google.com/intl/ja/photos/about/ (accessed on 27 August 2023).
15. Siltanen, S. Texture generation over the marker area. In Proceedings of the 2006 IEEE/ACM International Symposium on Mixed and Augmented Reality, Santa Barbara, CA, USA, 22–25 October 2006; pp. 253–254. [CrossRef]
16. Kawai, N.; Sato, T.; Yokoya, N. Augmented reality marker hiding with texture deformation. *IEEE Trans. Vis. Comput. Graph.* **2017**, *23*, 2288–2300. [CrossRef] [PubMed]
17. Mori, S.; Schmalstieg, D.; Kalkofen, D. Good Keyframes to Inpaint. *IEEE Trans. Vis. Comput. Graph.* **2023**, *29*, 3989–4000. [CrossRef]
18. Kikuchi, T.; Kukuda, T.; Yabuki, N. Diminished reality using semantic segmentation and generative adversarial network for landscape assessment: Evaluation of image inpainting according to colour vision. *J. Comput. Des. Eng.* **2022**, *9*, 1633–1649. [CrossRef]
19. Gsaxner, C.; Mori, S.; Schmalstieg, D.; Egger, J.; Paar, G.; Bailer, W.; Kalkofen, D. DeepDR: Deep Structure-Aware RGB-D Inpainting for Diminished Reality. In Proceedings of the International Conference on 3D Vision 2024, Davos, Switzerland, 18–21 March 2024.
20. Kari, M.; Puppendahl, T.G.; Coelho, L.F.; Fender, A.R.; Bethge, D.; Schutte, R.; Holz, C. TransforMR: Pose-Aware Object Substitution for Composing Alternate Mixed Realities. In Proceedings of the 2021 IEEE International Symposium on Mixed and Augmented Reality, Bari, Italy, 4–8 October 2021; pp. 69–79. [CrossRef]
21. Kawai, N.; Sato, T.; Yokoya, N. Diminished reality based on image inpainting considering background geometry. *IEEE Trans. Vis. Comput. Graph.* **2016**, *22*, 1236–1247. [CrossRef]
22. Fundamental Concepts | ARCore | Google Developers. Available online: https://developers.google.com/ar/discover/concepts (accessed on 27 August 2023).
23. Iizuka, S.; Simo-Serra, E.; Ishikawa, H. Globally and Locally Consistent Image Completion. *ACM Trans. Graph.* **2017**, *36*, 1–14. [CrossRef]
24. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18 June 2018. [CrossRef]
25. Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; Huang, T.S. Generative image inpainting with contextual attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18 June 2018. [CrossRef]
26. Quattoni, A.; Torralba, A. Recognizing Indoor Scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20 June 2009.
27. Pixabay. Available online: https://pixabay.com/ja/ (accessed on 27 August 2023).
28. Unsplash. Available online: https://unsplash.com/ (accessed on 27 August 2023).
29. GAHAG. Available online: https://gahag.net (accessed on 27 August 2023).