Mixed-Reality World Exploration Using Image-Based Rendering

FUMIO OKURA, MASAYUKI KANBARA, and NAOKAZU YOKOYA, Nara Institute of Science and Technology (NAIST), Japan

This paper describes a mixed-reality (MR) application that superimposes lost buildings of a historical site onto real scenes virtualized using spherical aerial images. The proposed application is set at a UNESCO World Heritage site in Japan, and is based on a novel framework that supports the photorealistic superimposition of virtual objects onto virtualized real scenes. The proposed framework utilizes image-based rendering (IBR), which enables users to freely change their viewpoint in a real-world virtualization constructed using pre-captured images. This framework combines the offline rendering of virtual objects and IBR to take advantage of the higher quality of offline rendering without the additional computational cost of online processing; i.e., it incurs only the cost of online lightweight IBR, which is simplified through the pre-generation of structured viewpoints (e.g., at grid points).

Categories and Subject Descriptors: H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*Artificial, augmented, and virtual realities*; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Virtual reality*

General Terms: Algorithms

Additional Key Words and Phrases: Image-based rendering, mixed reality, photorealistic rendering, historical tourism

ACM Reference Format:

Fumio Okura, Masayuki Kanbara, and Naokazu Yokoya. 2014. Mixed-Reality World Exploration Using Image-Based Rendering. *ACM J. Comput. Cult. Herit.* 0, 0, Article 0 (January 20xx), 24 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

This paper proposes a framework for generating and exploring a photorealistic mixed-reality (MR) world where virtual objects are seamlessly synthesized into a pre-captured image of the real world as if they actually exist. The framework is intended to offer users a freely changeable viewpoint in such MR worlds through lightweight online rendering that can be performed on mobile devices such as laptops and tablets.

MR is a technology for synthesizing virtual and real worlds. At various architectural heritage sites, certain structures that no longer exist have been manually replicated in detail as three-dimensional (3D) virtual models. MR-based techniques, which superimpose such virtual models onto real-world scenes, can provide a promising presentation of architectural heritage sites [Zoellner et al. 2009; Foni et al. 2010]. Virtual objects are occasionally superimposed onto the real world in real-time. This type of technique is referred to as augmented reality (AR) [Azuma 1997; Azuma et al. 2001], and has recently attracted significant attention from both digital artists and industrial people.

Author's address: F. Okura, M. Kanbara, N. Yokoya, Vision and Media Computing Lab, NAIST, Takayama-cho, 8916-5, Ikoma, Nara, Japan; email: {fumio-o,kanbara,yokoya}@is.naist.jp.

DOI: http://dx.doi.org/10.1145/0000000.0000000

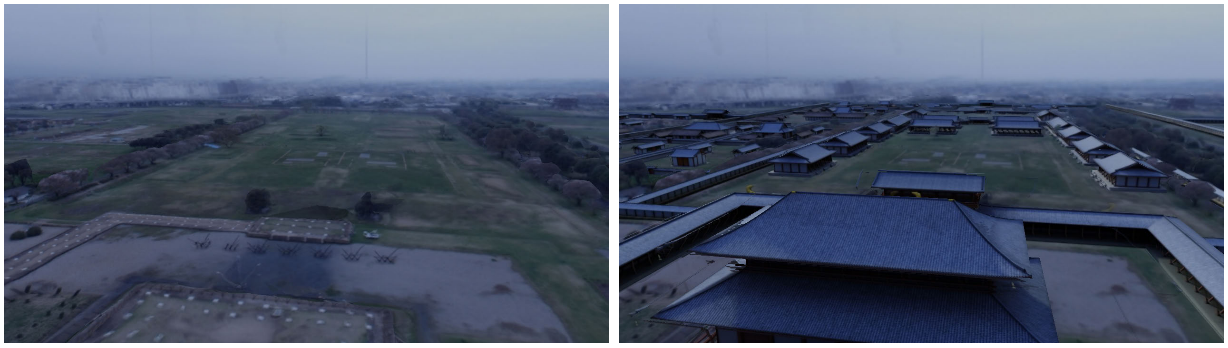

(a) Aerial scene of a large heritage site.
(b) MR scene generated using the proposed framework.

Fig. 1. MR scenes with a photorealistic superimposition of virtual objects generated using the proposed IBR framework.

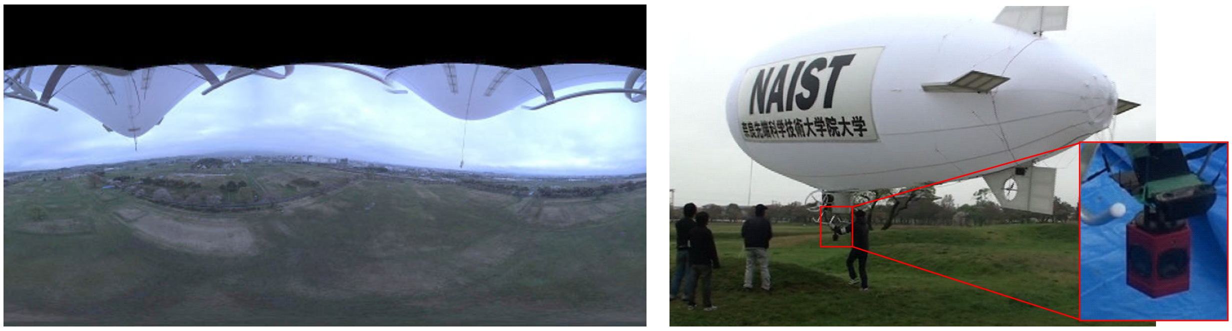
Aerial views are often employed for visualizing large areas (e.g., large heritage sites) reaching hundreds of meters in length. Exploration of the real world at a large scale by flying through a virtual simulation (virtualization) is very popular using virtual globe applications [Butler 2006] (e.g., Google Earth). Recent studies on computer vision (CV) have enabled us to virtualize real large-scale environments using modern 3D reconstruction methods, such as vision-based [Agarwal et al. 2009; Agarwal et al. 2010; Wu et al. 2011] and multi-sensor integration approaches [Banno et al. 2008; Pollefeys et al. 2008]. Along with AR applications using real-world views in real-time, virtual objects can be superimposed onto a (pre-generated) virtualized real world by rendering virtual objects onto the virtualized real world views in a frame-by-frame manner [Grosch 2005; Ghadirian and Bishop 2008; Wolk 2008; Xu et al. 2009; Okura et al. 2010]. In this paper, this type of MR application is referred to as MR world exploration.

This paper proposes a novel framework for realizing fly-through MR world exploration, where virtual objects are photorealistically superimposed onto a real-world virtualization, and a practical historical application based on the proposed framework. The application is set at a UNESCO World Heritage site, the Heijo-Palace in Nara, Japan. Aerial views used in the proposed application are shown in Figure 1. The Heijo Palace site is the location of an ancient palace of Heijo-Kyo, which was the capital of Japan from A.D. 710 to 784. The original palace buildings no longer exist at the site, which is approximately 1 km in length from east to west, and 1.3 km from north to south. The proposed application superimposes detailed 3D models of the original buildings over the palace site. These models, which were manually constructed by Toppan Printing Co., Ltd., consist of 4,255,650 polygons. In this type of MR application, the following functions are considered essential:

—**Photorealism**: The ultimate goal of virtual object superimposition is to realize an MR world where virtual objects are seamlessly synthesized to a real-world virtualization as if they actually exist.
—**Interactivity**: The user's viewpoint should be changeable.
—**Portability**: Similar to Google Earth, the application is intended for use not only on a large theater screen using a large mainframe computer, but also on a laptop PC or tablet device.

Unlike real-time AR, it is expected that MR world exploration will not require enforcing real-time changes in illumination because the illumination of a virtualized real world, which is constructed offline, remains static during the exploration. In addition, for the visualization of historical buildings in particular, we can assume that the virtual objects are static. Therefore, a novel approach for the superimposition of virtual objects suitable for MR world exploration applications should be developed; i.e., we need an approach for realizing a highly photorealistic image synthesis, despite the fact that it does not accommodate real-time changes in illumination.

In this paper, we propose a novel real-time rendering framework for MR world exploration that preserves the quality of offline efforts based on IBR, and a real-world virtualization technique for efficient rendering. IBR generates scenes at



(a) Aerial spherical image used as input for our application: A missing area appears in a portion of the sky.

(b) Unmanned airship and a spherical camera used for aerial imaging.

Fig. 2. Spherical image and aerial imaging equipment.

arbitrary viewpoints from multi-view images and rough 3D shapes. This technique is expected to successfully express the appearance of scenes using 3D shapes that include errors [Debevec et al. 1996]. IBR was originally designed for representing real-world scenes, but we expanded it for use in MR applications. In our framework, the appearance of both virtual and real objects is transformed through IBR using offline-rendered images at multiple viewpoints. The proposed framework resolves the problem of ordinary real-time rendering using a lightweight IBR method that preserves the efforts of offline rendering processes such as high-quality illumination, time-consuming rendering, and manual editing.

We employ spherical aerial images captured using an omnidirectional camera mounted on an aerial vehicle [Okura et al. 2010] to virtualize the real world for fly-through MR world exploration, as shown in Figure 2. Aerial imaging from omnidirectional cameras can be used to realize a more efficient virtualization of a large area as compared to the use of ordinary monocular cameras. Although the intensity of the sky is important for immersion and virtual-object rendering in MR world exploration applications, spherical aerial images may include missing areas in portions of the sky. Thus, the proposed framework includes a real-world virtualization technique with a completion of the spherical images. The contributions of this study can be summarized as:

—A novel photorealistic rendering framework for MR world exploration combining offline and online IBR.
—A lightweight IBR method that generates scenes at an arbitrary viewpoint using images at re-sampled viewpoints during the offline process.
—The development of a practical historical application that realizes a photorealistic superimposition of virtual buildings onto the real world and an immersive experience using spherical images.

The remainder of this paper is organized as follows. Section 2 describes other works related to MR world exploration. Next, the proposed framework is summarized in Section 3 using an example of an application which enables users to change their viewpoint on a 2D horizontal plane. Sections 4 and 5 detail the offline and online processes of the framework, respectively. Sections 6 and 7 describe experimental results and discussions using a practical historical application. The proposed framework can be extended for use with 3D structures to realize 3D viewpoint change; Section 8 describes an implementation of a 3D application and short discussions. Finally, Section 9 describes concluding remarks and future work.

2. RELATED WORK

2.1 Virtual Object Superimposition for MR World Exploration

Exploration applications that virtualize the real world have become popular with the growth of virtual globes [Butler 2006] such as Google Earth, and are among the most important applications based on real-world virtualization. In some applications, which are referred to as MR world exploration applications in this paper, virtual objects have been occasionally superimposed onto a real-world virtualization [Grosch 2005; Ghadirian and Bishop 2008; Okura et al. 2010], such as a landscape simulation. Google Earth has provided a framework for superimposing CG models using Google SketchUp (acquired by Trimble Navigation, Ltd. in 2012), which has been utilized for visualization studies [Wolk 2008; Xu et al. 2009]. In terms of the photorealism of superimposition, Okura et al. [2011] have developed a system that photorealistically superimposes virtual buildings onto aerial spherical videos. Di Benedetto et al. [2014] have recently developed a photorealistic exploration approach for a large virtual environment. This approach automatically determines the proper viewpoints, and then generates both photorealistic spherical views and spherical videos taken between nearby views using high-quality rendering. These systems [Okura et al. 2011; Di Benedetto et al. 2014] provide users with changes in viewpoint along pre-captured or pre-rendered paths. To realize free-viewpoint changes in MR world exploration applications, the appearance of virtual objects from the user's viewpoint must be rendered in a frame-by-frame manner. The issues inherent to virtual object superimposition in MR world exploration, specifically with the applications that provide changes in viewpoint, are also common with AR applications, i.e., both application types require real-time rendering techniques.

High-quality CG rendering methods for virtual object superimposition have been developed in the AR field. Most researches have applied pure CG field rendering methods to AR. Traditional approaches employ shadow maps [Kanbara and Yokoya 2002; Gibson and Murta 2000; Gibson et al. 2003] or shadow volumes [Haller et al. 2003] to express a local illumination, and do not consider complex physics, such as multiple reflections. The photorealistic appearance of virtual objects can be generated using global illumination (GI) rendering, such as photon mapping [Jensen 1996]. Image-based lighting (IBL), a real-world illumination acquisition and high-quality rendering framework that utilizes an imaging technique with a large field-of-view, which was proposed by Debevec [1998], is frequently employed in filmmaking. Such GI rendering techniques have been available as libraries [Ward 1994; Pharr and Humphreys 2010] and commercial software; however, in the past, it was difficult to employ them for AR applications because of their heavy computational requirements. With advancements in computer technology, recent studies on photometric AR have realized the use of real-time GI rendering, including the use of a pre-computed radiance transfer [Gierlinger et al. 2010], differential rendering [Knecht et al. 2010], real-time ray-tracing [Kán and Kaufmann 2012], and reflective shadow maps [Lensing and Broll 2012]. These approaches employ state-of-the-art real-time rendering methods, as studied in the CG field, for AR virtual-object representations. However, real-time rendering has not achieved the same quality as offline rendering, which is generally time-consuming. Even if real-world illumination conditions and virtual object reflectance properties can be estimated and/or determined accurately, part of their information is wasted because real-time rendering cannot represent all physical phenomena; that is, a real-time computation is achieved at the cost of quality. Notably, computational costs become greater when high-quality rendering techniques are used, or when virtual objects consisting of numerous polygons are applied. In these types of situations, it is difficult to perform state-of-the-art real-time rendering techniques on mobile devices. For this reason, *cinema-quality* virtual object superimposition on mobile devices is yet to be achieved. In filmmaking, on the other hand, CG effects are rendered using high-quality, time-consuming illumination methods, and a large number of manual operations. We contend that the efforts involved in such high-quality offline processes can be employed to improve virtual-object expression in MR world exploration.

The proposed framework, in principle, can employ any offline rendering method; that is, it can use scenes rendered photorealistically to the maximum extent. Therefore, it can include strict illumination settings, time-consuming rendering, and any manual adjustments. The offline-rendered images can subsequently be transformed using real-time IBR, which preserves the high-quality textures.

2.2 Image-Based Rendering

IBR, which is occasionally referred to as free-viewpoint (novel or arbitrary) image generation, has been actively studied in the CV and CG fields. In IBR, images are generated at arbitrary viewpoints using multi-view images and inaccurate 3D shapes. IBR can be categorized into a continuum of approaches, from physically based to appearance

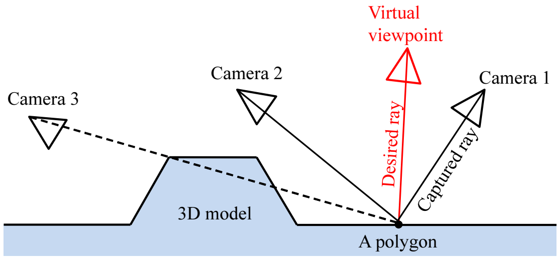


Fig. 3. View-dependent texture mapping (VDTM) [Debevec et al. 1996]. At the polygon, the weight for Camera 1 is larger than for Camera 2 according to the weight definition in the original VDTM. The polygon is invisible from Camera 3.

based, in terms of the input data and characteristics of the target scenes [Kang et al. 2000]. Most appearance-based IBR techniques do not use explicit 3D shapes; this technique includes view morphing [Seitz and Dyer 1996], light-field rendering [Levoy and Hanrahan 1996; Naemura et al. 1997], and lumigraph rendering [Gortler et al. 1996; Buehler et al. 2001]. Physically based IBR approaches [Debevec et al. 1996] basically use simple 3D scene models. Debevec et al. [1996] proposed view-dependent texture mapping (VDTM), which selects appropriate images from multi-view images, and then maps and blends these images into 3D models.

The calculation cost of most appearance-based approaches does not depend on the size of the target scene (e.g., the number of polygons). Physically based approaches enable a polygon reduction of virtual objects; they generate images that suppress the negative visual effects occurring from errors in the 3D shapes [Debevec et al. 1996]. In our proposed framework, these features contribute to a reduction in the number of computations.

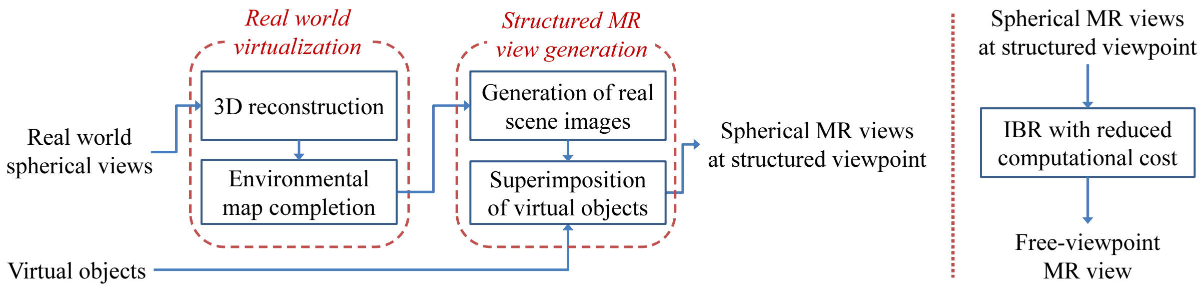
Our application employs a VDTM-based method to render MR scenes, although the proposed framework itself is not restricted to a specific IBR method. Because our application employs aerial views, it is expected that the 3D shapes reconstructed from the real world scenes are viewed from relatively distant positions. Physically based IBR, of which VDTM is a typical approach, is more appropriate for such environments compared with appearance-based IBR [Kang et al. 2000].

2.3 View-Dependent Texture Mapping (VDTM)

This section summarizes VDTM [Debevec et al. 1996] which is utilized in our application. VDTM is an effective technique for rendering photorealistic scenes using multiple images and an inaccurate 3D model. Multiple images are first projected onto a 3D model according to their position and orientation, unlike traditional texture mapping using only a single image. For each polygon in the model, the projected textures are blended using a given weighting function. Although the original implementation of VDTM [Debevec et al. 1996] employed the reciprocal of the angle between the desired ray and a captured ray of the *i*-th camera (see Figure 3), VDTM basically accepts various weight definitions. For example, the angle between the normal of the polygon and the *i*-th camera ray is occasionally utilized to select high resolution textures. In addition to the weighting functions, the visibility should be considered to acquire high-quality results. Camera 3 in Figure 3, for example, does not capture the polygon owing to the presence of an occlusion. A visibility test for determining whether the polygon is visible from the camera, is employed to avoid the texture of the camera being projected onto invisible polygons. Thanks to a wealth of studies on VDTM, the visibility of each polygon [Debevec et al. 1998] or pixel in a virtual view [Porquet et al. 2005] can be efficiently calculated.

3. OVERVIEW OF IBR FRAMEWORK FOR MR WORLD EXPLORATION

The proposed application generates scenes upon which virtual objects are photorealistically superimposed, as viewed from a position and direction that are freely configured on a designated two-dimensional (2D) plane in the sky, as shown in Figure 1. Spherical images (see Figure 2(a)) are captured from a number of positions in the sky using a spherical camera, Ladybug3 (Point Grey Research Inc.), and an aerial vehicle equipped with a GPS (see Figure 2(b)) [Okura


Real world virtualization
Structured MR view generation
Real world spherical views
3D reconstruction
Environmental map completion
Generation of real scene images
Superimposition of virtual objects
Virtual objects
Spherical MR views at structured viewpoint
Spherical MR views at structured viewpoint
IBR with reduced computational cost
Free-viewpoint MR view

(a) Offline process.

(b) Online process.

Fig. 4. Flow of the proposed IBR framework for MR world exploration.

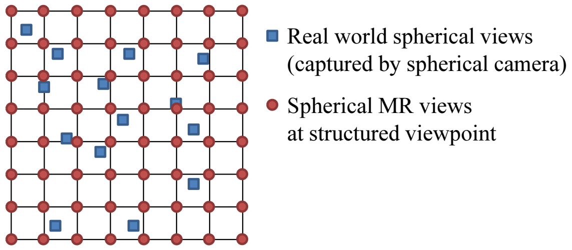

■ Real world spherical views (captured by spherical camera)
● Spherical MR views at structured viewpoint

Fig. 5. Pre-generation of real-scene images at grid points: unstructured viewpoints captured are re-sampled into a grid viewpoint structure using offline IBR.

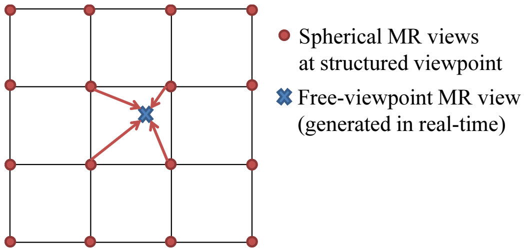

● Spherical MR views at structured viewpoint
✖ Free-viewpoint MR view (generated in real-time)

Fig. 6. Online IBR based on a bilinear weighting function. The weight is not affected by the position of the polygons.

et al. 2010]. Note that the proposed framework can easily be extended for use with 3D structures to realize 3D viewpoint change (Section 8 describes a straightforward extension for 3D viewpoint change).

Our framework can be divided into two parts: an offline process and an online process, as shown in Figure 4. The offline process inputs real-world aerial spherical images captured at various positions using a spherical camera, along with 3D models of virtual objects. This process generates spherical augmented views (MR views) at every structured viewpoint. Real-world views from structured viewpoints are generated through an IBR method using real-world 3D shapes estimated using 3D-reconstruction techniques and spherical images, in which any missing areas are completed. In our application based on the proposed framework, viewpoints are designated at every grid point, as illustrated in Figure 5. Virtual objects are then photorealistically rendered from every structured viewpoint using real-world illumination without any missing areas of illumination in the spherical images.

An MR view, which has a viewpoint that can be freely configured onto a 2D plane, is presented to users during the online process. The view is generated through a lightweight IBR from pre-rendered views at the structured viewpoints generated during the offline process. The computational cost of online IBR is reduced using a simple bilinear blending of the textures of the neighboring structured viewpoints (as illustrated in Figure 6) and using simplified 3D models. The online process requires only four times the number of weight calculations to blend the textures while rendering a frame, whereas the offline process needs the calculation for each polygon and camera.

4. OFFLINE PROCESS: PHOTOREALISTIC SUPERIMPOSITION OF VIRTUAL OBJECTS

During the offline process, aerial spherical MR views are generated at the structured viewpoints (at every grid point in our application) from real-world images captured at hundreds of different positions, and from 3D models of the virtual objects. The flow of the offline process is as follows:

(1) Real world virtualization
   (a) 3D reconstruction
   (b) Environmental map completion using a sky model
(2) MR view generation at structured viewpoints
   (a) Generation of real scene images using IBR
   (b) Photorealistic superimposition of virtual objects

The camera pose and the 3D shapes of real-world objects are used to generate real-world views at the grid points using VDTM [Debevec et al. 1996], which is a physically based IBR method. In aerial imaging, parts of the spherical images may be missing from occlusions by the camera or vehicle, for example. The intensity of these missing areas is estimated using the camera pose and a statistical model of the sky luminance. Using IBL [Debevec 1998], which is a high-quality offline superimposition technique using real-world illumination, the virtual objects are superimposed onto the completed environmental maps of the structured views, which are generated for every grid point. Although the processes for environmental map completion and virtual object superimposition are similar to [Okura et al. 2011], they did not achieve free-viewpoint exploration (i.e., IBR was not employed).

4.1 3D Reconstruction

A camera pose with six degrees-of-freedom (DoF) (three DoF positions and three DoF orientations) and dense 3D shapes are first reconstructed from the real world. We employ a vision-based structure-from-motion (SfM) approach using captured spherical images and GPS measurements. VisualSFM [Wu 2013], which is an SfM application using a multi-core bundle adjustment [Wu et al. 2011], is used to estimate the relative camera positions; however, SfM does not provide an absolute scale, which is required for geometric registration between real and virtual objects without a manual adjustment of their coordinates. For our situation, the camera position information using an absolute scale is acquired from the GPS installed on our aerial vehicle, and a non-linear minimization of the distance between the GPS measurement and the camera position estimated using SfM is conducted. Some approaches combining SfM and GPS measurements and using constrained/extended bundle adjustments [Kume et al. 2010; Lhuillier 2011] may also be suitable to our needs with slight improvements.

CMPMVS [Jancosek and Pajdla 2011], a state-of-the-art multi-view stereo (MVS) method, is used to further reconstruct the dense 3D polygonal shapes of real-world objects. The spherical images are preliminarily converted into six perspective images (cube maps) to prepare the input data for the SfM and MVS software, which uses perspective images as input.

4.2 Environmental Map Completion Using a Sky Model

Areas that a spherical camera cannot capture, or parts of the background scenery that are occluded by the aerial vehicle, may be included in the spherical images captured from a sky position. Missing areas must be filled in for the images to be used as a part of an environmental light map upon which virtual objects are rendered using IBL. An additional skyward camera can be used [Okura et al. 2014]; however, it is often difficult to capture all directions in a single pass using aerial imaging. As typically occurs in aerial imaging where the camera is mounted under the vehicle, in our case, a missing area of the sky, which is occluded by the vehicle, can be seen in the upper area of the spherical image (see Figure 2(a)). A sky model, which generates the statistical luminance of the sky, is used to fill in the missing area. We assume that both the spherical camera and the vehicle are in the same position relative to each other and that the missing areas are preliminarily identified by the user. Note that the position of the missing area does not change in the entire image set when the camera is securely mounted to the vehicle. The completion process of the environmental map can be divided into two parts:

(1) Completion based only on the intensity of the spherical images
(2) Completion of the remaining missing area using the All Sky Model [Igawa et al. 2004]

4.2.1 *Completion Based Only on the Intensity of the Spherical Images.* To estimate the intensity of the pixels in a missing area of the sky, we suppose that the illumination of the sky and clouds does not change during the image sequence capturing, and that the sky and clouds are infinite. Under these assumptions, the intensity of the sky can be generalized, and a single sky condition can be estimated for the entire image set because the intensity in the same direction does not change significantly while all images are being captured.

First, the spherical images are aligned using the camera pose information estimated as described in Section 4.1: the i-th image is mapped onto a sphere, and the sphere is rotated by $\mathbf{R}_i^{-1} = \mathbf{R}_i^{\mathrm{T}}$ using the estimated camera orientation $\mathbf{R}_i$ of the i-th image. The aligned spherical images are then converted into *sky images* in an equisolid angle projection, which have a uniform solid angle per pixel, as shown in Figure 7(a).

The intensity v_{uni} of a pixel in a particular direction in a sky image unified throughout the whole image set is estimated from the intensity v_i of the pixel in the same direction in the i-th image of the input image set.

$$\begin{aligned} v_{uni} &= \begin{cases} \frac{1}{\sum_i \alpha_i} \sum_i \alpha_i v_i & (\sum_i \alpha_i \neq 0) \\ \mathit{undefined} & (\sum_i \alpha_i = 0), \end{cases} \\ \alpha_i &= \begin{cases} 1 & (v_i \text{ is not in the missing area}) \\ 0 & (v_i \text{ is in the missing area}). \end{cases} \end{aligned} \tag{1}$$

Because this process assumes that the illumination of the sky and clouds does not change, using short sequences of only dozens or hundreds of frames is better when processing images of a drastically changing environment.

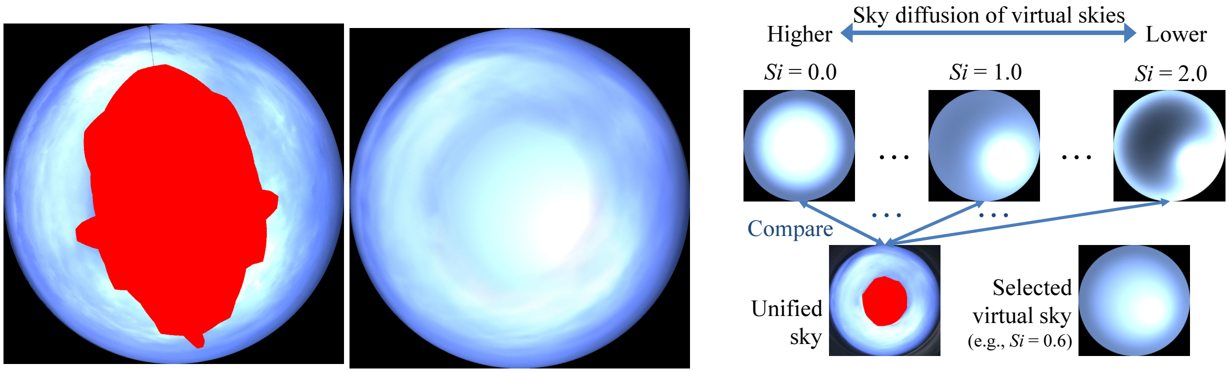
4.2.2 *Completion of the Remaining Missing Area Using the All Sky Model.* The same part of a missing area may be occluded in an entire image set (i.e., $\sum_i \alpha_i = 0$); in this case, the intensity v_{uni} cannot be determined through Eq. (1). In the All Sky Model [Igawa et al. 2004], the luminance (and radiance) distribution of the sky is statistically modeled, and is used to complete the remaining missing area. The All Sky Model is known to be a good approximation of the sky distribution under all-weather scenarios using a sky index, which is a variable used for indicating the sky conditions. When generating a complete sky using the All Sky Model, the generated sky image is represented as a high-dynamic range (HDR) image because the calculated intensities of the pixels can exceed an 8-bit value.

In this model, the sky luminance $La(\gamma_s, \gamma)$ is defined as the product of the zenith luminance $Lz(\gamma_s)$ and the relative sky luminance distribution $L(\gamma_s, \gamma, Si)$. The relative sky luminance distribution varies depending on the weather. Thus, the All Sky Model [Igawa et al. 2004] can be briefly described as follows:

$$La(\gamma_s, \gamma, Si) = Lz(\gamma_s) L(\gamma_s, \gamma, Si), \tag{2}$$

$La(\gamma_s, \gamma, Si)$: luminance of a sky element,
$Lz(\gamma_s)$: zenith luminance,
$L(\gamma_s, \gamma, Si)$: relative sky luminance distribution,
γ_s : direction (2 DoF polar coordinates) of the sun,
γ : direction (2 DoF polar coordinates) of the sky element, and
Si : sky index.

Here, the sky index Si has a value in the range of $0.0 \leq Si \leq 2.0$ depending on the weather. A larger Si indicates fine weather, and a lower value indicates a diffused (cloudy) sky. The value of Si is calculated using the amount of global solar radiation in the environment. Solar altitude γ_s is calculated automatically using the acquired time and date of the images, as well as the location in latitude-longitude coordinates. Normally, the luminance of the sky La and the zenith luminance Lz are represented using a physical unit, which in our case is an unknown value. In this study, we use the intensity of the pixels based on a linearized gamma, proportional to the physical value, instead of the physical luminance value.



(a) Sky image in an equisolid angle projection. The missing area is masked.

(b) Completed sky image.

(c) Estimation of sky index Si in the All Sky Model.

Fig. 7. Completion of an environmental map using the All Sky Model [Igawa et al. 2004].

The zenith luminance Lz required to calculate the intensity of the sky is usually unknown because the zenith is occluded during the entire sequence. Thus, Eq. (2) is modified to calculate Lz from La and L as follows:

$$Lz = \frac{La(\gamma, Si)}{L(\gamma, Si)}. \tag{3}$$

Here, we treat Si as a variable because the global solar radiation, which is required to calculate Si, cannot be acquired because of the existence of a missing area. In addition, γ_s is assumed to be a constant calculated based on the capture time and date. Although the intensity of the pixels not located in the missing areas can be used as the value of La, Si must still be estimated to calculate L. Once Si is estimated, Lz can be calculated using the acquired value of La. The calculated Lz does not vary ideally, but when calculated using the real intensity, it may include errors. We therefore estimate Si using the minimum variance of Lz calculated from the intensity of each pixel in the sky image, which is generated through the method described in Section 4.2.1. Here, V denotes a set of pixels fulfilling two conditions: they do not belong to the missing area, and they are not saturated. If γ_k denotes the direction of pixel k in a sky image, the optimal sky index Si_{opt} is estimated based on the minimum variance of Lz as follows:

$$\begin{aligned} Si_{opt} &= \underset{Si}{\arg\min} \sum_{k \in V} \left(Lz_{acq}(\gamma_k, Si) - \bar{L}z_{acq}(\gamma_k, Si) \right)^2, \\ Lz_{acq}(\gamma_k, Si) &= \frac{La_{acq}(\gamma_k)}{L(\gamma_k, Si)}, \end{aligned} \tag{4}$$

where Lz_{acq} denotes the calculated Lz from the real intensity acquired, La_{acq}, and $\bar{L}z_{acq}$ is the average of Lz_{acq} among all γ_k. Eq. (4) is equivalent to selecting Si_{opt} with the minimum error between a generated virtual sky using every Si, and a sky image unified as described in Section 4.2.1 (see Figure 7(c)). In our implementation, an optimal Si is searched from $Si = 0.0$ through $Si = 2.0$ in given increments (0.1 in our implementation) in a brute-force manner.

The intensity of the remaining missing area is calculated from Eq. (2), using Si_{opt} as Si. The intensity may vary at the boundary between areas filled in using the method described in Section 4.2.1, and areas filled in using the method detailed in Section 4.2.2; therefore, their boundary is alpha-blended into the estimated sky image. The complete sky image (see Figure 7(b)) is finally transformed into a panoramic spherical image and superimposed onto the missing areas in the input spherical images, in which the boundary of the missing area is also alpha-blended.

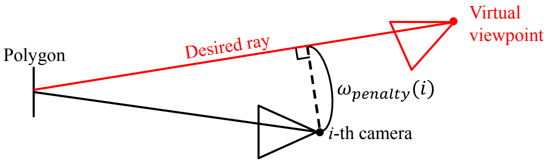


Fig. 8. Penalty definitions for offline IBR.

4.3 Generation of Real-Scene Images Using IBR

The reconstructed 3D models of large outdoor environments consist of millions of polygons, which causes a heavy amount of computations for the online rendering process. To reduce the amount of computations during the online process, our framework generates densely packed structured viewpoints during the offline process from the completed spherical images. By generating a dense arrangement of viewpoints, the 3D models are expected to be greatly simplified during the online process because of the small disparity between neighboring viewpoints; in addition, the structure of these viewpoints reduces the calculation cost for online texture blending. In our application, spherical views of the physical environment are pre-generated at every grid point on the designated plane, allowing changes in the user's viewpoint, as illustrated in Figure 5.

We implement an IBR method based on VDTM [Debevec et al. 1996], which is a common physically based IBR method. In VDTM, the images captured from multiple positions are projected and blended appropriately onto each polygon. A large hemispherical geometry, assumed to be infinite, is used as the background geometry for portions of the scene that are not reconstructed, such as the sky. The color Ip_j of the surface on the j-th polygon is determined by alpha-blending the texture color Ic_i projected from the i-th camera depending on the viewpoint to be generated. The value of Ip_j is calculated using the blending weight $\omega_{weight}(i)$, which is the reciprocal of the penalty function $\omega_{penalty}(i)$ as follows:

$$\begin{aligned} Ip_j &= \sum_i \frac{\omega_{weight}(i) \cdot Ic_i}{\sum_i \omega_{weight}(i)}, \\ \omega_{weight}(i) &= \frac{1}{\omega_{penalty}(i) + \varepsilon}. \end{aligned} \tag{5}$$

Here, $\omega_{penalty}(i)$ denotes the distance between the i-th camera and the desired ray, which is defined as a vector from the viewpoint to the center of the polygon (see Figure 8). In addition, ε is a tiny value used to avoid dividing by zero. Note that our definition of $\omega_{penalty}(i)$ is commonly used in selecting appropriate rays, such as in an image-based stereo-view generation approach based on light-ray selection [Hori et al. 2010]. This definition weights the cameras with a small angle between the desired and captured ray, as well as those close to the polygon. The visibility of the 3D shape from the i-th camera is calculated using the camera's depth maps, as in a study on VDTM using a per-pixel visibility test [Porquet et al. 2005].

The value of $\omega_{weight}(i)$ varies depending on the positions of the polygons, i.e., this process requires m penalty calculations for the i-th camera when there are m polygons. Even if the calculation process is reduced using only the k-th best cameras, as in [Buehler et al. 2001], or using a graphics processing unit (GPU) for parallelization, the rendering process with the binding and switching of thousands of textures generates a significant overhead for the graphic pipelines.

4.4 Photorealistic Superimposition of Virtual Objects

Virtual objects are rendered onto each structured real world spherical view using a commercial GI-rendering engine. We used Mentalray (Mental Images GmbH.) with 3ds Max (Autodesk, Inc.). Camera pose information estimated through SfM is used for geometric registration between the virtual objects and the real environment. Spherical MR

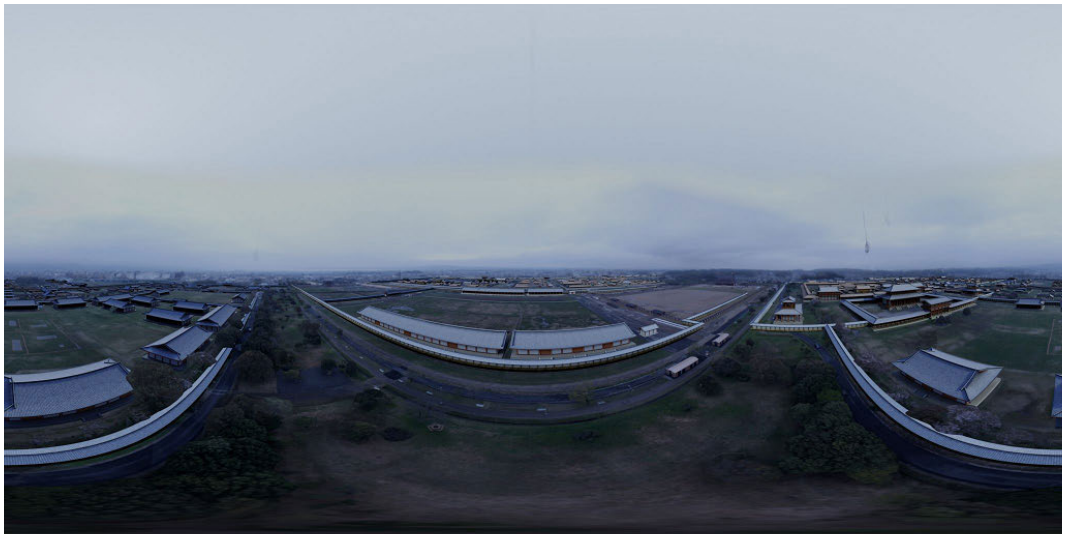


Fig. 9. Example of a spherical MR view at a structured viewpoint.

views are generated through IBL using GI rendering, as in commercial movies, using complete environmental maps and dense 3D shapes of the real world to create occlusion effects. Real world 3D shapes are also utilized for rendering shadows and ambient occlusions of virtual objects onto the real world using differential rendering. In our situation, we acquired the most plausible results using a combination of photon mapping and the final gathering of Mentalray. It should be noted that the illumination environment for virtual objects was manually edited from the original spherical images by setting up additional lights because we perceived that the illumination by the completed environmental map was slightly dark in the preliminary experiments. The allowance of manual editing is an important advantage of the proposed rendering framework.

Figure 9 shows an example of an augmented spherical image at a certain grid point. The real-world scene was generated by VDTM using completed spherical images, and the CG models of the buildings were rendered offline using IBL.

5. ONLINE PROCESS: IBR WITH REDUCED COMPUTATIONAL COST

The online process generates a planar perspective MR view from viewpoints configured freely in real-time, using spherical MR views at the structured viewpoints and simplified 3D models of both the real and virtual objects.

In this study, we enable a viewpoint change on a 2D plane using pre-generated structured viewpoints at the grid points. VDTM is improved to reduce the amount of online computations. During the online process, the views at four neighboring grid points are projected onto 3D surfaces and blended using bilinear weights, which are calculated based on the positions of the grid points and the viewpoint to be generated, as illustrated in Figure 6. Thus, Eq. (5) is modified to use the bilinear weight, $\omega_{bilinear}(k)$, as follows:

$$Ipr_j = \sum_{k=1}^{4} \omega_{bilinear}(k) \cdot Icr_k, \tag{6}$$

where Ipr_j denotes the surface color on the j-th polygon and Icr_k is the color of the pixel in the k-th $(1 \leq k \leq 4)$ neighboring view projected onto the surface. This process does not require per-polygon weight calculations because $\omega_{bilinear}(k)$ does not depend on the positions of the polygons, but only on the position of the viewpoint. It requires only four calculations for $\omega_{bilinear}(k)$ while generating an image. Using a constant weight among all polygons is beneficial

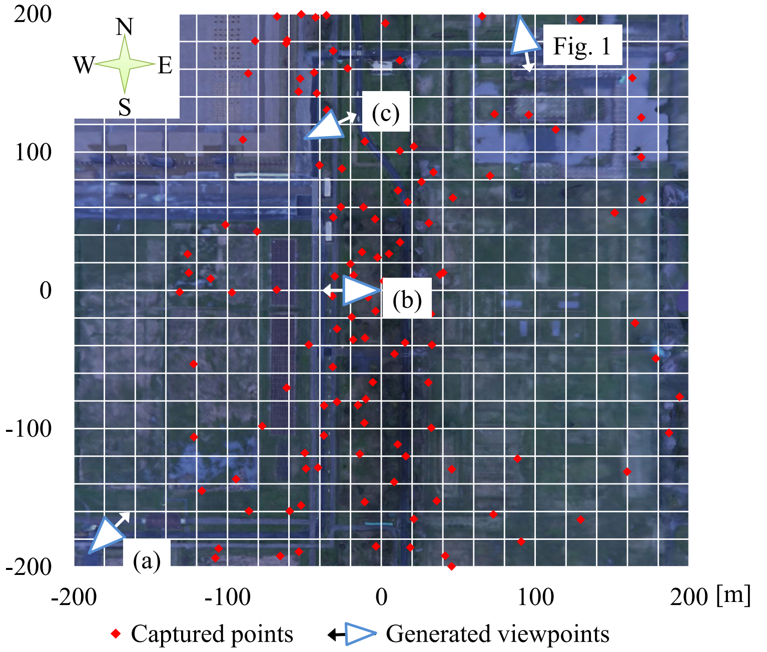


Fig. 10. Captured points of spherical images, structured viewpoints, and online-generated viewpoints. The background texture is a satellite image of the corresponding area. Structured viewpoints (the intersections of the grids) are designated in every 20m × 20m grid point area.

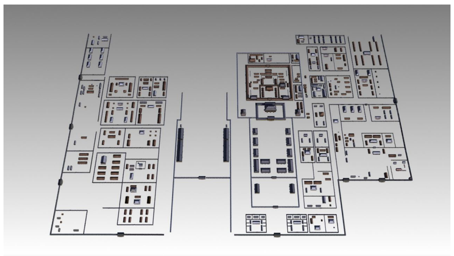


Fig. 11. 3D models of the virtual palaces in Heijo-Kyo, created by Toppan Printing Co., Ltd.

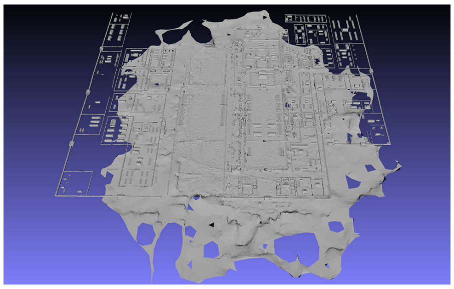


Fig. 12. Combined and simplified 3D models of the real and virtual worlds used for the online process.

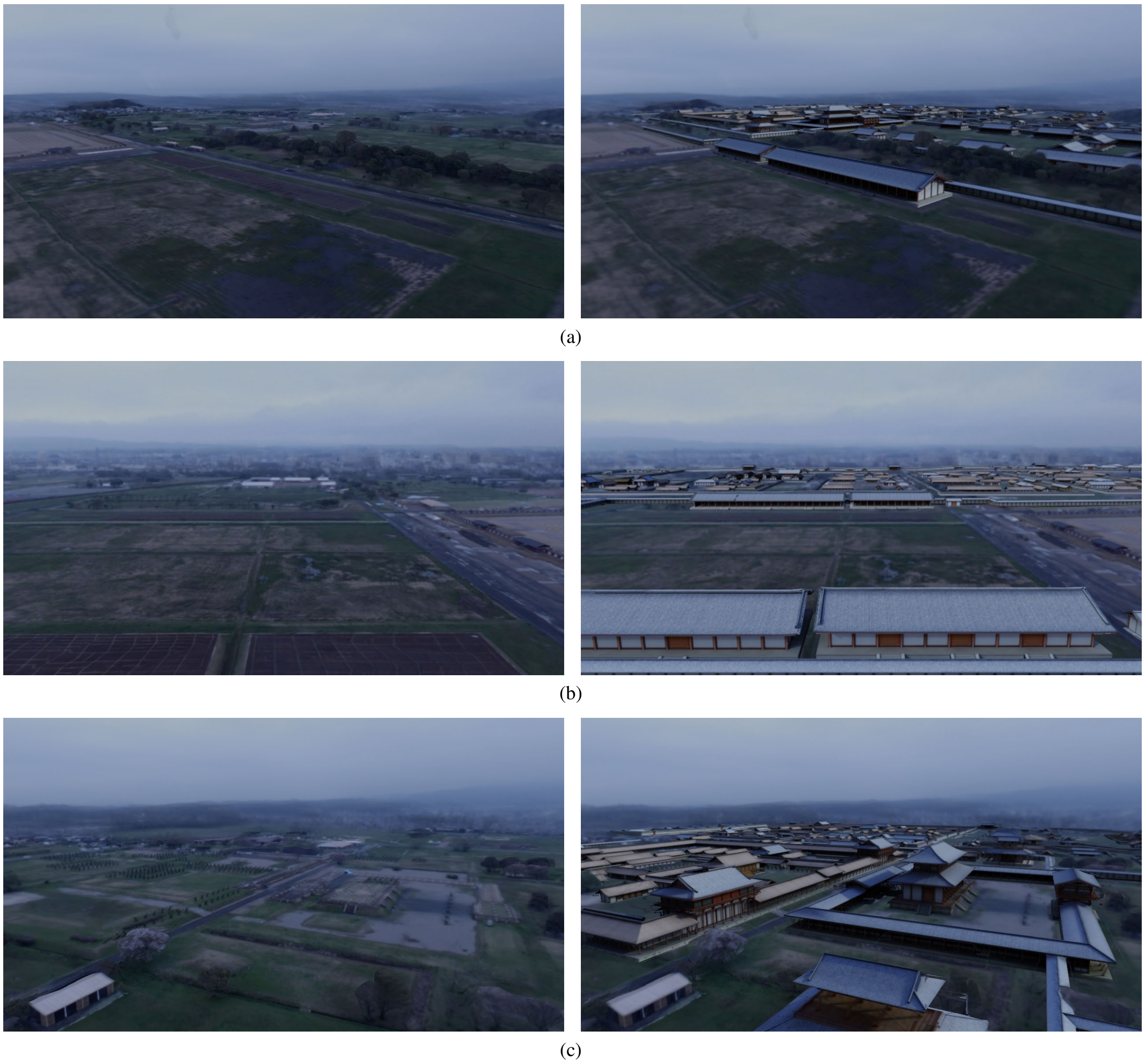
in terms of the running time. Because of this feature, the online process does not require switching and rebinding textures in GPU memory while rendering a frame, which are significant causes of overhead.

In our application, 3D models of both real and virtual objects are combined and simplified down to 1.4% of the original number of polygons using a quadric-based mesh decimation method [Garland and Heckbert 1997]. Although the simplification of the 3D models does not reduce the cost of calculating $\omega_{bilinear}(k)$, the rendering cost for the 3D models in a graphics pipeline is linearly reduced, depending on the number of polygons. The online process can be easily implemented on a GPU, such as through the use of OpenGL Shading Language (GLSL), with a per-pixel visibility test [Porquet et al. 2005] using the depth maps at the structured viewpoints.

6. EXPERIMENTS

6.1 Setting and Results

To confirm that the application based on the proposed framework generates the appropriate scenes in a practical environment, augmented free-viewpoint scenes were generated on a 400m × 400m plane at an altitude of approximately 50 m from the ground. The application setting was based on the palace site of Heijo-Kyo in Nara, the ancient capital city of Japan, where the original palaces no longer exist. The grid points, which were used for the positions of the structured viewpoints, were designated for every 20m × 20m area. The positions of the captured spherical images and the configured grid are shown in Figure 10. The input spherical images were captured at various altitudes, and the altitude of the grid plane was designated as the average altitude of the captured points. The superimposed virtual buildings of Heijo-Palace, shown in Figure 11, were approximately 1 km in length from east to west and 1.3 km from north to south, and consisted of 4,255,650 polygons. The real-world environment was reconstructed using 3,290,880 polygons from 174 spherical images. During the online process, 3D models of both real and virtual worlds were combined and



(a)

(b)

(c)

Fig. 13. Free-viewpoint images from various viewpoints, shown in Figure 10, with (right) and without (left) the photorealistic superimposition of virtual objects.

simplified into a model of 104,054 polygons (approximately 1.4% of the original 3D model), as shown in Figure 12. Examples of augmented images generated during the online process are shown in Figures 1 and 13. The viewpoint of each of these images is illustrated in Figure 10.

6.2 Performance

The entire offline process was performed on a desktop PC with an Intel Core i7-3930K (3.20GHz, six cores), 56.0GB of RAM, and an NVIDIA GeForce GTX 690 (2 GB of texture memory). Offline IBR described in Section 4.3 was implemented using CUDA. Table I describes the running time for each step of the offline process for this experiment.

The online process was implemented along with a per-pixel visibility test process using GLSL on two devices: 1) a desktop PC identical with that used in the offline process, and 2) a tablet device equipped with an Intel Core i7 3667U (2 GHz, two cores), 8 GB of RAM, and an Intel HD Graphics 4000 CPU-integrated graphics processor. Table II describes the performance of our online IBR using the simplified model as well as other rendering methods. The frame rates for rendering onto an internal frame buffer (1280×720 pixels) were measured and averaged under the same sequence of viewpoint movement. Simple rendering draws a texture-less model with a smooth shading of OpenGL. A full model refers to a combination of real and virtual models without simplification, which consist of $4,255,650 + 3,290,880 = 7,546,530$ polygons. The online performance of our application was 10- to 100-times faster than the simple rendering of the 3D model without simplification. This indicates that our approach combining the lightweight IBR and model simplification can be a good substitute for traditional model-based rendering in terms of performance. In addition, our online IBR with the full model showed a better performance (7.0 fps) than the offline IBR with a real world model without simplification (3,290,880 polygons), which performed approximately 0.7 fps with a CUDA implementation as shown in Table I. This implies that constant weights among all polygons in the online IBR contributed to the improvement in performance.

7. DISCUSSIONS

This section describes the quality of MR views and the effects of density of pre-generated viewpoints. We also discuss the presentation of non-Lambertian objects, as well as the interactivity, portability, and limitations of our framework.

7.1 Quality of Real-World Views on Structured Viewpoints

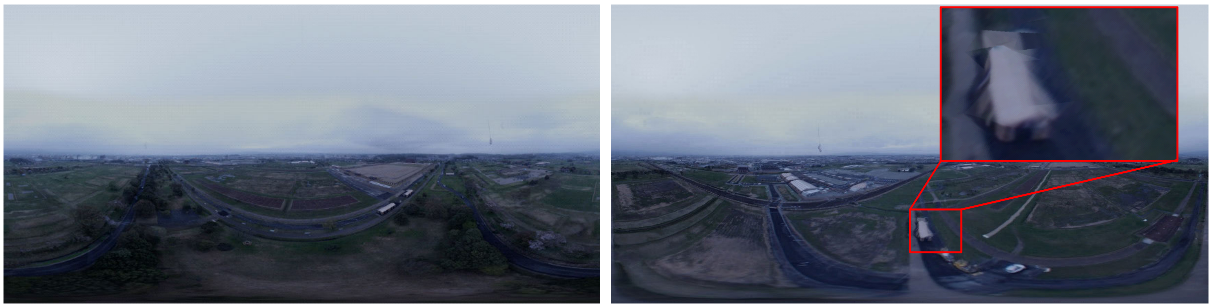
Because the density of the input spherical images may affect the quality of pre-generated real-world views, we compared the real-world views generated from a grid point where the input spherical images were densely captured to those of a grid point generated far from any captured points. Figure 14(a) shows a generated viewpoint where the input images were densely captured ((0, 0) in Figure 10); the viewpoint was successfully generated without artifacts owing to the use of IBR. Blurring appears in the images shown in Figure 14(b), far from any captured points ((-200, -200) in Figure 10), owing to the errors in the reconstructed 3D surfaces and the low-resolution textures that were projected from the distant cameras. This indicates that planning the capturing positions of the images is also important for improving the appearance of the real-world virtualization in our proposed framework. For 3D reconstruction using automated vehicles, view planning techniques for determining the next-best view and efficiently planning the

Table I. Running time of the offline processes. 3D reconstruction and environmental map completion take up the entire time spent for generating 174 spherical panoramas. Additional time required for the other processes include the total time taken for generating 441 grid points, each of which are cube-maps consisting of six images. Note that, in addition to this list, other manual processes required additional hours.

Process	Time (minutes)
3D reconstruction (SfM+MVS)	118
Environmental map completion	66
Generation of real scene images using offline IBR	63
Superimposition of virtual objects	1797

Table II. The frame rates (fps) of the online IBR using the simplified model and other rendering methods.

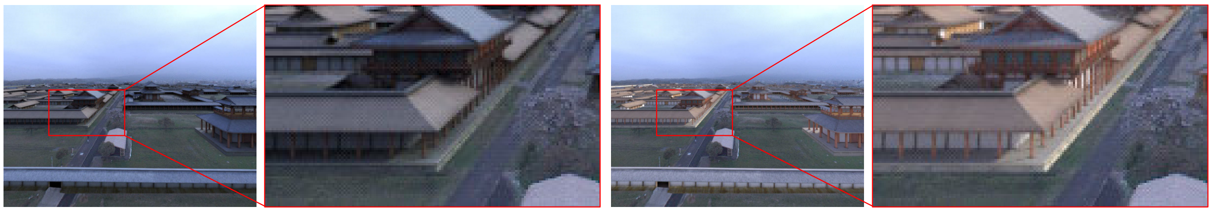
	Online IBR + simplified model (proposed)	Online IBR + full model	Simple rendering + simplified model	Simple rendering + full model
Desktop PC w/ NVIDIA GeForce GTX 690	639	7.0	721	8.7
Tablet PC w/ Intel HD Graphics	60	3.5	96	4.1



(a) Viewpoint at position (0, 0) within densely captured area shown in Figure 10.

(b) Viewpoint at (-200, -200) far from captured points shown in Figure 10.

Fig. 14. Spherical real-world images pre-generated at the grid points.



(a) Using real-world illumination, an average score of 4.4 was achieved.

(b) Using only parallel light, an average score of 2.1 was achieved.

Fig. 15. Sampled frames from the sequence used for an evaluation of the offline superimposition. Ten participants evaluated the naturalness of the synthesis of the real and virtual objects using a five-point scale (1, highly unnatural; 5, highly natural).

capturing paths have been studied [Pito 1999; Bottino and Laurentini 2006]. Such approaches can be adapted for data acquisition for IBR with further investigation on the quality of generated images.

7.2 Quality of Virtual Object Superimposition during Offline Process

We conducted a small subjective evaluation of the quality of offline superimposition of virtual objects. We compared an augmented video sequence generated by IBL using an environmental map that was completed by our application, with a conventional sequence created using light parallel to the angle of the sun as calculated based on the date and time of the captured sequence. Figure 15 shows two sampled frames in the sequence used in the evaluation. Note that to eliminate the effects of IBR and evaluate the superimposition quality itself, the IBR technique was not used to generate the sequences utilized in this evaluation; i.e., virtual objects were superimposed on completed spherical images. After watching both sequences, ten participants in their twenties or thirties evaluated the level of naturalness of the real- and virtual-object synthesis based on a five-point scale (1, highly unnatural; 5, highly natural). Consequently, the sequence generated using IBL and a completed environmental map received an average score of 4.4, while the conventional sequence received an average score of 2.1, which is a significant difference ($p < 0.01$, t-test). The process used for offline rendering based on IBL was more effective compared to the use of an ordinary illumination environment, in which the strength and distribution of the skylight were not carefully considered.

7.3 Density of Structured Viewpoints

Although the use of a larger interval of structured viewpoints can reduce the storage usage for pre-generated images and the memory usage during the online process, it can degrade the appearance of the resultant images. We performed a subjective experiment to investigate the effects of viewpoint density by changing the grid size.

Although the use of a larger interval of structured viewpoints can reduce the memory and storage usage during the online process, it can degrade the appearance of the resultant images. Table III shows the relation between grid size and the number of images. Because the proposed rendering requires a huge amount of data for a small grid size, it is important to investigate the trade-off between the data amount and image quality, and find useful trends for determining the appropriate grid size.

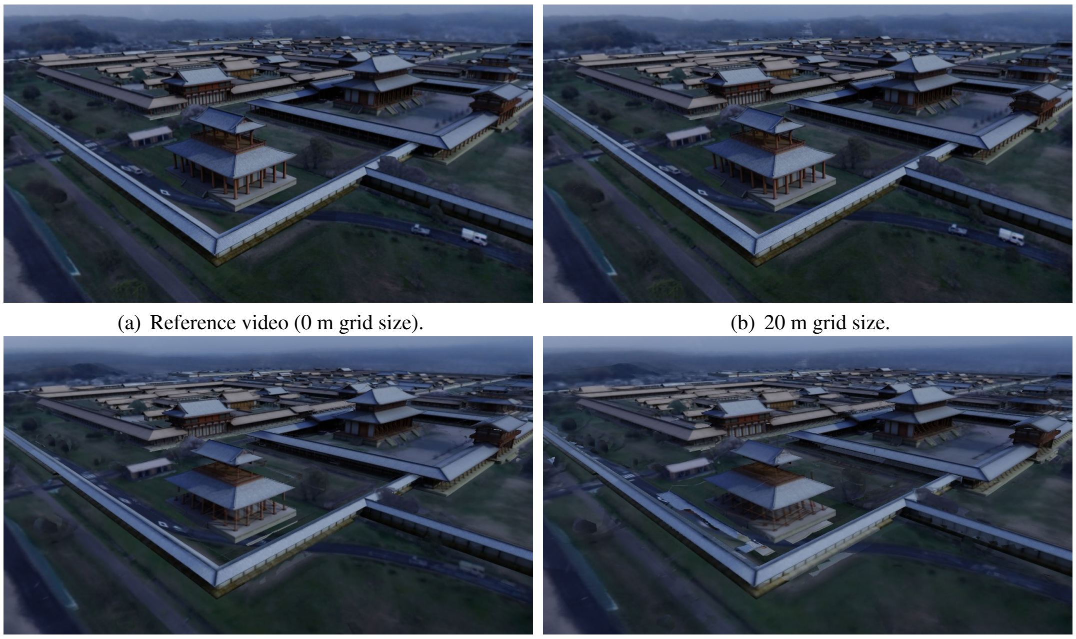
In this experiment, 14 participants in the age group of 20-30 years scored the naturalness of MR video sequences. The MR videos shown to the participants were generated by our application using six different grid sizes of 10, 20, 40, 60, 80, and 100 m. In addition, we prepared a reference video without utilizing structured viewpoints; that is, virtual objects were directly rendered onto every frame in video sequences where real-world views were generated using VDTM in a frame-by-frame manner during the offline process. This is equivalent to generating an infinite number of grid points for a 0 m grid size. We prepared two view directions for each grid size: downward and horizontal. Sampled frames from each video sequence are shown in Figures 16 and 17. The participants scored each sequence on a scale of 0 (worst) to 100 (best) using a trackbar interface. To evaluate the quality of our application relative to the reference video, the score of the reference video was fixed at 50.

Figure 18 shows the average naturalness scores achieved through the experiment. To remove the outliers, 20% of the scores (highest and lowest 10%) were ignored to calculate the average score and standard deviation. The results indicate that the naturalness decreased with a larger grid size. We performed a Williams' multiple comparison test ($p < 0.05$), in which the score of the control group (i.e., 50) was set as the score of the reference video, assuming a monotonic decrease in the score distribution. The scores for a grid size of larger than 60 m were significantly unnatural compared with that of the reference video, which was generated with grid size of 0 m. This trend also occurred between the horizontal and downward view directions. A 20 m grid size, which was used for our experiment described in Section 6, showed a similar performance as a smaller (i.e., 10 m) grid, as well as the reference video, under our experimental environment. Although this comparison result cannot be directly used for determining the best grid size, which is a trade-off between the quality and number of pre-generated images, it indicates that a larger grid size (i.e., larger than 60 m in this experiment) causes a significantly negative effect on the user experience.

Figure 19 shows the negative effects from the use of a larger grid, i.e., the low resolution of the entire scene, and the skewing/blurring of the virtual building posts. When a larger grid was used, the resolution of the resultant images was lower owing to the textures projected from the distant viewpoints. The skewing and blurring of the building posts were due to the simplified 3D models, whose posts were removed through a simplification; therefore, the effects became more pronounced for the larger grid. In addition, a significantly large grid occasionally generated clearly incorrect textures because some surfaces of the complex 3D shapes were not visible from all four neighboring grid points. As a trade-off between a reduction in the number of images and the appearance of the generated images, a promising

Table III. Amount of data required to prepare a 400m × 400m grid structure. The structured views are assumed to be stored as 8-bit 4-channel (RGB and depth) cube-maps of 512 × 512 pixels.

Grid size	# of grid points	Data amount (not compressed)
10m	1681	10086 MB
20m	441	2646 MB
40m	121	726 MB
80m	36	216 MB
100m	25	150 MB
200m	9	54 MB



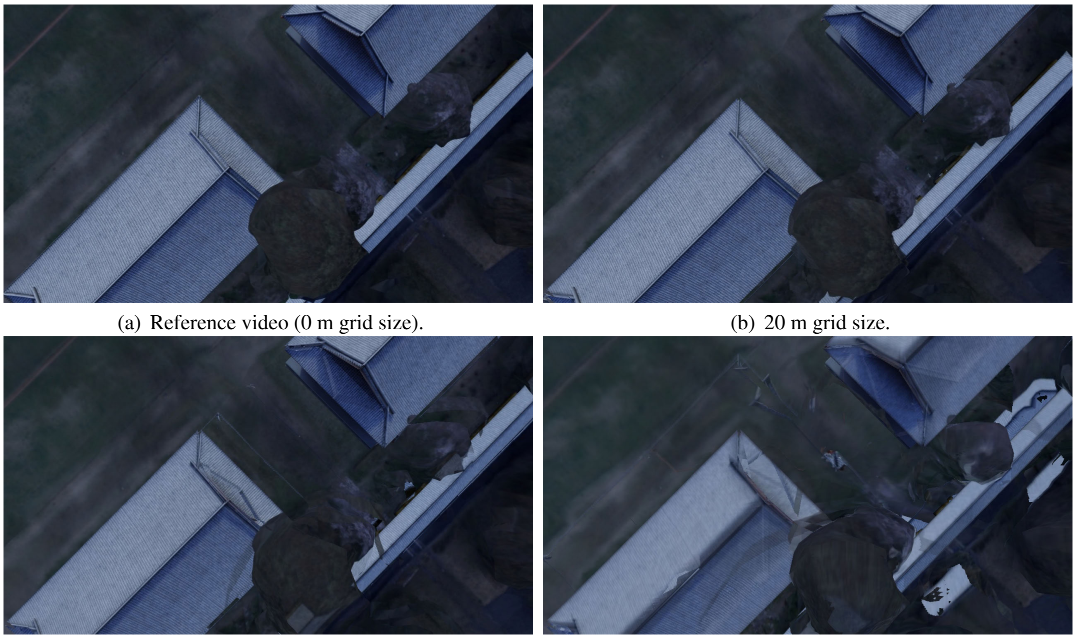
(a) Reference video (0 m grid size).

(b) 20 m grid size.

(c) 60 m grid size.

(d) 100 m grid size.

Fig. 16. MR views from the same viewpoint with a change in grid size (horizontal view direction).



(a) Reference video (0 m grid size).

(b) 20 m grid size.

(c) 60 m grid size.

(d) 100 m grid size.

Fig. 17. MR views from the same viewpoint with a change in grid size (downward view direction).

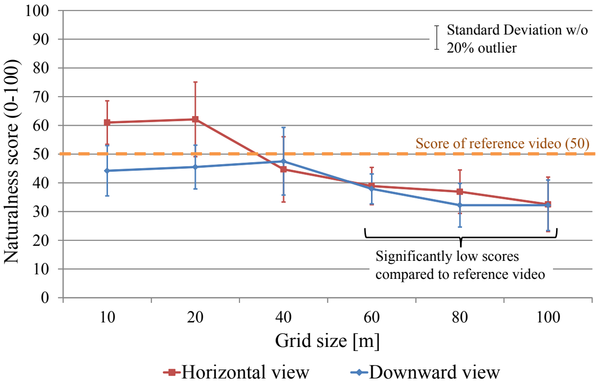


Fig. 18. Naturalness scores of augmented video sequences with a change in grid size.

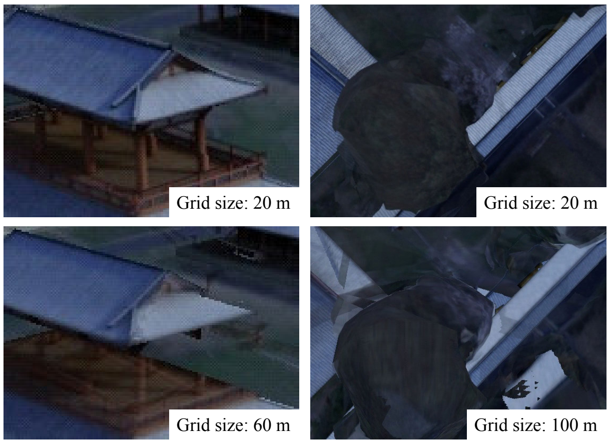


Fig. 19. Negative effects using larger grid sizes.

solution is to change the density of the structured viewpoints, depending on the complexity of the model and the distance to the objects, using quad-trees or similar structures.

7.4 Non-Lambertian Objects

The performance of the online IBR indicates that one benefit of our framework is a low computational cost owing to the use of simplified 3D models and the lightweight IBR. This benefit is largely due to an IBR feature that suppresses the negative visual effects occurring from errors in the simplified 3D shapes [Debevec et al. 1996]. Meanwhile, another inherent IBR feature offers an additional potential advantage to our framework, in which, unlike the traditional rendering of textured models, IBR can present non-Lambertian effects such as reflectance and specularity. Although the real and virtual objects used in our experiment consisted of nearly Lambertian surfaces, this issue is worth investigating further.

Although our framework can present non-Lambertian effects in principle, it is easy to assume that the presentation of sharp specular effects is impractical. The fidelity of IBR to non-Lambertian effects is basically determined by how densely the images are captured, i.e., if the environment is densely captured, IBR can present sharp specular effects. In our framework, it is impractical to capture a large outdoor environment in a dense manner as a way to handle sharp effects on real objects. Pre-generating densely structured viewpoints allows such effects to be realized on the surfaces of virtual objects; however, this greatly increases the amount of memory and storage usage required. The difficulty in presenting sharp non-Lambertian effects is common to most IBR approaches for large environments, which intend to suppress the negative effects of 3D shape errors.

7.5 Limitations

7.5.1 *Rendering framework.* The proposed framework has certain limitations. First, the number of pre-generated images may be large. This issue is common to IBR methods. To address this limitation, multi-view image compression approaches [Magnor et al. 2003; Shum et al. 2003; Lam et al. 2004] for IBR can be employed. Second, although dynamic objects can be superimposed using this framework if the whole animation is pre-rendered, it requires an enormous number of images to be prepared. Our framework can be combined with ordinary AR rendering methods; thus, a promising approach is to render dynamic objects using an ordinary rendering method, and static objects using the IBR framework. The third limitation is that artifacts can occur in the resultant images if the appearance of the real-world changes (e.g., by changes in illumination or dynamic objects) during the aerial imaging. In our application, we actually found that some dynamic objects on the ground (e.g., cars) faded in/out while moving the viewpoint.

The proposed rendering framework can be potentially employed for MR world exploration applications using a static environment. Although there are limitations when using the proposed framework under a dynamic environment, this framework is a promising approach for a variety of applications, such as landscape simulations. A historical scene is a typical example of such an application.

7.5.2 *Environmental map completion.* Synthetic skies estimated using the sky model differ from the real sky conditions, and thus IBL based on a completed environmental map can generate an unnatural synthesis. A recent study [Okura et al. 2014] reported that a synthetic sky, estimated in a similar manner [Okura et al. 2011] as described in this study, differed greatly from the actual intensity observed by the skyward camera. In their experiment conducted under a sunny sky, the average intensity of the synthetic model was approximately one quarter of the actual intensity. To address this, an additional skyward camera can be employed on top of the aerial vehicle [Okura et al. 2014] or on the ground to acquire the accurate radiance of the sky. Even without the use of additional cameras as in our situation, because of the inherent capability of our offline process for manual operations, lighting environments for virtual objects can be manually edited to achieve a satisfactory synthesis.

7.6 Interactivity and Portability

The essential functions required for MR world exploration applications are photorealism, interactivity, and portability, as discussed in Section 1. Our framework generates a photorealistic MR world. This section briefly discusses the two remaining functions of our framework: interactivity and portability.

7.6.1 *Interactivity.* Our framework offers users a certain amount of interactivity by realizing free-viewpoint navigation in a given space. This is valuable for landscape simulation applications. A possible direction for increasing the interactivity is to allow a modification of the world. As discussed in Section 7.5, combining the proposed technique with existing or additional techniques can help realize such modifications, e.g., the placing of additional objects. In terms of historical application, allowing users to change the time and age of an MR world is another promising option. Preparing multiple databases can help realize a change of age if the storage capacity allows it. With some modifications, relighting techniques [Laffont et al. 2013] may achieve a change in the time of day.

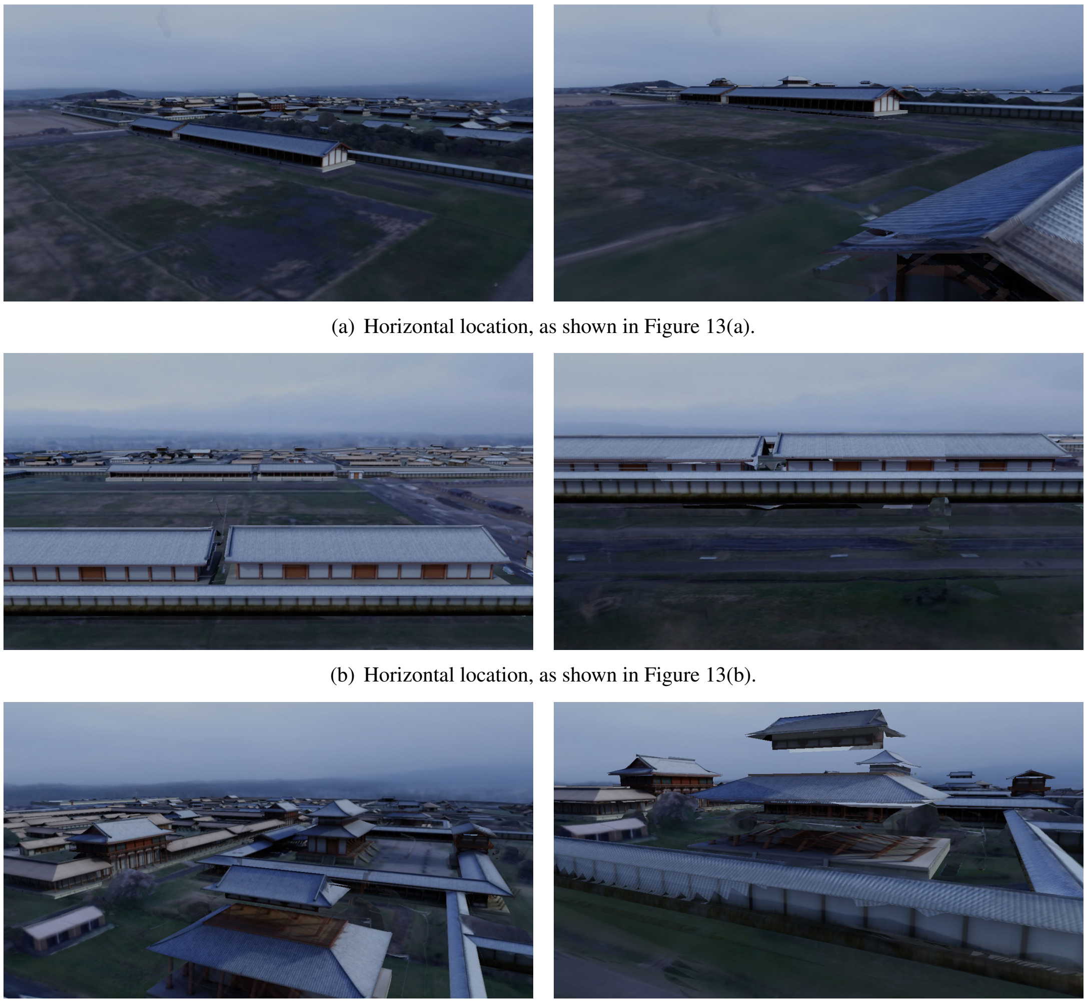
7.6.2 *Portability.* Our framework can be conducted on a high-end tablet device. It can also be potentially operated on general mobile devices such as an iPad or web-browser based interface after overcoming the memory and storage issues. Although a large network bandwidth may be required, a client server model similar to Google Earth, where the client receives pre-rendered images neighboring the user's viewpoint, is a promising means of addressing this issue.

8. EXTENSIONS FOR 3D VIEWPOINT CHANGE

The proposed framework can be extended for use with 3D structures through a straightforward approach generating multiple grids at multiple altitudes. This section describes the implementation and results of the 3D free-viewpoint MR application.

8.1 Implementation

Augmented free-viewpoint scenes were generated on three grids at multiple altitudes, each of which covers a 400m × 400m area. In addition to the grid used for the 2D application described in Section 6, with an altitude that was set at approximately 50 m from the ground, we also generated grids at 10 and 30 m altitudes in the same manner as in the offline process for the 2D application. The results of the experiment described in Section 7.3 show that there was no significant degradation in the free-viewpoint images for the 40 m grid, the grid points of which were designated for each 40m × 40m area, which is larger than that used in the previous application. The horizontal and vertical intervals of the structured viewpoints are different (40 m and 20 m). This is because the target environment includes some



(a) Horizontal location, as shown in Figure 13(a).

(b) Horizontal location, as shown in Figure 13(b).

(c) Horizontal location, as shown in Figure 13(c).

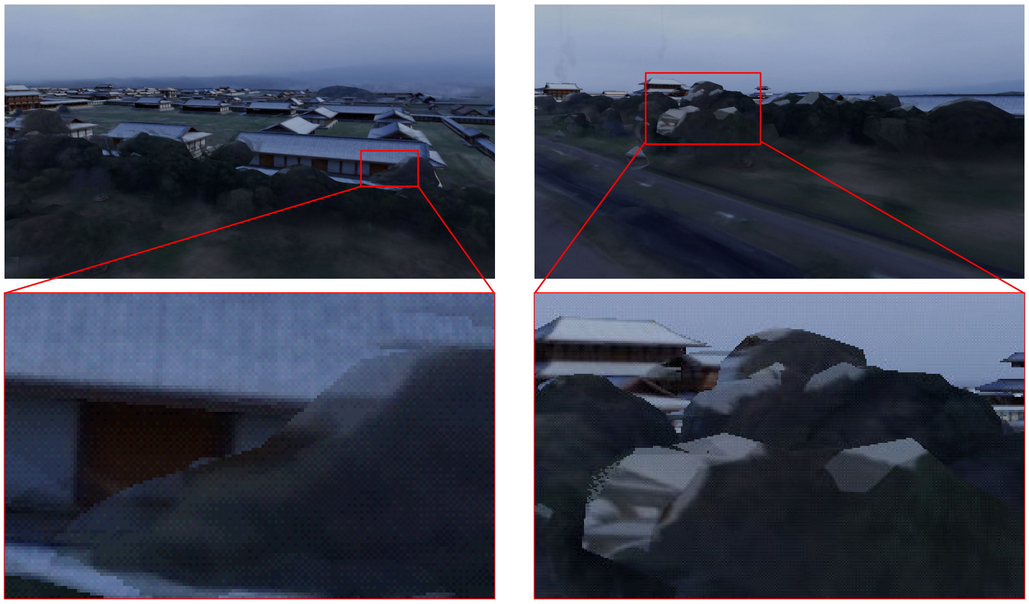
Fig. 20. Free-viewpoint images from altitudes approximately 40 m (left figure) and 20 m (right figure) from the ground.

two-story buildings, and missing textures clearly occurred for a large vertical interval of the viewpoints. Note that the differences in interval do not increase the complexity of the rendering process. In the application using a 3D structure, the online VDTM blends the textures at eight neighboring structured viewpoints, while the 2D application uses four textures as shown in Figure 6. The eight neighboring structured viewpoints are placed at each vertex of the cube containing the user's viewpoint. The blending weights are independent of vertex positions.

8.2 Results and Discussions

Examples of augmented images generated during the online process are shown in Figure 20. This application was implemented on the same desktop PC as used in the experiment described in Section 6, and achieved a similar performance as the 2D application (faster than 600 fps).

MR views were successfully generated, but the views from a low altitude include some visual artifacts. This is assumed to be caused by the following two problems: 1) the user's viewpoint is relatively close to the object shape at



(a) Viewing a complex shape from a high altitude (approximately 50 m from the ground).

(b) Viewing a complex shape from a low altitude (approximately 15 m from the ground).

Fig. 21. Artifacts on the occlusion boundary.

low altitude, and errors in shape owing to the 3D reconstruction and polygon reduction largely affect the appearance of the image, and 2) some parts of the complex 3D shapes were not visible from all neighboring structured viewpoints. Artifacts occurring from problem 1) appear in the occlusion boundary, as shown in Figure 21. The errors in shape cause blurring on the occlusion boundary from scenes viewed from a high altitude. The artifacts become larger at low altitude; i.e., clearly incorrect textures are generated. These negative visual effects can be reduced by designating a narrow grid if a larger number of images can be prepared.

9. CONCLUSIONS AND FUTURE WORK

This paper has proposed a novel framework for the photorealistic superimposition of static virtual objects into the setting of a real world virtualization. Offline rendering and physically based IBR are combined to preserve the quality of offline rendering in a free-viewpoint MR environment which provides users with freely configurable viewpoints. The computational cost of the online process is highly reduced through the pre-generation of structured viewpoints and the pre-rendering of virtual objects at these viewpoints.

We introduced a practical implementation of a fly-through application based on the proposed framework using spherical images captured from the sky. This application provides a free configuration of user's viewpoints on a 2D plane through a light-weight VDTM technique using pre-generated viewpoints at the grid points. In experiments conducted at a historical site, our implementation demonstrated a high performance. These experiments also illustrated the effects of rendering using IBL during the offline process, which improves the appearance of virtual objects, as well as some problems that should be examined further, such as the determination of the most effective grid sizes. Increasing the dimension easily leads to an enormous number of images. This issue should be considered when developing large applications.

In future work, real-time AR applications based on our framework should be developed because the proposed framework can be employed to resolve photometric registration problems in real-time AR applications using static virtual objects. In addition, by employing a web-based interface, the portability of our application can further be improved. We plan to develop a web-browser based MR world exploration application using WebGL.

ACKNOWLEDGMENT

This research was supported by the Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Scientific Research (A), No. 23240024, Grant-in-Aid for JSPS Fellows No. 25-7448, and by the "Ambient Intelligence" project funded by Ministry of Education, Culture, Sports, Science and Technology (MEXT). The CG models of the Heijo-Palace are courtesy of Toppan Printing Co., Ltd.

REFERENCES

Sameer Agarwal, Noah Snavely, Steven M. Seitz, and Richard Szeliski. 2010. Bundle adjustment in the large. In *Proc. 11th European Conf. on Computer Vision (ECCV'10)*. Crete, Greece, 29–42.

Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. 2009. Building Rome in a day. In *Proc. 12th IEEE Int'l Conf. on Computer Vision (ICCV'09)*. Kyoto, Japan, 72–79.

Ronald Azuma. 1997. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments* 6, 4 (1997), 355–385.

Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. 2001. Recent advances in augmented reality. *IEEE Computer Graphics and Applications* 21, 6 (2001), 34–47.

Atsuhiko Banno, Tomohito Masuda, Takeshi Oishi, and Katsushi Ikeuchi. 2008. Flying laser range sensor for large-scale site-modeling and its applications in Bayon digital archival project. *Int'l Journal of Computer Vision* 78, 2 (2008), 207–222.

Andrea Bottino and Aldo Laurentini. 2006. What's NEXT? An interactive next best view approach. *Pattern Recognition* 39, 1 (2006), 126–132.

Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. 2001. Unstructured lumigraph rendering. In *Proc. ACM SIGGRAPH'01*. Los Angeles, CA, 425–432.

Declan Butler. 2006. Virtual globes: The web-wide world. *Nature* 439, 7078 (2006), 776–778.

Paul Debevec. 1998. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proc. ACM SIGGRAPH'98*. Orlando, FL, 189–198.

Paul Debevec, Yizhou Yu, and George Borshukov. 1998. Efficient view-dependent image-based rendering with projective texture-mapping. In *Proc. Ninth Eurographics Workshop on Rendering (EGWR'98)*. Vienna, Austria, 105–116.

Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. 1996. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proc. ACM SIGGRAPH'96*. New Orleans, LA, 11–20.

Marco Di Benedetto, Fabio Ganovelli, M Balsa Rodriguez, A Jaspe Villanueva, Roberto Scopigno, and Enrico Gobbetti. 2014. Exploremaps: efficient construction and ubiquitous exploration of panoramic view graphs of complex 3d environments. *Computer Graphics Forum* 33, 2 (2014), 459–468.

Alessandro E Foni, George Papagiannakis, and Nadia Magnenat-Thalmann. 2010. A taxonomy of visualization strategies for cultural heritage applications. *ACM Journal on Computing and Cultural Heritage* 3, 1 (2010), 1:1–1:21.

Michael Garland and Paul S. Heckbert. 1997. Surface simplification using quadric error metrics. In *Proc. ACM SIGGRAPH'97*. Los Angeles, CA, 209–216.

Payam Ghadirian and Ian D. Bishop. 2008. Integration of augmented reality and GIS: A new approach to realistic landscape visualisation. *Landscape and Urban Planning* 86 (2008), 226–232.

Simon Gibson, Jon Cook, Toby Howard, and Roger Hubbold. 2003. Rapid shadow generation in real-world lighting environments. In *Proc. 14th Eurographics Symp. on Rendering (EGSR'03)*. Leuven, Belgium, 219–229.

Simon Gibson and Alan Murta. 2000. Interactive rendering with real-world illumination. In *Proc. 11th Eurographics Workshop on Rendering (EGWR'00)*. Brno, Czech Republic, 365–376.

Thomas Gierlinger, Daniel Danch, and André Stork. 2010. Rendering techniques for mixed reality. *Journal of Real-Time Image Processing* 5, 2 (2010), 109–120.

Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. 1996. The Lumigraph. In *Proc. ACM SIGGRAPH'96*. New Orleans, LA, 43–54.

Thorsten Grosch. 2005. PanoAR: Interactive augmentation of omnidirectional images with consistent lighting. In *Proc. Computer Vision/Computer Graphics Collaboration Techniques and Applications (Mirage'05)*. Rocquencourt, France, 25–34.

Michael Haller, Stephan Drab, and Werner Hartmann. 2003. A real-time shadow approach for an augmented reality application using shadow volumes. In *Proc. 10th ACM Symp. on Virtual Reality Software and Technology (VRST'03)*. Osaka, Japan, 56–65.

Maiya Hori, Masayuki Kanbara, and Naokazu Yokoya. 2010. Arbitrary stereoscopic view generation using multiple omnidirectional image sequences. In *Proc. 20th IAPR Int'l Conf. on Pattern Recognition (ICPR'10)*. Istanbul, Turkey, 286–289.

Norio Igawa, Yasuko Koga, Tomoko Matsuzawa, and Hiroshi Nakamura. 2004. Models of sky radiance distribution and sky luminance distribution. *Solar Energy* 77 (2004), 137–157.

Michal Jancosek and Tomás Pajdla. 2011. Multi-view reconstruction preserving weakly-supported surfaces. In *Proc. 2011 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'11)*. Colorado Springs, CO, 3121–3128.

Henrik Wann Jensen. 1996. Global illumination using photon maps. In *Proc. Seventh Eurographics Workshop on Rendering (EGWR'96)*. Porto, Portugal, 21–30.

Peter Kán and Hannes Kaufmann. 2012. High-quality reflections, refractions, and caustics in augmented reality and their contribution to visual coherence. In *Proc. 11th IEEE Int'l Symp. on Mixed and Augmented Reality (ISMAR'12)*. Atlanta, GA, 99–108.

Masayuki Kanbara and Naokazu Yokoya. 2002. Geometric and photometric registration for real-time augmented reality. In *Proc. First Int'l Symp. on Mixed and Augmented Reality (ISMAR'02)*. Darmstadt, Germany, 279–280.

Sing Bing Kang, Richard Szeliski, and P. Anandan. 2000. The geometry-image representation tradeoff for rendering. In *Proc. 2000 IEEE Int'l Conf. on Image Processing (ICIP'00)*, Vol. 2. Vancouver, BC, 13–16.

Martin Knecht, Christoph Traxler, Oliver Mattausch, Werner Purgathofer, and Michael Wimmer. 2010. Differential instant radiosity for mixed reality. In *Proc. Ninth IEEE Int'l Symp. on Mixed and Augmented Reality (ISMAR'10)*. Seoul, Korea, 99–107.

Hideyuki Kume, Takafumi Taketomi, Tomokazu Sato, and Naokazu Yokoya. 2010. Extrinsic camera parameter estimation using video images and GPS considering GPS positioning accuracy. In *Proc. 20th IAPR Int'l Conf. on Pattern Recognition (ICPR'10)*. Istanbul, Turkey, 3923–3926.

Pierre-Yves Laffont, Adrien Bousseau, and George Drettakis. 2013. Rich intrinsic image decomposition of outdoor scenes from multiple views. *IEEE Trans. on Visualization and Computer Graphics* 19, 2 (2013), 210–224.

Ping-Man Lam, Chi-Sing Leung, and Tien-Tsin Wong. 2004. A compression method for a massive image data set in image-based rendering. *Signal Processing: Image Communication* 19, 8 (2004), 741–754.

Philipp Lensing and Wolfgang Broll. 2012. Instant indirect illumination for dynamic mixed reality scenes. In *Proc. 11th IEEE Int'l Symp. on Mixed and Augmented Reality (ISMAR'12)*. Atlanta, GA, 109–118.

Marc Levoy and Pat Hanrahan. 1996. Light field rendering. In *Proc. ACM SIGGRAPH'96*. New Orleans, LA, 31–42.

Maxime Lhuillier. 2011. Fusion of GPS and structure-from-motion using constrained bundle adjustments. In *Proc. 2011 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'11)*. Colorado Springs, CO, 3025–3032.

Marcus Magnor, Prashant Ramanathan, and Bernd Girod. 2003. Multi-view coding for image-based rendering using 3-D scene geometry. *IEEE Trans. on Circuits and Systems for Video Technology* 13, 11 (2003), 1092–1106.

Takeshi Naemura, Takahide Takano, Masahide Kaneko, and Hiroshi Harashima. 1997. Ray-based creation of photo-realistic virtual world. In *Proc. Third Int'l Conf. on Virtual Systems and Multimedia (VSMM'97)*. Geneva, Switzerland, 59–68.

Fumio Okura, Masayuki Kanbara, and Naokazu Yokoya. 2010. Augmented telepresence using autopilot airship and omni-directional camera. In *Proc. Ninth IEEE Int'l Symp. on Mixed and Augmented Reality (ISMAR'10)*. Seoul, Korea, 259–260.

Fumio Okura, Masayuki Kanbara, and Naokazu Yokoya. 2011. Fly-through Heijo Palace Site: Augmented telepresence using aerial omnidirectional videos. In *Proc. ACM SIGGRAPH'11 Posters*. Vancouver, BC, 78.

Fumio Okura, Masayuki Kanbara, and Naokazu Yokoya. 2014. Aerial full spherical imaging and display. *Virtual Reality* (2014). DOI:http://dx.doi.org/10.1007/s10055-014-0249-x

Matt Pharr and Greg Humphreys. 2010. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, Burlington, MA.

Richard Pito. 1999. A solution to the next best view problem for automated surface acquisition. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 21, 10 (1999), 1016–1030.

Marc Pollefeys, David Nistér, Jan-Michael Frahm, Amir Akbarzadeh, Philippos Mordohai, Brian Clipp, Chris Engels, David Gallup, S-J Kim, and Paul Merrell. 2008. Detailed real-time urban 3D reconstruction from video. *Int'l Journal of Computer Vision* 78, 2 (2008), 143–167.

Damien Porquet, Jean-Michel Dischler, and Djamchid Ghazanfarpour. 2005. Real-time high-quality view-dependent texture mapping using per-pixel visibility. In *Proc. Third Int'l Conf. on Computer Graphics and Interactive Techniques in Australasia and South East Asia (GRAPHITE'05)*. Dunedin, New Zealand, 213–220.

Steven M. Seitz and Charles R. Dyer. 1996. View morphing. In *Proc. ACM SIGGRAPH'96*. New Orleans, LA, 21–30.

Heung-Yeung Shum, Sing Bing Kang, and Shing-Chow Chan. 2003. Survey of image-based representations and compression techniques. *IEEE Trans. on Circuits and Systems for Video Technology* 13, 11 (2003), 1020–1037.

Greg J. Ward. 1994. The RADIANCE lighting simulation and rendering system. In *Proc. ACM SIGGRAPH'94*. Orlando, FL, 459–472.

Robert M. Wolk. 2008. Utilizing Google Earth and Google Sketchup to visualize wind farms. In *Proc. 2008 IEEE Int'l Symp. on Technology and Society (ISTAS'08)*. Fredericton, NB, 1–8.

Changchang Wu. 2013. Towards linear-time incremental structure from motion. In *Proc. 2013 Int'l Conf. on 3D Vision (3DV'13)*. Seattle, WA, 127–134.

Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M. Seitz. 2011. Multicore bundle adjustment. In *Proc. 2011 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'11)*. Colorado Springs, CO, 3057–3064.

Hanwei Xu, Rami Badawi, Xiaohu Fan, Jiayong Ren, and Zhiqiang Zhang. 2009. Research for 3D visualization of digital city based on SketchUp and ArcGIS. In *Proc. SPIE Int'l Symp. Spatial Analysis, Spatial-Temporal Data Modeling, and Data Mining*, Vol. 7492. San Francisco, CA, 74920Z:1–74920Z:12.

Michael Zoellner, Jens Keil, Timm Drevensek, and Harald Wuest. 2009. Cultural heritage layers: Integrating historic media in augmented reality. In *Proc. 15th Int'l Conf. on Virtual Systems and Multimedia (VSMM'09)*. Vienna, Austria, 193–196.