Paper

Diminished Reality for AR Marker Hiding Based on Image Inpainting with Reflection of Luminance Changes

Norihiko Kawai [†], Masayoshi Yamasaki [†], Tomokazu Sato (member)[†], Naokazu Yokoya (member)[†]

Abstract This paper proposes a new diminished reality technique which removes AR markers from a user's view image. In order to achieve natural marker hiding, three factors should be considered; (1) naturalness of texture generated on a marker area. (2) geometric consistency between consecutive frames, (3) photometric consistency between a marker area and its surrounding. In this study, assuming that an area around a marker is locally planar, the marker area in the first frame image is inpainted using the rectified image to achieve high-quality inpainting. The unique inpainted texture is overlaid on the marker region in subsequent frames according to camera pose for temporal geometric consistency. Global and local luminance changes around the marker are reflected to the inpainted texture for photometric consistency.

Key words: Augmented Reality, Diminished Reality, Marker Hiding, Image Inpainting

1. Introduction

With the spread of tablet computers and portable game devices with cameras, it is becoming popular to experience applications of video see-through augmented reality (AR) which overlays virtual objects on video capturing the real world. In a number of AR applications, marker-based geometric registration is employed because of its robustness and easiness [1]. However, such marker-based AR applications have the limitation that seamless fusion between virtual objects and the real world is not achieved due to the existence of markers. To overcome this problem, diminished reality techniques, which remove real objects from video images by overlaying background texture onto the object regions, have been proposed. Diminished reality techniques can be classified into two categories: One uses actual background images and the other generates plausible background image from information around target objects.

The former methods [2)~7)] generate background texture to be overlaid on markers using actual background scenes. In these methods, there are several ways to capture actual background scenes. Cosco et al. [2] capture the background scene beforehand. Lepetit et al. [3] use past frames that captures the background in video sequence. Other methods [4)~7)] use multiple cameras. For the purpose of removing markers in AR applications, the method which captures the background scene beforehand is applicable. However, from a standpoint of a user, it is burdensome to capture background images before placing markers and starting an application.

On the other hand, in the latter methods [8] [9] [10] which generate plausible background image from information around target objects, a user can efficiently start an application without troublesome operations. The proposed method also pays attention on this approach.

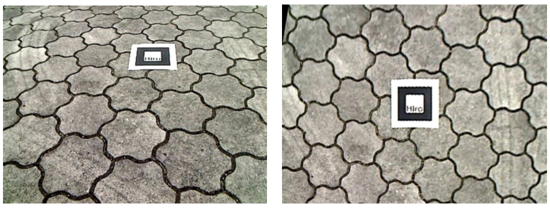
These methods visually remove real objects by applying image inpainting techniques which have been developed to remove undesired objects in a static image. In this approach, three factors should be considered to achieve natural marker hiding; (1) naturalness of texture generated on a marker area, (2) geometric consistency between consecutive frames, (3) photometric consistency between the marker area and its surroundings. We review conventional methods while paying attention to these factors and highlight our contribution in the following.

(1) Naturalness of generated texture

Siltanen [8] generates a texture on a marker area by determining each pixel value in the marker area from several pixel values of specific positions around the marker. This method can generate textures very fast. However, it is difficult to generate natural and complex

Received February 25, 2013; Revised May 7, 2013; Accepted June 11, 2013

† Nara Institute of Science and Technology
(Nara, Japan)



(a) Input Image (b) Rectified Image

Fig. 1 Correction of perspective distortion.

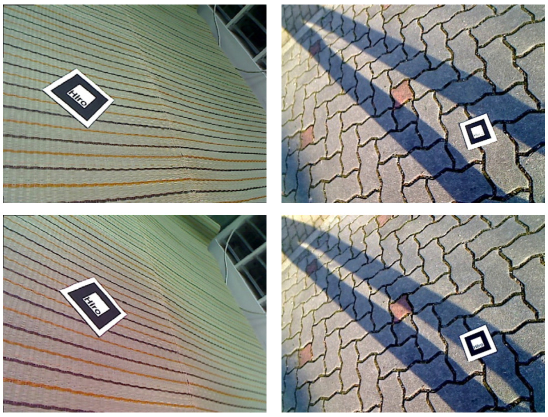
textures by such a simple approach. In order to synthesize more natural textures, Herling et al. [9] have applied exemplar-based image inpainting method [11] with high-speed approximate nearest neighbor search [12] to diminished reality. In this method, the whole input image except a marker area is searched for texture patterns similar to the pattern around the target marker area, and pixel values in the target area are determined using the similar patterns. Although exemplar-based inpainting methods yield good results in many cases, it is known that they are weak for perspective distortion such as shown in **Fig. 1**(a). To solve this problem, the proposed method inpaints a marker region using the image in which the perspective distortion is corrected as shown in Fig. 1(b) based on the assumption that an area around the marker is locally planar.

(2) Geometric consistency between consecutive frames

Siltanen [8] and Herling et al. [9] generate textures in every frame. Therefore, they tend to cause unnatural geometric changes between frames. Although Herling et al. [9] attempt to reduce texture flickering between frames by propagating patch correspondences in image inpainting, it is insufficient to achieve geometric consistency between frames taken with large camera motion. On the other hand, Korkalo et al. [10] overlay the unique texture inpainted in the first frame according to camera pose instead of inpainting every frame. Our proposed method also employs the same approach to achieve temporal geometric consistency.

(3) Photometric consistency between marker area and its surroundings

Siltanen [8] and Herling et al. [9] implicitly adjust the colors of texture according to global luminance changes, which are caused by changes of ambient illumination and photometric parameters of camera, as shown in **Fig. 2**(a), by using exemplar textures from the current frame image. On the other hand, Korkalo et al. [10] have employed a method that explicitly treats global luminance changes. This method generates a low-resolution texture for the current frame and combine the texture with high-resolution texture generated using the first frame. As far as we know, there has been no method that explicitly solves the problems of both global and local luminance changes, which are caused by cast shadow, like in Figs. 2(a) and (b). In the proposed method, both global and local luminance changes around the marker are explicitly detected and the changes are reflected to the inpainted texture to achieve photometric consistency.



(a) Global changes (b) Local changes

Fig. 2 Types of luminance changes.

In summary, our solutions for above problems are; (1) correction of perspective distortion in image inpainting for natural texture generation, (2) use of the unique inpainted texture in all the frames for temporal geometric consistency, and (3) explicitly considering both global and local luminance changes for photometric consistency. In the proposed method, extrinsic camera parameters are calculated using AR markers, and the positions of markers need to be fixed while hiding markers.

2. Overview of Proposed Method

Fig. 3 illustrates the proposed pipeline. The method consists of two processes: (A) image inpainting for generating texture for marker areas, and (B) marker hiding by reflecting luminance changes and overlaying the inpainted textures on marker areas for every frame. Because image inpainting process (A) requires several seconds in order to generate a high-quality background texture, we perform processes (A) and (B) concurrently not to keep a user waiting. If multiple markers exist in a scene, a set of processes (A) and (B) is independently and concurrently performed for each marker in order not to decrease the frame rate.

In this section, we describe processes (A) and (B) briefly. We then describe process (B) in detail in Sec-

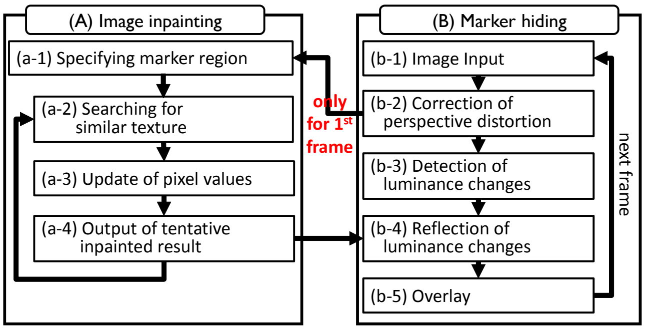

(A) Image inpainting
(a-1) Specifying marker region
(a-2) Searching for similar texture
(a-3) Update of pixel values
(a-4) Output of tentative inpainted result
only for 1st frame
(B) Marker hiding
(b-1) Image Input
(b-2) Correction of perspective distortion
(b-3) Detection of luminance changes
(b-4) Reflection of luminance changes
(b-5) Overlay
next frame

Fig. 3 Flow of the proposed method.

tion 3.

2.1 Image Inpainting

For process (A), an exemplar-based inpainting method is applied to the first frame image whose perspective distortion is corrected by process (B). The correction of perspective distortion is effective for exemplar-based inpainting methods to give better results as shown later in **Fig. 13** because that unifies the size and shape of texture patterns used as exemplars.

As an inpainting method, an arbitrary exemplar-based inpainting method using global optimization such as Wexler's method [13] and our previous method [14] is applicable in the proposed framework. Generally, in such exepmlar-based image inpainting methods, an energy function is defined based on the similarity between a missing region and the remainder of image. The missing region is then iteratively restored so as to minimize the energy function. As the concrete steps of process (A), the marker area in the rectified first frame is specified as a missing region (a-1). The region is inpainted by alternately searching for similar patterns (a-2) and updating pixel values (a-3) until the energy converges. In this process, the tentative inpainted result is passed to process (B) at each iteration (a-4).

2.2 Marker Hiding

In process (B), a scene containing a marker is first captured (b-1). The marker is detected and the input image is rectified by calculating homography matrix H as if the scene is observed with the viewing direction perpendicular to the marker (b-2). Luminance changes around the marker are then detected (b-3), and the luminance of the texture generated by process (A) is adjusted according to the changes around the marker (b-4). Finally, the adjusted texture is overlaid onto the marker area in the original input image using inverse homography matrix H^{-1} (b-5). In the following section, we describe steps (b-2) to (b-5) in detail.

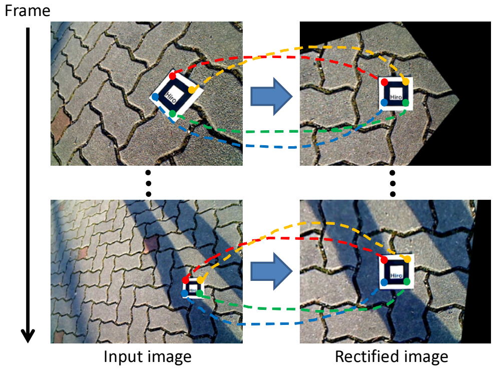


Fig. 4 Correction of perspective distortion using homography matrix.

3. Marker Hiding with Reflection of Luminance Changes

3.1 Correction of Perspective Distortion

As shown in **Fig. 4**, perspective distortion of an input image is corrected by converting the image as if the scene is captured by a camera in front of the marker in step (b-2). Concretely, four corner points of a marker are extracted from an input image, and homography matrix H is calculated from four pixel pairs between the extracted corners and four corner points of the fixed size square through all frames on a rectified image [15]. The whole input image is then converted using the homography matrix H.

By correcting the perspective distortion, the size and shape of a texture pattern are unified. That not only results in generating high-quality texture by image inpainting in process (A), but also easily determines pixel correspondences between different frames of input images for detecting luminance changes of textures in steps (b-3) and (b-4).

3.2 Detection and Reflection of Luminance Changes

The luminance of the inpainted marker region generated by process (A) is adjusted by detecting both local luminance changes (e.g. by cast shadows) and global luminance changes (e.g. by ambient illumination changes and camera parameter changes) in order to preserve photometric consistency with high-quality. In these steps, a region including a marker is divided into regular grids as shown in **Fig. 5**. Here, we do not directly use the amplitude of luminance changes in the marker region but estimate it from the marker's surrounding region Ψ (grid with a diagonal line in Fig. 5) because photometric properties of markers and the sur-

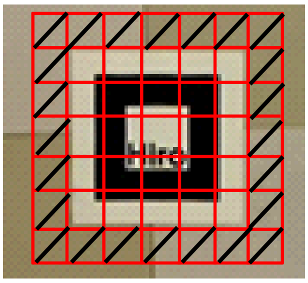


Fig. 5 Divided grids.

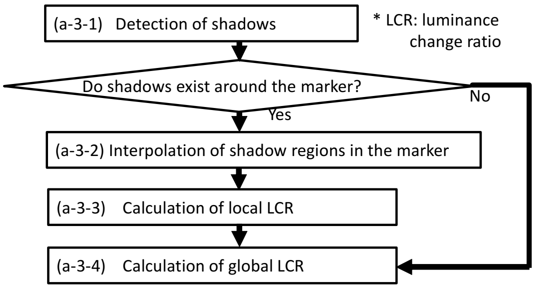

(a-3-1) Detection of shadows
* LCR: luminance change ratio
Do shadows exist around the marker?
No
Yes
(a-3-2) Interpolation of shadow regions in the marker
(a-3-3) Calculation of local LCR
(a-3-4) Calculation of global LCR

Fig. 6 Detailed steps in step (b-3) in Figure 3.

rounding materials are different.

Fig. 6 illustrates the detailed steps of detection and reflection of luminance changes. Cast shadow regions around a marker are first detected (b-3-1). The existence of cast shadows is then checked. If cast shadows exist around the marker, shadow regions in the marker region are interpolated (b-3-2) and local luminance change ratio β between the first frame and the current frame is calculated (b-3-3). Then global luminance change ratio α_i for each pixel $\mathbf{x}_i$ (i is a pixel index) in the remainder of the shadow region is calculated while excluding the influence of cast shadows (b-3-4). If no cast shadows exist around the marker, only the global luminance change ratio is calculated (b-3-4). Here, the local and global luminance change ratios of RGB are separately calculated. Finally, as shown in **Fig. 7**, the luminance of the texture generated by process (A) is adjusted by multiplying local luminance change ratio β to the interpolated shadow region and global luminance change ratio α_i to the remainder of the marker region (b-4) as follows:

$$I_n(\mathbf{x}_i) = \begin{cases} \beta J(\mathbf{x}_i) & (\textit{cast shadow regions}) \\ \alpha_i J(\mathbf{x}_i) & (\textit{otherwise}), \end{cases} \tag{1}$$

where $I_n(\mathbf{x}_i)$ is a pixel value of $\mathbf{x}_i$ in the marker region of the rectified n-th frame image and $J(\mathbf{x}_i)$ is a pixel value of $\mathbf{x}_i$ in the image generated by process (A).

In the following, we describe the methods to detect cast shadows around a marker and calculate local and global luminance change ratios in detail.

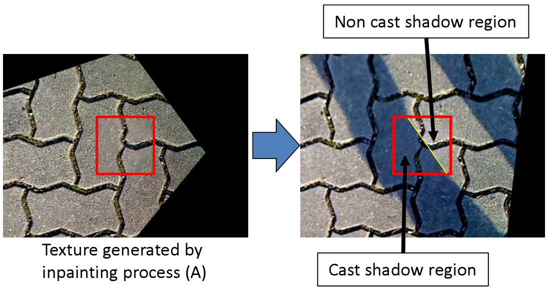


Fig. 7 Adjustment of luminance of inpainted texture.

3.2.1 Detection of Shadow Regions around Marker

In shadow detection step (b-3-1), assuming that cast shadows do not exist in the first frame and the intensities of cast shadow regions are much smaller than other regions, we use divided grids in surrounding region Ψ as shown in **Fig. 8**. Provisional global luminance change ratio δ for the regions in which cast shadows do not exist in Ψ is then calculated in an iterative manner, and cast shadow regions are detected using the calculated change ratio δ.

Concretely, images of the first frame and the n-th frame are first smoothed with Gaussian filter to reduce the influence of noise. Provisional global luminance change ratio δ_t in the t-th iteration is then calculated as follows:

$$\delta_t = \frac{\sum_{\mathbf{u}_p \in \Psi} \omega(\mathbf{u}_p)\bar{I}_n(\mathbf{u}_p)}{\sum_{\mathbf{u}_p \in \Psi} \omega(\mathbf{u}_p)\bar{I}_1(\mathbf{u}_p)}, \tag{2}$$

where $\bar{I}_n(\mathbf{u}_p)$ is the average of pixel values in grid $\mathbf{u}_p$ (p is a grid index) in the rectified n-th frame image. Here, because weight $\omega(\mathbf{u}_p)$ should be small in non-shadow regions, $\omega(\mathbf{u}_p)$ is defined using luminance change ratio δ_{t-1} in the $(t-1)$-th iteration as follows:

$$\omega(\mathbf{u}_p) = \frac{1}{\mathrm{dif}(\mathbf{u}_p) + 1}, \tag{3}$$

$$dif(\mathbf{u}_p) = max(\delta_{t-1}\bar{I}_1(\mathbf{u}_p) - \bar{I}_n(\mathbf{u}_p), 0), \tag{4}$$

where we set $\delta_0 = 1$ in the first iteration. After iterating the above processes several times, the grid whose dif value computed by Eq. (4) is larger than a threshold is determined as the cast shadow region.

After classifying each grid into shadow region and non-shadow region, the above processes using Eqs. (2) to (4) are performed pixel-wise by replacing grids in Eqs. (2) to (4) with pixels in two adjacent grids between which a shadow boundary exists. As a result, each pixel is classified into shadow region and non-shadow region.

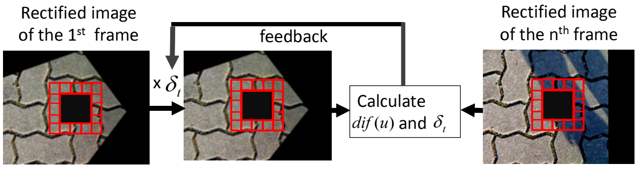

Rectified image of the 1st frame
feedback
x δ_t
Calculate $dif(u)$ and δ_t
Rectified image of the nth frame

Fig. 8 Detection of cast shadows.

3.2.2 Interpolation of Shadow Regions

Assuming that a boundary of a cast shadow in a marker is straight, the shadow region is interpolated in the marker in step (b-3-2). From an image representing shadow regions as shown in **Fig. 9**(a), the boundary pixels of the cast shadow on the boundary line between region Ψ and the marker are detected. Then pairs of shadow boundary pixels are decided and connected. Concretely, to decide pairs of the shadow boundaries, the following cost function is used.

$$C = |\theta_i - \theta_{ij}| + |\theta_j - \theta_{ij}|, \tag{5}$$

where θ_{ij} indicates the gradient direction of shadow boundary in connecting pixel $\mathbf{x}_i$ and $\mathbf{x}_j$ (j is a pixel index and $i \ddagger j$). θ_i and θ_j indicate the gradient direction of shadow at pixel $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. These directions are calculated by the following equation.

$$\theta = \begin{cases} \tan^{-1}\left(\frac{L_y}{L_x}\right) & (L_x > 0) \\ \tan^{-1}\left(\frac{L_y}{L_x}\right) + \pi & (otherwise) \end{cases}, \tag{6}$$

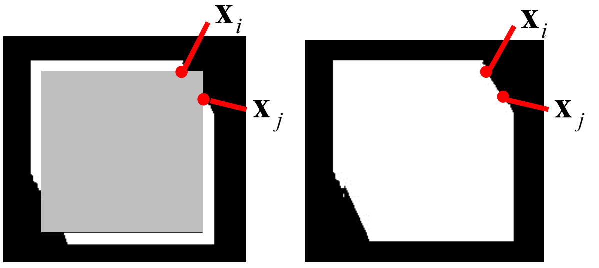
where L_x and L_y represent the magnitude of horizontal and vertical edges and calculated by convolving a 3 × 3 region including a boundary pixel in the image whose pixel values are dif with horizontal and vertical Prewitt filters, respectively, as shown in **Fig. 10**. The optimal pair which gives the smallest cost C is decided, and the next pair is decided after removing the optimal pair. This process is iterated until all pairs are decided. Next, as shown in Fig. 9(b), all the pairs are connected with straight lines and the area surrounded by shadow boundaries is set as shadow region.

3.2.3 Calculation of Local Luminance Change Ratio

Local luminance change ratio β is calculated using pixels in the shadow region in region Ψ as follows:

$$\beta = \frac{\sum_{\mathbf{x}_i \in S} I_n(\mathbf{x}_i)}{\sum_{\mathbf{x}_i \in S} I_1(\mathbf{x}_i)}, \tag{7}$$

where S is a set of pixels in the shadow region in Ψ.



(a) Shadow around marker (b) Interpolated shadow region

Fig. 9 Interpolation of shadow (White is shadow region, gray is marker area, black is other region).

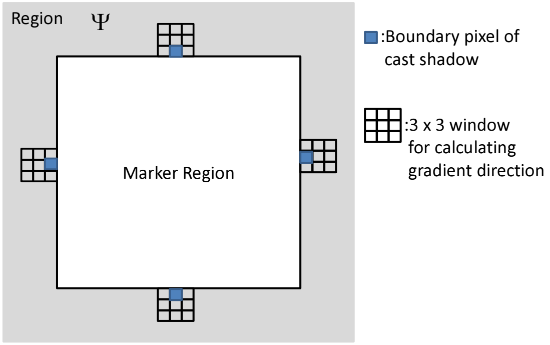

Region Ψ
Marker Region
:Boundary pixel of cast shadow
:3 x 3 window for calculating gradient direction

Fig. 10 Calculation of gradient direction of shadow boundary.

3.2.4 Calculation of Global Luminance Change Ratio

In step (b-3-4), a luminance change ratio of each grid in marker's surrounding region Ψ is calculated, and luminance change ratios in the marker are then estimated.

Calculation of global luminance change ratio around marker

Luminance change ratio $\hat{\alpha_p}$ in each grid $\mathbf{u}_p$ in region Ψ is ordinarily calculated with the ratio of sum of pixel values in a grid between the first frame and the current n-th frame as follows:

$$\hat{\alpha_p} = \frac{\sum_{\mathbf{x}_i \in \mathbf{u}_p} I_n(\mathbf{x}_i)}{\sum_{\mathbf{x}_i \in \mathbf{u}_p} I_1(\mathbf{x}_i)}, \tag{8}$$

Here, it should be noted that, in case that cast shadows exist in some grids, it is difficult to properly estimate global luminance change ratios in a marker because of the influence of cast shadows. To exclude the influence of cast shadows, as shown in **Fig. 11**, luminance change ratio $\hat{\alpha_p}$ of the grid $\mathbf{u}_p$ which includes cast shadows in region Ψ is replaced using luminance change ratios $\hat{\alpha_q}$ and $\hat{\alpha_r}$ of both ends of the cast shadow grids $\mathbf{u}_q$ and $\mathbf{u}_r$ (q and r are grid indexes and $p \ddagger q \ddagger r$) as follows:

$$\hat{\alpha_p} = \frac{(D-d)\hat{\alpha_q} + d\hat{\alpha_r}}{D}, \tag{9}$$

where D represents the Manhattan distance between grids $\mathbf{u}_q$ and $\mathbf{u}_r$. d is the Manhattan distance between grids $\mathbf{u}_p$ and $\mathbf{u}_q$.

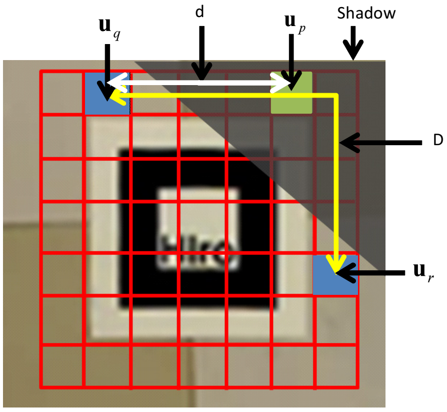


Fig. 11 Replacement of global luminance change ratios for excluding influence of cast shadows.

Estimation of global luminance change ratio in marker

Global luminance change ratios in a marker are interpolated using the calculated change ratios in region Ψ. On the assumption that adjacent grids have close luminance changes under global luminance changes such as ambient illumination changes, global luminance change ratio of each grid is calculated so as to minimize the following cost function E.

$$E = \sum_{(p,q)\in R} (\hat{\alpha_p} - \hat{\alpha_q})^2, \tag{10}$$

R is a set of all index pairs of neighboring grids. The luminance change ratio which minimizes E satisfies the following equation.

$$\frac{\partial E}{\partial \hat{\alpha_p}} = 0, \tag{11}$$

where grid $\mathbf{u}_p$ is in the marker. Each global luminance change ratio for each grid is uniquely calculated with a system of equations because the number of the above equations is equal to the number of parameters. After calculating a change ratio for each grid, the change ratio is set to the central pixel of the grid and global luminance change ratio α_i for each pixel $\mathbf{x}_i$ in the marker is calculated by bilinear interpolation.

3. 3 Overlay of color-adjusted texture

The color-adjusted texture in step (b-4) is overlaid on the marker region using inverse matrix of homography H, which is calculated in step (b-2). Concretely, as shown in **Fig. 12**, the rectified image in which the color of inpainted texture is adjusted is converted using inverse homography matrix H^{-1}. The pixel values in the target region of the converted image are then copied to the pixels in the original input image.

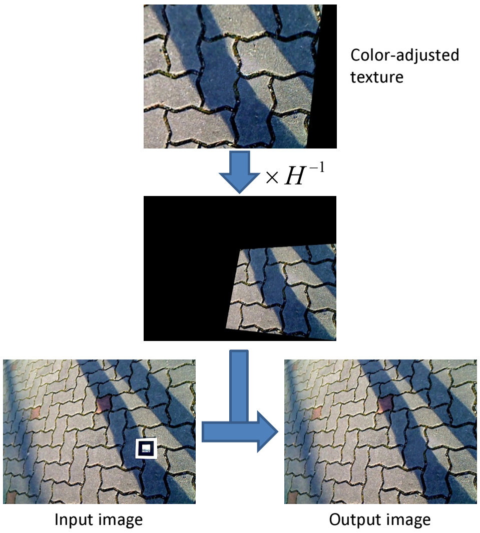


Fig. 12 Overlay of color-adjusted texture.

4. Experiments

To demonstrate the effectiveness of the proposed method, we have carried out experiments by placing markers on several environments. In the first experiment, we verify the effectiveness of correcting perspective distortion for image inpainting by comparing inpainted results using original and rectified images as inputs. Next, we verify the effectiveness of the proposed method comparing results with several approaches. Finally we show two examples of augmented reality applications using the proposed method.

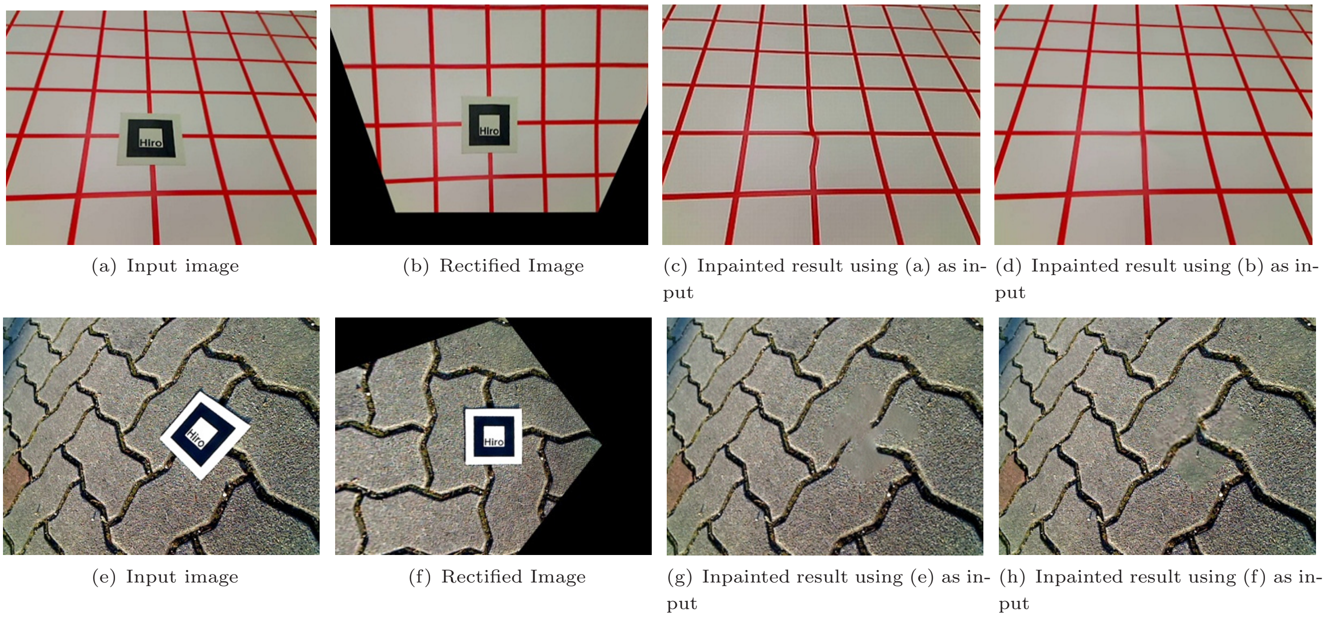
In experiments, we used a PC with Core i7 3.47 GHz CPU and 12 GB of memory and a USB camera with automatic white balance and gain. As an inpainting method, we used our inpainting method [14] with high-speed searching method [12]. The resolution of input images is 640 × 480. We set parameters as shown in **Table 1** balancing the quality of image inpainting and processing cost.

4. 1 Verification of Effectiveness of Correcting Perspective Distortion

Fig. 13 shows the comparison of inpainted results for two kinds of images (a) and (e) using original and rectified images as input for image inpainting. From the results in figures (c) and (g), it can be seen that unnatural textures are generated from original images because appropriate samples for inpainting do not exist due to

Table 1 Parameters in experiments.

Marker size in rectified image	80 × 80 pixels
Target region for inpainting	140 × 140 pixels
Grid size	14 × 14 pixels
Number of iterations in shadow detection	5



(a) Input image
(b) Rectified Image
(c) Inpainted result using (a) as input
(d) Inpainted result using (b) as input
(e) Input image
(f) Rectified Image
(g) Inpainted result using (e) as input
(h) Inpainted result using (f) as input

Fig. 13 Comparison of inpainting using original input and rectified images.

the perspective distortion. On the other hand, as shown in figures (d) and (h), natural textures are successfully generated by the proposed method using rectified images (b) and (f). From these results, we confirm that perspective distortion correction is effective to inpaint marker regions.

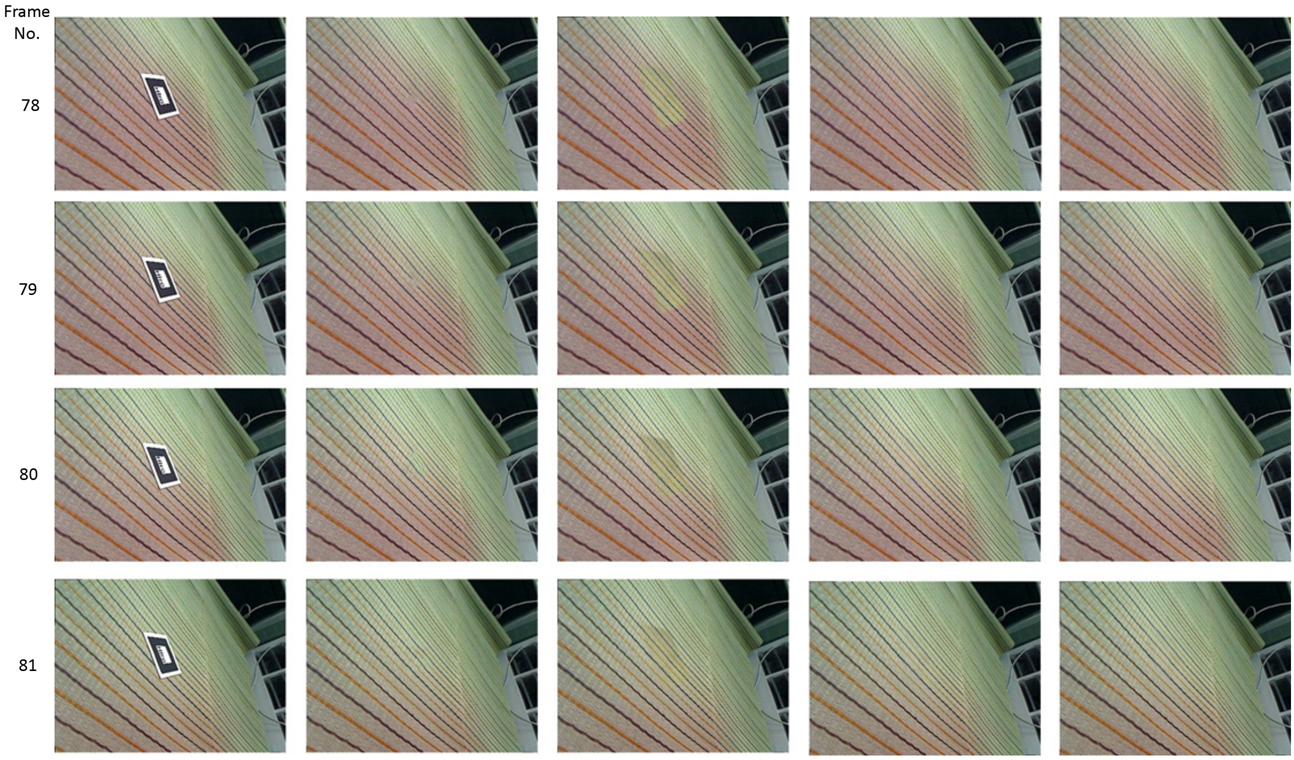
The processing time for image inpainting was eight seconds on average. Therefore, in the prototype system, we can experience AR applications with completely inpainted results in about eight seconds after starting applications.

4. 2 Comparison of Marker Hiding Results

We compared results by four approaches using two sequences. Sequence 1 consists of 360 frames and includes ambient illumination changes as shown in **Fig. 14**. Sequence 2 consists of 340 frames and includes cast shadows as shown in **Figs. 15** and **16**. Figure (a) indicates input images, (b) indicates results by inpainting for every frame, which is conventional approach 9), (c) indicates results by the proposed method without reflection of luminance changes, (d) indicates results by the proposed method with reflecting only global luminance changes, and (e) indicates results by the proposed method with reflecting both local and global luminance changes.

For sequence 1 including ambient illumination changes, in the results by the inpainting for every frame shown in Fig. 14(b), it can be seen that some frames give good results but some frames give unnatural textures on the marker region due to the perspective distortion, resulting in not preserving temporal geometric consistency. On the other hand, temporal geometric consistency is preserved by the proposed method as shown in Figs. 14(c), (d) and (e). Although photometric consistency is not achieved without reflection of luminance changes as shown in Fig. 14(c), natural textures are generated with reflecting only global luminance changes as shown in Fig. 14(d). Combining adjustment methods for local and global luminance changes also gives great results without adverse effects as shown in Fig. 14(e).

For sequence 2 including cast shadows, the geometry of results by inpainting for every frame is different from each other as shown in Figs. 15 and 16(b). Unlike this result, the temporal geometric consistency is preserved by the proposed method. On the other hand, we can also confirm that photometric consistency is not achieved without the reflection of luminance changes as shown in Figs. 15 and 16(c). In the scene including cast shadows, reflecting only global luminance changes gives blurry shadow edges as shown in Fig. 15(d). However, combining the methods for local and global changes gives clear cast shadows in the marker region as can be seen in Fig. 15(e). From the results with various viewpoints shown in Fig. 16, we can see that the proposed method which uses the unique inpainted result for all the frames can preserve temporal geometric consistency in case of a largely moving camera as well as photometric consistency. In the 4th frame of sequence 2 in Fig. 16, we can recognize that inpainting has not finished yet, but in the 20th frame, the whole texture has mostly been completed although the inpainting has not finished yet at the time. This is because the in-



(a) Input image

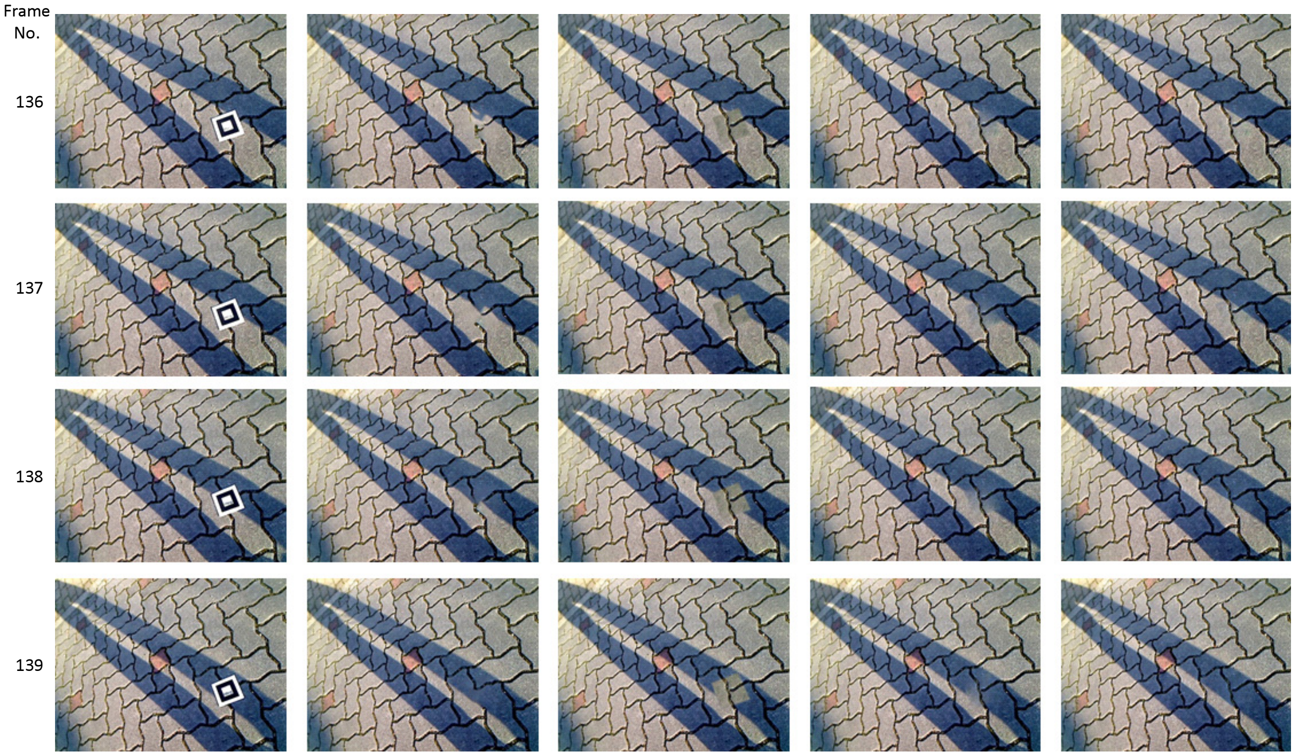
(b) Output image (Inpainting every frame)

(c) Output image (No reflection of luminance changes)

(d) Output image (Reflecting global luminance changes)

(e) Output image (Reflecting local and global luminance changes)

Fig. 14 Sequence 1 including ambient illumination changes (continuous frames)



(a) Input image

(b) Output image (Inpainting every frame)

(c) Output Image (No reflection of luminance changes)

(d) Output image (Reflecting global luminance changes)

(e) Output image (Reflecting local and global luminance changes)

Fig. 15 Sequence 2 including cast shadows (continuous frames)

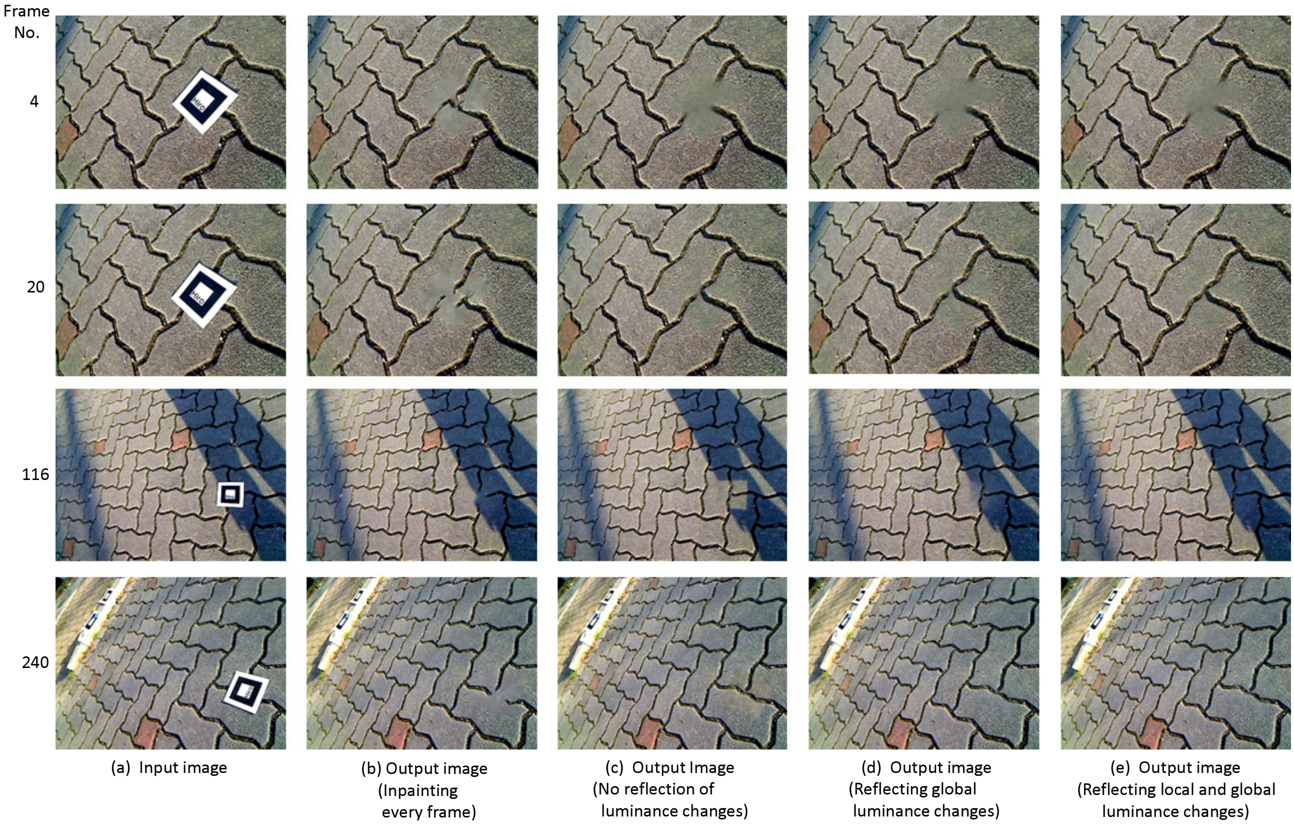

Frame No.
4
20
116
240
(a) Input image
(b) Output image (Inpainting every frame)
(c) Output Image (No reflection of luminance changes)
(d) Output image (Reflecting global luminance changes)
(e) Output image (Reflecting local and global luminance changes)

Fig. 16 Sequence 2 including cast shadows from various viewpoints

painting method [14] with searching method [12] employs a multi-scale approach, resulting in mostly generating geometry of texture for the whole target region in a few seconds.

The frame rate was 19.7 fps for sequence 1 and 18.8 fps for sequence 2, respectively. The reason why the frame rate for sequence 2 is a little smaller is that it costs an additional time to detect cast shadows and interpolate shadow regions in the marker.

4. 3 Examples of AR Applications by Hiding Markers

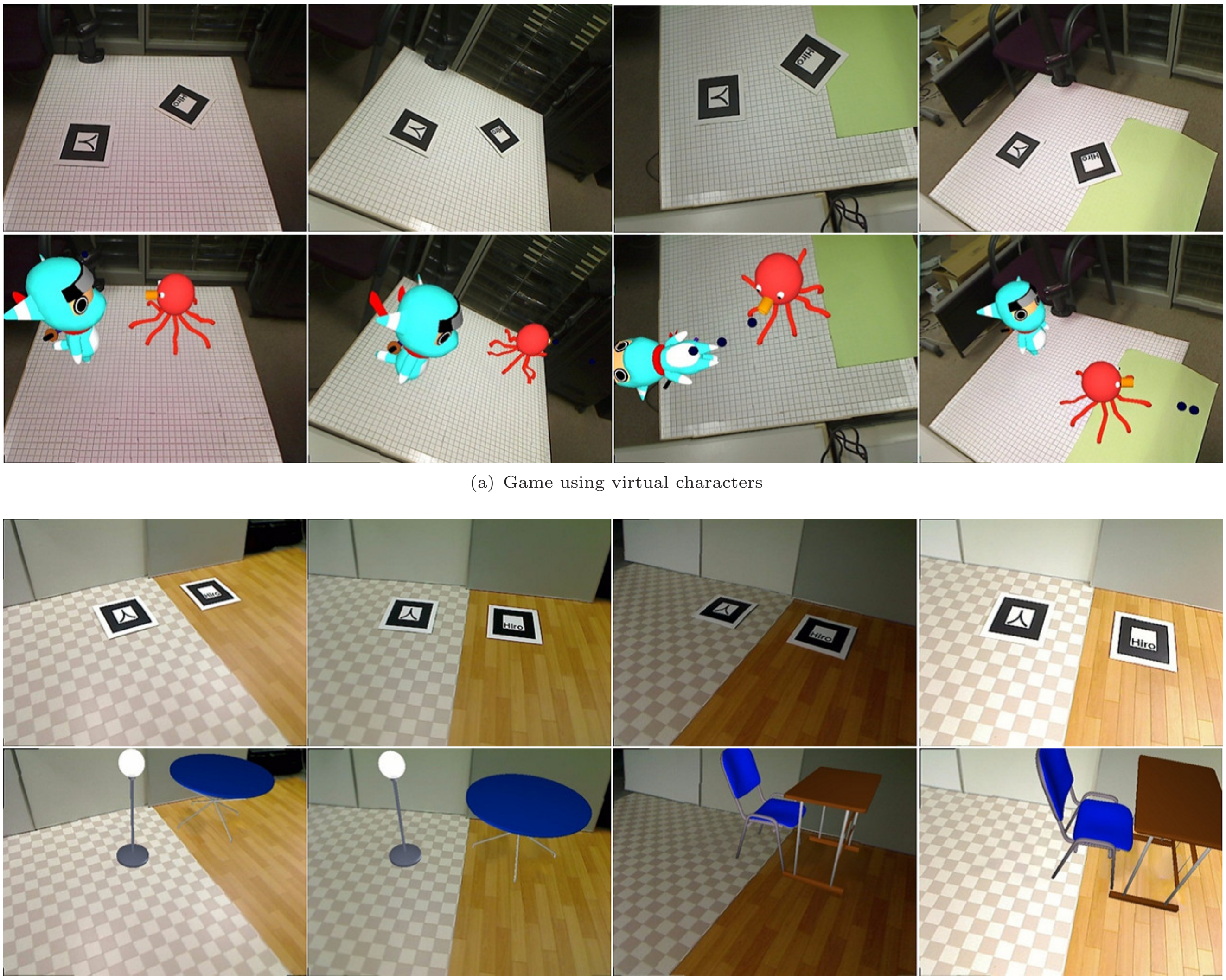
We have constructed prototype AR systems using the proposed marker hiding technique considering scenarios of hiding markers in playing games and simulating furniture arrangement with portable devices. In the AR systems, virtual objects are overlaid after markers are hidden every frame. Top and bottom images in **Figs. 17**(a) and (b) show the input images with markers and the output images in which markers are removed and virtual characters and furniture are overlaid, respectively. As can be seen from the result in Fig. 17(a), we can be absorbed in playing games without noticing the existence of markers. As can be seen from the result in Fig. 17(b), the markers do not ruin the atmosphere of the room when we confirm whether the furniture matches the room or not.

5. Conclusion

We have proposed a new diminished reality technique for removing AR markers. Correcting the perspective distortion of images enables high-quality image inpainting. Geometric and photometric consistencies of generated textures for marker regions are achieved by reflecting both local and global luminance changes separately and overlaying the texture on the marker according to camera motion. In experiments, we have demonstrated the effectiveness of the proposed method using several sequences. In future work, we will develop a marker hiding method for a non-planer background and consider more complex luminance changes such as curved cast shadows. We will also aim at developing a diminished reality method for removing various real objects from video images in conjunction with a localization and mapping method.

Acknowledgment

This work was partially supported by Grant-in-Aid for Scientific Research Nos. 23240024 and 24700118 from the Ministry of Education, Culture, Sports, Science and Technology.



(a) Game using virtual characters

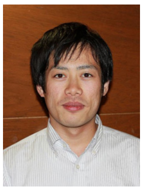
(b) Furniture arrangement simulation

Fig. 17 Examples of AR applications using the proposed marker hiding (Top: input images with markers, bottom: output AR images).

References

1) H. Kato and M. Billinghurst: "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System", Proc. Int. Workshop on Augmented Reality, pp.85–94 (1999)
2) F. I. Cosco, C. Garre, F. Bruno, M. Muzzupappa and M. A. Otaduy: "Augmented Touch Without Visual Obtrusion", Proc. Int. Symp. on Mixed and Augmented Reality, pp.99–102 (2009)
3) V. Lepetit and M.-O. Berger: "An Intuitive Tool For Outlining Objects in Video Sequences: Applications To Augmented And Diminished Reality", Proc. Int. Symp. on Mixed and Augmented Reality, pp.159–160 (2001)
4) A. Enomoto and H. Saito: "Diminished Reality Using Multiple Handheld Cameras", Proc. ACCV'07 Workshop on Multidimensional and Multi-view Image Processing, pp.130–135 (2007)
5) S. Jarusirisawad, T. Hosokawa and H. Saito: "Diminished Reality Using Plane-sweep Algorithm with Weakly-calibrated Cameras", Progress in Informatics, 7, pp.11–20 (2010)
6) T. Tsuda, H. Yamamoto, Y. Kameda and Y. Ohta: "Visualization Methods for Outdoor See-Through Vision", Proc. Int. Conf. on Artificial Reality and Telexistence, pp.62–69 (2005)
7) S. Zokai, J. Esteve, Y. Genc and N. Navab: "Multiview Paraperspective Projection Model for Diminished Reality", Proc. Int. Symp. on Mixed and Augmented Reality, pp.217–226 (2003)
8) S. Siltanen: "Texture Generation over the Marker Area", Proc. Int. Symp. on Mixed and Augmented Reality, pp.253–254 (2006)
9) J. Herling and W. Broll: "Advanced Self-contained Object Removal for Realizing Real-time Diminished Reality in Unconstrained Environments", Proc. Int. Symp. on Mixed and Augmented Reality, pp.207–212 (2010)
10) O. Korkalo, M. Aittala and S. Siltanen: "Light-weight Marker Hiding for Augmented Reality", Proc. Int. Symp. on Mixed and Augmented Reality, pp.247–248 (2010)
11) D. Simakov, Y. Caspi, E. Shechtman and M. Irani: "Summarizing Visual Data Using Bidirectional Similarity", Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.1–8 (2008)
12) C. Barnes, E. Shechtman, A. Finkelstein and D. B. Goldman: "PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing", ACM Trans. on Graphics, 28, 3, pp.1–11 (2009)
13) Y. Wexler, E. Shechtman and M. Irani: "Space-Time Completion of Video", IEEE Trans. on Pattern Analysis and Machine Intelligence, 29, 3 (2007)
14) N. Kawai, T. Sato and N. Yokoya: "Image Inpainting Considering Brightness Change and Spatial Locality of Textures and Its Evaluation", Proc. Pacific-Rim Symp. on Image and Video Technology, pp.271–282 (2009)
15) R. Hartley and A. Zisserman: Multiple View Geometry in Computer Vision, Cambridge University Press, ISBN: 0521540518 (2004)

Norihiko Kawai received his B.E. degree in informatics and mathematical science from Kyoto University in 2005. He received his M.E. and Ph.D. degrees in information science from the Nara Institute of Science and Technology in 2007 and 2010, respectively. He was a research fellow of the Japan Society for the Promotion of Science and a postdoctoral fellow at the University of California at Berkeley in 2010-2011. He has been an assistant professor at the Nara Institute of Science and Technology since 2011. He is a member of IEEE, IEICE, IPSJ and VRSJ



Masayoshi Yamasaki received his B.E. degree from Sasebo National College of Technology in 2010. He received his M.E. degree in information science from the Nara Institute of Science and Technology in 2012. He has been working at Canon Inc. since 2012.

Tomokazu Sato received his B.E. degree in computer and system science from the Osaka Prefecture University in 1999. He received his M.E. and Ph.D. degrees in information science from the Nara Institute of Science and Technology in 2001 and 2003, respectively. He was an assistant professor at the Nara Institute of Science and Technology in 2003-2011. He was a visiting researcher at Czech Technical University in Prague in 2010-2011. He has been an associate professor at the Nara Institute of Science and Technology since 2011. He is a member of IEEE, IEICE, IPSJ, VRSJ and ITE

Naokazu Yokoya received his B.E., M.E., and Ph.D. degrees in information and computer sciences from Osaka University in 1974, 1976, and 1979, respectively. He joined Electrotechnical Laboratory (ETL) of the Ministry of International Trade and Industry in 1979. He was a visiting professor at McGill University in Montreal in 1986-87 and has been a professor at the Nara Institute of Science and Technology since 1992. He is a fellow of IPSJ, IEICE and VRSJ and a member of IEEE, ACM SIGGRAPH, JSAI, JCSS and ITE.