X-ray Visualization Based on Characteristics of the Human Visual System

TAIKI FUKIAGE[†1] TAKESHI OISHI[†2]
KATSUSHI IKEUCHI[†2]

Abstract: This paper presents a visualization method to show virtual objects while handling the occlusion problem in Mixed Reality (MR) scenes. The occlusion handling is still a challenging problem in the MR applications. It is hard to precisely segment the foreground contour from the real scene in particular when the foreground has a complicated shape such as leaves and branches. We overcome this problem by semi-transparent visualization methods based on the human perception models estimated by the psychophysical experiments. Our method is composed by two blending methods. One is the visibility-based blending, which semi-transparently renders the virtual object with a constant and uniform visibility across different scenes. The other is the bistable-transparency blending, which blends a virtual object such that the virtual object can be seen naturally as being behind the real foreground objects. We show that the proposed method is robust for MR scenes where complicated foreground objects exist.

Keywords: Mixed Reality, Augmented Reality, Occlusion problem, Transparency Perception, Human Visual System

1. Introduction

In Mixed Reality (MR) applications, overlaying virtual objects on real scenes 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. [20] and [21] reconstructed depth information in a real scene to detect occlusion using a stereo vision-based technique. [24], [2], and [14] handled occlusion by constructing the visual hull [26] 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, [8] and [37] proposed an algorithm that enabled real-time foreground segmentation from a monocular video sequence. [18] and [40] 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.

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. Since computational cost of foreground segmentation will increase with the complexity of the scene, we think it is necessary to consider a solution that can work in cases where only rough foreground information is available.

As such a solution, we propose a semi-transparent visualization method that blends a virtual object with a real scene so that the virtual object naturally appears to be behind the foreground region. Unlike the conventional approach, our method does not strictly mask the virtual object at foreground pixels. Thus, the method is more tolerant about errors of the given foreground information.

In realizing the semi-transparent visualization method, we addressed two problems. First one is that the visibility of a semi-transparently rendered object depends significantly on its background. Second one is that semi-transparently rendered objects can often be hard to see as being behind the foreground region. In this study, we solved these problems by designing blending methods based on the characteristics of the human visual system.

The paper is organized as follows. The following section addresses the visibility issue, and proposes the visibility-based blending, which renders a virtual object semi-transparently with a constant and uniform visibility. The third and fourth sections address the latter issue. Specifically, in the third section we conduct a psychophysical experiment to make a model that can predict the perceived depth order of overlapping semi-transparent objects. Based on that model, in the fourth section we propose the bistable-transparency blending, which blends a virtual object so that it appears to be behind a foreground region. Finally, we complete this paper with summary and conclusion.

2. Visibility-based blending

In many interactive applications, one sometimes needs to render an object half-transparently on a background scene image. For example, in portable augmented reality systems, rendering virtual information in 100% opacity can be dangerous because obstacles in the real world are often occluded. Virtual objects

†1 *Graduate School of Interdisciplinary Information Studies, the University of Tokyo*
†2 *the Institute of Industrial Science, the University of Tokyo.*

may also be rendered semi-transparently to be visualized as if they were behind the foreground real objects like in our case [12], [39]. In addition, when showing virtual objects in optical see-through systems, or structured augmented reality systems, virtual information is usually perceived semi-transparently.

Under all of those situations, one often wants to keep visibility of a rendered object constant. However, there is still no established method that can blend two images according to a subjective measure of visibility. In the conventional alpha blending method [34], we can change opacity of one image relative to another image by an alpha value. However, the size of the alpha value does not necessarily correspond with the visibility of an image against another image. For example, given a situation in which a virtual object is blended with a background image, visibility of the virtual object largely depends on intensities and textures of the background scene and the virtual object (left column in Figure 1).

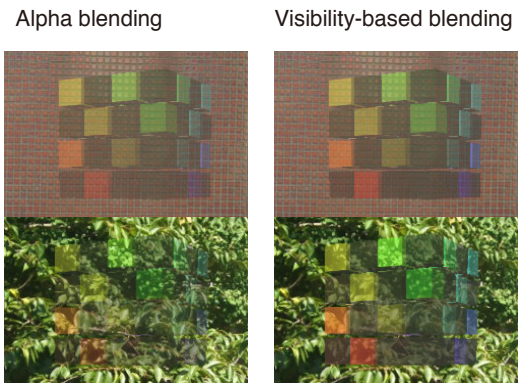

Figure 1. A virtual object is blended with two different background images by (Right column) the visibility-based blending, and (Left column) the conventional alpha blending with a constant alpha value (=0.4).

One possible solution to this problem is to predict the visibility, and optimize a blending parameter. In this work, we employed one of the error visibility models to predict visibility, which have been developed for the purpose of image quality assessment [25], [29]. In the error visibility model, visibility of image distortion is predicted by comparing simulated neural responses for an original image, and those for a distorted image. The simulation of neural responses is based on the computational model of the primary visual area (referred to as V1). In our case, the input images are replaced with an image before blending, and an image after blending; visibility of the blended image is predicted by comparing simulated responses for the two input images.

Based on the visibility model, we propose two blending methods. One is the visibility-based blending, which locally optimizes a blending parameter such that the visibility of the blended object achieves the arbitrarily targeted level. The other method is the visibility-enhanced blending for optical see-through systems, in which visibility of a virtual object is adaptively and locally enhanced. Using the proposed method, we can blend an object with constant visibility across different background scenes (right column in Figure 1).

2.1 Basic features of the Human Visual System

In MR (or augmented reality, AR) visualizations, several studies have worked on improving legibility of virtual information rendered on background scenes [13], [17], [19], [35]. However, any of the methods proposed in those studies doesn't apply to estimating a correct visibility level of a semi-transparent object.

To correctly predict the visibility level of a semi-transparent object on an arbitrary background, we adopted the framework of error visibility metrics for image quality assessment [5], [9], [25], [28], [43]. Those error visibility metrics usually take into account basic features of the human visual system that are particularly important for predicting visibility. Hereafter, we introduce each of the two features and their underlying mechanisms.

2.1.1. Contrast Sensitivity

One of the key features that contribute to visibility can be observed as contrast sensitivity for stimuli with various spatial frequencies (known as contrast sensitivity function, CSF). As shown in Figure 2, contrast sensitivity of the human visual system has a band-pass nature, with its peak at around 2-5 cycles per degree [6], [7]. Evidence from psychophysical and physiological studies has shown that several different neural mechanisms, each of which is tuned to separate, and a more limited band of spatial frequencies, underlie the CSF [6], [16]. Each of the spatial frequency detection mechanisms is also tuned to a specific range of orientations at a local position. It is believed that those mechanisms are implemented by neurons in V1. Thus, it can be said that in the early stage of the visual processing, visual stimuli are linearly decomposed by several different neural channels, each of which are tuned to a specific band of spatial frequencies, a specific range of orientations, and a specific location in the visual field.

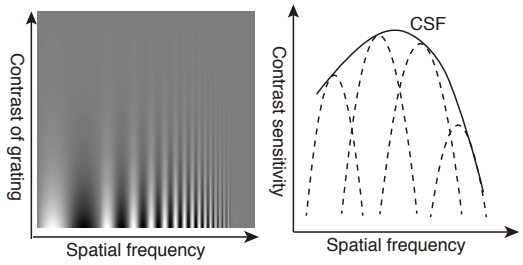

Figure 2. Contrast sensitivity function. (Left) Contrast sensitivity depends on spatial frequency. (Right) Schematic illustration of the contrast sensitivity function (solid line) and its underlying spatial frequency channels (broken lines).

2.1.2. Contrast Masking

Visibility of a visual stimulus also depends on contrast of its background (a phenomenon known as contrast masking [27]). In Figure 3, a sinusoidal target with the same contrast is embedded on different backgrounds. In the leftmost image, the target is presented on a plain background, but in the center image, the same target is added on a background with a similar sinusoidal

pattern. Here, physical intensity increment and decrement relative to background, is exactly the same between both images. However, visibility of the target is lower in the center image. This contrast masking also occurs if the orientation of the background pattern is different from that of the target (see the rightmost image) though the effect becomes relatively smaller [11].

The contrast masking can be explained by a non-linear contrast gain control process in V1. Currently, the most influential model of the gain control mechanism is the divisive normalization model [15]. According to the divisive normalization model, a response of each neuron is divisively normalized by the weighted sum of the responses of neurons that are tuned to the same location (including the neuron whose response is being normalized). Because the response to the target stimulus (or increment of the response regarding the target stimulus) is reduced due to the normalization when another pattern is added on the same location, perceived contrast of the target would also be reduced. The model can explain a vast variety of data, including physiologically measured neural responses and psychophysically measured contrast masking data [15], [38], [43].

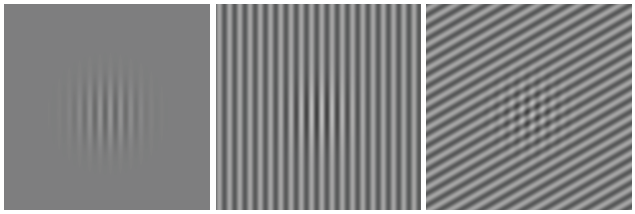

Figure 3. Examples of the contrast masking effect. When the sinusoidal target is embedded on textured backgrounds, visibility of the target decreases though the physical intensity increment or decrement is kept constant across images.

2.2 Visibility model

The error visibility metrics have been developed using computational models of V1 (V1 model) that can simulate the basic features of the human visual system described in the previous section. In this work, we developed the visibility model based on one such metric proposed by [25]. The blending methods proposed in this paper optimize a blending parameter according to the visibility of a blended object predicted by the visibility model.

A schematic of the visibility model is shown in Figure 4. In the visibility model, two input images, an image before blending and an image after blending, are first converted to a color space that is more appropriate to simulate the behaviors of the visual system. Next, the converted images are processed in the computational model of the visual mechanisms in V1, (V1 model) and simulated neural responses of several neural channels are obtained for each location of each image. Then, differences of those neural responses between the two images are pooled across neural channels. Finally, the pooled difference is used as a measure of the subjective amount of visibility for that location.

Although most of the mathematical formulations are common between the visibility model used in this paper and that in [25], some modifications are incorporated to obtain better results, as well as to reduce computational cost. Those modifications are as follows:

1. Using CIE *L*a*b** color space instead of *YUV*
2. Considering local lightness difference in addition to contrast difference
3. Ignoring chromatic contrast difference
4. Ignoring inhibition from surrounding pixels in the divisive normalization process

In the following part, we show the details of the visibility model including explanations for these modifications.

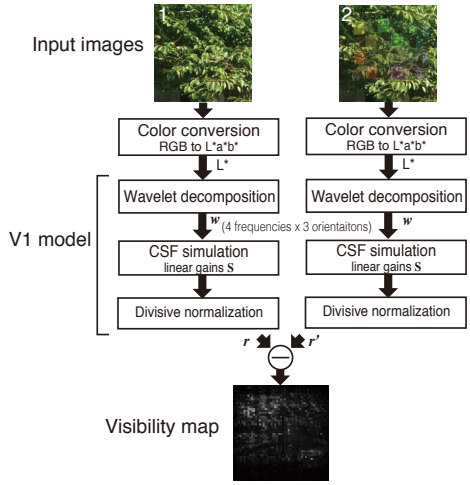

Figure 4. Schematic of the visibility model. The visibility of blending image (the right image) is calculated by comparing simulated neural responses for the blending image and a background image before blending (the left image).

2.2.1. Color Conversion

In the first stage of the visibility model, input images are converted from *RGB* to the CIE *L*a*b** color space. Although [25] used the *YUV* color space, the *L*a*b** is better because the *L** channel in the *L*a*b** is more perceptually linear than the Y channel in the *YUV*. In addition, we only used *L** channel to calculate visibility because sensitivity for iso-luminant color contrast is small compared to that for luminance contrast [30]. Since we assume use of the blending method for real-time applications, we gave priority to efficiency at the expense of a presumably small contribution of the color channels.

2.2.2. Simulation of the Contrast Sensitivity Function

The input images are then linearly decomposed into several oriented frequency domains to simulate behaviors of the neural channels; each tuned to a specific range of spatial frequency bands and a specific range of orientation bands. In [25], the separable QMF wavelet transform (proposed in [36]) was used for the image decomposition. The QMF wavelet filter decomposes an image into 4 frequency bands and 3 orientation bands (horizontal, vertical, and diagonal), giving a vector $\boldsymbol{w}$ composed of 12 coefficients for each location. Although two

diagonal orientations (i.e., 45º and -45º) are confounded with each other in the separable QMF wavelet transform, it fits real time applications quite well since the calculation speed is very fast.

After the transformation, each of the 12 coefficients $\boldsymbol{w}$ is multiplied with linear gains $\boldsymbol{S}$ as follows:

$$c_i = S_i w_i \tag{1}$$

where c_i and w_i denote a wavelet coefficient of the ith filter, after and before the linear gain process, respectively. S_i is a linear gain for the ith filter to simulate the CSF. In [25], S_i is modeled by the following function:

$$S_i = S_{(e,o)} = A_o \exp\left(-\frac{(4-e)^\theta}{s^\theta}\right) \tag{2}$$

where e and o denotes the scale (e can be 1, 2, 3, and 4, from fine to coarse), and the orientation (o=1, 2, 3, each of which stands for horizontal, diagonal, and vertical, respectively). A_o is the maximum gain for the orientation o, s controls the bandwidth, and θ determines the sharpness of the decay. Here, the parameters A_o, s, and θ are given in [25]. The values of those parameters are shown in Table 1.

2.2.3. Simulation of the Contrast Masking Effect

The coefficients are then divisively normalized to simulate the contrast masking effect. According to [25], we used the following equation to obtain the normalized response of neural channel i:

$$r_i = sign(c_i)\frac{|c_i|^\gamma}{\beta^\gamma + \sum_{k=1}^{n} H_{ik}|c_k|^\gamma} \tag{3}$$

where γ is a constant given in [25]. β_i is a saturation constant for the ith filter, which defines the point at which saturation begins (this is also necessary to prevent division by zero). The saturation constants are determined according to a standard deviation of each wavelet coefficient of 100 natural images sampled from a calibrated image database [31]. Since the standard deviations of wavelet coefficients can differ between different color spaces ($L^*a^*b^*$ in our model and YUV in [25]), we recalculated the standard deviations, and multiplied them by a scaling constant b to obtain β_i. The scaling constant b was determined via optimization described in section 3.6.

In Equation 3, H_{ik} denotes a weight that defines the size of influence of the kth filter to the ith filter. H_{ik} is assumed to be larger if the kth filter is neighboring the ith filter in its dimension, and is defined as follows:

$$H_{ik} = H_{(e,o),(e',o')} = K\exp\left\{-\left(\frac{(e-e')^2}{\sigma_e^2} + \frac{(o-o')^2}{\sigma_o^2}\right)\right\} \tag{4}$$

where (e, o) and (e', o') indicates the frequency level and orientation to which each of the ith and kth filters is tuned. K is a normalization factor, which ensures that summation of H_{ik} for all k equals one. σ_e and σ_o are given in [25]. In [25], they assumed not only interactions from nearby frequency levels or orientations, but also interactions from nearby pixels. However, it is quite time consuming to access surrounding pixels every time we calculate each of the divisive normalization responses. Since we need to iteratively calculate the visibility to optimize a blending parameter, in this work, we approximated the weight function H_{ik} as in Equation 4, omitting the term related to the spatial interaction. In section 2.5.1, we show that the approximated model can predict visibility of a blended pattern quite well. A previous study also suggested that spatial pooling over space was very localized [43].

2.2.4. Responses for Local Lightness

In [25], only 4 band-pass subbands are taken into consideration for visibility calculation. Thus, the visibility model in [25] cannot correctly predict visibility if the differences exist in the frequency range lower than that covered by those subbands. This defect can cause incorrect blending results due to visibility underestimation around pixels where both virtual object and background real scene have smooth surfaces (e.g. sky, less textured walls, darkly shaded regions, etc.).

To prevent this, in this work, we additionally consider responses for local lightness by using low-pass residual in the result of the QMF wavelet transform. We modeled the response for local lightness r_L as follows:

$$r_L = \omega w_L \tag{5}$$

where w_L denotes a wavelet coefficient of the low-pass residual and ω denotes a linear gain.

2.2.5. Pooling Simulated Responses

After simulated responses are obtained for both input images, the differences of the responses between the two images are pooled across neural channels for each location. This process is modeled as an l_p norm:

$$d_{xy} = \frac{1}{n+1}\left(|r_L - r_L'|^p + \sum_{i=1}^{n}|r_i - r_i'|^p\right)^{\frac{1}{p}} \tag{6}$$

where d_{xy} denotes the pooled difference of simulated responses for a local position (x,y). r_i and r_i' are the simulated responses of the ith neural channel (filter) for each of the two input images. n is the number of neural channels and thus is equal to 12. r_L and r'_L are the simulated responses for local lightness for each of the two input images.

2.2.6. Parameter Optimization

In [25], the parameters in the visibility model were optimized via fitting to a set of subjectively rated image quality data. They demonstrated that the optimized model not only explains a larger set of image quality data, but also reproduces basic trends in psychophysical data (i.e., contrast sensitivity and contrast masking). So as not to impair the compatibility of their optimized model, we used the parameters given in [25], except for the saturation constants $\boldsymbol{\beta}$ (in Equation 3), and a linear gain ω for local lightness (in Equation 5). Thus, in this paper, only two parameters (the scaling constant b and the linear gain ω) were optimized.

The parameters were optimized via fitting to the subjectively rated visibility of a pattern that was blended with various natural textures and with various transparencies. The detail of the data acquisition procedure is described in section 2.5.1. To compare

the visibility predicted by the model simulation with a subjective visibility score, the local visibility values d_{xy} (Equation 6) were pooled across pixels according to the following equation.

$$d = \frac{1}{m}\left(\sum_{(x,y)\in O} d_{xy}^{q}\right)^{\frac{1}{q}} \quad (7)$$

where $\boldsymbol{O}$ denotes a group of pixels that belong to the pattern, and m is the number of pixels in $\boldsymbol{O}$. Here, we used q=2.2, according to [25]. The parameters (b, ω) were optimized by minimizing the residual sum of squares as a result of linear regression between the subjective visibility scores and the predicted visibility d.

The obtained parameters (b, ω) were (10.3, 0.35). The saturation constants $\boldsymbol{\beta}$ scaled by b are shown in Table 1. It should be noted that the saturation constants $\boldsymbol{\beta}$ obtained in this paper are quite similar to those obtained in [25]. Thus, the changes in those parameters did not affect the predictability of the model optimized in [25].

Parameters	Optimized values				
A_o	40 when o=1 or 3 (horizontal or vertical) 36.6 when o=2 (diagonal)				
s	1.5				
θ	6				
γ	1.7				
σ_e	0.25				
σ_o	3				
p	4.5				
ω	0.35				
		e=1	e=2	e=3	e=4
$\beta_i=\beta_{(e,o)}$	o=1,3	0.3	0.8	1.9	4.6
	o=2	0.2	0.5	1.1	2.7

Table 1. Parameters of the visibility model used in this study. In these parameters, ω and β were optimized via fitting to the subjectively rated visibility data obtained in this work (section 2.5.1). The other parameters were obtained from [25].

2.3 Visibility-Based Blending

Based on the visibility model described in the previous section, we propose the visibility-based blending. The visibility-based blending locally optimizes a blending parameter (α) such that the visibility of the blended object achieves the arbitrarily targeted level. The blending equation we assumed is as follows:

$$I = \alpha I_1 + (1-\alpha) I_2 \quad (8)$$

where I_1 denotes an image intensity of the to-be-blended object and I_2 denotes an image intensity of the background scene (both colors are in the $L^*a^*b^*$ color space).

A schematic of the visibility-based blending is shown in Figure 5. In the first stage, the two input images are converted into CIE $L^*a^*b^*$ color space and the images in L^* channel are decomposed by the 4-scale separable QMF filter. Those two images are a background image before blending and an image in which a to-be-blended object is rendered on the background image with 100% opacity. Since the QMF transform is a kind of

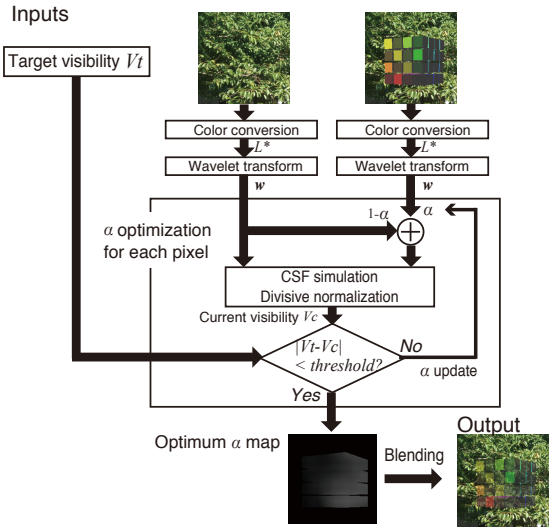

Figure 5. Overview of the visibility-based blending.

linear transform, we can generate decomposed image data of any blending image with arbitrary transparency level by linearly combining these two decomposed images.

After the QMF transform, we have 12 coefficients (4 frequency levels by 3 orientations) for every location of the input images. The next step is to find an optimum blending parameter to realize the target visibility for every location. The optimum α is searched for by the binary search method. In every step of the search algorithm, the visibility of the rendering result by the current α is calculated and whether the visibility is higher than the target visibility is checked.

The visibility at the current α is obtained as follows. Firstly, the coefficients of the blending image at the current α are generated by linealy combining the coeffients of the two input images using the current α and Equation 8. Here, I in Equation 8 denotes the combined coefficients. I_1 and I_2 denote the coefficients of the input image 1 (background scene) and the coefficients of the input image 2 (the background + an opaque object), respectively.

The combined coeffients are then processed by the linear gains $\boldsymbol{S}$, and divisively normalized according to Equation 3. The coeffients of the background image (the input image 1) are also processed by Equation 3. The responses for local lightness are also calculated for both images by Equation 5. Then, the pooled difference of those simulated responses is calculated by Equatioin 6.

The value d obtained in Equation 6 is used in comparison to the target visibility. The next α is decreased if the visibility d is higher than the target and the next α is increased if d is not higher than the target. The size of increment/ decrement is initially set 0.25, but it is halved at the end of every step. The initial blending parameter α_0 is 0.5. The search is finished after 8 iterations.

Finally, the blending is conducted according to Equation 8, using the optimized α. However, a locally optimized α can often cause artificial edge or discontinuity in appearance of a blended object because optimization is independent across pixels. Therefore, we averaged each α within a predifined window. The

size of the window is emperically given.

2.4 Visibility-Enhanced Blending for OST displays

In usual optical see-through devices using half-mirrors, colours of a virtual object are added on colours of a real scene. Therefore, a virtual object the observer sees is always transparent to a certain extent. Under such circumstances, the visibility of a virtual object depends not only on incoming light intensity from the real scene and the display device, but also on textures or structures of the virtual object and its background real scene. Using the visibility model, we are able to take into consideration such attributes to predict visibility. Here, we propose a blending method that can adaptively enhance the visibility of a virtual object added on a real scene in OST displays. In our method, the visibility is enhanced by increasing intensities of local pixels where visibility is lower than the targeted level.

To accurately predict visibility of virtual objects in optical see-through systems, we need to know the exact location of the object in the scene in the user's visual field. Moreover, we have to know the adaptation level of the user's eyes to the current light level in the real scene. However, in the present work we assumed that the simulated MR/AR scene image under accurate calibrations is already given, and focused on describing the visibility enhancement method itself.

A schematic of the visibility enhanced blending is shown in Figure 6. As shown in the figure, the process of the visibility-enhanced blending is almost the same as that of the visibility-based blending in the previous section. The major difference is that we need three input images: (1) a background real scene image, (2) a simulated mixed reality scene, as an original rendering result, and (3) a simulated mixed reality scene in which the object is rendered with the maximum lightness level. To obtain the object's color of the maximum lightness, the object's image is first converted to CIE $L^*a^*b^*$ space and then the values in L^* channel are replaced with the maximum value. The final blending result is obtained by linearly combining the original rendering result and the maximally enhanced result with a locally optimized weight (α) for each pixel using Equation 8. Here, I_1 denotes the maximally enhanced image, and I_2 denotes the original rendering result.

In the optimization process, the optimum α, which shows the nearest visibility to the target visibility, is searched for within a range between 0 and 1 by the binary search. In each step of the search algorithm, the visibility with the current α is calculated by Equation 6. Here, simulated responses for a simulated mixed reality scene, generated by a linear combination of the input images (2) and (3) with the current α, are compared with simulated responses for the background real scene. The other details in the optimization process are exactly the same as those of the visibility-based blending method.

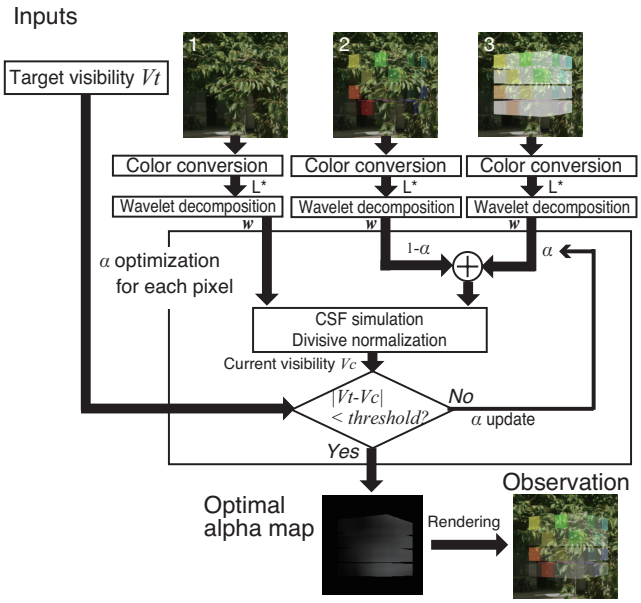

Figure 6. Overview of the visibility-enhanced blending.

2.5 Experiment

In this subsection, we firstly test the validity of the visibility model, which we described in section 2.2 and used in the proposed blending methods. As for the original visibility model proposed in [25], they demonstrated that their model can explain subjective error visibility data for a large variety of image distortions. However, how well the model can explain perceived visibility of a blended object was not explicitly studied. Moreover, we modified their model in several points. Thus, we need to validate our version of the visibility model. After the validation of the visibility model, we tested the proposed blending methods using several real scene images and virtual objects.

2.5.1. Validation of the Visibility Model

We conducted an experiment in which human observers rated the visibility of a pattern blended by various levels of transparency on various textures. The rated visibility data was used to test the visibility model as well as to optimize a couple of parameters in the model (see section 2.2.6 for the details of the parameter optimization). Here, we show the details of the data acquisition procedure and the results of comparison between the visibilities obtained from the visibility model and subjectively rated visibility data.

2.5.1.1. Methods

- *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 114 cm. According to [25], [29], the visibility model assumes that images are observed at a distance where the images are sampled at 64 cycles per degree. The viewing distance was determined by following this assumption.
- *Stimuli.* In every stimulus, a checkerboard pattern was blended on a natural texture image (Figure 7). The

checkerboard pattern subtended 200 pixels (a visual angle of 3.1 deg) both horizontally and vertically, and was composed of two colors, whose RGB values are (0, 0.8, 0) and (0.2, 0, 0.2). 50 different photo images were used as the texture images. The resolution of the textures was 512 x 512 and subtended 8 deg in visual angle. The texture images were mostly taken from [33]. 48 homogeneous textures in frontal perspective were chosen from the database. Those textures included photos of bark, brick, fabric, flowers, food, grass, leaves, metal, sand, stone, and tile. 2 photo images of leaves were additionally taken by one of the authors. The checkerboard and the textures were blended by the simple alpha blending (Equation 8). For each natural texture image, 5 blending images were produced using different α's. The α was modulated approximately on a logarithmic scale so that visibility of the checkerboard varied as equally and broadly as possible.

Figure 7. Examples of the stimuli. The observers rated the visibility of the checkerboard pattern blended on the natural texture image.

- *Static and Dynamic conditions.* In addition to the static condition in which both the checkerboard pattern and the texture image were fixed at the center of the display, we also tested the dynamic condition in which the checkerboard pattern and the texture image were moving at different speeds, assuming practical situations. Under the dynamic condition, both the checkerboard pattern and the texture image were swinging horizontally in the same direction. Their speeds were modulated sinusoidally in the same temporal frequency, 1 Hz, but the widths of the swings were different: 0.8 deg for the checkerboard and 1.6 deg for the texture.
- *Participants.* Ten observers, unaware of the purpose of the experiment (9 male and 1 female, aged 22–27), participated in the study. 9 of the observers completed both static and dynamic conditions. The other male observer participated only in the dynamic condition.
- *Procedure.* Before starting the experiment, a training session was conducted. In training, the approximate range of visibility of the stimuli was presented, and the observers were told to make a consistent criterion to judge visibility. In the experiment, one of the stimuli was presented for 1.6 seconds in each trial. After disappearance of the stimulus, the observer evaluated visibility of the checkerboard pattern in a numerical scale of 1 to 5, where 1 denotes "invisible," 2 denotes "barely visible," 3 denotes "visible," 4 denotes "fairly visible," and 5 denotes "very clear." Those words were always presented beside the corresponding numerical values. The observer could also choose an intermediate scale between arbitrary abutting scales. The observer performed the task by using a mouse. For each of the static and dynamic conditions, there were in total 250 stimuli. The 250 stimuli were presented in a random order. For those who participated in both of the conditions, the observers completed the dynamic condition first, and the static condition was conducted on another day. A training session was conducted every time they started the experiment in that day.

2.5.1.2. Results

We compared the visibility estimated by the visibility model described in section 3 with the subjectively evaluated visibility. The subjective data was converted into Z scores within observers using the following equation:

$$z = \frac{v - \mu_v}{\sigma_v} \tag{9}$$

where v denotes a raw score of visibility. μ_v and σ_v denote the average and the standard deviation of the raw scores for the 250 stimuli, respectively. The z scores of individual observers were then averaged across observers for each stimulus, which was used as representatives for subjective visibility.

We calculated the visibility by the visibility model described in section 3 for each of the 250 stimuli. In calculating visibility, a stimulus image and a texture image of the stimulus were used as the input images. To obtain a representative value of visibility of the pattern as a whole, we pooled d_{xy} in Equation 6 using Equation 7.

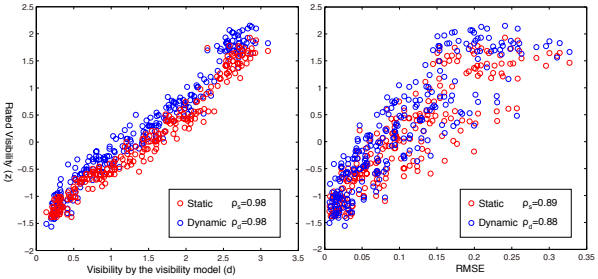

Figure 8. Subjectively rated visibility (z scores) plotted as a function of Predicted visibility d (left) and RMSE (right). ρ_s and ρ_d shown in each plot denote Pearson's correlation of the static condition and the dynamic condition, respectively.

In the left plot of Figure 8, the subjective visibility (z scores) was plotted as a function of the predicted visibility (d values in Equation 7) for each of the 250 stimuli. Red circles show the data of the static condition, and blue circles show the data of the dynamic condition. As a comparison, in the right plot of Figure 8, we also plotted the same subjective visibility data as a function of Root Mean Squared Error (RMSE) between a blending image and a texture-only image calculated in L^*. The Pearson correlation of each plot was also shown in Figure 8.

As shown in the scatter plot and its Pearson correlation, the prediction by the visibility model was remarkably good, despite the fact that most of the parameters of the visibility model were obtained from [25]. Although the data of the subjective visibility

was slightly higher in the dynamic condition than in the static condition, the predicted visibility linearly correlated with those data in both conditions.

The reason why the subjective visibility was higher in the dynamic condition may be that the perceived visibility was temporally pooled in a winner-take-all fashion across frames in the dynamic condition. Another possibility is that adaptation of the detection mechanisms in the visual system may reduce responses to the checkerboard pattern in the static condition. Taking into consideration those behaviors in the visual system would further improve predictability of the model.

However, given the linearity and high correlation between the prediction and the subjective data, we can conclude that the model used in the present study was accurate enough for practical uses.

2.5.2. Evaluation of the Proposed Blending Methods

In this subsection, we firstly describe the details about the implementation and evaluate the efficiency of the blending methods. Then, we show the effectiveness of each of the proposed methods using several experimental images.

2.5.2.1. Imperimentation

We implemented all calculations in both of the proposed blending methods in the GLSL shader. The QMF transform in each scale was implemented in GLSL as shown in Figure 9. In the 1st and 2nd passes, the original image is horizontally convolved by a one-dimensional low-pass (1st pass) or high-pass (2nd pass) filter kernel, and down-sampled in the same direction. Those convolved images are rendered in the same frame buffer. Then, in the 3rd and 4th passes, the combined convolved images are vertically convolved by the low-pass (3rd pass) or high-pass (4th pass) filter kernel and down-sampled. A resultant low-pass image "LL" is then processed into the convolution process in the next scale. In this way, 6 convolutions in each frequency level are accomplished by 4 passes. The 4-scale QMF transform was thus completed after 16 convolution passes.

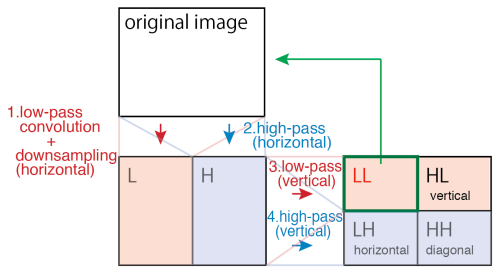

Figure 9. Processes of the QMF wavelet transform in each scale.

In a preliminary experiment, however, we found that downsampling noises in the lower frequency subband images can cause temporal inconsistency in the blending result across frames. To reduce the downsampling noise while keeping the computational speed as fast as possible, we modified the algorithm of the QMF wavelet transform such that the downsampling is only applied in the two higher frequency levels. Accordingly, distances between sampling pixels for the convolution kernel were doubled in the lowest frequency level.

In the visibility-based blending, L^* channel of the two input images are rendered in different channels of a single image, and every convolution is conducted together for both of the images. To reduce degradation of convolved image values due to quantization, we preserved the data in each pass using 2 channels (16 bit) for each input image (i.e., R and G channels for one image, B and alpha channels for the other image). In the case of the visibility-enhanced blending, two of the three input images are rendered within a single image and the other input image is rendered on another image. Therefore, the QMF transform is conducted twice to obtain wavelet coefficients of the three input images.

2.5.2.2. Computational Efficiency

In the experiment, we used a personal computer (OS: Windows 7, CPU: Corei7 2.93 GHz, RAM: 8GB, GPU: nVIDIA GTX 550Ti 1024MB). The resolution of the input images was 640x480. The size of the window to average each optimized α was 65 x 65. Under this condition, both of the proposed blending methods worked at a frame rate higher than 100 FPS.

2.5.2.3. Experiment on the Visibility-Based Blending

We tested the blending method assuming a situation in which a virtual object is blended with a static real-scene image. The resolution of the image was 640x480.

Firstly, we tested the visibility-based blending by blending a virtual object with two different real scene images (one had a relatively smooth texture and the other had a high-contrast texture) using 4 different target visibilities (v_t=0.6, 1.2, 1.8, and 2.4). As a comparison, we also blended the same virtual object with the same real scene images using the conventional alpha blending using 4 different alpha values (α=0.2, 0.4, 0.6, and 0.8). The results are shown in Figure 10. In the results of the visibility-based blending, the visibility of the virtual object (the colorful cubes) looks similar between the two vertically aligned images (an image pair in which the same target visibility was used). By contrast, in the results of the alpha blending, the visibility looks significantly different between the two vertically aligned images though the blending parameters (α) are the same for both of them.

In Figure 11, we show additional experimental results including a more practical situation. Here, a virtual model (a colorful cube or a tower-like building) was blended by the visibility-based blending (left column) and by the alpha blending (right column). In the results of the alpha blending, a constant alpha value was used for every region of the same scene. However, the visibility of the virtual object blended by the alpha blending looks different between regions within the image.

This problem of non-uniform visibility is found in both results of the alpha blending. On the other hand, in the results of the visibility-based blending, the problem is mitigated, and every part of the virtual object looks almost uniform in every image (the target visibility was 1.5). Therefore, the visibility-based blending will be useful when one wants to show

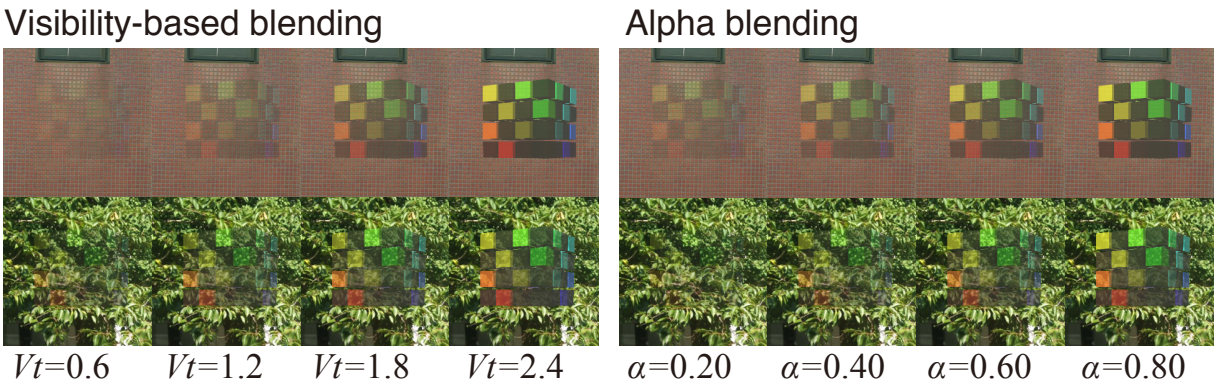

Figure 10. Blending results by the visibility-based blending with 4 different target visibilities, v_t (left images) and by the conventional alpha blending with 4 different alpha values (right images).

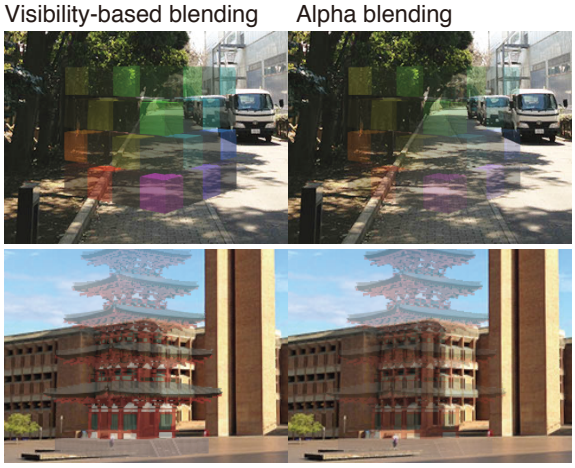

Figure 11. Examples of the visibility-based blending (Top images) and comparison results of the conventional alpha blending (Bottom images).

a virtual object with constant and uniform visibility across different scenes as well as across local regions within the same scene, irrespective of textures or structures in the scene.

2.5.2.4. Experiment on the Visibility-Enhanced Blending

Here, we tested the visibility-enhanced blending described in section 2.4. To see how our blending method works under ideal calibrations, we first simulated rendering results in an optical see-through system within an intensity range between 0 and 1 using Equation 8. Here, α represents relative influence of the light from the device to that of the incoming light from the real scene. I_1 and I_2 denote linearized RGB colors of a virtual object and a real scene, respectively. The parameter α we used in the experiment was 0.5. In Figure 12A, we show experimental results obtained by the simulation. In each image, a virtual object (a colorful cube or an ancient building) was blended on a real scene. In each row of the figure, the left image shows the result by the visibility-enhanced blending (target visibility=1.5), and the right image shows the original scene without enhancement. In the original images (right column), the virtual objects are partially hard to see. In the results of the visibility-enhanced blending (left), the visibility is improved in those regions, and we can perceive the whole contour of the virtual object.

Secondly, we tested the visibility-enhanced blending using an actual OST glasses (MOVERIO BT-200, EPSON). To analyze the real scene, we captured the real scene by a camera (Grasshopper2, Point Gray Research). In this experiment, the calibrations were manually conducted such that appearance of the input images for the blending pipeline and that of the actual scene seen through the glasses became as similar as possible (both photometrically and geometrically). Then, the optimized virtual scene was presented on the glasses. The resultant AR scene was captured from outside of one of the glasses by the camera (Grasshopper2). The results are shown in Figure 12B. Again, we can see that the visibility is improved in the result with visibility enhancement.

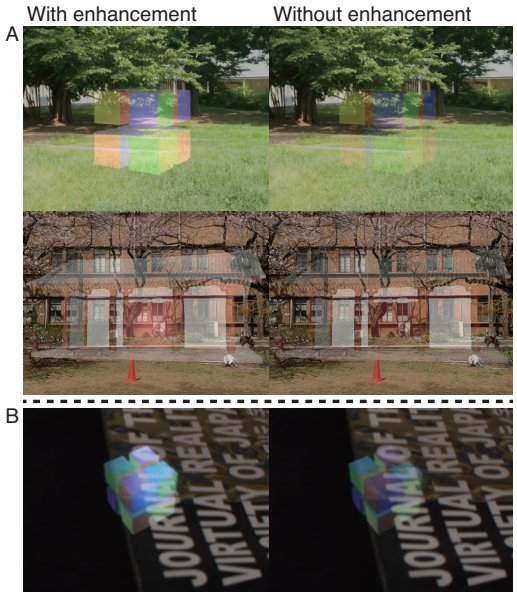

Figure 12. Examples of the visibility-enhanced blending for optical see-through systems. (A) The experimental results obtained by simulation assuming ideal calibrations. (B) The experimental results obtained using an actual optical see-through device.

2.6 Section Conclusion

In this section, we proposed two blending methods based on the visibility model. One is the visibility-based blending, which locally optimizes a blending parameter α such that the visibility

of the blended object achieves an arbitrarily targeted level. The other is the visibility enhanced-blending for optical see-through systems, in which visibility of a virtual object is adaptively and locally enhanced to an arbitrary targeted level. In the experiment, we demonstrated that the visibility model can linearly predict the visibility of a blended object on various natural texture images. Then, we showed that the proposed blending methods are effective to blend images with constant and uniform visibility. Since the proposed blending methods work at a sufficiently fast frame rate, they will not violate interactivity even in combination with other computations indispensable for constructing AR/MR scenes (e.g. tracking).

3. The model of perceived depth order of bistable-transparency pattern

The visibility-based blending proposed in the previous section enables us to blend a virtual object with a constant visibility across different regions or different scenes. However, showing the object semi-transparently does not necessarily make it appear to be behind the foreground region in the real scene (see Figure 13). We think that the characteristics of human transparency perception are a key to resolve this issue.

Figure 13. A failure case of the simple semi-transparent rendering. The virtual object does not appear to be behind the foreground tree.

3.1 Phenomenal classification of perceptual transparency

The human visual system decomposes a 2D retinal image in the same location into two surfaces at different depths, even when a very simple pattern is presented. One of the major issues in this "perceptual transparency" is what photometric condition is important for the depth stratification. Regarding this problem, [1] and [3] proposed that the luminance pattern around an X-junction (a junction where 4 regions meet together) plays the main role in perceptual transparency, and argued that the perceived state of the surface decomposition depends on categories of the X-junction (the contrast polarity rule). They classified X-junctions into three categories according to polarity relationships of aligned contours. Those junctions are termed non-reversing junction, single-reversing junction, and double-reversing junction.

For example, the X-junction in Figure 14A is classified as a single-reversing junction since contrast polarity along vertical contours is reversed while contrast polarity along horizontal contours is preserved (see the magnified X-junction in the figure). In this case, the surface composed by the regions *p* and *q* (the bottom-left square) is always perceived as transparent and being in front according to their theory. This special case induced by the single-reversing junction was thus termed *unique transparency*. On the other hand, the X-junction in Figure 14B is classified as a non-reversing junction since contrast polarity along both horizontal and vertical contours is preserved. In this case, which surface is perceived as being in front remains ambiguous; sometimes the bottom-left square may appear to be transparent and in front, but sometimes the top-right square may appear to be transparent and in front. Thus, the perceptual transparency under this condition was termed *bistable transparency*. Finally, the X-junction in Figure 14C is classified as a double-reversing junction since contrast polarity along both horizontal and vertical contours is reversed. For the double-reversing junction, one barely experiences transparency perception.

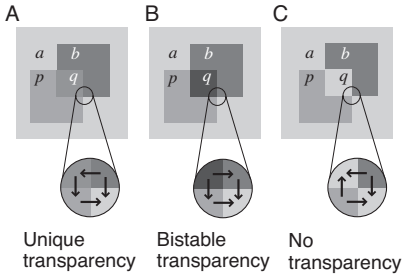

Figure 14. Schematic explanation of the contrast polarity rule.

When blending a virtual object using the conventional blending equation (e.g. alpha blending), the resulting pattern often creates unique transparency such that the virtual object is always perceived as being in front (see X-junctions in Figure 13). Thus, it would be ideal if we could find a new blending method that produces unique transparency such that a virtual object always appears to be 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 paper, 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, the 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 [4], [10], [22], [23], [32]. 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, in this section we examine the situation and make a model of perceived depth ordering.

3.2 Related works

Several researchers have already investigated how our

perception of transparency varies with luminance patterns around x-junctions by using stimuli inducing bistable transparency. For example, some previous studies [4], [22], [32] suggested the following tendency in perceived depth ordering: the more similar the lightness of abutting regions is to each other, the more often the surface composed by those abutting regions appears to be in front as a transparent filter. However, Delogu et al. [10] argued that a mathematical model incorporating this tendency alone could not explain their data.

In this study, we show that a model formulating the above-mentioned tendency in a more perceptually realistic way can explain the perceived depth ordering of various bistable-transparency patterns very well.

3.3 Proposed model of perceived depth-order

Given a bistable-transparency pattern like Figure 15, the proposed model predicts the likelihood of occurrence of the perception that the left disk is in front of the right (left-in-front) as proportional to the following value ρ:

$$\rho = \frac{|b-q|-|p-q|}{|b-q|+|p-q|} \tag{10}$$

where b, p, and q denotes lightness of each corresponding region in the pattern. The model basically indicates that the smaller the contrast between the left region (p) and the shared region (q) relative to the contrast between the right region (b) and the shared region (q), the larger the likelihood of "left-in-front" perception. Thus, the model is following the tendency suggested in [4], [22], [32]. The division by the sum of those contrast values (denominator) simulates the non-linear nature of the visual system (i.e., the visual system tends to overestimate the difference when the absolute levels of the contrasts are small, and underestimate the difference when they are large), which is not considered in Delogu et al.'s model [10].

We tested this model using a large number of bistable-transparency patterns in a psychophysical experiment.

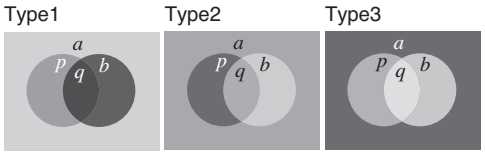

Figure 15. Three types of bistable transparency patterns. Bistable transparency patterns can be classified into three categories based on the contrast polarity along the edges of each region.

3.4 Psychophysical Experiment

3.4.1. Methods

- *Stimuli.* A stimulus was composed of two disks of the same size (diameter was 5.1 deg). In the presentation we made the stimulus move horizontally in a symmetrical fashion because a previous study suggested that motion can reduce inconsistency in depth-order perception without distorting the mean response (Experiment 1 in [10]). For each presentation, the two disks first appeared at both sides of the screen. Immediately after the onset of the disks, the disks started moving horizontally toward the center of the screen. The movement of the disks was sinusoidally modulated and the disks reversed their motion direction when their center locations reached 0.63 deg away from the screen center. The disks disappeared when they returned to their initial onset locations. Thus, when the two disks were overlapping, the whole image of the stimuli had four different regions: background region (a), right-disk region (b), left-disk region (p), and shared region (q). When classified based on the contrast polarity along the edges of each region, each stimulus can be classified into 3 different types (Type1 to 3, see Figure 15). Of all the 562 stimuli, 180 stimuli belonged to Type 1, 208 stimuli belonged to Type 2, and 174 stimuli belonged to Type 3 of the bistable transparency pattern. We generated the stimulus patterns so that each luminance of the regions a, b, p, and q was independently and uniformly modulated in lightness domain.
- *Procedure.* In each trial the stimulus was presented for 1.3 seconds. After that a blank with a fixation point followed, during which time the observer performed a task of judging whether the left disk appeared behind or in front of the right disk by button press. However, the possible perceptual patterns were not restricted to those two alternatives. For example, the whole surface with a hole shaping a disk might be perceived as being in front. In this case and other such cases, the observer was told to cancel that trial by pressing the third button. The next trial started immediately after the observer pressed a key. In one session, 281 stimuli were randomly chosen from all of the 562 stimuli, and tested in a random order. The remaining 281 stimuli were tested in the next session. Twelve observers, who were unaware of the purpose of the experiment, (aged 22–42) completed 6 sessions. Therefore, 36 responses were collected for each stimulus.
- *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/m2). Each subject placed his/her head on a chin-rest and used both eyes to view the stimuli. The viewing distance was 86 cm.

3.4.2. Results

We calculated the probability that the left disk was perceived as being in front of the right disk (%"left in front") for each stimulus from the responses in the trials which the subjects did not cancel. The percentage of canceled trials in all the responses for each stimulus was 2.8% on average, and 22.2% at most. Because enough responses for calculating %"left in front" were obtained for every stimulus, we used the data from all the stimuli in the following analysis.

We examined how much the model defined in Equation 10 can explain the data of %"left in front." Here, we expected that lightness difference, not luminance difference, should be used as the contrast between the two abutting regions since lightness is the perceptually uniform scale. Here, we used the following equation to translate luminance into lightness:

$$l' = l^n \tag{11}$$

where l represents normalized luminance level (luminance divided by the maximum luminance 89.2 cd/m^2), and l' represents lightness value. We left the translation exponent be a free parameter and estimated the best one because what exponent best predicts the depth-order perception should also be an empirical matter. In order to estimate the best exponent as well as to establish a quantitative measure of goodness of prediction of the model, we fitted a sigmoid function (Equation 12) to the data in logistic regression.

$$y = \frac{100}{1+\exp\left(-\frac{x-m}{s}\right)} \tag{12}$$

The thick gray curve in Figure 16 shows the best-fit sigmoid function for the entire data set including all of the stimulus types. The best-fit parameters of the sigmoid function and R^2 are shown in Table 2. The result showed that the proposed model can explain the data very well. The exponent obtained by the fitting analysis was 0.46. This is very close to the square-root exponent that was used to explain perceived lightness in some previous studies [41], [42]. Using the best-fit exponent n=0.46, we also fitted different sigmoid functions to different stimulus type data separately. The best-fit parameters of these sigmoid functions and R^2s are also shown in Table 2. Those best-fit curves were very similar to each other, indicating that the visual system is indifferent to the stimulus types defined in Figure 15.

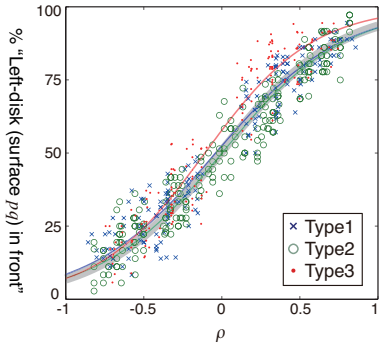

Figure 16. % "left in front" plotted as a function of the relative contrast ratio ρ (Equation 10). The thick gray curve shows the best-fit sigmoid function for the entire data set. The red, green, and blue curves show the best-fit sigmoid functions for the data of Type 1, Type 2, and Type 3 stimuli, respectively.

Stimulus type	Best-fit parameters	R^2
All	n=0.46, m=-0.01, s=0.38	0.88
Type1	m=-0.04, s=0.40	0.91
Type2	m=0.00, s=0.39	0.90
Type3	m=-0.11, s=0.35	0.89

Table2. The results of the fitting analysis.

3.5 Section Conclusion

To render a virtual object so that the virtual object appears to be behind a foreground region, we decided to utilize the bistable-transparency perception. In this section, we conducted a psychophysical experiment and estimated a model that can predict perceived depth order of bistable transparency patterns. The model indicated that the more similar the contrast between two abutting regions is to each other, the more often the surface composed by those abutting regions appears to be in front. The likelihood that the surface is perceived as in front was closely related to the perceived size of difference between the contrast within one surface and the contrast within the other surface.

4. Semi-transparent visualization for occlusion handing

In the previous section, we estimated a model that can predict the perceived depth ordering of bistable-transparent layers. Based on this model, in this section we first propose "bistable-transparency blending," which blends a virtual object on a real scene image such that the virtual object can appear to be behind the foreground region in the real scene. Then, we implement the semi-transparent visualization method, combining the bistable-transparency blending and the visibility-based blending proposed in the section 2.

4.1 Bistable-Transparency Blending

We designed the algorithm of the bistable-transparency blending based on the model we estimated in the previous section. The simple bistable-transparency pattern used in the psychophysical experiment could correspond to a MR scene as shown in Figure 17.

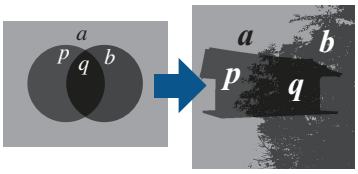

Figure 17. Topological correspondence between the abstract stimuli used in the experiment and a MR scene.

To make a bistable transparency pattern, regions with which a virtual object is blended must consistently become darker, or inversely, consistently become brighter after the blending. As such blending equations, here we used multiplicative blending, which is as follows:

$$I_M\left(I_r, I_v\right) = I_r I_v \tag{13}$$

and inversed-multiplicative blending, which is as follows:

$$I_I\left(I_r, I_v\right) = 1 - \left(1 - I_r\right)\left(1 - I_v\right) \tag{14}$$

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. Multiplicative blending applied to a real scene where the intensity of a foreground object is lower than that of the background leads to the type-1 pattern in Figure 15. Multiplicative blending applied to a real scene where the intensity of a foreground object is higher than that of the background leads to the type-2 pattern in Figure 15. 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 the type-2 pattern in Figure 15. 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 the type-3 pattern in Figure 15.

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

$$I_M = \lambda I_r I_v + (1-\lambda) I_r \tag{15}$$

for multiplicative blending, and

$$I_I = \lambda\{1-(1-I_r)(1-I_v)\} + (1-\lambda) I_r \tag{16}$$

for inversed-multiplicative blending. Because our model predicts that the likelihood of "surface-*pq* behind" increases as the contrast |*p*-*q*| increases against the contrast |*b*-*q*|, we can monotonically increase the likelihood of "virtual behind" by decreasing λ. However, we want the transparency of the virtual object to be as low as possible at the same time. Therefore, we choose the largest λ among those that can make %"virtual behind" larger than 50%.

The calculation of λ is conducted as follows. In the case of the multiplicative blending, given the lightness of the virtual object I_v, the lightness of the background region of the real scene I_b, and the lightness of the foreground region of the real scene I_f, the lightness of the regions (b, p, q) can be described as:

$$\begin{cases} b = I_f \\ p = \lambda I_v I_b + (1-\lambda) I_b \\ q = \lambda I_v I_f + (1-\lambda) I_f \end{cases} \tag{17}$$

In the case of the inversed-multiplicative blending, (b, p, q) can be described as:

$$\begin{cases} b = I_f \\ p = \lambda\{1-(1-I_v)(1-I_b)+(1-\lambda)I_b\} \\ q = \lambda\{1-(1-I_v)(1-I_f)+(1-\lambda)I_f\} \end{cases} \tag{18}$$

By substituting those values into |*p*-*q*| ≥ |*b*-*q*| and solving the inequality for λ, we can find the largest λ that render the virtual object so that %"virtual behind" is not less than 50%. In the case of the multiplicative blending, such λ is specified as follows:

$$\begin{cases} \lambda = \min\left(\dfrac{I_b - I_f}{I_b(1-I_v)}, 1\right) & (\ if \quad I_f < I_b\ ) \\ \lambda = \min\left(\dfrac{I_f - I_b}{(2I_f - I_b)(1-I_v)}, 1\right) & (\ if \quad I_f > I_b\ ) \end{cases} \tag{19}$$

In the case of the inversed-multiplicative blending:

$$\begin{cases} \lambda = \min\left(\dfrac{I_b - I_f}{(1+I_b-2I_f)I_v}, 1\right) & (\ if \quad I_f < I_b\ ) \\ \lambda = \min\left(\dfrac{I_f - I_b}{(1-I_b)I_v}, 1\right) & (\ if \quad I_f > I_b\ ) \end{cases} \tag{20}$$

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. In this paper, we defined the visibility as |*a*-*p*|, a contrast between the virtual object and the background region. In Figure 18, we show the values |*a*-*p*| as a function of various combinations of I_b and I_f. Here, I_v was set constant to 0.5, but the qualitative patterns of the results did not change depending on I_v.

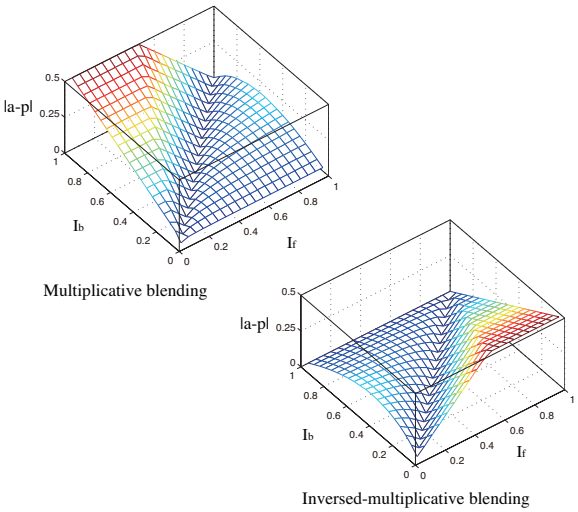

Figure 18. Visibility of the virtual object as a function of (I_f, I_b).

As shown in Figure 18, 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 used multiplicative blending when the intensity of the foreground region is lower than that of the background, and used inversed-multiplicative blending when the intensity of the foreground region is higher than that of the background.

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

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

2. Selection of blending equation If $I_f > I_b$, multiplicative blending (Equation 15) is selected. If $I_f < I_b$, inversed-multiplicative blending (Equation 16) is selected.

3. Determining the blending parameter λ When the multiplicative blending is selected, λ is determined by Equation 19. When the inversed-multiplicative blending is selected, λ is determined by Equation 20.

4. Output Using the blending equation selected in 2 and the blending parameter λ obtained in 3, the final blending between the virtual object and the real-scene image is conducted.

4.2 Implementation and Experiment

Based on the bistable-transparency blending and the visibility-based blending, 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.

4.2.1. Implementation

Basically, the bistable-transparency blending 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 λ 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:

$$I_v = \frac{1}{\sum_{(p,q)\in W} A_v(p,q)} \sum_{(p,q)\in W} P_v(p,q) A_v(p,q) \tag{21}$$

$$I_b = \frac{1}{\sum_{(p,q)\in W} \{1 - P_m(p,q)\}} \sum_{(p,q)\in W} P_r(p,q)\{1 - P_m(p,q)\} \tag{22}$$

$$I_f = \frac{1}{\sum_{(p,q)\in W} P_m(p,q)} \sum_{(p,q)\in W} P_r(p,q) P_m(p,q) \tag{23}$$

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). Using those values as inputs for the bistable-transparency blending, the blending result at the current pixel (x,y) is obtained as:

$$P_{blend} = \begin{cases} \alpha P_v(x,y) P_r(x,y) + (1-\alpha) P_r(x,y), & \text{if} \quad I_f \le I_b \\ \alpha\left[1 - \{1 - P_v(x,y)\}\{1 - P_r(x,y)\}\right] + (1-\alpha) P_r(x,y), & \text{if} \quad I_f > I_b \end{cases} \tag{24}$$

When most of the pixels in a window of the current pixel are within the foreground region, we blend the virtual object by the visibility-based blending proposed in section 2. One might think that we should not render the virtual object at all when it is completely occluded by the foreground region. In this work, however, we assume situations in which accurate and reliable foreground information is not available; it is possible that background scene is seen through gaps between leaves or branches even if the system judge that the area is covered by the foreground object. If we do not render the virtual object on such areas of the scene, the user might perceive the MR scene as if a part of the virtual object disappears around the foreground object. To keep showing a consistent MR scene, therefore, the virtual object should be always presented at least semi-transparently at a certain visible level.

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. To make a blending result smooth between these pixels and the other pixels, we introduce the next equation:

$$P_{output}(x,y) = (1 - \psi_{in} - \psi_{out}) P_{blend}(x,y) + \psi_{in} P_{vis}(x,y) + \psi_{out} P_v(x,y) \tag{25}$$

where $P_{output}(x,y)$ is the final output of our blending algorithm at the current pixel (x,y). $P_{vis}(x,y)$ is the result of the visibility-based blending at the current pixel (x,y). ψ_{in} and ψ_{out} are weight functions that switches different blending results smoothly. ψ_{in} and ψ_{out} are defined as:

$$\psi_{in} = t(r;m,s) \tag{26}$$

$$\psi_{out} = t(1-r;m,s) \tag{27}$$

where

$$t(x;m,s) = \begin{cases} 0 & \text{if} \quad x \le m - s \\ \frac{1}{2} + \frac{1}{2}\sin\frac{\pi(x-m)}{2s} & \text{if} \quad m - s < x < m + s \\ 1 & \text{if} \quad m + s \le x \end{cases} \tag{28}$$

$$r = \frac{\sum_{(u,v)\in W} P_m(u,v)}{N} \tag{29}$$

In Equation 29, N denotes the number of pixels in the window. m and s are empirically given.

4.2.2. Experiment

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 65x65. m and s (in Equations 26 and 27) were both set to 0.1. The target visibility of the visibility-based blending was set to 0.8. 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 4th pixel within a window when calculating I_f, I_b, and I_v. The actual frame rate depended on the number of pixels around and within a foreground region, but it worked at a frame rate higher than 60 FPS on most of the cases.

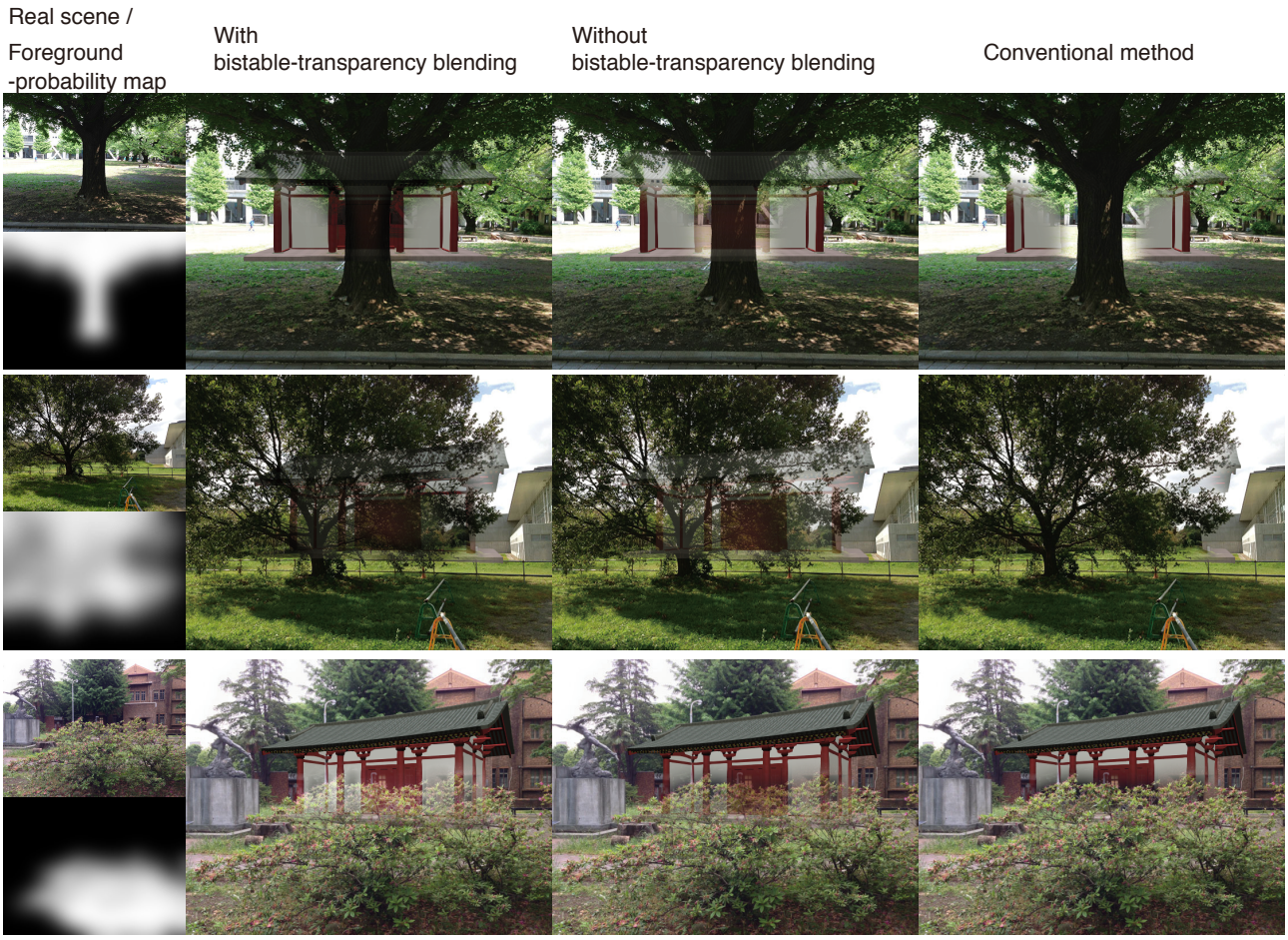

Figure 19. Comparison between the proposed semi-transparent visualization method (images in the second column) and others (images in the third and the fourth column).

The blending results are shown in the second column in Figure 19. Their neighbors to the right are images for comparison and were obtained without bistable-transparency blending. We made those results by just substituting the result of the visibility-based blending instead of that of the bistable-transparency blending in Equation 25. We also generated results in which the semi-transparent visualization is not used at all (the rightmost column). 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. On the other hand, the comparison results obtained without bistable-transparency blending (the third column) sometimes showed the impression of contradiction (i.e., the virtual object appeared to be in front of the foreground object). In the rightmost column, the virtual object appears to unnaturally fade away around the foreground region. Thus, when correctly segmented foreground information is not available, the proposed semi-transparent visualization works most robustly.

4.3 Section Conclusion

Based on the model of perceived depth order of bistable transparency, we made the bistable-transparency blending that can effectively reduce the contradictory occlusion information in MR scenes. The proposed method blends a virtual object such that the virtual object is perceived as behind a foreground region in the real scene given only an obscured foreground probability map. The experimental results showed that our 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.

5. Conclusion

This paper proposed the semi-transparent visualization method for occlusion handling in MR scenes. In the section 2, we first developed the visibility-based blending, which render a virtual object semi-transparently with a constant and a uniform visibility on any arbitrary scenes. In the section 3, we conducted a psychophysical experiment and modeled the perceived depth ordering of partially overlapping semi-transparent surfaces. Based on the model, in the section 4 we developed the bistable-transparency blending, which blends a virtual object such that the virtual object appears to be behind a foreground region in a real scene.

Combining the visibility-based blending with the bistable-transparency blending, we implemented the semi-transparent visualization method, which renders a virtual

object semi-transparently with a constant visibility while modulating its lightness such that it appears to be behind foreground regions. We showed that the proposed method works robustly in cases where accurate foreground segmentation is not available. 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 more arbitrary scenes in real time.

Reference

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

[5] A. P. Bradley. A wavelet visible difference predictor. *IEEE Transaction on Image Processing*, 5: 717–730, 1999.

[6] F. W. Campbell and J. G. Robson. Application of fourier analysis to the visibility of gratings. *Journal of Physiology*, 197: 551-566, 1968.

[7] C. R. Carlson, R. W. Cohen, and I. Gorog. Visual processing of simple two-dimensional sine-wave luminance gratings. *Vision Research*, 17:351-358, 1977.

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

[9] S. J. Daly. Visible differences predictor: an algorithm for the assessment of image fidelity. In *Proceedings of SPIE* 1666, pages 2-15, 1992.

[10] 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.

[11] J. M. Foley and G. M. Boynton. A new model of human luminance pattern vision mechanisms: analysis of the effects of pattern orientation, spatial phase, and temporal frequency. In *Proceedings of SPIE* 2054, pages 32-42, 1994.

[12] T. Fukiage, T. Oishi, and K. Ikeuchi. Reduction of contradictory partial occlusion in Mixed Reality by using characteristics of transparency perception. In *ISMAR*, pages 129-139, 2012.

[13] J. L. Gabbard, J E. Swan, D. Hix, S. Jung Kim, and G. Fitch. Active text drawing styles for outdoor augmented reality: A user-based study and design implications. In *IEEE Virtual Reality*, pages 35-42, 2007.

[14] 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.

[15] D. J. Heeger. Normalization of cell responses in cat striate cortex. *Visual Neuroscience*, 9(2): 181-192, 1992.

[16] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160:106–154, 1962.

[17] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11): 1254-1259, 1998.

[18] 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.

[19] D. Kalkofen, E. Veas, S. Zollmann, M. Steinberger, and D. Schmalstieg, Adaptive Ghosted Views for Augmented Reality. In *ISMAR*, pages 1-9, 2013.

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

[21] 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.

[22] 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.

[23] J. Koenderink, A. J. van Doorn, S. C. Pont, and W. Richards. Gestalt and phenomenal transparency. *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, 25, 190–202, 2008.

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

[25] V. Laparra, J. Muñoz-Marí, and J. Malo. Divisive normalization image quality metric revisited. *Jounrnal of Optical Society of America A*, 27(4): 852-64, 2010.

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

[27] G. E. Legge and J. M. Foley. Contrast masking in human vision. *Jounrnal of Optical Society of America*, 70(12): 1458-1471, 1980.

[28] J. Lubin. A human vision system model for objective picture quality measurements. In *International Broadcasting Convention*, pages 498-503, 1997.

[29] J. Malo and V. Laparra. Psychophysically tuned divisive normalization approximately factorizes the PDF of natural images. *Neural computation*, 22(12): 3179-3206, 2010.

[30] K. T. Mullen. The contrast sensitivity of human color vision to red-green and blue-yellow chromatic gratings. *The Journal of Physiology*, 359: 381–400, 1985.

[31] A. Olmos and F. A. A. Kingdom. Mcgill calibrated colour image database. http://tabby.vision.mcgill.ca., 2004.

[32] T. Oyama and J. Nakahara. The effects of lightness, hue and area upon the apparent transparency. *Japanese Journal of Psychology*, 31, 35–48, 1960.

[33] R. Picard, C. Graczyk, S. Mann, J. Wachman, L. Picard, and L. Campbell. The MIT Vision Textures database. http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html, 1995.

[34] T. Porter and T. Duff. Compositing Digital Images. *Computer Graphics*, 18(3): 253-259, 1984.

[35] 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.

[36] E. Simoncelli and E. Adelson. Subband Image Coding. *Norwell, MA: Kluwer Academic Publishers*, pages 143–192, 1990.

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

[38] P. C. Teo and D. J. Heeger. Perceptual image distortion. In *Proceedings ICIP*, pages 982-986, 1994.

[39] T. Tsuda, H. Yamamoto, Y. Kameda, and Y. Ohta. Visualization methods for outdoor see-through vision. In *ICAT*, pages 62-69, 2005.

[40] 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.

[41] R. M. Warren and E. C. Poulton. Basis for lightness judgments of grays. *American Journal of Physiology*, 73, 380–387, 1960.

[42] R. M. Warren and E. C. Poulton. Lightness of grays: Effects of background reflectance. Perception and Psychophysics, 1, 145–148, 1966.

[43] A. B. Watson and J. A. Solomon. Model of visual contrast gain control and pattern masking. *Jounrnal of Optical Society of America A*, 14(9): 2379-2391, 1997.