Foreground and Shadow Occlusion Handling for Outdoor Augmented Reality

Boun Vinh Lu* The University of Tokyo
Tetsuya Kakuta† The University of Tokyo
Rei Kawakami‡ The University of Tokyo
Takeshi Oishi§ The University of Tokyo
Katsushi Ikeuchi¶ The University of Tokyo

Figure 1: Foreground and shadow occlusions are handled correctly with our proposed solution. The two images on the left shows the original frame in campus sequences and its corresponding augmented result. The two ones on the right shows the original frame in Asuka sequences and its corresponding result.

ABSTRACT

Occlusion handling in augmented reality (AR) applications is challenging in synthesizing virtual objects correctly into the real scene with respect to existing foregrounds and shadows. Furthermore, outdoor environment makes the task more difficult due to the unpredictable illumination changes. This paper proposes novel outdoor illumination constraints for resolving the foreground occlusion problem in outdoor environment. The constraints can be also integrated into a probabilistic model of multiple cues for a better segmentation of the foreground. In addition, we introduce an effective method to resolve the shadow occlusion problem by using shadow detection and recasting with a spherical vision camera. We have applied the system in our digital cultural heritage project named Virtual Asuka (VA) and verified the effectiveness of the system.

Index Terms: I.3.7 [Computer graphics]: Three-dimensional Graphics and Realism—virtual Reality; I.4.6 [Image Processing and Computer Vision]: Segmentation—Pixel Classification

1 INTRODUCTION

The occlusion problem in AR challenges researchers with some issues. The first problem is foreground occlusion in which 3D models should be rendered correctly behind the foreground if necessary. Another one is shadow casting with respect to the coherence of the location and the illumination condition. In an outdoor AR system like the digital museum, there is necessity to handle the mentioned problems robustly in an outdoor environment. These two issues are the main concerns of our work and are addressed in the remaining part of the paper. Regarding the former problem, our goal is to handle the foreground occlusion in an outdoor environment with complicated illumination conditions. In the AR literature, there are some noticeable works which partially addressed the problem. Pilet *et al.* proposed a method to re-texture the deformable surface with the presence of complex illumination and occlusion in [20] but outdoor illumination was not taken into account. While Kanbara [9] and Kim [11] applied stereo vision for depth information for which multiple cameras are required, Ladikos *et al.* used a system of 16 ceiling-mounted cameras to let users interact with virtual objects [12]. However, those works rely on accurate stereo vision which is time-consuming. In commercial applications, although the augmentation of lines or banners into live videos has been used in the television broadcasting industry, the augmented contexts are simple. Furthermore, the illumination variation within the environment is not correctly handled.

In order to reach that goal, our approach is to segment the foregrounds and to estimate their depths. Foreground segmentation has attracted much attention in the literature of computer vision ranging from manual processing to totally automatic systems. Among these methods, image matting is widely known to give highly accurate segmentation results [24]. Nevertheless, a fully automatic matting approach for outdoor scene with complex illumination condition is still a challenging problem. Meanwhile, among graph-cut based methods, Criminisi *et al.* probabilistically combined multiple models including temporal continuity, spatial continuity, color likelihood, and motion likelihood to segment foreground from monocular video sequences in real time [3]. Sun *et al.* proposed another real-time foreground segmentation with only color and contrast cues [23]. Kakuta *et al.* combined the methods proposed in [3], [23] and [10] to solve the foreground occlusion in AR [8]. These methods are applied in indoor applications with certain robustness. The segmentation results, on the other hand, are rather sensitive to the illumination condition especially in an outdoor environment.

Concerning the illumination change in segmentation, while a

*e-mail:lbvinh@cvl.iis.u-tokyo.ac.jp
†e-mail:kakuta@cvl.iis.u-tokyo.ac.jp
‡e-mail:rei@cvl.iis.u-tokyo.ac.jp
§e-mail:oishi@cvl.iis.u-tokyo.ac.jp
¶e-mail:ki@cvl.iis.u-tokyo.ac.jp

IEEE International Symposium on Mixed and Augmented Reality 2010
Science and Technolgy Proceedings
13 -16 October, Seoul, Korea
978-1-4244-9345-6/10/$26.00 ©2010 IEEE

gradual change can be solved by an adaptive learning of a background model, a sudden change is a challenging problem and remains an active research area. Rosin proposed a well-known threshold-based method for change detection [22]. Li *et al.* improved the detection by combining the intensity with texture cues [15]. In [18], O'Callaghan *et al.* successfully detected foreground from indoor background changing in illumination by using normalized gradient-correlation between two successive frames. Pilet *et al.* also showed impressive results regarding the abrupt change of background illumination by using a statistical model on illumination ratio [21]. However, all above methods lack the sound consideration of outdoor illumination, change of which is in a rather different manner. Moreover, the statistical model of illumination in [21] is inaccurate since we cannot assume that the ratio of two Poisson or Gaussian distributions has a Gaussian distribution.

Contrary to the above approaches, we introduce a set of illumination constraints to handle both the sudden illumination change and the gradual one. Furthermore, we extend the multiple cue-based segmentation in [8] by adding the illumination and the motion cue with background attenuation. Due to the inaccuracy of optical flow regarding the aperture problem, our heuristic background attenuation proves its effectiveness in estimating motion areas of the scene. Then, a spherical vision camera is applied to estimate the depth of the segmented foregrounds as proposed in [8]. Therefore, the foreground occlusion problem is resolved by setting those far foregrounds behind the virtual objects.

In addition, an effective method to solve shadow occlusion in AR is also introduced in this paper. First, with the camera sensitivity, shadow region is detected by using the illumination invariant constraint, and refined by using the energy minimization method. The detected shadow area is then used to estimate light direction and to recast shadow onto the virtual object by using the spherical vision camera. Finally, real-time shading and shadow in [7] is also applied to render virtual objects without using any additional camera for scene illumination.

In this paper, two significant contributions to outdoor AR are introduced. First, we propose novel outdoor illumination constraints which make the foreground occlusion handling more robust with respect to the abrupt illumination change. Secondly, a simple yet effective solution to shadow occlusion is also presented. These two key contributions provide us the solution to both foreground and shadow occlusion issues in outdoor AR, which we believe will be promising in the literature.

The paper is divided into eight sections. After presenting an overview of the system in Sect. 2, we introduce the foreground occlusion handling in Sect. 3. Then, Sect. 4 provides us insight into the illumination constraints and how they improve the foreground segmentation by handling sudden change in illumination. Sect. 5 explains how shadow is detected and recast with a spherical vision camera. Subsequently, an overall system is implemented and experimented in Sect. 6. Experimental results in Sect. 7 prove that the proposed methods work effectively and practically with high resolution outdoor video sequences. The last section completes this paper with summary and discusses ideas for future improvement and further research.

2 Overview

In our occlusion handling system as illustrated in Fig. 2, the input from the spherical vision camera is first processed by foreground segmentation to obtain a foreground image. However, the obtained foregrounds include shadows which need to be separated in order to solve the shadow occlusion. Thus, shadows are then detected as shadow regions in the second stage. Depth and height of foregrounds are also estimated to correct depth rendering and to calculate the direction of the light source which causes the shadow. With known shadow and directional light, shadow casters can be obtained and used to cast the shadows onto virtual objects with a traditional shadow mapping method.

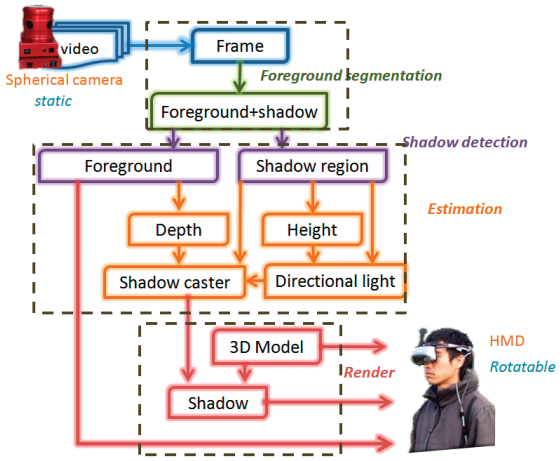

Figure 2: An overview of the whole system.

3 Foreground Occlusion Handling

In our system, foreground occlusion is handled by using online foreground segmentation and foreground depth estimation. In order to make this section clear and easy to understand, we will explain the depth estimation later in section 5 together with height estimation by using a spherical vision camera. In Sect. 3.1, online foreground segmentation will be introduced first to explain the basic framework of the probabilistic fusion of multiple cues, which is inherited from the work in [8]. We then present a new motion cue with background attenuation in Sect. 3.2 and finally identify the main problem caused by outdoor illumination in Sect. 3.3.

3.1 Online foreground segmentation

Regarding the foreground segmentation process in occlusion handling, besides illumination changes, the outdoor scene challenges the task due to the occupation of moving foreground and moving background such as trees, leaves and clouds. Therefore, it is wise to combine different available cues together to segment the foreground online because a single cue is not reliable. We extend the work of Kakuta *et al.* [8] by adding the illumination cue and the motion cue with background attenuation which will be explained in the following sub-sections.

Let I^{B^t} and $I^t = (I_{1^t}, .., I_{n^t})$ denote the estimated background image at time t, and the image at time t where I_{i^t} indicates the i^{th} pixel in the image, respectively. Our desired binary output is denoted by $X^t = (x_{1^t}, .., x_{n^t})$ where $x_{i^t} \in \{F(foreground), B(background)\}$. As in [8], we apply the most widely used energy function which is in the form of data and smoothness term as follows:

$$E(X^t, X^{t-1}, X^{t-2}, I^{t-1}, I^t) = E_{data}(X^t, X^{t-1}, X^{t-2}, I^{t-1}, I^t) + \lambda E_{smooth}(X^t, I^t) \quad (1)$$

where λ is a smoothness factor.

In our fusion-based model, the data term E_{data} in Eq. (1) can be used to integrate different cues together to give the final likelihood

of the foreground to segment.

$$E_{data}(X^t,X^{t-1},X^{t-2},I^{t-1},I^t) = \alpha E_{color}(X^t,I^t) + \beta E_{temp}(X^t,X^{t-1},X^{t-2}) + \gamma E_{motion}(X^t,I^t,I^{t-1},I^{B^t}) + \theta E_{illum}(X^t,I^t,I^{B^t}) \quad (2)$$

where E_{color}, E_{temp}, E_{motion} and E_{illum} are color term using the background model, temporal prior term, motion term and illumination term, respectively. α, β, γ and θ are corresponding mixing factors.

- Color term $E_{color}(X^t,I^t)$: The color likelihood is computed from the online Mixture of Gaussian (oMoG) model which is learned from the input I^t online as proposed in [14]. The energy for this term here is defined as the negative log of the foreground likelihood.

$$E_{color}(X^t = F,I^t) = \sum_{i_r \in I} -\log(1 - p_{oMoG}(i_r^t|B)) \quad (3)$$

where

$$p_{oMoG}(i_r^t|B) = \sum_{i=1}^{n^r} w_i^r G(x_r^t,\mu_i^r,\sigma_i^r) \quad (4)$$

where w_r^t, μ_i^r, σ_i^r are weight, mean and variance of the i^{th} Gaussian distribution, respectively.

- Temporal prior $E_{temp}(X^t,X^{t-1},X^{t-2})$: The temporal likelihood is computed from the temporal prior transition table which represents the estimation of the coming result based on the previous ones as proposed in [3]. The energy for this term is defined as the negative log of the temporal prior likelihood.

$$E_{temp}(X^t = F,X^{t-1},X^{t-2}) = \sum_{x_r \in X} -\log(p_{temp}(x_r^t = F|x_r^{t-1},x_r^{t-2})) \quad (5)$$

- Motion term $E_{motion}(X^t,I^t,I^{t-1},I^{B^t})$: The motion likelihood is computed from the motion cue with background attenuation in the next sub-section 3.2. The energy for this term is also defined as the negative log of the motion likelihood.

$$E_{motion}(X^t = F,I^t,I^{t-1},I^{B^t}) = \sum_{i_r \in I} -\log(p_{motion}(i_r^t,i_r^{t-1},I^{B^t})) \quad (6)$$

- Illumination term $E_{illum}(X^t,I^t,I^{B^t})$: The illumination likelihood indicates how similar the current frame I^t is likely to be with the learned background, although there exists illumination change. To estimate this likelihood, the background image I_B^t from the online background model is used in Eq. (16). The energy for this term is also defined as negative log of the illumination distance.

$$E_{illum}(X^t = F,I^t) = \sum_{\substack{i_r^t \in I^t \\ i_r^{B^t} \in I^{B^t}}} -\log(d_{inv}(i_r^t,i_r^{B^t})) \quad (7)$$

Meanwhile, the second term in Eq. (1) called smoothness indicates the tendency that the same label is assigned to the neighboring pixels in an image. In general, the labels are spatially continuous in the foreground area but different at the segmentation boundaries. We define the energy for this smoothness term with background attenuation proposed by Sun *et al.* in [23].

The labels $(\hat{X}^1,\ldots,\hat{X}^t)$ that minimize the energy $E(X^t,X^{t-1},X^{t-2},I^{t-1},I^t)$ shown in Eq. (1) are computed by estimating the current label $\hat{X}^t$ using the old labels $(\hat{X}^1,\ldots,\hat{X}^{t-1})$ that are already estimated.

$$\hat{X}^t = arg\min E(X^t,\hat{X}^{t-1},\hat{X}^{t-2},I^t) \quad (8)$$

It is widely known that the optimum label $\hat{X}^t$ in Eq. (8) can be estimated by using graph cut [2], .i.e. the solution to the energy minimization in a Markov Random Field (MRF) corresponds to the minimal cut of the corresponding graph, which is also equivalent to the max-flow of the graph according to the max-flow min-cut theorem.

There are some algorithms in the literature to find the maximum flow including linear programming, Ford-Fulkerson, Edmonds-Karp and push-relabel algorithms. However, most of the algorithms fall into one of the two groups known as augmenting paths and push-relabel. In this work, we apply the algorithm based on augmenting paths proposed in [1], which has been proved to be the fastest and the most efficient one.

3.2 Motion cue with background attenuation

The motion cue provides us some important hints about moving objects which are potential foregrounds. Among the methods for motion estimation, optical flow is widely used for two consecutive frames. Since our goal is to obtain a motion cue for foreground segmentation, not for accurate motion, dense optical flow is preferred among different approaches to determine motion flow. Thus, for the motion cue, we use the most basic model to estimate dense optical flow based on [5].

Since moving parts are more likely to be moving foregrounds, we use the length of estimated motion vectors as a motion cue, i.e. the larger motion it is, the more likely it belongs to the foreground. Thus, for each pixel x_r^t and x_r^{t-1} in frame I^t and I^{t-1} respectively, we have:

$$p_{motion}(x_r^t,x_r^{t-1}) = \frac{\|\vec{v}(x_r)\|}{M} \quad (9)$$

where M is a constant indicating the largest displacement allowed in dense optical flow. However, with such simple model, there are some well-known problems such as the occlusion and aperture problem. Rather than using the time-consuming but accurate flow estimation, we propose a simple strategy to attenuate the estimated background since the learned background I_B^t is available. We attenuate the background area in the flow by replacing Eq. (9) with Eq. (10) below, i.e. removing those pixels with the intensity which is close to that of the corresponding background pixel.

$$p_{motion}(x_r^t,x_r^{t-1}) = \frac{\|\vec{v}(x_r)\|}{M} \frac{d(I^t(x,y),I_B^t(x,y))}{\sigma_D} \quad (10)$$

where $d(I^t(x,y),I_B^t(x,y))$ and σ_D denote how the new pixel and the corresponding background pixel differ and the attenuation parameter respectively. Fig. 3 illustrates how the motion model works.

3.3 Problems with outdoor illumination change

Fig. 4 illustrates how the abrupt change of outdoor illumination causes problems in the traditional foreground segmentation with the color distribution and motion model. Although we can obtain better results for moving foreground by adjusting the mixing factor of the motion cue, it still suffers from the cases when the foreground temporarily stands still.

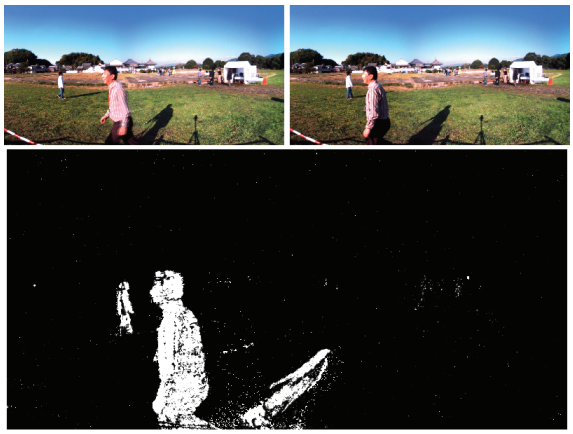

Figure 3: Motion with background attenuation from two successive frames. The whiter the pixel is, the higher probability that it belongs to the foreground.

Figure 4: The upper row shows how the illumination changes abruptly. The lower left is the corresponding probability from the Mixture of Gaussian model while its corresponding result from segmentation is in the lower right image.

4 Handling Outdoor Illumination Change in Foreground Occlusion

In an outdoor scene, changes in illumination are inevitable and challenging to most researchers in the computer vision field. Therefore, it is impossible to have a robust approach to any online segmentation task in an outdoor scene without significant understanding of its complex illumination condition. In this section, we propose an effective solution to handle sudden changes in illumination in most cases of day light condition ranging from a sunny one to a heavy cloudy one. The illumination constraints will be explained in Sect. 4.1 and 4.2. Both constraints can be used as an illumination cue which can be integrated into a multiple-cue background model in Sect. 3. In addition, we also introduce an illumination constraint-based algorithm to check two image irradiances regarding illumination change in Sect. 4.3.

4.1 Illumination invariant constraint

Let's assume that a surface patch S_t at time t is Lambertian with normal $\vec{n}$ and the corresponding *surface reflectance* ρ. Our proposed outdoor illumination model Eq. (35) with the visible portion of the sky in the appendix A reads

$$I_t = \rho E_t B \tag{11}$$

where

$$E_t = \left(g E_t^{sun}(\vec{n}\vec{D}_t) + E_t^{sky} \cos^2 \frac{\beta}{2} \right)$$

and I_t denotes the corresponding *image irradiance* and E for the *irradiance* at the surface patch.

If we assume that the camera sensitivity is sufficiently narrow and that daylight is blackbody radiation, we can apply Wiens approximation to Plancks formula as in [6], i.e.

$$E_t^{\lambda} \approx c_1 \lambda^{-5} e^{-\frac{c_2}{\lambda T_t^{\lambda}}} \tag{12}$$

where c_1 and c_2 are two constants and T_t is the color temperature with the center wavelength λ of the camera sensitivity. By applying Eq. (11) and Eq. (12) at two different times t_1 and t_2, we obtain the ratio as :

$$R_{t_1,t_2}^{\lambda} = \frac{I_{t_1}^{\lambda}}{I_{t_2}^{\lambda}} = \frac{E_{t_1}^{\lambda}}{E_{t_2}^{\lambda}} = e^{-\frac{c_2}{\lambda}\left(\frac{1}{T_1}-\frac{1}{T_2}\right)} \tag{13}$$

Finally, we can easily infer the constraint among the log-ratios $\ln R_{t_1,t_2}^{\lambda_R}$, $\ln R_{t_1,t_2}^{\lambda_G}$ and $\ln R_{t_1,t_2}^{\lambda_B}$ as :

$$\ln R_{t_1,t_2}^{\lambda_R} - C \ln R_{t_1,t_2}^{\lambda_G} + (C-1) \ln R_{t_1,t_2}^{\lambda_B} = 0 \tag{14}$$

where $C = \left(\frac{1}{\lambda_R} - \frac{1}{\lambda_B}\right) / \left(\frac{1}{\lambda_G} - \frac{1}{\lambda_B}\right)$. We call Eq. (14) the illumination invariant constraint. Furthermore, it is possible to modify the illumination constraint in Eq. (14) into a form of distant measure as :

$$d_c(I_1,I_2) = \| \ln R_{t_1,t_2}^{\lambda_R} - C \ln R_{t_1,t_2}^{\lambda_G} + (C-1) \ln R_{t_1,t_2}^{\lambda_B} \| \tag{15}$$

Since $d_c(I_1,I_2) \leq 2(C+1)D_c$ where $D_c = \ln \max(Intensity)$, Eq. (15) can be rewritten in normalized form as :

$$d_{inv}(I_1,I_2) = \frac{\| \ln R_{t_1,t_2}^{\lambda_R} - C \ln R_{t_1,t_2}^{\lambda_G} + (C-1) \ln R_{t_1,t_2}^{\lambda_B} \|}{2(C+1)D_c} \tag{16}$$

4.2 Illumination ratio constraint

First, by taking *logarithm* of Eq. (13), we obtain

$$\lambda \ln R_{t_1,t_2}^{\lambda} = -c_2 \left(\frac{1}{T_1} - \frac{1}{T_2} \right) \tag{17}$$

Since it is obvious that the right side of Eq. (17) is independent of the wavelength, another constraint which we call the illumination ratio constraint can be obtained as :

$$\lambda_R \ln R_{t_1,t_2}^{\lambda_R} = \lambda_G \ln R_{t_1,t_2}^{\lambda_G} = \lambda_B \ln R_{t_1,t_2}^{\lambda_B} \tag{18}$$

For the illumination ratio constraint in Eq. (18), we can also rewrite it in a similar distant measure as :

$$\begin{aligned} d_r(I_1,I_2) = \max \big(& \| \lambda_R \ln R_{t_1,t_2}^{\lambda_R} - \lambda_G \ln R_{t_1,t_2}^{\lambda_G} \|, \\ & \| \lambda_G \ln R_{t_1,t_2}^{\lambda_G} - \lambda_B \ln R_{t_1,t_2}^{\lambda_B} \|, \\ & \| \lambda_B \ln R_{t_1,t_2}^{\lambda_B} - \lambda_R \ln R_{t_1,t_2}^{\lambda_R} \| \big) \end{aligned} \tag{19}$$

Furthermore, we can also normalize the ratio constraint as :

$$d_{ratio}(I_1,I_2) = \frac{d_r(I_1,I_2)}{D_c \max(\lambda_R,\lambda_G,\lambda_B)} \tag{20}$$

where D_c is defined as in Eq. (16).

4.3 Illumination constraint-based algorithm

Before introducing the algorithm based on the two proposed constraints, let's discuss the cases of outdoor illumination changes. We can divide the cases into three main categories as follows:

- Heavy cloudy day: The Sun is absent and changes in illumination are mainly caused by the changes of the brightness and/or the color of the sky. However, in reality, the case that the sky color dramatically changes is rare. For instance, clouds with strange color appear after the rain. For most cases, changes in the sky color are very small and insignificant in comparison with changes of the surface reflectance due to the thickness of the cloud.

- Partial cloudy day: The Sun is partially occluded and changes are mainly caused by the occlusion routine of the Sun or the reappearance routine of the Sun after being occluded. Casting shadow also falls into this case when the Sun is occluded regarding the surface geometry. Although the Sun is partially occluded, the saturated cloud area is so bright that it can be considered as the Sun with lower brightness. Therefore, changes in this case are also very small and insignificant in comparison with the changes of the surface reflectance due to the saturated cloud areas.

- Sunny day: In this case, the Sun light is dominant and changes are mainly caused by changes in the brightness and/or the position of the Sun. Thus, changes in this case are also very small and insignificant in comparison with the changes of the surface reflectance due to the fact the Sun just changes its brightness and/or its location.

From the above observation, we combine two constraints to propose a simple algorithm (Alg. 1) to check whether the two *image irradiances* are from the same point on the surface patch regarding a large change in outdoor illumination.

Algorithm 1 Illumination constraint-based algorithm

Input: Image irradiance I_1 and I_2
Output: A boolean value indicating whether I_1 and I_2 are from the same point on the surface patch after illumination changes

$$d_c \longleftarrow \| \ln R^{\lambda_R}_{t_1,t_2}(I_1,I_2) - C \ln R^{\lambda_G}_{t_1,t_2}(I_1,I_2) + (C-1) \ln R^{\lambda_B}_{t_1,t_2}(I_1,I_2) \|$$

$$\begin{aligned} d_r = \max \big(& \| \lambda_R \ln R^{\lambda_R}_{t_1,t_2} - \lambda_G \ln R^{\lambda_G}_{t_1,t_2} \|, \\ & \| \lambda_G \ln R^{\lambda_G}_{t_1,t_2} - \lambda_B \ln R^{\lambda_B}_{t_1,t_2} \|, \\ & \| \lambda_B \ln R^{\lambda_B}_{t_1,t_2} - \lambda_R \ln R^{\lambda_R}_{t_1,t_2} \| \big) \end{aligned}$$

Return: $d_c < \varepsilon$ **and** $d_c < \delta$

Furthermore, the flow of segmentation process in Fig. 5 illustrates how the algorithm is integrated into the robust segmentation to handle large changes in illumination. Along with the time line, a fusion of multiple cues including background model, motion, temporal prior and illumination is applied to estimate the foreground. However, if the estimated result is inaccurate, i.e. the whole foreground areas are larger than a certain threshold due to the sudden change in illumination, the system will switch into the extraordinary stage using Alg. 1 above. While doing this abnormal routine, the background model is also learned concurrently either till it converges or till the illumination condition becomes stable again. After that, the system switches back into the normal routine as before. The algorithm can be found in detail in Alg. 2. The corresponding result of Fig. 4 is shown in Fig. 6.

5 Shadow Occlusion Handling

In order to handle the shadow occlusion problem, it is necessary to separate the shadow region from detected foreground and then

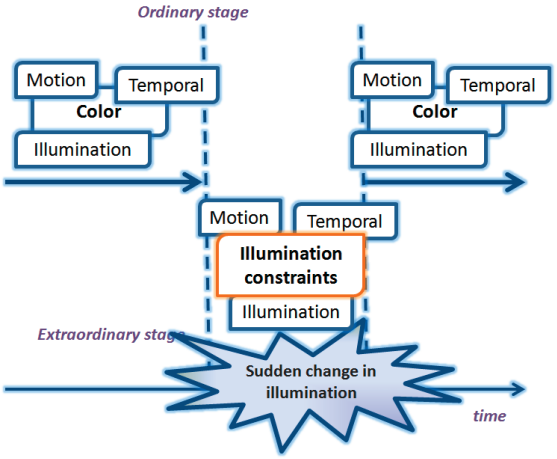

Figure 5: An overview of the foreground segmentation with illumination constraints.

to recast the shadow onto virtual objects correctly. In addition, the foreground depth and height, and the light direction should also be estimated. Shadow detection comes next in section 5.1 and then it is followed by how we recast the detected shadow in section 5.2.

5.1 Shadow detection

Natural shadow in an outdoor scene is normally formed by the fact that only sky light reaches the surface area without the participation of the Sun light. The proposed illumination constraints in section 4 can also be applied with a larger threshold.

In addition, it can be easily proved in appendix B that the proposed illumination invariant constraint in section 4.1 is equivalent to F-value of the shadow invariant in [17] of Marchant *et al.*

With the assumption of blackbody radiation and the narrow-banded camera, we can apply the proposed illumination invariant constraint in Eq. (16) to detect shadow at per-pixel level with the observation that the brightness of shadow is lower than that under sunlight (*brightness constraint*). The process is described in Alg. 3 below.

Nevertheless, the proposed per-pixel approach provides us the shadow point clouds which are unexpected (Fig. 7). Although one can think of a morphology operator to enhance the result, we introduce a region-based optimization by using energy minimization. The shadow energy can be represented as :

$$E_{shadow} = E_{data}(X^t, I^t, I^t_B) + \alpha E_{smooth}(X^t, I^t, I^t_B) \tag{21}$$

Lu *et al.* proposed a method which detects the shadow region using energy minimization in [16]. However, in [16], the authors discard the data term which indicates the likelihood of shadow and non-shadow for each pixel. On the contrary, we use the distance from Eq. (16) to form the shadow likelihood as :

$$E_{data}(X^t = B, I^t, I^t_B) = \sum_{i_r \in I} -\log(d_{inv}(i^t_r, i^t_B)) \tag{22}$$

The smoothness term in (21) is defined as

$$E_{smooth}(X^t, I^t, I^t_B) = \sum_{(p,q)\in N} -\log(Contrast_{atten}(i^t_p, i^t_q, i^{B^t}_p, i^{B^t}_q)) \tag{23}$$

where $Contrast_{atten}$ is defined as in section 3. Finally, the optimum label can also be optimized using energy minimization with graph

Algorithm 2 Segmentation with illumination change

Input: Image irradiance I^t and I_b^t of the current frame and the learned background respectively
Output: Foreground I_f^t from the current frame I^t
Estimate color cue $Prob_{color}$ as in Eq. (4)
Estimate temporal prior cue $Prob_{temp}$ as in Eq. (5)
Estimate motion cue $Prob_{motion}$ as in Eq. (6)
Estimate illumination cue $Prob_{illum}$ as in Eq. (7)
Estimate I_f^t from $Prob_{color}$
$count_{color} \longleftarrow CountForegroundPixel(Prob_{color})$
$count_{illum} \longleftarrow CountForegroundPixel(Prob_{illum})$
if $abs(count_{color} - countillum) > threshold$ **then**
  for each location(x,y) **where** $I_f^t(x,y)$ is foreground **do**
    if Constraint$(I_f^t(x,y), I_b^t(x,y))$ from Alg. 1 **then**
      $Prob_{color}(x,y) \longleftarrow d_r(I_f^t(x,y), I_b^t(x,y))$ from Eq. (20)
    end if
  end for
end if
$I_f^t \longleftarrow DoGraphCut(Prob_{color}, Prob_{temp}, Prob_{motion}, Prob_{illum})$
Return: I_f^t

Algorithm 3 Per-pixel shadow detection

Input: Image irradiance I_f and I_b of foreground and background respectively
Output: A boolean value indicating whether I_f belongs to the shadow region or not
$d_c \longleftarrow \| \ln R^{\lambda_R}(I_f, I_b) - C \ln R^{\lambda_G}(I_f, I_b) + (C-1) \ln R^{\lambda_B}(I_f, I_b) \|$
Return: $d_c < \sigma$ **and** $I_f < I_b$

cut as in Sect. 3. Fig. 8 illustrates how the shadow regions are refined.

5.2 Shadow recasting

With the help of a spherical vision camera, the estimation of the foreground depth and height as well as the directional light can be done with simple spherical geometry. The shadow mapping method is then applied. The process is described in detail in Alg. 4.

First, in order to superimpose virtual objects with respect to foregrounds in the real scene and to cast the detected shadows correctly, the foreground depth should be known and virtual objects should be rendered correctly whether in front of or behind the foreground according to the estimated depth. With the assumption that the ground is relatively flat and that the camera height is known, the foreground depth can be easily estimated by using simple spherical geometry as in Fig. 14 in Appendix C. Let h denote the camera height and d for the estimated depth. We have

$$d = h \sin \alpha \tag{24}$$

where α denotes the first components in the spherical coordinate $I(\alpha, \phi)$ of the foreground bottom. Fig. 9 illustrates how the depth map is estimated.

Additionally, in order to recast the detected shadows, the light direction should be known; hence, the foreground height should be taken into account. Inheriting from the depth estimation, the foreground height is estimated by using simple geometry as illustrated in Fig. 10. Certainly, we also assume that the ground is relatively flat and that the camera height is known. There are two cases in this calculation. One is when the height is over the camera height:

$$h_2 = h(1 + \tan \beta_1 \tan \beta_2) \tag{25}$$

Figure 6: Result of illumination constraints. The uppler left image shows the illumination invariant distance while the one on the upper right shows the illumination ratio distance. The whiter the pixel is, the larger the distance is. The lower image indicates the final segmentation result.

Figure 7: The image on the left is the original frame and its corresponding foreground is in the middle image. The one on the right shows detected shadow in per-pixel level.

And another is when the height is below that of the camera:

$$h_1 = h(1 - \tan \alpha_1 \tan \alpha_2) \tag{26}$$

where h denotes the camera height and the angles are defined as in Fig. 10.

Next, we assume that the detected foreground shadows are all casting shadows caused by the Sun light. Thus, the light source can be safely assumed to be directional. In order to estimate the directional light vector, foreground and its corresponding shadow should be projected onto the ground from the spherical image by using the transformation in Eq. (39) in appendix C. Let $\vec{P}_0$, $\vec{P}$ and h denote the position of the shadow of the foreground head, the position of the foreground bottom and the foreground height estimated as in previous section, respectively. The directional light vector can be simply calculated as :

$$\vec{d_{light}} = \vec{P}_0 - (\vec{P} + \vec{h}) \tag{27}$$

At this stage, we have estimated the required components for shadow recasting, which are the directional light, shadow regions and their corresponding projections onto the ground from the spherical image. Whether we apply the shadow mapping or the shadow volume method, the silhouette of the shadow casters should be known. For each point in known shadow regions on the ground, we calculate the intersection of the vector toward $\vec{d_{light}}$ with the

Figure 8: The per-pixel shadow detection provides results as in the image on the left and the one on the right shows how it is refined.

Algorithm 4 Shadow recasting algorithm

Input: Image irradiance I^t, I^t_f and I^t_b of the current frame, the segmented foreground and the learned background respectively
Output: The render context with recast shadows
$Shadow_{region} \longleftarrow EnergyBasedShadow(I^t_f, I^t_b)$ from Sect. 5.1
for each $fore$ in I^t_f **do**
  $Depth_{fore} \longleftarrow$ DepthEstimation($fore$) from Eq. (24)
  $Height_{fore} \longleftarrow$ HeightEstimation($fore$) from Eq. (25) and (26)
  $DirectionalLight_{fore} \longleftarrow$ DirectionalLight($fore$, $ShadowRegion$, $Depth_{fore}$, $Height_{fore}$) from Eq. (27)
end for
$Light_{direction} \longleftarrow Average(DirectionalLight_{\{I^t_f\}})$
for each $fore$ in I^t_f **do**
  $Caster_{fore} \longleftarrow$ CasterEstimation($Light_{direction}$, fore, ShadowRegion, $Depth_{fore}$, $Height_{fore}$)
end for
Render shadow casters and virtual objects with shadow mapping
Return

vertical plane at foreground bottom $\vec{P}$. The border of all intersection points will provide us the silhouette of the shadow caster. Traditional shadow mapping is then used to correct the shadows on virtual objects.

6 Implementation and Experiment Setup

6.1 Implementation

Since the proposed foreground segmentation system is used to process high resolution frames online in a per-pixel level, it is too time-consuming to be implemented in any practical application. We take advantage of the current available Graphics Processing Unit (GPU) to accelerate the calculation of each cue for foreground segmentation.

In addition, we obtain the solution of the optimum label by using graph cut on the whole image. Although this provides us optimized results, it also yields poor performance especially for high resolution images. By observing that the foreground regions tend to fall into the joint areas provided by multiple cues, we propose a simple heuristic approach by applying the graph cut on areas which are more likely to be foreground. Alg. 5 below explains how the heuristic process is done.

6.2 Experiment setup

In our experiment, the spherical vision camera Ladybug2 by Point Grey Research Inc. is used and fixed on a tripod at the height of around 160cm. First, we experimented on video sequences captured in our campus and then applied in our VA project. In the system, we use a notebook, the spec of which is, OS: Windows 7, CPU: Core2Quad 2.0 GHz, RAM: 4GB, GPU: nVIDIA GTS 160M 1024MB.

Figure 9: Depth estimation using a spherical vision camera. The image on the left is the original frame. The middle image shows segmented foreground. The one on the right is the estimated depth map. The shadow in the last image is considered to be foreground and not separated yet.

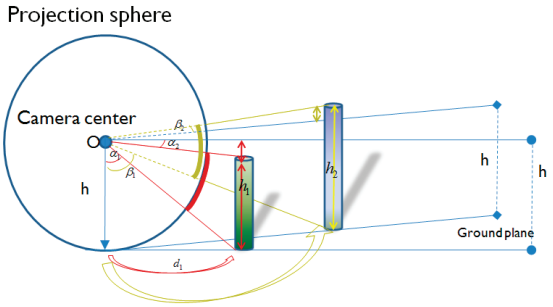

Figure 10: Depth and height estimation using a spherical vision camera.

As introduced in Fig. 2 in Sect. 2, we fix the spherical vision camera during the experiment while the viewers can move the head-mounted display (HMD).

7 Experimental Results

Regarding the experiments on illumination change, we compare our foreground segmentation results with respect to the previous method in [8] which is a combination of [3] and [23]. Since current public dataset is not available, we use our own video data. Fig. 11 and Fig. 12 show how the illumination changes in reality in Asuka and campus sequences. In our experiment, the mixing factors of the energies to minimize (i.e. α, β, γ and θ in Eq. (2)) are set to 0.25, 0.125, 0.125 and 0.5 respectively. Fig. 11 and Fig. 12 below show that previous methods fail when the outdoor illumination changes. Meanwhile, our proposed outdoor illumination handling achieves correct foreground segments.

In order to evaluate the correctness of our proposed method, we

Algorithm 5 Heuristics for segmentation

Input: Image irradiance I^t and I^t_b for the current frame and the learned background respectively
Output: An area A^t_{cut} to segmentation with graph cut instead of the whole frame I^t
$A^t_{cut} \longleftarrow$ NULL
for each cue in $\{Color, Temporal, Motion, Illumination\}$ **do**
  mask $\longleftarrow MaskOf(Prob_{cue})$
  $A^t_{cut} \longleftarrow A^t_{cut} \cup$ Dilate(mask)
end for
Return: A^t_{cut}

Figure 11: Asuka sequence. The images in the first row are original ones. The second row shows how the previous approach fails. The third row indicates probability of foreground from illumination constraints. The bottom row shows the results by our approach.

Figure 12: Campus sequence. The images in the first row are original ones. The second row shows how the previous approach fails. The third row indicates probability of foreground from illumination constraints. The bottom row shows the results by our approach.

use *precision* and *recall* defined in [19] as :

$$precision = \frac{t_p}{t_p + f_p} \tag{28}$$

$$recall = \frac{t_p}{t_p + f_n} \tag{29}$$

where t_p, f_p and f_n denote *true positive*, *false positive* and *false negative*, respectively. The table 1 and 2 below show that our proposed method achieves stable results regardless of illumination changes. However, our system has some limitations which need to be improved. Since our proposed illumination constraints are heavily based on the assumption of narrow-banded cameras and Lambertian surfaces, either non-Lambertian surfaces or a wider band will violate the constraints. Nevertheless, such cases are so rare that we can safely make the assumption for practical applications.

Results from Fig. 16 to Fig. 18 illustrate how the shadow is recast. The recast shadow and the self-shadows on virtual objects match well with the natural ones of the foregrounds, from which we estimate the directional light. However, since we rely on the spherical vision camera and the assumption that the ground is relatively flat, this effective solution is suitable only for outdoor augmented reality. Furthermore, in our approach, we only use detected shadows to estimate the directional light without maintaining the light with temporal consistency. As they can be seen from the demonstration video, the cast shadows and self-shadows flicker due to the unstable estimation the directional light. It is possible and necessary to consider the stability of the directional light which changes very slowly in reality. A combination of multiple cues as in [13] should be applied for more accurate estimation. Nevertheless, the estimation in [13] requires offline processing which makes it impractical to be used in outdoor AR.

Table 1: Asuka sequence.

Method	*Precision*(%)	*Recall*(%)
Proposed method	92	97
Combination in [8]	7	99

Table 2: Campus sequence.

Method	*Precision*(%)	*Recall*(%)
Proposed method	94	97
Combination in [8]	56	99

Finally, with the spec in Sect. 6, our segmentation system can work at a frame rate of 5 frames/sec with the image resolution at 2048 by 1024 and at 2 frames/sec with shadow recasting. Improvement for the faster implementation is required to handle the foreground and shadow occlusion problem in outdoor augmented reality in real-time.

8 Conclusion

Our proposed illumination constraints and foreground segmentation strategy handle well the cases of outdoor illumination changes ranging from the gradual change to the sudden one. Thus, foreground occlusion problem is resolved for outdoor AR. In addition, shadow recasting results show that our solution to the shadow occlusion works effectively. Real experiments on campus and in the VA project prove the effectiveness of the introduced system which can be applied in practical outdoor augmented reality application. Further improvement in foreground and shadow separation should be done to provide more accurate foreground location and light direction. Finally, we will extend our system to use dynamic cameras which allow viewers moving in the virtual world.

Acknowledgements

This work was, in part, supported by the Japanese Ministry of Education, Culture, Sports, Science and Technology, under the program, "Development of High Fidelity Digitization Software for Large Scale and Intangible Cultural Assets."

References

[1] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, 2004.

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

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

[4] B. K. P. Horn. *Robot vision*. MIT Press, Cambridge, MA, USA, 1986.

[5] B. K. P. Horn and B. G. Schunck. "determining optical flow": A retrospective. *Artif. Intell.*, 59(1-2):81–87, 1993.

[6] D. B. Judd, D. L. Macadam, G. Wyszecki, H. W. Budde, H. R. Condit, S. T. Henderson, and J. L. Simonds. Spectral distribution of typical daylight as a function of correlated color temperature. *J. Opt. Soc. Am.*, 54(8):1031–1036, 1964.

[7] T. Kakuta, T. Oishi, and K. Ikeuchi. Shading and shadowing of architecture in mixed reality. In *ISMAR*, pages 200–201, 2005.

[8] 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.

[9] M. Kanbara and N. Yokoya. Geometric and photometric registration for real-time augmented reality. In *ISMAR*, pages 279–280, 2002.
[10] R. Kawakami, R. T. Tan, and K. Ikeuchi. Consistent surface color for texturing large objects in outdoor scene. In *ICCV*, pages 1200–1207, 2005.
[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. Ladikos and N. Navab. Real-time 3d reconstruction for occlusion-aware interactions in mixed reality. In *ISVC (1)*, pages 480–489, 2009.
[13] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Estimating natural illumination from a single outdoor image. In *ICCV*, 2009.
[14] D.-S. Lee. Effective gaussian mixture learning for video background subtraction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(5):827–832, 2005.
[15] L. Li and M. K. H. Leung. Robust change detection by fusing intensity and texture differences. In *CVPR (1)*, pages 777–784, 2001.
[16] C. Lu and M. S. Drew. Shadow segmentation and shadow-free chromaticity via markov random fields. In *13th Color Imaging Conference*, 2005.
[17] J. A. Marchant and C. M. Onyango. Shadow-invariant classification for scenes illuminated by daylight. *J. Opt. Soc. Am. A*, 17(11):1952–1961, 2000.
[18] R. O'Callaghan and T. Haga. Robust change-detection by normalised gradient-correlation. In *CVPR*, 2007.
[19] D. L. Olson and D. Delen. *Advanced Data Mining Techniques*. Springer, 2008.
[20] J. Pilet, V. Lepetit, and P. Fua. Retexturing in the presence of complex illumination and occlusions. In *ISMAR*, pages 1–8, 2007.
[21] J. Pilet, C. Strecha, and P. Fua. Making background subtraction robust to sudden illumination changes. In *ECCV (4)*, pages 567–580, 2008.
[22] P. L. Rosin. Thresholding for change detection. In *ICCV*, pages 274–279, 1998.
[23] J. Sun, W. Zhang, X. Tang, and H.-Y. Shum. Background cut. In *ECCV (2)*, pages 628–641, 2006.
[24] J. Wang and M. F. Cohen. Image and video matting: a survey. *Found. Trends. Comput. Graph. Vis.*, 3(2):97–175, 2007.

A LAMBERTIAN SURFACE UNDER VISIBLE PORTION OF THE SKY

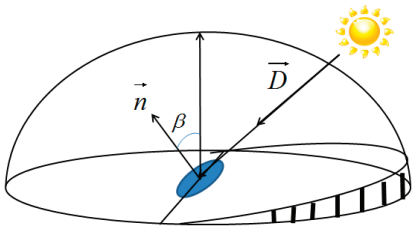

Figure 13: A visible portion of the sky in outdoor illumination model.

Considering a surface patch S with normal $\vec{n}$ and the corresponding *surface reflectance* ρ, we assume that the surface is *Lambertian* and that there are two main sources of light in outdoor scene which are the Sun and the sky. Next, let L denote the *scene radiance* and E as the *irradiance* at the surface patch. Since the Sun light can be safely assumed to be directional and white, we obtain the Sun model from the Lambertian assumption as :

$$L_{sun} = \rho E_{sun}(\vec{n}\vec{D}) \tag{30}$$

where $\vec{D}$ denotes the incident direction of the Sun light as in Fig. 13. Also from Lambertian assumption, *Bidirectional reflectance distribution function* (BRDF) becomes a constant for each surface patch regardless of the direction of viewing. Thus, from [4], under a uniform sky, the radiance of the surface patch can be obtained as :

$$L_{sky} = \int_{-\pi}^{\pi}\int_{0}^{\frac{\pi}{2}} \frac{\rho E_{sky}}{\pi} \sin\theta_i \cos\theta_i d\theta_i d\phi_i \tag{31}$$

where (θ_i, ϕ_i) denotes the incident polar angle of the incoming light. However, in reality, just a portion of the sky is visible to the surface patch. We can modify Eq. (31) by considering the visible hemisphere regarding the surface normal as in Fig. 13. Thus, Eq. (31) becomes

$$\begin{aligned} L_{sky} &= \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}}\int_{0}^{\frac{\pi}{2}} \frac{\rho E_{sky}}{\pi} \sin\theta_i \cos\theta_i d\theta_i d\phi_i \\ &\quad + \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}}\int_{0}^{\theta'} \frac{\rho E_{sky}}{\pi} \sin\theta_i \cos\theta_i d\theta_i d\phi_i \\ &= \frac{\rho E}{2}\left(1 + \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} \frac{1}{1+\tan^2\beta\cos^2\phi} d\phi\right) \\ &= \frac{\rho E(1+\cos\beta)}{2} = \rho E \cos^2\frac{\beta}{2} \end{aligned} \tag{32}$$

where β denotes the angle between the surface normal and the vertical vector as in Fig. 13. Thus, Eq. (30) and Eq. (32) provide us

$$L = \rho\left(gE_{sun}(\vec{n}\vec{D}) + E_{sky}\cos^2\frac{\beta}{2}\right) \tag{33}$$

where $g \in [0,1]$ is to determine whether the point on the surface patch is shadowed or not.

Meanwhile, we can obtain the proportion between the *image irradiance* I and the *scene radiance* L as :

$$I = L\frac{\pi}{4}\left(\frac{d}{f}\right)^2 \cos^4\alpha \tag{34}$$

where d and f denote the lens diameter and the distance from the camera lens to the image plane respectively, and α denotes the angle between the optical axis and the ray from the surface patch to the center of the lens. Thus, we can obtain the outdoor illumination model from Eq. (33) and Eq. (34) as :

$$I = \rho\left(gE_{sun}(\vec{n}\vec{D}) + E_{sky}\cos^2\frac{\beta}{2}\right)B \tag{35}$$

where

$$B = \frac{\pi}{4}\left(\frac{d}{f}\right)^2 \cos^4\alpha \tag{36}$$

It is obvious that B only depends on the location of the surface patch and is independent of the irradiance of any incident light.

B ILLUMINATION INVARIANT

From Eq. (14), we have

$$\begin{aligned} &\ln R^{\lambda_R}_{t_1,t_2} - C\ln R^{\lambda_G}_{t_1,t_2} + (C-1)\ln R^{\lambda_B}_{t_1,t_2} = 0 \; with \; C = \frac{\frac{1}{\lambda_R} - \frac{1}{\lambda_B}}{\frac{1}{\lambda_G} - \frac{1}{\lambda_B}} \\ &\Longleftrightarrow \left(\ln\frac{I_1^{\lambda_R}}{I_1^{\lambda_B}} - C\ln\frac{I_1^{\lambda_G}}{I_1^{\lambda_B}}\right) - \left(\ln\frac{I_2^{\lambda_R}}{I_2^{\lambda_B}} - C\ln\frac{I_2^{\lambda_G}}{I_2^{\lambda_B}}\right) = 0 \\ &\Longleftrightarrow \ln F_1 - \ln F_2 = 0 \end{aligned} \tag{37}$$

Therefore, the illumination invariant constraint is equivalent to the log of F value in [17].

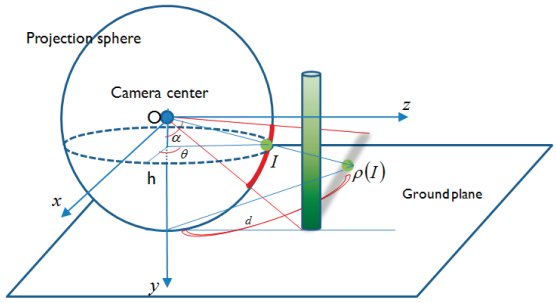

Figure 14: 3D projection onto the ground plane.

C SPHERICAL VISION

In a spherical vision camera system, the surrounding scene is mapped into the spherical image I, in which each member is represented by $I(\alpha,\theta)$ as illustrated in Fig. 14. Given a traditional image coordination $I^{'}(x,y)$, we can transform it into the spherical coordination $I(\alpha,\theta)$ as :

$$\begin{aligned}\alpha &= \pi\frac{I_{height}-y}{I_{height}}\\ \theta &= 2\pi\frac{I_{width}-y}{I_{width}}\end{aligned} \tag{38}$$

On the contrary, each point $I(\alpha,\phi)$ in a spherical image can also be projected into the scene. For example, in Fig. 14, I is projected onto the ground as :

$$\rho(I(\alpha,\theta)) = \begin{cases} x = h\tan\alpha\sin\theta \\ y = h \\ z = h\tan\alpha\cos\theta \end{cases} \tag{39}$$

where h denotes the height of the camera.

Figure 15: Input frames from Campus sequence.

Figure 16: The final result of occlusion handling.

Figure 17: Input frames from Asuka sequence.

Figure 18: The final result of occlusion handling.