Reduction of contradictory partial occlusion in Mixed Reality by using characteristics of transparency perception

Taiki Fukiage* Takeshi Oishi† Katsushi Ikeuchi‡
The University of Tokyo The University of Tokyo The University of Tokyo

ABSTRACT

One of the challenges in mixed reality (MR) applications is handling contradictory occlusions between real and virtual objects. The previous studies have tried to solve the occlusion problem by extracting the foreground region from the real image. However, real-time occlusion handling is still difficult since it takes too much computational cost to precisely segment foreground regions in a complex scene. In this study, therefore, we proposed an alternative solution to the occlusion problem that does not require precise foreground-background segmentation. In our method, a virtual object is blended with a real scene so that the virtual object can be perceived as being behind the foreground region. For this purpose, we first investigated characteristics of human transparency perception in a psychophysical experiment. Then we made a blending algorithm applicable to real scenes based on the results of the experiment.

Keywords: Mixed Reality, Augmented Reality, transparency perception.

Index Terms: I.5.1 [Information interfaces and presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities; H.1.2 [Models and Principles]: User/Machine Systems—Human factors

1 INTRODUCTION

In mixed reality (MR) applications, overlaying virtual objects on real images often causes contradictory occlusion in which a real foreground object is occluded by a virtual object that should be behind the real object. In such cases, users of the application often underestimate the depth and scale of the virtual object or simply perceive that the virtual object does not belong to the scene, resulting in the collapse of the original impact or presence of the MR scene.

Many previous studies have tried to solve the occlusion problem. [10] and [11] reconstructed depth information in a real scene to detect occlusion using a stereo vision-based technique. [13], [2], and [8] handled occlusion by constructing the visual hull [14] of objects using multiple cameras. Although some of these studies enabled real-time interaction in an MR scene without contradictory occlusion, the use of the applications was restricted to a specific local space and not applicable to arbitrary outdoor scenes. As for the methods that do not limit its use to a restricted space, [6] and [18] proposed an algorithm that enabled real-time foreground segmentation from a monocular video sequence. [9] and [19] further extended these methods and handled the occlusion problem caused by moving objects in an outdoor scene. Despite these efforts, however, there is still a difficulty in constructing a natural MR scene with an arbitrary environment especially when a complex object, which is difficult to precisely segment out in real time, exists in the real scene.

Figure 1: Contradictory occlusion seen in the left image is handled well with our blending method in the right image.

When walking around the outdoor scene, we frequently encounter many natural objects such as trees or bushes. All these objects are possible candidates for the occluder to be handled for an MR application used in an arbitrary scene. The goal of our research is to realize a system that can handle such situations and reduce contradictory occlusions robustly regardless of contents in the scene. Considering that computational cost of foreground segmentation will increase with the complexity of the scene, we think it is necessary to find a solution other than improving segmentation methods. In this study, therefore, we focused on designing a method that allows natural visualization of depth ordering between a virtual and a real object for arbitrary scenes without precise segmentation of the foreground regions. Our approach is quite different from other studies in that we took advantage of characteristics of human transparency perception. We utilized the behavior of perceived depth ordering between transparent surfaces for handling contradictory occlusions in a synthesized scene (Fig.1).

The paper is divided into five sections. In the following section, we first show the basic characteristics of human transparency perception, and continue to explain a psychophysical experiment that we conducted to make a model of perceived depth ordering. In the third section, we propose a model of transparency that predicts the results of the experiment. The fourth section provides the blending method designed based on the model of the transparency. Subsequently, in the fifth section, we implement the blending method and test it by simulated MR scenes using several real-scene images. Finally, in the last section, we complete this paper with summary and conclusion.

2 EXPERIMENTAL INVESTIGATION OF HUMAN TRANSPARENCY PERCEPTION

2.1 X-junction model of perceptual transparency

The human visual system simultaneously sees two translucent surfaces at different depths when even a very simple 2D-pattern is presented (Figs. 2A and 2B). According to previous studies [1, 3, 4], the visual system employs simple heuristics related to the luminance pattern around an “x-junction” where four surfaces meet together to stratify the 2D-image into different

* fukiage@cvl.iis.u-tokyo.ac.jp
† oishi@cvl.iis.u-tokyo.ac.jp
‡ ki@cvl.iis.u-tokyo.ac.jp

layers. There are three categories in possible perception depending on the patterns of the x-junction: when a line that progressively passes from brighter to darker regions around an x-junction creates a C-shape, the same surface is always perceived as transparent and in front of the other surface (Fig.2A, unique transparency); when the line creates a Z-shape, either surface can be perceived as in front (Fig.2B, bistable transparency); when the line creates a crisscross pattern, neither surface appears transparent (Fig.2C, no transparency). It is also known that these heuristics roughly reflect physical photometric constraints [1, 12].

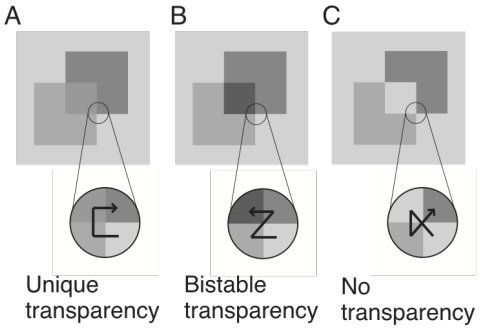

Figure 2: Transparency perception classified based on patterns around an x-junction.

Thus, it would be ideal if we could find a new blending method that produces unique transparency such that a virtual object always appears behind a real foreground object. To create such a situation, however, we have to change the blending equation exactly at the border between a foreground region and a background region in the real scene because the contrast polarity at the edge of the foreground object must be reversed between outside and inside of the virtual object. Thus, unique transparency does not meet the purpose of this study since this kind of algorithm requires an accurate foreground mask.

In this study, therefore, we focused on utilizing bistable transparency. This type of transparency can be easily obtained by a simple blending algorithm because contrast polarities at both edges around the x-junction are retained. As described above, bistable transparency makes perceived depth ordering ambiguous, but previous studies showed that the probability that one surface is perceived as being in front of the other depends on contrasts between edges forming the x-junction [7, 12]. If we know the behavior of the perceptual transparency as a function of contrasts around an x-junction, we will be able to control the perceived depth ordering of a virtual object. Thus, the first step of this study was to examine the situation and make a model of perceived depth ordering.

2.2 Psychophysical Experiment

Several researchers have already investigated how our perception of transparency varies with luminance patterns around x-junctions by using stimuli inducing bistable transparency [5, 7, 12]. In these studies, Delogu et al. have also made a model that predicts perceived depth ordering as a function of lightness around an x-junction [7]. However, these previous studies adopted only parts of all possible patterns in their experiments, and the model may not be applicable to those that they did not examine. In our experiment, therefore, we used a number of stimuli covering various possible luminance patterns and made a more general model that predicts the depth stratification of transparent layers in the human visual system.

2.2.1 Methods

Participants

Eight observers unaware of the purpose of the experiment (7 male and 1 female, aged 22–25) participated in the study.

Apparatus

Stimuli were presented in a dark room on a CRT monitor (Sony Trinitron Multiscan CPD-17SF9, 17 inch, 1024 × 768 pixels, refresh rate 75 Hz, mean luminance 44.6 cd/m^2). Each subject placed his/her head on a chin-rest and used both eyes to view the stimuli. The viewing distance was 86 cm.

Stimuli

The stimulus was composed of a disk (diameter: 4 deg in visual angle) and a rectangle. The disk was presented at the center of the display. The rectangle was presented in the right part of the display and subtended a visual angle of 4 deg horizontally and 8 deg vertically. The disk was split into two regions of the same size by the left-side edge of the rectangle, so the whole image of the stimuli had four different regions: background region (B), rectangle region (R), disk region (D), and overlapping region (O). By setting the luminance values of these four regions adequately, the situation of bistable transparency, in which depth ordering of the disk and the rectangle is ambiguous, was obtained. The patterns of the bistable transparency can be further classified into four cases depending on the contrast polarity at the edges around an x-junction (Fig.3). We tested a total of 438 stimuli including these 4 cases. Every stimulus had a different combination of various luminance values so that as many patterns as possible could be tested. The actual luminance values of each stimulus are available online (http://www.cvl.iis.u-tokyo.ac.jp/~fukiage/ISMAR2012/SupplTable.pdf).

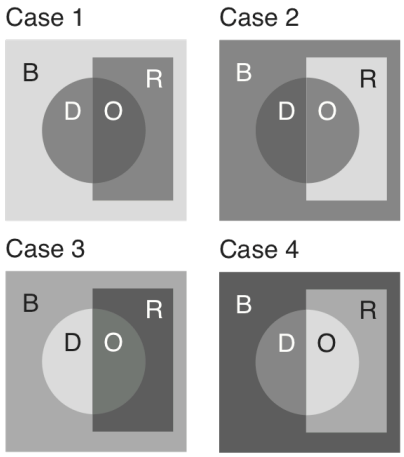

Figure 3: Examples of the experimental stimuli.

Procedure

In each trial, the stimulus composed of a disk and a rectangle was presented for 0.75 seconds. After that, a blank with a random-dot pattern (99% contrast) was followed, during which the observer pressed one of two keys to respond to a two-alternative forced-choice question about whether the disk was perceived as behind or in front of the rectangle. The random-dot pattern was used to prevent adaptation to the specific luminance intensity. The next trial started immediately after the observer pressed a key. The fixation point was presented at the center of the screen at the beginning of each session and during every blank period.

In each session, all of the 438 stimuli were tested in a random order. Eight observers repeated the session 6 times, and 48 responses were collected for each stimulus to estimate the probability that the disk was perceived as behind the rectangle.

2.2.2 Results

The obtained probabilities of disk-behind perception for all the tested stimuli are available online (http://www.cvl.iis.u-tokyo.ac.jp/~fukiage/ISMAR2012/SupplTable.pdf). The results indicated which of the two surfaces appears behind varied largely with the luminance patterns of the four regions (B, R, D and O). In some cases, the disk was always perceived as behind, and in other cases, it was always perceived as in front even though the luminance patterns were those of bistable transparency. This is consistent with the previous studies [5, 7, 12].

3 Modeling Perceived Depth Ordering of Bistable-Transparent Layers

3.1 Delogu et al.'s model

The next problem is to find the determinant of the perceived depth ordering. For this problem, Delogu et al. proposed a model that could predict their results obtained in an experiment similar to ours though they tested only 20 patterns from case 2 and case 3 in Fig. 3 [7]. In their model, preferences for each depth ordering depended on lightness contrast between several abutting regions. Here, "lightness" means perceptually scaled value of luminance. Delogu et al. adopted the equation proposed in [20] to get a lightness value from a luminance level, which is described as follows:

$$W = 25Y^{1/3} - 17, \tag{1}$$

where W is the lightness value, and Y is the luminance level. They argued that a surface that has the highest lightness contrast against all abutting regions is perceived as behind (Highest Contrast Model: Rule 1). The contrast of the disk is defined as $|D-B| + |D-R| + |D-O|$, and the contrast of the rectangle is defined as $|R-B| + |R-D| + |R-O|$, where B, D, R, and O denote lightness values of each region. Thus, the percentage of disk-behind perception would increase with the difference of both contrasts: $(|D-B| + |D-O|) - (|R-B| + |R-O|)$. In addition, they also argued that a surface that has the highest contrast against the overlapping region is more likely perceived as behind if the contrasts of both surfaces defined above are the same (Highest Contrast Model: Rule 2). Thus, the percentage of disk-behind perception would increase with $|D-O| - |R-O|$ if $(|D-B| + |D-O|) = (|R-B| + |R-O|)$.

First, we applied Delogu et al.'s model to our results. In Fig. 4A, we plotted the percentages of disk-behind perception as a function of the lightness contrast difference: $(|D-B| + |D-O|) - (|R-B| + |R-O|)$. Blue, cyan, green, and red circles represent data measured with the stimuli in cases 1, 2, 3, and 4 (see Fig. 3), respectively. The percentages data from cases 2 and 3 gradually increase with the abscissa, which is consistent with Delogu et al.'s model. However, the same model could not explain the variance in cases 1 and 4 because the operation of $(|D-B| + |D-O|) - (|R-B| + |R-O|)$ always outputs zero in these cases. Thus, we plotted the data from cases 1 and 4 as a function of $|D-O| - |R-O|$ in Fig. 4B. This time, the data seemed proportional to the abscissa, but the plot still could not explain the variances induced by changes of luminance values in region O. In Fig. 4B, the data from the stimuli that have the same luminance value except for region O are joined in one line. A line extended vertically indicates that the model could not explain variances within the line. Therefore, we concluded that the Delogu et al.'s model could explain only part of our results.

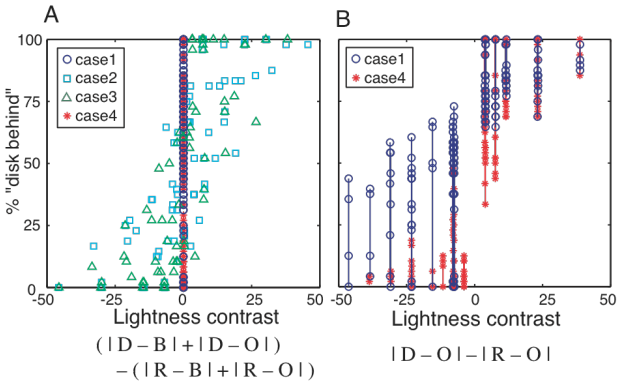

Figure 4: The percentages of disk-behind perception plotted according to Delogu et al.'s highest contrast model. A: Data are plotted according to Highest Contrast Model Rule 1. B: Data are plotted according to Highest Contrast Model Rule 2.

3.2 The model proposed in this study

To explain our results, we made a more flexible model. In our model, we assumed that the preferences for the interpretation of "disk behind" depend on the following value X:

$$X = w\frac{c(D,O)}{c(D,O)+c(R,O)} + (1-w)\frac{c(D,B)}{c(D,B)+c(R,B)}, \tag{2}$$

where w is a weight, and B, R, D, and O are the four regions identified in the previous section. $c(D,O)$ denotes Michelson contrast between luminance values of a region D and O , and is obtained as follows (the same is true for the other pairs):

$$c(D,O) = \frac{|D-O|}{D+O}. \tag{3}$$

Michelson contrast has often been used to scale luminance contrast to perceptual level [15, 17]. An advantage of using Michelson contrast in this study is that it does not require absolute luminance value of the screen in its calculation. We only need to get intensity of each region, which is proportional to luminance if the monitor is linearized.

As shown in Eq. (2), we assumed that the probability of disk-behind perception increases with the weighted sum of two components: one represents how large the contrast between D-O is compared with the contrast between R-O; the other represents how large the contrast between D-B is compared with the contrast between R-B. We replotted our results based on Eq. (2) and fitted the sigmoid function to approximate the data and to get the best-fit w. The sigmoid function (S) we used was as follows:

$$S(x;a,b) = \frac{1}{\exp\left(-\frac{x-a}{b}\right)}. \tag{4}$$

The results we obtained showed small differences between cases 1-4 in Fig. 3, so we fitted different sigmoids for the data from different cases. The data plotted based on our model are shown in Fig. 5. The best-fit parameter of w was 0.93 and the best-fit parameters of sigmoid functions were $(a, b) = (0.41, 0.11)$ for the data from case 1, $(a, b) = (0.57, 0.14)$ for the data from case 2, $(a, b) = (0.55, 0.13)$ for the data from case 3, and $(a, b) = (0.52, 0.12)$ for the data from case 4. Our model clearly provides a better prediction than Delogu et al.'s model. The fact that the best-fit parameter of w was close to 1 suggests that the perceived depth ordering largely depends on the contrasts

against an overlapping region. By using this model, we can estimate the preferences for disk-behind interpretation given the intensities of B, R, D, and O.

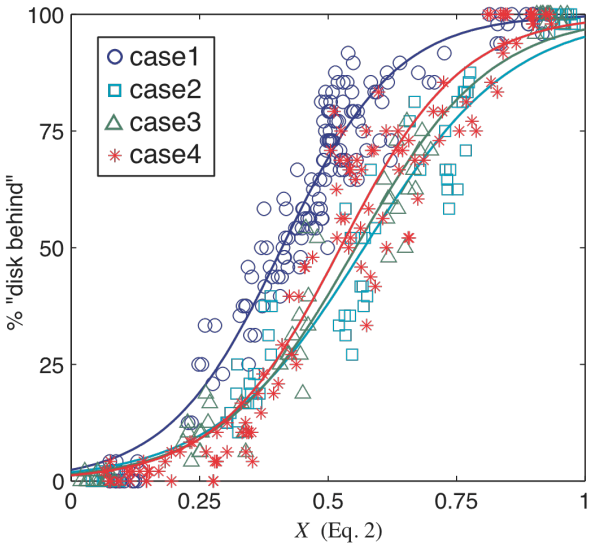

Figure 5: The percentage of disk-behind perception plotted according to our original model. The abscissa is defined in Eq. (2).

4 Blending Method Based on the Model of Transparency

In the previous section, we found a model that can predict the perceived depth ordering of bistable-transparent layers. The next step is to make the best algorithm to blend a virtual object with images of a real scene based on the perceptual model.

There are several ways to make bistable transparency when blending a virtual object with a real image. For example, additive blending is simply adding the intensity of two images, and the obtained result always leads to bistable transparency. Likewise, subtractive blending leads to bistable transparency by subtracting one image from the other image. In these methods, however, there is a problem that the intensity of a resulting blending image often exceeds the maximum intensity or falls below the minimum intensity. The better blending methods we propose here are multiplicative blending, which is as follows:

$$I_M(I_r, I_v) = I_r I_v, \tag{5}$$

and inversed-multiplicative blending, which is as follows:

$$I_I(I_r, I_v) = 1 - (1 - I_r)(1 - I_v), \tag{6}$$

where I_M and I_I are the intensities resulting from each blending method, and I_r and I_v denote the intensity of a real-scene image and a virtual object, respectively. Here, the range of the intensity should be scaled within 0-1. Both of these blending methods always lead to bistable transparency. For example, multiplicative blending applied to a real scene where the intensity of a foreground object is lower than that of the background leads to case 1 in Fig. 3. Multiplicative blending applied to a real scene where the intensity of a foreground object is higher than that of the background leads to case 2 in Fig. 3. Likewise, inversed-multiplicative blending applied to a real scene where the intensity of a foreground object is lower than that of the background leads to case 3 in Fig. 3. Finally, inversed-multiplicative blending applied to a real scene where the intensity of a foreground object is higher than that of the background leads to case 4 in Fig. 3.

Next, we introduced a new parameter α to modify the blending results based on the model we proposed in the previous section. α modulates transparency of a blended virtual object as follows:

$$I_M = \alpha I_r I_v + (1 - \alpha) I_r \tag{7}$$

for multiplicative blending, and

$$I_I = \alpha\{1 - (1 - I_r)(1 - I_v)\} + (1 - \alpha) I_r \tag{8}$$

for inversed-multiplicative blending. Because our experimental data indicated that the perceived depth ordering largely depends on contrasts against an overlapping region, we can monotonically modulate the virtual-behind perception by the transparency of a virtual object. Our model of transparency requires the intensity of four regions to obtain the probability of disk-behind perception. These abstract classes in the experimental stimuli, previously identified as background region (B), rectangle region (R), disk region (D), and overlapping region (O) in our experiments, can be translated into four regions in the actual MR scene as follows: B is a background region of the real scene, R is a foreground region of the real scene, D is a region where a virtual object is blended with the background of the real scene, and finally, O is a region where a virtual object is blended with the foreground of the real scene (Fig. 6).

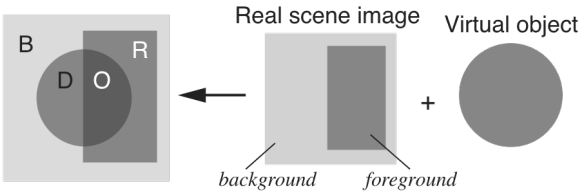

Figure 6: How an actual MR scene can be translated into the abstract stimuli used in the psychophysical experiment.

Therefore, given the intensity of the virtual object I_v, the intensity of the background region of the real scene I_b, and the intensity of the foreground region of the real scene I_f, the intensity of each region can be obtained by Eqs. (7) and (8) as:

$$\begin{cases} B = I_b \\ R = I_f \\ D = \alpha I_b I_v + (1 - \alpha) I_b \\ O = \alpha I_f I_v + (1 - \alpha) I_f \end{cases} \tag{9}$$

for multiplicative blending, and

$$\begin{cases} B = I_b \\ R = I_f \\ D = \alpha\{1 - (1 - I_b)(1 - I_v)\} + (1 - \alpha) I_b \\ O = \alpha\{1 - (1 - I_f)(1 - I_v)\} + (1 - \alpha) I_f \end{cases} \tag{10}$$

for inversed-multiplicative blending. By substituting these values into the model of transparency, we can get the probability that the virtual object is perceived as behind the foreground region as a function of parameter α. In other words, we can determine the best parameter α so that the virtual object

is more likely perceived as behind. As an example of what α value should be chosen with various real scene images, in Fig. 7, we plotted upper limits of α that makes an observer perceive the virtual object as behind for more than a 50% chance as a function of I_b and I_f for both multiplicative blending (Fig. 7A) and inversed-multiplicative blending (Fig. 7B). Here, the intensity of a virtual object I_v was set to 0 for multiplicative blending and 1 for inversed-multiplicative blending, but the qualitative patterns of the results did not change depending on I_v.

The next problem to consider is which of the two blending equation we should use. One of the important determinants to take into account is the visibility of the virtual object. Lower α makes the visibility of the virtual object also lower. Thus we should choose the better blending method depending on the visibility at given I_b and I_f. However, the selected α value is not necessarily a quantitative measure of visibility since the contrast of a virtual object depends largely on the intensity of the real scene image with which the virtual object is blended. In the case of multiplicative blending, for instance, the contrast of a virtual object becomes substantially lower under the condition in which the intensity of a real image is close to 0, even if α equals 1. Likewise in the case of inversed-multiplicative blending, the contrast of a virtual object becomes substantially lower under the condition in which the intensity of a real image is close to 1, even if α equals 1. Therefore, we used the equations below as quantitative measures of visibility:

$$V_M = \alpha I_b \tag{11}$$

$$V_I = \alpha\left(1 - I_b\right) \tag{12}$$

where V_M denotes the visibility measure for multiplicative blending and V_I denotes that for inversed-multiplicative blending. We converted the α values in Fig. 7 using Eqs. (11) and (12), and the results are plotted in Fig. 8.

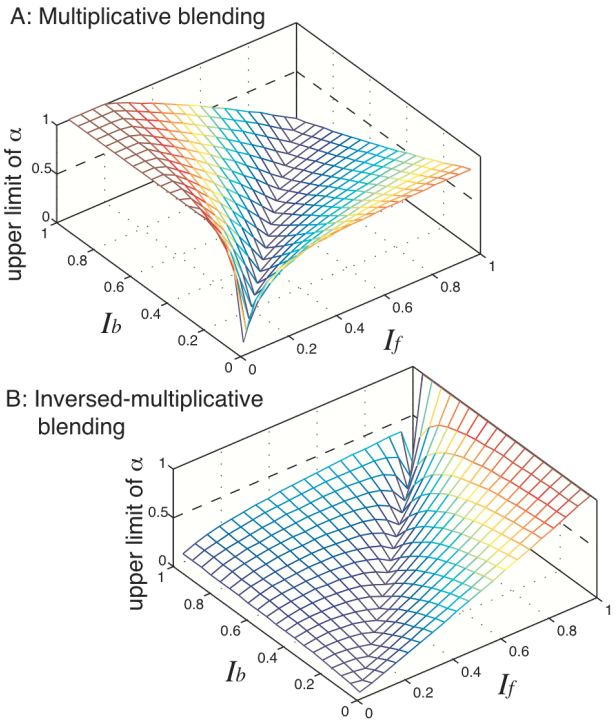

Figure 7: Upper limits of α in which an observer perceives a virtual object as behind in more than 50% chances plotted as a function of I_b and I_f.

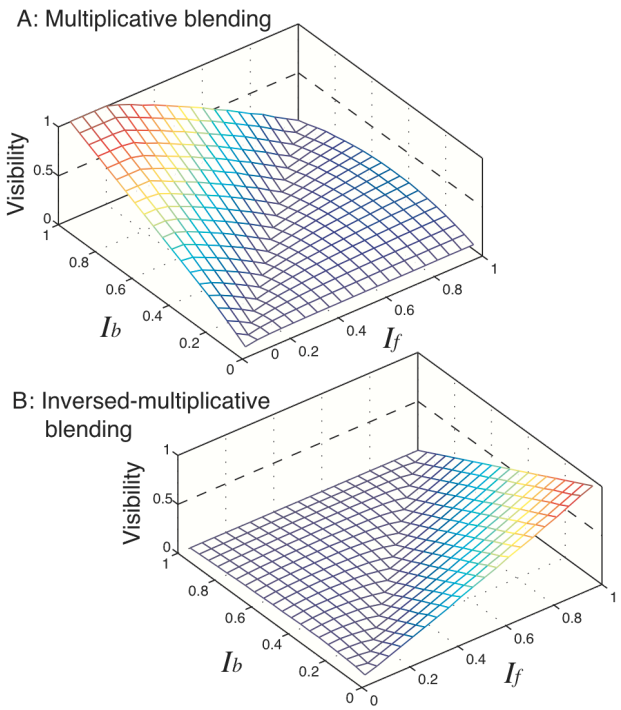

Figure 8: Visibility of the virtual object blended with α in Fig. 7.

As shown in Fig. 8, it is clear that the visibility becomes quite low when $I_f > I_b$ in the case of multiplicative blending, and when $I_f < I_b$ in the case of inversed-multiplicative blending. Thus, we should use multiplicative blending when the intensity of the foreground region is lower than that of the background, and should use inversed-multiplicative blending when the intensity of the foreground region is higher than that of the background. We ensured this simulated result using real-scene images (Fig. 9).

Moreover, the blending equation chosen in this manner always makes the contrast of the virtual object's texture lower within the foreground region than that within the background region. This may also make the virtual object more likely to be perceived behind the foreground region because it is known that the human visual system has a tendency to interpret a high-contrasted region as in plain view and a low-contrasted region as in seen-through view [17].

Figure 9: Comparison between multiple blending and inversed-multiple blending. In this figure, α is set to 0.8 for both blending methods. Multiplicative blending shows a better result when foreground region is darker than background (top left). On the other hand, Inversed-multiplicative blending shows a better result

when foreground region is brighter than background (bottom right).

As an overview, our blending method can be described as follows:

Input I_f (Intensity of a foreground region in the real scene image), I_b (Intensity of a background region in the real scene image), and I_v (Intensity of a virtual object).

Selection of blending equation If $I_f > I_b$, multiplicative blending (Eq. 7) is selected. If $I_f < I_b$, inversed-multiplicative blending (Eq. 8) is selected.

Determining the blending parameter α First, intensities of 4 regions around the x-junction are calculated depending on the blending equation selected above (Eq. 9 for multiplicative blending, and Eq. 10 for inversed-multiplicative blending). Second, these four values are substituted into Eq. 2. Now the probability that the virtual object appears behind can be described as a function of the parameter α. The probability is varied according to the best-fit sigmoid as shown in Fig. 5 (the curve of case 1 is used for multiplicative blending, and that of case 4 is used for inversed-multiplicative blending). Finally, the largest α in those that yield a probability larger than 50% is used as the blending parameter.

5 Implementation and Experiment

5.1 Implementation of the blending algorithm

Based on the results in the previous section, we developed a blending algorithm that is applicable to any real scene with any virtual object. Our method requires a probability map of foreground regions in the real scene, but the map does not need to be accurate. Theoretically, the probability map can be obtained by various ways including depth map, foreground segmentation, and optical flow. Hereafter we assumed that an image of the probability map, which shows the probability density of the existence of occluders at each pixel, is already obtained.

Basically, our method provides the best blending results when a foreground or background region in a given real scene image has a single color. However, such a case is quite rare in the actual outdoor scene to which we want to apply our method. Thus, we overcame this limitation by applying our blending method in a pixel-wise fashion. The algorithm we propose here scans along pixels where virtual objects exist and calculates the best blending equation and parameter α based on the information within a local window centered at that pixel. Since neighboring pixels share most of the pixels within their windows, the blending parameter varies smoothly over pixels. Even transition between different blending equations does not cause any noticeable problem in appearance because the virtual object becomes completely transparent at the area around the switching pixel.

Hereafter we show the details of our blending algorithm. Let (x,y) denote the current coordinates in the scanning pixels and let P_r, P_v, and P_m denote an image of a real scene, virtual object, and probability map, respectively. For each pixel at $P_r(x,y)$, $P_v(x,y)$, and $P_m(x,y)$, the intensities within a square window of a specific size centered at that pixel are examined, and the averaged intensity of the virtual object I_v, the background region I_b, and the foreground region I_f at the current pixel are calculated as follows:

$$I_v = \frac{1}{\sum_{(p,q)\in W} A_v(p,q)} \sum_{(p,q)\in W} P_v(p,q) A_v(p,q) \tag{13}$$

$$I_b = \frac{1}{\sum_{(p,q)\in W} \{1 - P_m(p,q)\}} \sum_{(p,q)\in W} P_r(p,q)\{1 - P_m(p,q)\} \tag{14}$$

$$I_f = \frac{1}{\sum_{(p,q)\in W} P_m(p,q)} \sum_{(p,q)\in W} P_r(p,q) P_m(p,q) \tag{15}$$

where W denotes a group of pixels in the window, and A_v denotes an alpha-channel array of the virtual objects' image, which indicates the existence of a virtual object at each pixel (we assume that the virtual object is rendered on an off-screen frame buffer). These three values are substituted into Eq. (9) if $I_f < I_b$, or substituted into Eq. (10) if $I_f > I_b$. Obtained values of B, R, D and O are then substituted into Eq. (2), and a parameter α is determined so that the probability of "disk-behind" perception becomes larger than 50%. Using this parameter α, the blending result at the current pixel (x,y) is obtained as:

$$P_{blend} = \begin{cases} \alpha P_v(x,y) P_r(x,y) + (1-\alpha) P_r(x,y), & \text{if} \quad I_f \le I_b \\ \alpha\left[1 - \{1 - P_v(x,y)\}\{1 - P_r(x,y)\}\right] + (1-\alpha) P_r(x,y), & \text{if} \quad I_f > I_b \end{cases} \tag{16}$$

If most of the pixels in a window of the current pixel are within the foreground region, the above-mentioned algorithm cannot determine the blending equation or blending parameter reliably. In such cases, the size of the window is repeatedly extended by 1.5 times until a certain ratio (τ) of the pixels is assigned to a background region. This magnified window is used only for calculating the intensity of a background region (I_b). However, it will cause an abrupt change in the blending result among abutting pixels. Thus, we used weighted sum of all intensity values from every window size as an input for Eqs. (9) or (10), which is:

$$I'_b = \left(1 - \sum_{k=0}^{n-1} \rho_k\right) I_{b:W_n} + \sum_{k=0}^{n-1} \rho_k I_{b:W_k} \tag{17}$$

where W_n denotes the window 1.5^n times larger than the original window (W_0), and n denotes the number of repetitions to obtain the satisfactory window size. $I_{b:Wn}$ denotes the I_b calculated within W_n. The weight ρ is defined as follows:

$$\rho_k = \frac{r_k^2}{\tau \sum_{l=0}^{n-1} r_l} \tag{18}$$

where r_k denotes the ratio of background pixels within W_k:

$$r_k = \frac{1}{N} \sum_{(p,q)\in W_k} \{1 - P_m(p,q)\} \tag{19}$$

By this operation, we could switch the window size smoothly among pixels.

By contrast, if all the pixels in a window of the current pixel are within a background region, the color of the virtual object is directly substituted for this pixel since there should be no contradictory occlusion. To make a blending result smooth between these pixels and the other pixels, we introduce the next equation:

$$P_{output}(x,y) = (1-\lambda) P_{blend}(x,y) + \lambda P_v(x,y) \tag{19}$$

where $P_{output}(x,y)$ is the final output of our blending algorithm at the current pixel (x,y). λ in Eq. (19) is a weight function that switches non-blending pixels with the other pixels smoothly. It can be obtained as:

$$\lambda = S\left(\frac{\sum_{(p,q)\in W}\left(1 - P_m(p,q)\right)}{N}; threshold, slope\right) \quad (20)$$

where N denotes the number of pixels in the window, and S is the sigmoid function shown in Eq. (4).

5.2 Experiment

5.2.1 Experiment setup

We tested our method using several static images of real scenes in which foreground objects can cause the occlusion problem. The resolution of the images was 640x480. Images of the probability map of foreground regions were manually generated, but we intentionally made it not so precise that they were not appropriate for usual methods that simply cut out the overlapping region from the virtual objects. In the experiment, we used a personal computer (OS: Windows 7, CPU: Corei7 2.93 GHz, RAM: 8GB, GPU: nVIDIA GTX 550Ti 1024MB). The size of the averaging window (W) was 60x60. The smallest ratio of background pixels (τ), which is used to determine the maximum window size at that pixel, was set to 0.1. In Eq. (20) *threshold* and *slope* were set to 0.9 and 0.05, respectively. Because our algorithm proposed in the previous section calculates the blending parameter in a pixel-wise fashion, we could implement it on the programmable shader (GLSL). To keep an interactive frame rate, we slightly modified the algorithm so that it sampled every 6th pixel within a window when calculating I_f, I_b, and I_v. The actual frame rate largely depends on the number of pixels around and within a foreground region, but it works at a frame rate higher than 25 FPS on most of the cases. Although the frame rate drops to about 8 FPS when a virtual object subtends all pixels and most of the pixels are in a foreground region, this will be easily improved by decreasing the sampling rate adaptively.

5.2.2 Experimental results

The blending results are shown in the leftmost images in Fig. 10. Their neighbors to the right are images for comparison and were obtained by simple alpha blending using the foreground probability map as an alpha-channel mask. Other results are also available in Fig.12 in Appendix. Although the borders between foreground and background are uncertain in the probability maps, the blending results obtained by our proposed method did not cause any sense of contradictory occlusion in most of the cases. On the other hand, the comparison results obtained by simple alpha blending can more likely cause the impression of contradiction if the foreground mask is slightly smaller than the actual foreground region (the second image from the left in Fig. 10A). If the foreground mask is larger than the actual foreground region, a virtual object becomes totally invisible around the edges of the foreground region (the second images from the left in Fig. 10C and Fig. 12A). One of the merits of using our blending method is thus its robustness for the uncertainty of foreground-background borders.

If we use a low-cost foreground detector, the obtained probability maps may be more ambiguous and not saturated. Assuming such situations, we manually generated probability maps in which no pixel indicates "100% foreground" and used those maps to make blending results. The parameter settings were the same as in the previous experiment except for τ (the ratio of background pixels that need to be included within each averaging window). τ was changed from 0.1 to 0.4 to optimize the blending results. The comparison between our method and the simple alpha blending is shown in Fig. 11. The results of our blending method are not so different from those using less ambiguous probability maps in Fig. 10 though the visibility becomes slightly lower. On the other hand, the results of the simple alpha blending cause more or less a sense of contradictory occlusion for all example images. Thus, our method is robust for the ambiguity of a probability map if the parameter τ is appropriately chosen according to the degree of the ambiguity.

Precise segmentation has been one of the bottlenecks in solving occlusion problem in real time. By using our method with an inaccurate, but lower-cost, foreground detector, it becomes possible to reduce contradictory occlusions in MR applications implemented in a hand-held device that does not have a high throughput. For the same reason, our method has the advantage in rendering a virtual object to a real scene where complex foreground objects (e.g., bushes or leaves of a tree) exist. It takes too much computational cost and is quite difficult to precisely segment such a complicated foreground region in real time. In this study, our blending method provides an alternative solution to handle such situations. In addition, our algorithm can make a virtual object seen through a foreground region that is actually an opaque object. Therefore, our blending method may be useful as a new X-ray visualization technique. X-ray visualization methods are extensively studied to make virtual information seen through real foreground surfaces in AR or MR applications [16, 21]. It should be noted, however, that our model of perceptual transparency gives no assurance to provide good results when a virtual object is completely occluded. To handle such situations reliably, we have to make use of other depth cues related to perceptual transparency (e.g., blur, contrast), which is a subject of our future study.

5.2.3 Limitations of our method

Despite these advantages, our method still has several problems to be solved in the future. First, our blending method makes a virtual object almost invisible when the intensity of a foreground region is close to that of a background region. In such cases, our method could produce no better results than the standard alpha-blending method (Fig. 10D and Fig. 12C). To improve this, we will have to combine some other perceptual cues that can reinforce desirable depth perception without lowering the visibility of the virtual object. Second, our algorithm could not provide the optimal blending results when the intensity of a foreground or background region has a very large variance within a local window because our algorithm uses the averaged intensity within a window to determine the blending equation and its parameter (Fig. 1E and Fig. 12D). Since the size of the local window at the foreground pixel becomes larger as the distance from the foreground-background border increases, optimal blending may not be provided especially for such pixels. Third, in the experiment, we manually determined parameters like window size so that the blending results for the experimental images appeared as smooth as possible while keeping the correct depth perception. For example, a smaller window may be able to keep the correct depth perception for a finely textured image, but it will break a spatial consistency of a virtual object across pixels. Considering that the scale or apparent texture of a foreground or background region varies across each scene, it will be desirable to determine the window size and other parameters automatically. Fourth, the algorithm proposed in the previous section only assumes cases for blending virtual objects with a single real scene and is not appropriate for a dynamic scene since it does not compensate for any resulting temporal inconsistencies. However, we think that if we extend the averaging window into the temporal domain using a 3-dimensional window, our blending method will show spatiotemporally smooth results even with dynamic image sequences. For example, if the foreground region is moving, the contrast polarity at the edge of the foreground region may change. In such cases, the blending equation will

also change abruptly. Nevertheless, using a 3-dimensional averaging window, the polarity can change smoothly from positive to negative or vice versa with several frames delay. Thus the blending results would also change smoothly from multiple blending to inversed-multiple blending, straddling the completely invisible state. Finally, we have not yet conducted any user study to confirm the validity of using the psychophysical experimental data obtained in a strictly controlled condition for blending a virtual object with a real scene. Since in a real scene there are many more depth cues, the optimal blending parameter, which is now determined based on the psychophysical experimental data, may be more or less different in value. In the future we have to further optimize our blending method by another psychological experiment using more realistic synthesized images or movies.

6 Conclusion

In this study, we examined the behavior of human transparency perception in a psychophysical experiment and made a model that can predict the results. Based on the model of perceptual transparency, we made a blending algorithm that can effectively reduce the contradictory occlusion information in arbitrary MR scenes. Our proposed method blends a virtual object such that the virtual object is perceived as behind a foreground region in the real scene given only a moderately accurate foreground mask image. The experimental results showed that our method, as compared with the simple alpha blending method, is robust for an MR scene where very complicated foreground objects exist. By combining our method with a low-cost foreground detector, we will be able to make an MR application that can handle occlusion problems in arbitrary scenes in real time.

Acknowledgements

This research received support from Digital Museum Project of Ministry of Education, Culture, Sports, Science and Technology, and Strategic Information and Communications R&D Promotion Programme of Ministry of Internal Affairs and Communications.

References

[1] E. H. Adelson and P. Anandan. Ordinal characteristics of transparency. In *AAAI-90 Workshop on Qualitative Vision*, 1990.

[2] J. Allard, C. Menier, B. Raffin, E. Boyer, and F. Faure. Grimage: Markerless 3d interactions. In *ACM SIGGRAPH Emerging Technologies*, 2007.

[3] B. L. Anderson. A theory of illusory lightness and transparency in monocular and binocular images: the role of contour junctions. *Perception*, 26(4): 419-453, 1997.

[4] J. Beck and R. Ivry. On the role of figural organization in perceptual transparency. *Perception & Psychophysics*, 44(6): 585-594, 1988.

[5] J. Beck, K. Prazdny, and R. Ivry. The perception of transparency with achromatic colors. *Perception & Psychophysics*, 35(5): 407-422, 1984.

[6] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. In *CVPR* (1), pages 53–60, 2006.

[7] F. Delogu, G. Fedorov, M. O. Belardinelli, and C. van Leeuwen. Perceptual preferences in depth stratification of transparent layers: Photometric and non-photometric factors. *Journal of Vision*, 10(2): 1-13, 2010.

[8] J. M. Hasenfratz, M. Lapierre, F. Sillion. A real-time system for full body interaction with virtual worlds. *Eurographics In Symposium on Virtual Environments*, pages 147-156, 2004.

[9] T. Kakuta, L. B. Vinh, R. Kawakami, T. Oishi, and K. Ikeuchi. Detection of moving objects and cast shadows using a spherical vision camera for outdoor mixed reality. In *VRST*, pages 219-222, 2008.

[10] M. Kanbara and N. Yokoya. Geometric and photometric registration for real-time augmented reality. In *ISMAR*, pages 279-280, 2002.

[11] H. Kim, S. J. Yang, and K. Sohn. 3d reconstruction of stereo images for interaction between real and virtual worlds. In *ISMAR*, pages 169-177, 2003.

[12] A. Kitaoka. A new explanation of perceptual transparency connecting the X-junction contrast-polarity model with the luminance-based arithmetic model. *Japanese Psychological Research*, 47(3): 175-187, 2005.

[13] A. Ladikos and N. Navab. Real-time 3d reconstruction for occlusion- aware interactions in mixed reality. In *ISVC* (1), pages 480-489, 2009.

[14] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16: 150-162, 1994.

[15] J. Rovamo, O. Luntinen, and R. Näsänen. Modelling the dependence of contrast sensitivity on grating area and spatial frequency. *Vision Research*, 33(18): 2773–2788, 1993.

[16] C. Sandor, A. Cunningham, A. Dey, and V.-V. Mattila. An Augmented Reality X-Ray system based on visual saliency. In *ISMAR*, pages 27-36, 2010.

[17] M. Singh and B. L. Anderson. Toward a perceptual theory of transparency. *Psychological Review*, 109: 492-519, 2002.

[18] J. Sun, W. Zhang, X. Tang, and H. Y. Shum. Background cut. In *ECCV* (2), pages 628-641, 2006.

[19] L. B. Vinh, T. Kakuta, R. Kawakami, T. Oishi, and K. Ikeuchi. Foreground and Shadow Occlusion Handling for Outdoor Augmented Reality. In *ISMAR*, pages 109-118, 2010.

[20] G. Wyszecki. Proposal for a new color-difference formula. *Journal of the Optical Society of America*, 53: 1318-1319, 1963.

[21] S. Zollmann, D. Kalkofen, E. Mendez, and G. Reitmayr. Image-based Ghostings for Single Layer Occlusions in Augmented Reality. In *ISMAR*, pages 19-26, 2010.

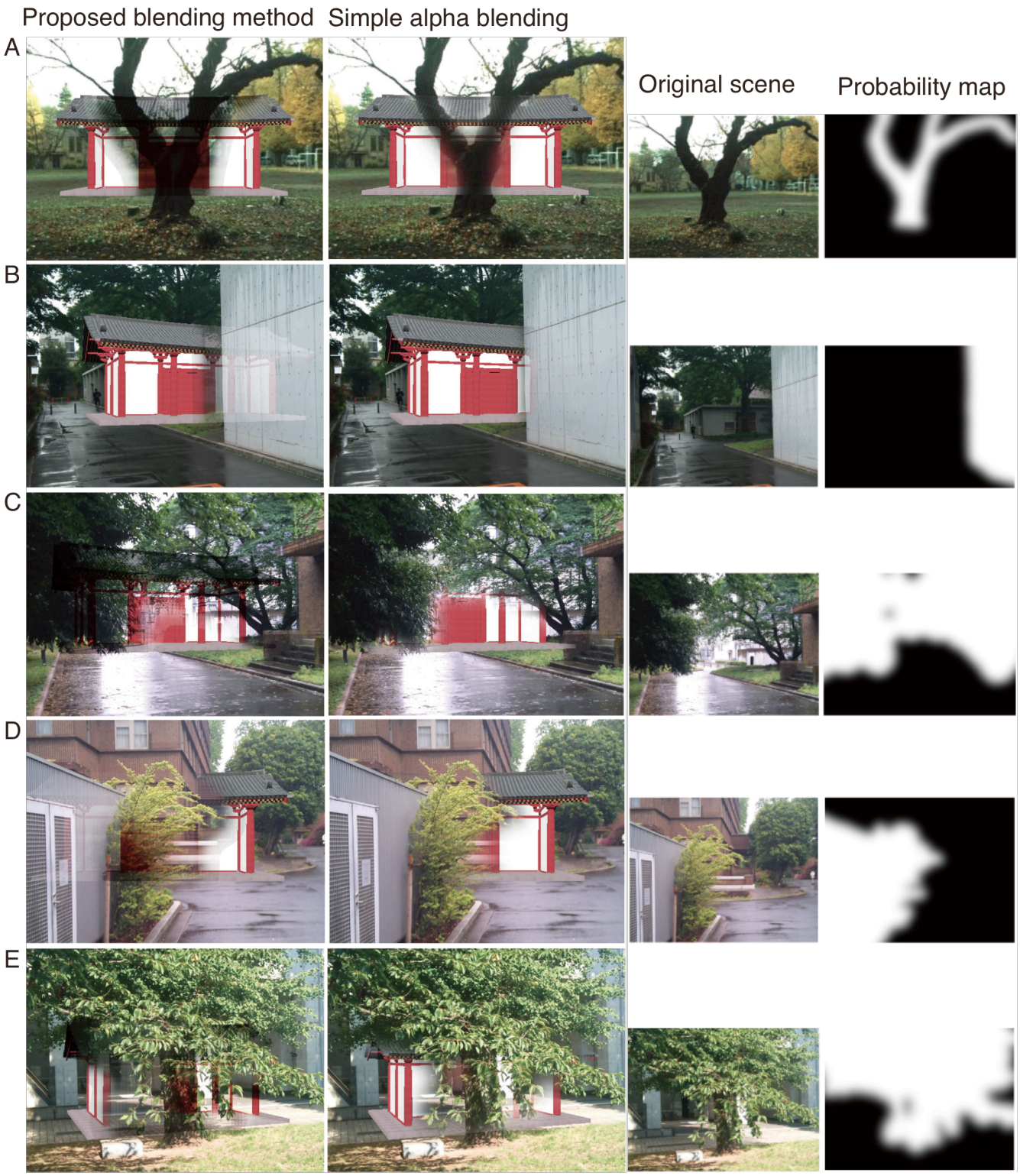

Figure 10: Comparison between the results of our blending method and those of simple alpha blending. Left-most column shows the results of our blending algorithm. The second column shows the comparison results obtained by simple alpha-blending using the probability map as an alpha-channel mask. The third column shows original images of real scenes. Right-most column shows foreground mask images that were manually generated for the simulation. Our method shows better results as compared with simple alpha blending in A, B, and C. In D, however, a virtual object blended by our method becomes almost invisible around the edges of foreground region since the intensity of the foreground is similar to that of the background region. In addition, our method cannot provide optimal results when the intensity of a foreground region or a background region has a large intensity variance within a local window in which the averaged intensity value is calculated. In such cases, the results are often not better than those obtained from the simple alpha blending (E).

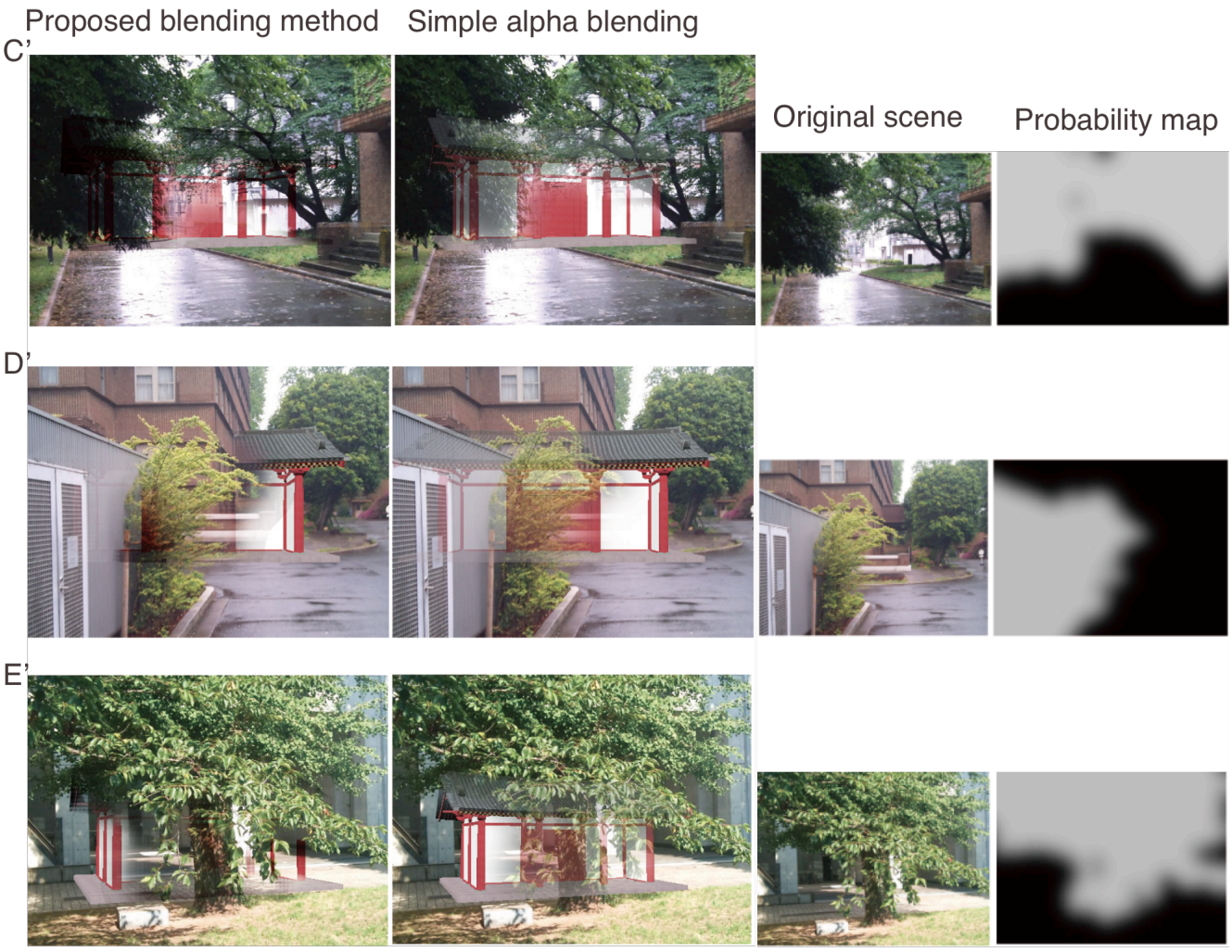

Figure 11: Comparison between the results of our blending method and those of simple alpha blending, using more ambiguous probability maps. If we use a low-cost foreground detector, the obtained probability maps may be more ambiguous and not saturated. Assuming such situations, we manually generated probability maps in which no pixel indicates “100% foreground” and used those maps to make blending results. The results of our blending method are not so different from those using less ambiguous probability maps in Fig. 10 though the visibility becomes slightly lower. On the other hand, the results of the simple alpha blending cause more or less a sense of contradictory occlusion for all example images.

APPENDIX

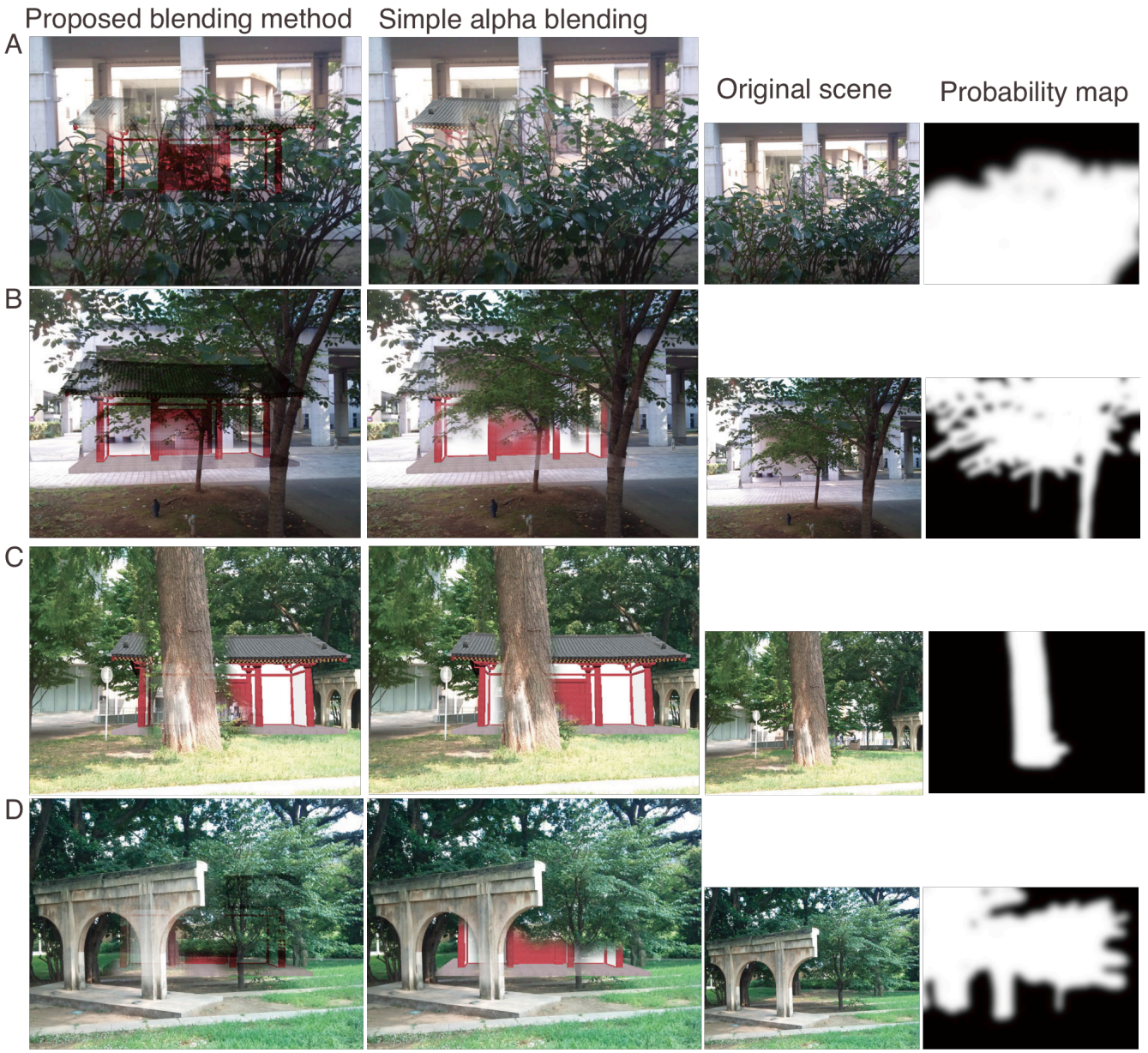

Figure 12. Additional examples of comparison between the results of our method and those of simple alpha blending. Left-most column shows the results of our blending algorithm. The second column shows the comparison results obtained by simple alpha blending using the probability map as an alpha-channel mask. The third column shows original images of real scenes. Right-most column shows foreground mask images that were manually generated for the simulation. Our method shows better results as compared with simple alpha blending in A and B. In C, however, the result of our blending method is no better than that of simple alpha blending because the intensity of a foreground region and that of a background region calculated within the local window are similar to each other. For the same reason, the visibility of a virtual object blended by our method becomes very low in most of the pixels in D.