Approximated User-Perspective Rendering in Tablet-Based Augmented Reality

Makoto Tomioka* Sei Ikeda† Kosuke Sato‡

Graduate School of Engineering Science, Osaka University

Abstract

This study addresses the problem of geometric consistency between displayed images and real scenes in augmented reality using a video see-through hand-held display or tablet. To solve this problem, we present approximated user-perspective images rendered by homography transformation of camera images. Homography approximation has major advantages not only in terms of computational costs, but also in the quality of image rendering. However, it can lead to an inconsistency between the real image and virtual objects. This study also introduces a variety of rendering methods for virtual objects and discusses the differences between them. We implemented two prototypes and designed three types of user studies on matching tasks between real scenes and displayed images. We have confirmed that the proposed method works in real time on an off-the-shelf tablet. Our pilot tests show the potential to improve users' visibility, even in real environments, by using our method.

Keywords: User-perspective rendering, augmented reality, geometric consistency, video see-through, tablet-based AR.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities; H.5.2 [Information Interfaces and Presentation]: User Interfaces—Evaluation/methodology I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms

1 Introduction

Hand-held computers with a video see-through display, hereafter referred to as tablets, are one of the few commercialized augmented-reality (AR) interfaces which can enable us to interact with virtual objects through a touch panel screen. Even though tablet devices with an optical see-through display are being used experimentally, research on image presentation techniques for AR based on a video see-through display will continue to be important as long as there remain good applications where processed real images are displayed, such as diminished reality [1] and fog see-through techniques [2].

In AR, the most fundamental factor is usually to create an illusion to users that virtual objects drawn by computer graphics (CG) actually exist in the real world. This illusion potentially improves the users' visibility in the sense that users can easily recognize what the CG information is related to in the real scene. In order to create the illusion, AR systems, including tablet-based AR systems, should satisfy geometric, photometric and temporal consistencies.

However, most studies on tablet-based AR have dealt with the problems of consistency between the real image and CG (RI-CG), rather than that between the real scene and displayed image (RS-DI). In most AR systems using tablets, CG images are rendered on the captured real images by exact perspective projection at the optical center of the rear camera, as shown in Figure 1 (a). This is called device-perspective rendering (DPR). Even if the RI-CG consistency can be completely satisfied, unless the consistencies between the real scene and real image (RS-RI) are not satisfied, the users will be required to find RS-DI correspondences to know what the virtual objects are related to in the real scene. This may decrease some sort of visibility or task performance.

(a) device-perspective (b) user-perspective (c) ground truth

Figure 1: Comparison of device-perspective rendering, user-perspective rendering, and a grand truth.

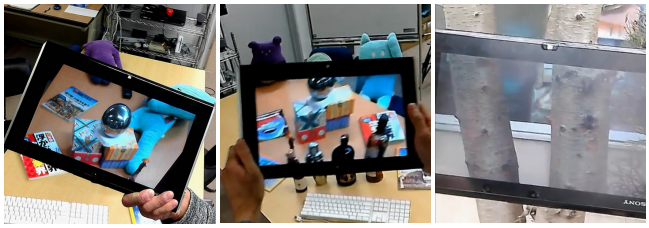


Figure 2: Operation test on an off-the-shelf tablet.

The RS-RI geometric inconsistency is caused by the camera-eye discrepancy, not only in terms of the field of view (FOV) or lens distortion [3, 4] but also of viewpoint. In order to create images without this type of inconsistency, AR systems should be able to map the real scene in 3D and generate images according to the user's viewpoint, which is called user-perspective rendering (UPR) i.e., it should create an impression as if the screen of the tablet is transparent, as shown in Figures 1 (b) and 2. Therefore, reducing the geometric discontinuity or rendering user-perspective images in a tablet frame is a problem equivalent to a novel view synthesis or image-based rendering, depending on both the shape of the environment and the user's viewpoint. Since the novel view synthesis from multiple images captured from sparse viewpoints is obviously an ill-posed problem, it is intrinsically difficult to render the exact user-perspective view of any scene. To this effect, it is important to validate the UPR for tablets in real environments where image generation errors might negatively affect the users' visibility.

The first work that achieved UPR for AR/MR hand-held display was the ARScope developed by Yoshida et al [5]. In this system, the user holds a special display device that contains a rear camera on the opposite side to the user. The user wears a head-mounted projector and a camera, whose optical centers are close to each other

*e-mail: tomioka@sens.sys.es.osaka-u.ac.jp
†e-mail: ikeda@sys.es.osaka-u.ac.jp
‡e-mail: sato@sys.es.osaka-u.ac.jp

IEEE International Symposium on Mixed and Augmented Reality 2013
Science and Technology Proceedings
1 - 4 October 2013, Adelaide, SA, Australia
978-1-4799-2869-9/13/$31.00 ©2013 IEEE

and to the user's viewpoint. The system acquires real images from both the rear and head-mounted cameras, and warps the rear camera image to the head-mounted one by homography transformation estimated via SIFT feature matching. The advantages of this system are that it is not necessary to track the user's viewpoint, and the retro-reflective material screen surface can be of any shape. On the other hand, unavoidable disadvantages are that the user needs to wear the projector and camera, and SIFT feature matching can be performed only in the common region of the two camera images. There is no mention of the effectiveness in any of the tasks or of what happens if the environment is non-planar.

Hill et al. [3] implemented the first video see-through AR system with UPR. Similar points between their study and ours are the fact that the device has both rear and front cameras, and that the UPR is based on homography. The system generates user-perspective images according to the user's viewpoint under the assumption that the scene lies on a certain single depth or on a flat surface parallel to the screen. In this system, the RS-RI inconsistency cannot be ignored if the real scene depth is qwith respect touite different from the specified one or has changed.

Baričević et al. [6] conducted a user study in a simulator using virtual reality techniques to validate UPR on a tablet. Their user study indicates that there are some situations in which the user-perspective view is preferred by users or effective in manipulating real objects depending on virtual ones. These results are supported by Steinicke et al.'s work [7], which shows the virtual camera's view frustum should be the same as a human's. Baričević et al.'s user study was designed to explore the limits of general UPR, because all the operations were performed in a VR environment. However, there is still room for a considerable measure of disagreement about the validity of the UPR. First, their user study was not performed in a real environment but in a simulator. The simulator does not cover unavoidable negative factors such as inconsistency of focus or rendering artifacts because it is difficult to completely reproduce the complexity of real environments. Second, there were some problems in their experimental design. In their experiment comparing the visibility between DPR and UPR, each participant was required to engage in a task in which the participant held a 3D pointer with his/her hand, and moved the pointer into a target virtual object displayed only in the tablet screen. In this task, the merits of UPR do not fully appear. First, when the pointer is out of the tablet frame, it is easy to move the pointer into the frame without watching the real images displayed on the screen. This operation is possible without any trouble, even if there is nothing on the screen. Second, once the pointer enters the tablet frame, the participant can adjust the position of the pointer in the real image to the virtual object while observing only the screen. This operation is also possible without looking at the real scene outside of the screen at all. In conclusion, there is almost no need to find the RS-DI correspondence, and the correspondence is required only at the moment when the pointer is entering the frame from the outside. While they were able to observe the UPR slightly faster than the DPR, it remains uncertain if the UPR is capable of improving the visibility in real environments.

It is difficult to design a user study to confirm the validity of the UPR in real environments. This is because the real scene must be simple enough for participants to be able to find the correspondences complex enough that they cannot remember the scene while repeating the tasks. In practical use, such as navigation applications, this problem does not occur because most users use this AR interface in an environment that they do not remember in detail. In experiments, it is necessary to use the same environment repeatedly in order to maintain the same conditions. In addition, task design should be also considered to effectively measure the visibility. There are two possible methods to confirm the visibility in finding the RS-DI correspondence. One method is that participants point to the environmental point corresponding to an augmented marker on the real image. The other is that participants point at the screen location corresponding to a projected marker in the real scene. However, in both methods, participants can complete the tasks by observing only at the screen if the markers appear on it.

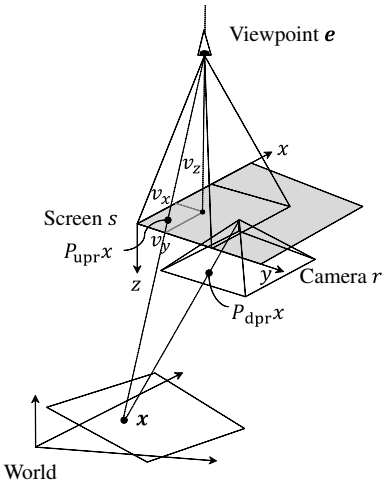


Figure 3: User-perspective projection.

In this study, we propose a novel method for presenting user-perspective images in a tablet AR device. For avoiding artifacts in novel view synthesis, the user-perspective view is generated by homography transformation that approximates the strict perspective projection, rather than by the perspective projection of each 3D measurement point. Although the homography approximation has some benefits, it causes RI-CG inconsistency if the virtual objects are rendered by a simple perspective projection at the user's viewpoint. We show that there are at least three rendering methods for virtual objects. Moreover, we show three types of experimental designs to confirm the validity of UPR, even in real environments. The tasks of these experiments are strongly associated with the visibility in finding RS-RI correspondences. Through the pilot tests of these experiments, we confirm the tendency and clarify advantages and disadvantages of these experimental methods.

2 User-perspective rendering by homography

2.1 Approximated user-perspective projection

In the proposed method, each user-perspective image is generated by the homography transformation of captured images. This section reviews where each 3D point measured by the rear camera or other sensors should be displayed if perspective views at the user's viewpoint are computed by the exact perspective projection.

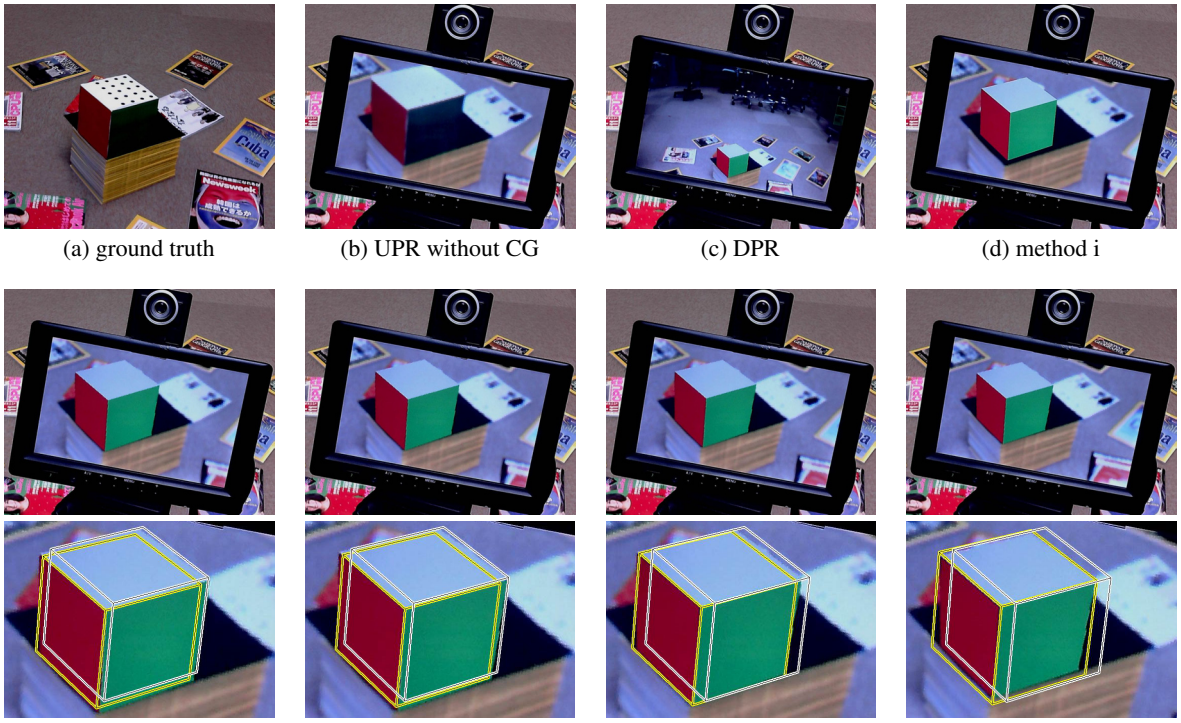
Suppose there is a tablet with a flat screen and two cameras f, r at the front and rear, respectively. We assume that the intrinsic $\boldsymbol{K}_c$ and extrinsic $\boldsymbol{M}_c$ parameters of each camera $c \in \{r, f\}$ are known, and the relative pose $\boldsymbol{M}_{c\to s}$ of each camera c to the screen s is also known. For simplicity, lens distortion of all images is assumed to be corrected in advance.

The perspective projection at the user's viewpoint can be represented by the following equation:

$$\boldsymbol{P}_{\mathrm{upr}} = \boldsymbol{P}_e \boldsymbol{M}_{r\to s} \boldsymbol{M}_r, \tag{1}$$

where $\boldsymbol{P}_e$ is the perspective projection matrix of a view frustum formed by the viewpoint $\boldsymbol{e} = [e_x, e_y, e_z]^{\mathrm{T}}$ and the screen surface, as shown in Figure 3. This matrix can be represented as follows:

$$\boldsymbol{P}_e = \begin{bmatrix} -e_z & 0 & e_x & 0 \\ 0 & -e_z & e_y & 0 \\ 0 & 0 & 1 & -e_z \end{bmatrix}. \tag{2}$$



(a) ground truth (b) UPR without CG (c) DPR (d) method i

(e) method ii (f) method iii ($w_y = 2^3$) (g) method iii ($w_y = 2^6$) (h) method iii ($w_y = 2^9$)

Figure 4: Comparison of different augmentation methods.

An exact user-perspective image can be theoretically generated by this perspective projection $\boldsymbol{P}_{\rm upr}$ of the 3D points if the positions of the 3D points corresponding to all the screen pixels can be obtained by passive or active sensing. However, as mentioned in Section 1, in the case that 3D points are sparse or partly missing because of occlusions or violation of the Lambertian assumption, the simple UPR of the missing parts is impossible. Even if a dense 3D surface can be obtained, the effects of smoothness regularization [8] and measurement errors may locally distort the displayed real images.

To avoid the above problems, the proposed method finds the optimal homography transformation closest to the exact UPR. In particular, for a given set F of 3D points $\boldsymbol{x}$, we minimize the following error function E to obtain the optimal $\boldsymbol{H}$:

$$E(\boldsymbol{H}) = \sum_{\boldsymbol{x}\in F} w_{\boldsymbol{x}} d(\boldsymbol{P}_{\rm upr}\boldsymbol{x}, \boldsymbol{H}\boldsymbol{P}_{\rm dpr}\boldsymbol{x}), \tag{3}$$

where $w_{\boldsymbol{x}}$ is the weight of the point $\boldsymbol{x}$ and $d(,)$ is the squared Euclidean distance between two points. Handling this weight enables generation of images suitable for various applications. $\boldsymbol{P}_{\rm dpr}$ is a matrix of the device-perspective projection, defined as $\boldsymbol{P}_{\rm dpr} = \boldsymbol{K}_r\boldsymbol{M}_r$. By transforming captured images with $\boldsymbol{H}$, the colors of all the screen pixels are assigned.

2.2 Different augmentation methods

While homography approximation has several benefits, using this approximation for real image rendering causes another problem, namely RI-CG inconsistency, if the virtual objects are rendered by an exact perspective projection $\boldsymbol{P}_{\rm upr}$ at the user's viewpoint. Here, we introduce three possible methods to augment virtual objects on the approximated user-perspective images.

i. UPR: This method directly renders virtual objects by the exact UPR $\boldsymbol{P}_{\rm upr}$ on the real image transformed by $\boldsymbol{H}$. In this method, the RI-CG geometric inconsistency or misregistration can occur.

ii. DPR + homography: This method renders virtual objects by the DPR on the rear camera image before the rendered image is transformed by homography $\boldsymbol{H}$. The above misregistration does not occur. Instead, virtual objects can be distorted.

iii. UPR + homography: This method first projects vertices of the virtual objects V by the exact UPR $\boldsymbol{P}_{\rm upr}$, then uses both measured points $\boldsymbol{x} \in F$ and vertices $\boldsymbol{y} \in V$ to estimate the matrix $\boldsymbol{H}'$. Next, the virtual objects are rendered on the captured image by the DPR. Finally, the rendered images are transformed by the estimated homography $\boldsymbol{H}'$.

$$E'(\boldsymbol{H}') = E(\boldsymbol{H}') + \sum_{\boldsymbol{y}\in V} w_{\boldsymbol{y}} d(\boldsymbol{P}_{\rm upr}\boldsymbol{y}, \boldsymbol{H}'\boldsymbol{P}_{\rm dpr}\boldsymbol{y}) \tag{4}$$

In this method, the RI-CG misregistration does not occur. Virtual objects can be distorted and the RS-RI misregistration occurs. However, the gaps from the UPR are totally minimized. The behavior of this method can be customized by adjusting the weights $w_{\boldsymbol{x}}$ and $w_{\boldsymbol{y}}$.

Figure 4 shows the apparent differences among the augmentation methods. Figure (a) shows the real scene without using the tablet as a reference. (b) is the same scene as (a) but captured through the tablet. (c) shows the DPR with a virtual cube. (d) and (e) are the results of method i and ii, respectively. (f), (g) and (h) show the results of method iii using different values of the vertex weights $w_{\boldsymbol{y}} = 2^3$, 2^6 and 2^9, respectively. The lower row of (e) - (h) shows the magnified screens of the upper row. The white wireframe indicates the position of the real cube. The yellow lines are the same, but they are translated so that its top-front corner coincides with that of the three-colored faces to show the difference of the virtual object's size. All the weights $w_{\boldsymbol{x}}$ of the feature points are set as 1. The real and virtual cubes were placed along the two side edges of the stacked magazines.

These figures clearly represent the characteristics of each method. Since the set of feature points F did not include points on the real cube and the stacked magazines, the estimated homography

$\boldsymbol{H}$ was optimized for the UPR of the flat floor. Therefore, the real image shown in Figure (b) continues in the real scene without large misregistration, and the cube is slightly distorted compared with the ground truth (a). Regarding Figure (d), despite the fact that the real image part and the size and position of the virtual object are the same as (b), small RI-CG misregistration appears around the bottom edges of the cube. The displayed images in Figures (e) - (h) were made of the image in (c) by using different homography transformations. There is no large RS-RI misregistration. Figure (e) of method ii is equivalent to method iii using $w_y = 0$. As shown in Figures (e) - (h), increasing the weights w_y reduces the distortion or size difference but increases the misregistration. However, the remarkable point is that the misregistration and the distortion of the virtual objects appear significantly small in methods i - iii at first glance. This effect is mainly caused by the goodness of homography approximation.

3 Prototype Systems

This section shows two prototypes. One was used in the preliminary experiments described in Section 4 and was designed to achieve high performance and to be light weight. The other one demonstrates that the proposed method works in real time and in a standalone manner on an off-the-shelf tablet without remodeling any hardwares.

3.1 Hardware and software configuration

Hardware configuration of Prototype 1 The prototype system consists of a touch panel screen (Hanwha HM-TL7T, 800 × 480 pixels, 152.4 × 91.44 mm), a rear camera (Point Gray Research Dragonfly, 640 × 480 pixels, 30 fps), a front camera (ELECOM UCAM-DLW500TA, 640 × 480 pixels, 30 fps) and a desktop PC (CPU: Intel Core2 2.33 GHz, RAM: 4 GB, GPU: NVIDIA GeForce 8800GTX). The horizontal FOVs of both rear and front cameras are approximately 60° and 70° , respectively.

In order to reduce the overall weight for experiments, all the computations are performed not on the hand-held device but on the desktop PC. The images captured by the rear camera are transmitted through an IEEE1394 cable to the PC, and then, augmented images are transferred through an HDMI cable to the display. The weight of the prototype, except the cables, is 464 g. If the cables are connected and one end of the bunch of cables is fixed at the same height as the tablet, the weight is 525 g. This weight saving enables us to perform user studies using a tablet whose weight is close to typical off-the-shelf tablet devices such as Apple iPad with Retina Display (652 g) and Google Nexus 10 (603 g).

Hardware configuration of Prototype 2 In order to demonstrate that our method does not require high computational costs, we implemented one more prototype using an off-the-shelf tablet, Sony Vaio Duo 11 (CPU: Intel Core i7-3687U, RAM: 8GB, GPU: Intel HD Graphics 4000, size: 319.9 × 199 × 17.85 mm, weight: 1665 g). The tablet has two full HD cameras that work simultaneously only at the smaller resolutions (front: 320 × 240 pixels, 30 fps, rear: 640 × 480 pixels, 30 fps). The total weight of the prototype is 1.665 kg and its size is 319.9 × 199 × 17.85 mm. As it is heavier and larger than Prototype 1, we did not use Prototype 2 for the experiments.

Software configuration For tracking and estimating 3D points, we used PTAMM by Castle et al. [9], which can save and reuse obtained 3D point data so that the experiments can be performed for multiple participants under the same conditions. Since PTAMM is based on structure-from-motion, the scale of the entire reconstruction cannot be decided in advance. In our prototype system, we provide a constant length as the baseline between the first two frames of the initialization. For calculating user's viewpoint $\boldsymbol{e}$, we used a face tracking method using a non-rigid model developed by Saragih et al. [10]. For the user's viewpoint, the 3D position of the right eye (the dominant eye) was computed by providing the face size. The intrinsic parameters $\boldsymbol{K}_r$ and $\boldsymbol{K}_f$ were obtained by a classic calibration method. As extrinsic parameters $\boldsymbol{M}_{r\to s}$ and $\boldsymbol{M}_{f\to s}$, we used design values of each prototype.

Figure 5: Sampled frames of operation test video.

3.2 Performance of prototype systems

The performance of Prototype 1 used in the experiments can directly affect the results. This section mainly describes the performance of Prototype 1, although Prototype 2 is also briefly mentioned. In this section, first we demonstrate how the approximated user-perspective images appear when the target scene is a cluttered environment. Second, the registration error and the jitter size are evaluated by using a marker, where the weights of all the points are set as a constant value $w_{\boldsymbol{x}} = 1$ because we do not consider any specific applications here.

Operation test As shown in Figure 5, we performed an operation test using Prototype 1 in a cluttered environment (our laboratory) with non-planar, specular objects, many occlusions and a walking person as a dynamic scene. The depth of the scene was in the range of 1 - 3.5 m, which is not a distant scene. The images in Figure 5 are sampled frames from a video captured with a mannequin-shaped camera (Buffalo Technology BSW20K10HBK, 1600 × 1200 pixels, horizontal FOV: 40°). Figure 7 shows this mannequin. Each frame is rotated by 90° because there was not enough space to place the camera inside the mannequin horizontally, and we placed it rotated by 90°. While capturing the video, we moved the tablet and the mannequin simultaneously. The motion of the tablet included rotation and translation.

As shown in Figure 5, we can observe that the displayed image and the real scene appear geometrically continuous, while there are indistinct frames because of reflections, and marginally discontinuous parts. The important point is that this continuity does not seem to fail, even when the tablet focussed at non-planar parts or a moving object.

Next, we confirmed the operating rate (fps). In both prototypes, the camera tracking, the viewpoint tracking and the rendering operate in parallel. The operating rates of the camera and the viewpoint tracking were more than 30 Hz. The rate of the UPR was more than 60 Hz. However, Prototype 2 does not have the stability to perform the experiments because the FOV of the front camera is too narrow.

Registration error We first evaluated the positional error of the RS-RI geometric registration. The target of this evaluation was

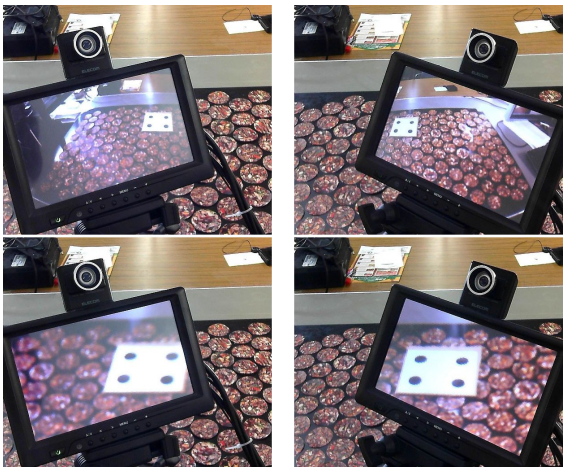


Figure 6: Registration errors in the device-perspective (top) and user-perspective (bottom) renderings.

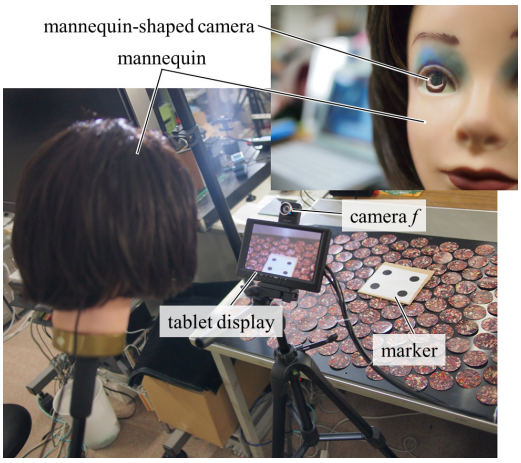


Figure 7: Experimental setup for registration error evaluation.

a marker shown in Figures 6 and 7, in a planar scene where theoretically correct images could be generated. The centers of gravity of the marker were detected in the real images displayed on the tablet frame in five poses. The ground truth was created by directly detecting the marker without looking through the tablet. We compared the mean μ and the standard deviation σ of the centers of gravity between the DPR and the UPR.

The results are shown in Figure 6. The registration errors ($\mu \pm \sigma$) in pixel units are 45.29 ± 0.38 in the UPR and 111.81 ± 0.031 in the DPR on the mannequin camera images. As shown in Figure 6, although the lack of resolution in the UPR is not negligible, the markers in the different poses are displayed at the same position and with the same size, and the border line between the black and gray parts in the background is almost seamless between the real scene and the real image. In fact, the mean μ of the registration errors is almost independent to that of the tablet poses, although the error values are much worse than sub-pixel accuracy. These tendencies are not observed in the DPR. However, the standard deviations σ in the UPR, corresponding to the size of jitter, are larger than that in the DPR.

3.3 Discussion

Operation test Initially, we predicted that an apparently large discontinuity would occur in the case of non-planar or dynamic scenes. In contrast, the presented images appear continuous in most cases. This can be explained by the following two reasons. First, in the small tablet frame, the scene can be partially approximated to be a plane. Second, if the viewpoint is around the optical axis of the rear camera, the DPR might be almost the same as the exact UPR around the center of the captured image. In most cases, users' viewpoints would be around the optical axis.

Registration error We confirmed that the registration errors in the UPR were smaller than those in the DPR, as predicted. However, even in the UPR, the average of the registration errors was approximately 38 pixels in the screen pixel unit or 11 pixels in the rear camera pixel unit, which significantly exceeds the relative pixel size of the screen or the rear camera images, despite experimenting in a flat environment, in which correct UPR is theoretically possible by using homography. The main causes for this misregistration could be the estimation errors of the feature points and camera pose, the calibration and installation errors of the cameras, and the detection errors of the viewpoint.

Since feature points with large re-projection errors are removed by robust estimation in PTAMM, it is unlikely that the estimation errors of feature points or camera poses are the main cause. Here the matter of importance is that the scale of this estimation was decided manually in our prototypes despite the fact that the UPR requires an exact scale. In our empirical study, the results were sensitive to scale error. The simplest method to determine the scale is to integrate an accelerometer [11] built in typical off-the-shelf tablet devices.

The calibration and installation errors of the two cameras may have caused the registration errors. The intrinsic parameters are estimated by a classic calibration method. The extrinsic ones of the front camera $\boldsymbol{M}_{f\to s}$ can be computed by the screen-camera calibration using a mirror [12] or user's eyes [13]. The extrinsic parameters of the rear camera $\boldsymbol{M}_{r\to s}$ can be indirectly estimated by multi-camera calibration [14] of the two cameras.

Detection errors in the viewpoint may also cause the registration errors. These prototypes use a single face model not only for this evaluation but for the experiments. We found a small misalignment between the model and the mannequin face. Pupil detection may improve the accuracy of the viewpoint detection.

Jitter The standard deviations σ in the UPR were larger than those in the DPR, in other words the homography estimation was more unstable. This was caused by the instability of feature matching in PTAMM at each frame, because even if a viewpoint value was fixed, this instability did not change. To improve the stability of the homography estimation, the system can choose only stable feature points.

Latency The latency of image rendering may have affected the visibility. However, it is difficult to remove the latency in capturing and displaying processes. To avoid the negative effects caused by the latency, the captured real images can be displayed at a predicted position by using an accelerometer [11] or gyroscope [15].

Limitations in the prototypes One of the advantages of our prototypes over the existing implementation methods using an RGBD camera [6] is that it is possible to create user-perspective images even if the scene contains distant parts that coded lights cannot reach. One disadvantage of our implementation is that it is not possible to obtain 3D information about dynamic parts, because the visual SLAM algorithms such as PTAM/PTAMM fundamentally assume a static environment. The depth image estimation technique [16] is one possible method to overcome this limitation.

4 Preliminary Experiments

This section proposes three types of experimental design and reports the results of their pilot tests to clarify the existence of limitations. Two of the three experiments were designed to evaluate the visibility in finding RS-DI correspondences. The other experiment focused on the visibility in recognizing the pose of a virtual

object. In these experiments, participants were recruited from our laboratory to obtain their technical opinions.

4.1 Visibility test in a planar environment

4.1.1 Method

The target of the first experiment is a planar scene where the exact UPR can be realized by homography. The task is to select the target, which is randomly and automatically pointed at with an augmented circle marker. The participants are instructed as follows:

- Select one target as quickly and accurately as possible.
- Move the pointing stick and the tablet so that the pointing stick does not get captured in the screen image.
- Move the tablet so that the frame of the target area does not appear in the screen image.

The movements of the pointing stick and the tablet are restricted in the third instruction to prevent participants from identifying the correspondences only in the screen image.

As targets, 175 pieces (30 different types) of circular paper, shown in Figure 7, are used. Each target is made by clipping a 6 cm circle from a large picture of fallen leaves. This size was decided by considering the balance between ease in pointing and difficulty in finding correspondences. In order to avoid the participants from remembering the arrangement of the patterns, the patterns are designed to be complex, as described in Section 1.

Our interest variable in this experiment is the task completion time. The task completion time is defined as the time interval between the marker appearing on the screen and the participant successfully pointing at the correct target circle. In addition to the above measurements, we conducted a small questionnaire to obtain free comments from the participants.

4.1.2 Pilot test

Eight participants (six males and two females), between the ages of 21 and 23, were recruited from our laboratory. In order to enhance fairness, each participant was trained to use the prototype with both the DRP and the UPR before the testing. The training typically took 5 minutes. During the training, we advised the participants to slide the tablet in the UPR when comparing the real scene and the displayed image. After the training, the participants engaged in a test session of tasks.

The average task completion time of all the trials was 5.95 secs in the UPR and 7.73 secs in the DPR. The completion time in the UPR was generally shorter than that in the DPR, although for one of the eight, this was reversed. This was because the participant had made a random guess to select the target without following the instructions. We obtained the following comments:

- It was difficult to identify correspondences because the resolution of the screen images was low.
- The screen did not look transparent. It was difficult to move the fixation point between the screen and the target because their focuses were quite different.
- The registration errors when viewing with both eyes were larger than those with just the dominant eye.

4.1.3 Discussion

Most participants predictably completed their tasks faster in the UPR than in the DPR, despite some negative factors that reduced the UPR score as described below. This result at least indicates the possibility of confirming the validity of the exact UPR in a real environment.

One negative factor could be low resolution caused by magnifying a part of a 640×480 pixel image captured by the rear camera in the current prototypes. For fairness, we can improve the resolution of the UPR by using a high-resolution camera for presenting and downsampled images for tracking.

The second and third comments are related to the focusing inconsistency and the binocular rivalry, respectively. Although they can occur in both UPR and DPR, the participants tended to be more distracted in the UPR. It is not possible to solve these problems with simple geometric transformation.

However, there remain factors that cannot be generalized for the above results. First, since the participants already knew about augmented reality techniques, it is possible that the general public might not be able to perform operations such as sliding the tablet in UPR easily. In fact, there was one participant whose time in the DPR was shorter than that in the UPR. The scene could have been too complicated for this participant to find the correspondences precisely. Second, in the DPR, the third instruction could be a type of interference in the task. The participants were required to pay attention to the movement of the tablet only in the DPR because the FOV in the DPR is wider than that in the UPR and the target environment was not large enough to move the tablet freely. Finally, the lens distortion of the screen image in the DPR could be too large for the participants to recognize the patterns. Therefore, a DPR method without lens distortion should be evaluated for comparison.

4.2 Visibility test in a non-planar environment

4.2.1 Method

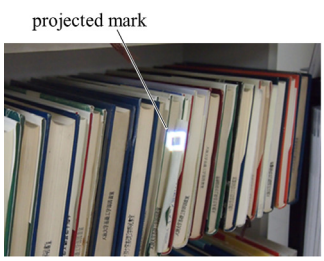
In this section, we describe an experiment using a non-planar target, which is closer to a practical situation. The purpose of this experiment is to confirm that the UPR improves users' visibility against the approximation errors. To evaluate the visibility, we measure the task completion time again. Therefore, the task in this experiment is also to find the target which was randomly and automatically indicated. To remove the restrictions for the movement of the tablet and the pointing stick, we changed the method for finding the target. The target part in the real environment is indicated not with an augmented marker but with a projected one, as shown in Figure 8 (a), and each participant is required to find the corresponding point on the screen. Once the participant finds the corresponding point, he/she has to move the tablet so that the marker fixed on the screen, as shown in Figure 8 (b), coincides with the position of the projected marker in the real scene. This task is designed to be close to authoring applications using augmented reality [17]. While the marker is projected, the tablet screen is blacked out to avoid finding correspondences only from the screen images. The instructions to the participants are as follows:

1. Push the key to start each trial. The system projects the marker at the target position and blacks out the tablet screen for 1 sec. Remember the target position.
2. The screen marker appears at one of the five positions and the projected marker disappears. Move the tablet so that the screen marker coincides with the target in the real image.
3. Once you have finished, push the key again. Perform each trial as quickly and accurately as possible.

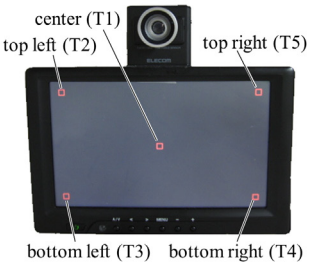
For each screen marker and method, each participant performs the task 20 times. Their dominant eye is confirmed by Dolman's method [18]. Similar to the previous experiment, each participant is trained to use the prototype with both the DRP and the UPR in advance. For each method, the training is performed 20 times. After the training, the participants perform the task, and then comment on this experimental design. We measure the time interval between the two key inputs as the task completion time.

4.2.2 Pilot test

We had 11 male participants, between the ages of 21 and 24. The marker was projected by a projector (Acer X1261, 1,024 $\times$ 768 pixels, 30°, 2,500 lumen) mounted behind the participants and was



(a) projected mark.



(b) displayed mark.

(c) experimental environment.

Figure 8: Non-planar environment for visibility test.

a white square, bright enough to see in the environment. The projection points were manually decided from the feature points whose 3D positions were estimated by PTAMM. The correspondences between the projector and camera image coordinates were obtained by gray-code light projection.

We created a more realistic environment where homography transformation was no longer equivalent to the exact perspective projection. As shown in Figure 8 (c), we placed two bookshelves as a workspace and arranged books on them. In order to prevent the participants from remembering the pattern of the target objects, the books were placed backwards because the fore edges have less texture and fewer characters than the bindings.

In this experiment, we used Prototype 1 as the tablet device. On the tablet screen, a 20 pixel square marker was overlaid at one of the five positions, on the real images rendered by the DPR or the approximated UPR. All the failure frames in either camera or face tracking were recorded. The trials with more than 50% failure frames were excluded. After the tracking failure frames were removed, there remained no trials in which an incorrect target was chosen.

The average task completion times are shown in Table 1. We found that the average completion time in the UPR was shorter than that in the DPR for each position. From the questionnaire, we obtained the following comments.

- In the DPR, it was necessary to observe and remember the entire pattern. In the UPR, the task can be performed without remembering anything.
- The jitters of the real images displayed by the UPR caused difficulty in adjusting the tablet pose.
- It was confusing to find correspondences because of the differences in color and contrast between the scene and the image.

4.2.3 Discussion

The average time of UPR was shorter than that of DPR, as per our prediction. This result indicates the potential to prove the validation

Table 1: Task completion time [sec] in a non-planar environment.

Position	T1	T2	T3	T4	T5	AVG
DPR	1.54	2.24	1.97	1.92	1.97	1.92
UPR	1.53	1.70	1.68	1.63	1.52	1.62

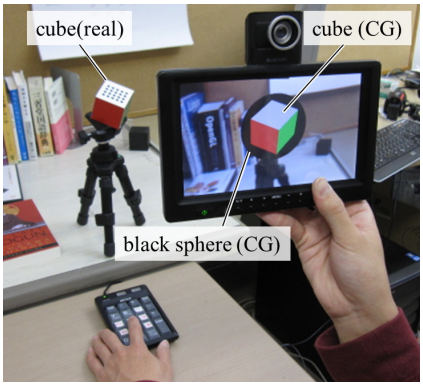


Figure 9: Experimental setup for testing augmentation methods.

of the approximated UPR. The first comment supports this possibility despite the jitter. The participants were confused in UPR because of the jitters, according to the second comment. Since the size of the jitter in the approximated UPR was larger than that of the DPR, the score of the UPR may increase if the jitter is reduced.

Through this experiment, we found two points that we did not expect. First, the non-planar environment was not complex enough to prevent the participants from remembering the pattern. In fact, the difference in the average task completion time between DPR and UPR was smaller in this experiment than that in a planar environment. Thus, it is better to build up a more complex environment or to change the scene more frequently. Second, the difference in the task completion time of T1 was smaller than that of any other position. The time could be dependent on the position of the screen. It is better to improve the experimental method so that the evaluation can be performed not only at discrete positions but also in the entire screen.

4.3 Comparison of virtual object augmentations

4.3.1 Method

This section shows the comparison of augmentation methods. The target methods in this experiment are methods i, ii, and iii (described in Section 2.2) and the simple DPR. These methods correspond to (d), (e), (g) and (c) in Figure 4, respectively. For simplification, we only used the weights $w_{\boldsymbol{x}} = 1, w_{\boldsymbol{y}} = 2^6$ in method iii. To observe some tendencies among these methods, the task in this experiment is to recognize the 3D pose of virtual and real objects and to input its pose.

As shown in Figure 9, we used a real cube whose top is a marker to estimate its pose from camera images. The cube is mounted on a small tripod placed on a desk. The height of the cube is about 25cm. The pose of the real cube can be changed arbitrarily by adjusting the tripod. Some magazines are placed on the desk to add texture.

In addition, we used a virtual cube with the same color faces and size as the real one. Note that the real cube is hidden by a black sphere on the screen so that participants cannot compare both cubes only in the screen images. The virtual cube is placed in the same position (+0 mm) as the real one, or 100 mm to the right (+100 mm). In the case of +0 mm, the participants cannot observe both cubes with the dominant eye simultaneously. In the case of +100 mm , they can observe both cubes simultaneously. The task in each trial is to rotate the virtual cube by using six buttons of a numeric keypad to match the pose of the real one. The initial pose of the virtual cube is given with random noise of less than 20°.

The variable evaluated in this experiment is the angular error between the real and the virtual cubes. We are interested in how accurately the participants can recognize the pose of the virtual cube and not the task completion time. They are instructed to adjust the rotation as accurately as possible.

4.3.2 Pilot test

Nine male participants, between the ages of 22 and 24, were recruited from our laboratory. Table 2 shows the results of this experiment. The angular error is defined using the axis-angle of Rodrigues' rotation formula. We found that the average angular error in (d) was the lowest and that in (c) was the highest. The average error of all trials for the +100 mm position was smaller than that for the +0 mm.

4.3.3 Discussion

The average angular error in (d), (e) and (g) was shorter than that in (c). This result indicates the possibility that the approximated UPR improves the visibility in recognizing the object pose irrespective of the augmentation method used. However, the above-mentioned factors such as jitter and distortion still remain.

In contrast, the error in (d) was smaller than that in (e) and (g). The reason can be considered to be a task design problem, although the task actually demanded that the participants observe both the real scene and the screen. For this experiment, the participants could manipulate the virtual cube without observing the real images. The reason why the error in (d) was the least (best) can be explained by the fact that the virtual object is rendered without distortion in (d). To prove the validity of the approximated UPR, we must improve the overall experimental design for the situations where the participants are required to recognize the tripartite relation among the real scene, the real image and the virtual object.

In addition, we did not expect the small difference between the error in (e) and (g). The cause of this result can be considered to be the goodness of the homography approximation, the task design problem or the balance of the weights. However, isolating the first two problems is difficult by only using experiments in real environments. Virtual reality environments such as that in Baričević et al.'s study may be suitable for this purpose.

Finally, we obtained the tendency that the average error of +100 mm was smaller than that of +0 mm in every rendering method, although the same tendency is present on the magnitude relationship in both results. This result can be explained by the following two factors. First, the participants had to move the tablet because the real and the virtual cubes could not be observed simultaneously in the +0 mm setup. The latency or jitter could affect the score. Second, the participants might see the same appearance by parallel viewing in the +100 mm setup. In fact, a distance of 100 mm is close to the size of the parallax of human eyes. To this effect, some task with recognition of the tripartite relation should be required.

5 Conclusions

This study proposed a novel method for presenting real images on a tablet display for video see-through AR. In this method, the real images are rendered by homography transformation, thus creating an illusion of the screen being transparent from the user's viewpoint. Homography approximation enables real-time rendering without any reconstruction artifact, even in a complex environment. Through the pilot tests of our proposed experiments, we observed the tendency that the users' visibility in the approximated rendering is better than that in the device-perspective rendering. Moreover, we obtained a number of concrete policies to improve the experimental design. Our future work is to improve the prototype system and to conduct a user study based on the discussion of the pilot tests.

Table 2: Results of the comparison of augmentation methods.

	Angular error [°]			
Method	(c)	(d)	(e)	(g)
+0 mm	15.1	8.7	11.6	11.9
+100 mm	11.1	6.8	7.4	7.8

Acknowledgements

The main issue in this research was originally raised in 2009 by KISEGAWA Takuya, a graduate student. He contributed in the implementation of the early prototype under the supervision of Professor CHIHARA Kunihiro and Professor KATO Hirokazu from the Nara Institute of Science and Technology. This research was also supported in part by a grant from JSPS KAKENHI (No. 40432596) and JST A-Step (No. AS242Z03754H).

References

[1] S. Zokai, J. Esteve, Y. Genc, and N. Navab. Multiview paraperspective projection model for diminished reality. In *Proc. 2nd IEEE/ACM Int. Symp. on Mixed and Augmented Reality (ISMAR2003)*, pages 217–226, 2003.

[2] L. Schaul, C. Fredembach, and S. Susstrunk. Color image dehazing using the near-infrared. In *Proc. 16th IEEE Int. Conf. on Image Processing (ICIP2009)*, pages 1629–1632, 2009.

[3] A. Hill, J. Schiefer, J. Wilson, B. Davidson, M. Gandy, and B. MacIntyre. Virtual transparency: Introducing parallax view into video see-through AR. In *Proc. 10th IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR2011)*, pages 239–240, 2011.

[4] E. Kruijff, J. Swan, and S. Feiner. Perceptual issues in augmented reality revisited. In *Proc. 9th IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR2010)*, pages 3–12, 2010.

[5] T. Yoshida, S. Kuroki, H. Nii, N. Kawakami, and S. Tachi. ARScope. In *Proc. ACM SIGGRAPH 2008*, 2008.

[6] D. Baričević, C. Lee, M. Turk, T. Hollerer, and D. A. Bowman. A hand-held AR magic lens with user-perspective rendering. In *Proc. 11th IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR2012)*, pages 197 –206, 2012.

[7] F. Steinicke, G. Bruder, and S. Kuhl. Realistic perspective projections for virtual objects and environments. *ACM Trans. Graph.*, 30(5):1–10, 2011.

[8] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *Proc. Int. Conf. on Computer Vision (ICCV2011)*, pages 2320–2327, 2011.

[9] R. Castle, G.Klein, and D. W. Murray. Video-rate localization in multiple maps for wearable augmented reality. In *Proc. 12th IEEE Int. Symp. on Wearable Computers (ISWC2008)*, pages 225–234, 2008.

[10] J. M. Saragih, S. Lucey, and J. Cohn. Face alignment through subspace constrained mean-shifts. In *Proc. Int. Conf. on Computer Vision (ICCV2009)*, pages 1034–1041, 2009.

[11] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart. Fusion of IMU and vision for absolute scale estimation in monocular SLAM. *Jour. of Intelligent and Robotic Systems*, 61(1-4):287–299, jan 2011.

[12] T. Bonfort, P. Sturm, and P. Gargallo. General specular surface triangulation. In *Proc. 7th Asian Conference on Computer Vision (ACCV2006)*, volume 2, pages 872–881, 2006.

[13] C. Nitschke, A. Nakazawa, and H. Takemura. Display-camera calibration using eye reflections and geometry constraints. *Computer Vision and Image Understanding*, 115(6):835–853, 2011.

[14] G. Carrera, A. Angeli, and A. Davison. SLAM-based automatic extrinsic calibration of a multi-camera rig. In *IEEE Int. Conf. on Robotics and Automation (ICRA2011)*, pages 2652–2659, 2011.

[15] K. Satoh, M. Anabuki, H. Yamamoto, and H. Tamura. A hybrid registration method for outdoor augmented reality. In *Proc. 2nd IEEE/ACM Int. Symp. on Augmented Reality (ISAR2001)*, pages 67–76, 2001.

[16] K. Karsch, C. Liu, and S. B. Kang. Depth extraction from video using non-parametric sampling. In *Proc. 12th European Conf. on Computer Vision (ECCV2012)*, volume 5, pages 775–788, 2012.

[17] G. Reitmayr, E. Eade, and T. W. Drummond. Semi-automatic annotations in unknown environments. In *Proc. 6th IEEE/ACM Int. Symp. on Mixed and Augmented Reality (ISMAR2007)*, pages 1–4, 2007.

[18] W. H. Fink. The dominant eye: Its clinical significance. *Archives of Ophthalmology*, 19(4):555–582, 1938.