Multiview Foreground Extraction and Composition to Multiview Background Using Trimap Sharing for Natural 3D Scene Generation

Myung-Han Hyun,[1] Sung-Yeol Kim,[2] Yun-Suk Kang,[3] Yo-Sung Ho[3]

[1] The 1st R&D Institute-4, Agency for Defense Development, Daejeon 305-600, Korea

[2] Imaging, Robotics, and Intelligent Systems Lab, The University of Tennessee (UTK), Knoxville, TN 37996-2100

[3] Department of Information and Communications, Gwangju Institute of Science and Technology (GIST), Gwangju 500-712, Korea

Received 13 May 2008; accepted 17 May 2010

ABSTRACT: Digital matting for extracting foreground objects from an image is an important process to generate special effects in the movie industry and the broadcasting center. Recently, a digital matting algorithm has been developed to create an alpha matte using a well-focused image generated from multiview images. However, this method could generate only a single-view alpha matte, even though it used multiple cameras. In this article, we propose a new estimation scheme for multiview alpha mattes by sharing the trimap of the reference view. Furthermore, we use the motion vector to update the trimap for video matting. After we extract foreground objects from all view images, we composite the foreground objects with the corresponding background images captured in the same multiview camera arrangement. Experimental results demonstrate that multiview composite images can generate reasonably natural 3D views through the stereoscopic monitor. © 2010 Wiley Periodicals, Inc. Int J Imaging Syst Technol, 20, 285–293, 2010; View this article online at wileyonlinelibrary.com. DOI 10.1002/ima.20251

Key words: multiview foreground extraction; multiview camera system; trimap sharing

I. INTRODUCTION

Efficient image and video compositing techniques are required to make special effects in a movie industry or a broadcasting center. In general, a composite image is divided into two layers: foreground object and background. To extract a foreground object, referred to as a foreground matte, we remove the background of the original image by considering an alpha value α that represents the pixel opacity of the image. This technique is known as digital matting (Chuang et al., 2001). Meanwhile, digital compositing is to combine a foreground matte with an arbitrary background by using the alpha value α (Porter and Duff, 1984).

Blue screen matting is widely used for digital matting (Smith and Blinn, 1996). As the blue screen matting algorithm uses monotonous blue or green backgrounds, it is easy to extract a foreground object from them. However, if the foreground object contains the background constraint color, it is hard to pull out the foreground object efficiently. Furthermore, a virtual studio environment is required for the blue screen matting. To overcome the limitation of blue screen matting, natural image matting, which has fewer constraints for backgrounds, has been actively studied in the field of computer vision (Hillman et al., 2001; Wexler et al., 2002; Hillman and Hannah, 2005). However, the natural image matting requires user assistances and more complicated algorithms than the blue screen matting. Moreover, we need to make strenuous efforts to extract the foreground object from complex background scenes (Ruzon and Tomasi, 2000). To obtain the enhanced alpha matte in the natural image matting, we can also use secondary operations by an image gradient (Sun et al., 2004). Although the previous works enable us to represent complex boundaries correctly, they usually take too tedious operations.

Recently, a digital matting algorithm using multiview cameras has been developed to create an accurate alpha matte (Joshi et al., 2006). It can generate an alpha matte fast and automatically. However, even though this work uses a multiview camera system, it can only generate a single-view alpha matte because it assumed that the alpha value α is fixed in all view images. As a result, the work cannot produce multiview composite images from the multiview camera system. In addition, as previous work suffers from a large amount of ad-hoc operations to create a trimap, such as double-thresholding and the selection of a structuring element, the unknown region of the trimap is tended to be isolated and broadened. Therefore, it is difficult to estimate the alpha value in the unknown region.

In this article, we propose a new digital matting algorithm to estimate multiview alpha mattes using multiview images. The main contribution of our work is that we first propose the concept and

Correspondence to: Myung-Han Hyun; e-mail: drwise@add.re.kr

Grant sponsor: MKE under the ITRC support program supervised by NIPA (NIPA-2010-(C1090-1011-0003))

methodology of the framework for multiview video matting and compositing. In this work, we generate view-dependent alpha mattes to extract each foreground object from multiview images by sharing a trimap of a reference view. For video matting, we use the motion vector to update the trimap. Furthermore, we consider multiview background images captured from the identical camera system for digital compositing. Thus, we can generate 3D scenes using multiview composite images. We can also reduce the overall processing time in comparison with the conventional matting methods that independently extracts the foreground object from each camera.

This article is organized as follows. In Section "Alpha Channel Estimation," we discuss the conventional alpha channel estimation, and we describe the multiview video matting and compositing in Section "Multiview Video Matting and Compositing." After experimental results and analysis are presented in Section "Experimental Results and Discussion," we conclude the article in Section "Conclusion."

II. ALPHA CHANNEL ESTIMATION

Most existing methods for natural image matting require the input image to be accompanied by a trimap [Chuang et al., 2001; Hillman and Hannah, 2005; Hillman et al., 2001; Ruzon and Tomasi, 2000; Sun et al., 2004). The goal of the method is to solve the compositing equation $I = \alpha F + (I - \alpha)B$, where I, F, and B are the composite, foreground, and background colors, respectively, for the unknown pixels. Given the composite color I, matting solves the inverse problem with seven unknowns (α, F_r, F_g, F_b, B_r, B_g, B_b) and three constraints (I_r, I_g, I_b) (Joshi et al., 2007; Levin, et al., 2008). This is typically done by exploiting some local regularity assumptions on F and B to predict their values for each pixel in the unknown region. To make a composite image, we first extract a foreground object by estimating an alpha channel. An alpha matte, which is composed of alpha values, can separate and blend pixels according to its values ranged 0 to 1. Then, the extracted foreground object can be composited with an arbitrary background.

In this Section, we first introduce the previous alpha channel estimation techniques briefly and explain their inherent problems for estimation of multiview alpha mattes. Then, we propose a new scheme to estimate multiview alpha mattes by sharing the trimap of a reference view that has significantly a better achievement in foreground extraction as compared to the conventional method in next Section.

A. Variance-Based Alpha Estimation. Joshi et al. (2006) extended the compositing equation to deal with the variance of pixel measurements using a camera array for alpha matting. Given n images of a scene, we consider the following matting equation of a given scene point i by

$$I_i(p) = \alpha(p)F_i(p) + (1 - \alpha(p))B_i(p) \tag{1}$$

where $I_i(p)$ corresponds to the intensity of point p recorded in image I_i. $F_i(p)$ and $B_i(p)$ are the foreground and background pixel values that, as a function of the transparency $\alpha(p)$, are mixed to give $I_i(p)$. We consider $\{I_i(p)\}_{i=1}^n$, $\{F_i(p)\}_{i=1}^n$ and $\{B_i(p)\}_{i=1}^n$ as sampling the random variables $\mathbf{I}$, $\mathbf{F}$, and $\mathbf{B}$, respectively, and rewrite the compositing equation using these variables as

$$I = \alpha F + (1 - \alpha)B, \tag{2}$$

By deriving the variance-based matting equation to solve for α, we take the variance of Eq. (2)

$$\mathrm{var(I)} = \mathrm{var}[\alpha \mathrm{F} + (1 - \alpha)\mathrm{B}]. \tag{3}$$

If we assume that B and F are statistically independent, then

$$\begin{aligned}\mathrm{var(I)} &= \mathrm{var}[\alpha F + (1-\alpha)B] \\ &= \left\langle [\alpha F + (1-\alpha)B - \langle \alpha F + (1-\alpha)B\rangle]^2 \right\rangle \\ &= \left\langle [\alpha(F - \langle F\rangle) - (1-\alpha)(B - \langle B\rangle)]^2 \right\rangle \\ &= \alpha^2 \left\langle (F - \langle F\rangle)^2 \right\rangle \left\langle (B - \langle B\rangle)^2 \right\rangle \\ &= \alpha^2 \mathrm{var}(F) + (1-\alpha)^2 \mathrm{var}(B),\end{aligned} \tag{4}$$

where $\langle X\rangle$ denotes the mean value of X. The assumption that $\mathbf{B}$ and $\mathbf{F}$ are statistically independent is manifested in going from the third to the fourth line of Eq. (4), where the expected value of term $\alpha(1 - \alpha)(E - \langle F\rangle)(B - \langle B\rangle)$ is assumed to be equal to zero. To compute α, we need to solve a quadratic equation as

$$[\mathrm{var}(F) + \mathrm{var}(B)]\alpha^2 - 2\mathrm{var}(B)\alpha + [\mathrm{var}(B) - \mathrm{var}(I)] = 0. \tag{5}$$

The solutions to this quadratic equation are

$$\alpha = \frac{\mathrm{var}(B) \pm \sqrt{\Delta}}{\mathrm{var}(F) + \mathrm{var}(B)}, \tag{6}$$

where

$$\Delta = \mathrm{var}(I)[\mathrm{var}(F) + \mathrm{var}(B)] - \mathrm{var}(F)\mathrm{var}(B). \tag{7}$$

However, the variance-based alpha estimation suffers from one problem that prevents it to become a complete method for multiview alpha matte generation. The source of this problem arises from this fact that the transparency of the point α is view-independent and fixed across all images. Consequently, it is hard to make view-dependent alpha mattes using the variance-based alpha estimation even though we use camera arrays.

B. Color-Cluster-Based Alpha Estimation. A vast literature is devoted to the alpha estimation based on color clusters. Chuang et al. [2001] used mixture of oriented Gaussians to learn the local distribution and then α, foreground color F, and background color B are estimated as the most probable ones given that distribution. Hillman et al. [2001] used principal component analysis to represent color clusters with oriented line segments.

Basically, we can estimate a clean foreground color f and a clean background color b from the color clusters of foreground and background. By using them, we can make a blended pixel p in the set of unknown pixels. If we assume that f is within the foreground cluster and b within the background cluster extracted from the image, then, the pixel p will lie on the line. Thus, α will be the ratio $\overrightarrow{bp}/\overrightarrow{bf}$ (Hillman et al., 2001).

As shown in Figure 1a, we first cluster the area of foreground and background according to the central pixel of unknown area. The size of the window should be flexible with respect to the width of unknown regions. Second, we calculate representative values of the foreground and background cluster by quantizing R, G, and B coordinates with level 10 and then make a histogram (ISO/IEC JTC1/SC29/WG11 M11014, 2004). We chose histogram value over 10. Finally, we take the mean of both clusters' chosen pixel RGB coordinate. From the vector in Figure 1b, we perform dot product to get a proportion of $\vec{f} - \vec{b}$ and $\vec{s} - \vec{b}$, where $\vec{f} - \vec{b}$ and $\vec{s} - \vec{b}$ specify

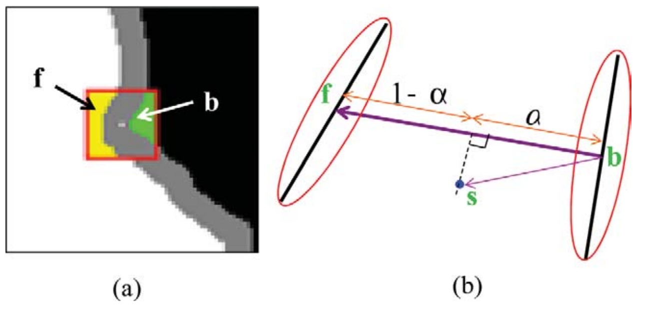


Figure 1. Color clusters and its geometrical representation: (a) pixel sets in color cluster, (b) color space. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

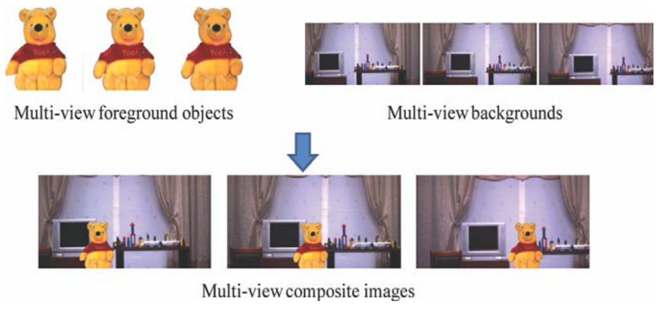


Figure 3. Outline of multiview matting and compositing. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

the vectors whose origin is point b and terminals are point f and s, respectively. From the representative values of the foreground, background and unknown area, we can recover α by using a simple geometrical formulation in Eq. (8)

$$\alpha = \frac{(\vec{s} - \vec{b}) \cdot (\vec{f} - \vec{b})}{\left|\vec{f} - \vec{b}\right|^2}. \tag{8}$$

However, the previous color cluster based alpha estimations work well only when the color distributions of the foreground and background do not overlap and the unknown region in the trimap is small (Levin et al., 2008). As the trimap is not made by user-assist for the proposed multiview matting, the unknown region in the trimap is wide. As shown in Figure 2, color cluster based alpha estimation is not a reliable approach to estimate multiview alpha mattes.

III. MULTIVIEW VIDEO MATTING ANDCOMPOSITING

This Section denotes to the presentation of a new scheme for the multiview video matting and compositing, where we efficiently utilize the concept of the trimap sharing and view-dependent alpha matte generation for this purpose.

A. Concept of Multiview Video Matting and Compositing. For natural 3D scene editing system, we should handle foreground objects of other views for multiview matting. We define the multiview matting as a method that considers multiview foreground objects extraction. For efficient multiview matting, we do not independently extract the foreground object from each camera but adapt a trimap sharing concept. By trimap sharing, we do not need to make manual multiview trimaps. We also define the multiview compositing as a method that composes the extracted multiview foreground objects with the corresponding multiview backgrounds to make multiview composite images. Through the multiview composite images, we can generate natural 3D scenes using the stereoscopic monitor.

Figure 3 illustrates the proposed multiview video matting and compositing. From the camera arrays, we capture multiview images, and then segment foreground objects in the multiview images. Finally, we composite them with multiview background images from the same camera environment.

Figure 4 shows the proposed procedure of the multiview video matting and compositing using trimap sharing. First, we film multiview images using a multiview camera system as shown in Figure 4a. Second, we make a synthetic aperture image (SAI) in Figure 4b by moving other view images to the central view image, i.e., the reference view, of the multiview camera system (Joshi, et al., 2006). SAI will be specified by the following subsection. After creating SAI, we convert it into a variance image in Figure 4c by calculating a variance of its corresponding pixels. Third, to generate a trimap in Figure 4d, we apply dilation and erosion operations into a binary image generated by the variance image. The trimap contains the foreground object, background, and unknown areas. Fourth, the trimap is shared with other views and used to create multiview alpha mattes by estimating the boundary of unknown areas. Finally, we can extract and composite the multiview foreground objects from multiview images using multiview alpha mattes in Figure 4e.

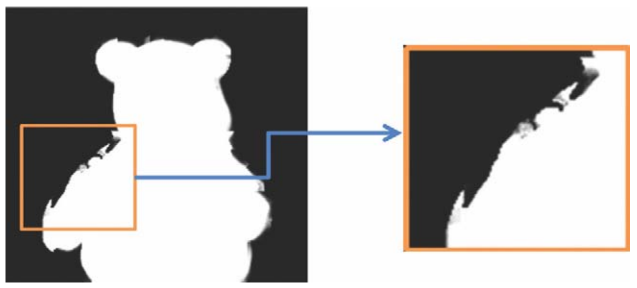


Figure 2. Drawbacks of color-cluster-based alpha estimation. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

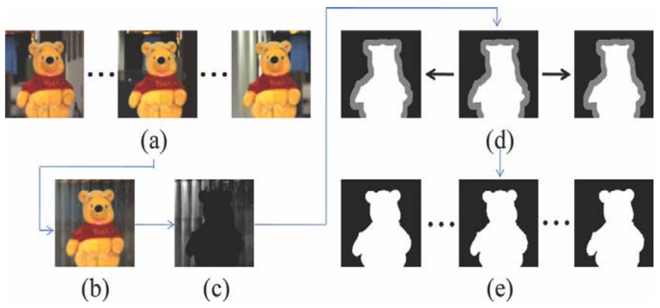


Figure 4. Multiview matting procedure: (a) multiview images, (b) synthetic aperture image, (c) variance image, (d) trimap sharing, (e) multiview alpha mattes. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

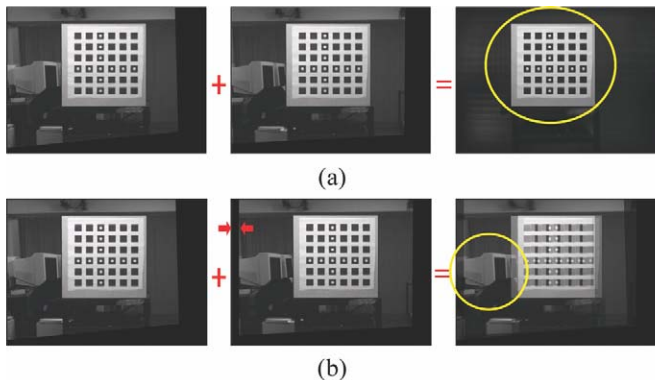


Figure 5. Focused depth change according to parallax shift: (a) focused on the calibration chart, (b) focused on the monitor. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

In the multiview image matting, we make the reference trimap from SAI. We share the reference trimap to the side views. For next frame's trimap, we need to update the reference trimap. By using the temporal information of the video sequence, we do not need to make SAI at every frame. As a result, we can save time to perform the video matting.

To update the reference trimap, we need to find a motion vector using the present and previous frame. To calculate the motion vector, we split the frames as 16×16 block size, and then use 48×48 block size search range. By using the motion vector, we update the trimap. However, there exist some holes in the trimap. To reduce the holes, we perform closing operation. Finally, we can make an updated trimap for next frame.

B. Multiview Image Rectification. Image rectification is the geometrical transformation of two or more images, i.e., multiview images. Using the multiview image rectification, we can efficiently overlap each view to make SAI easily. Basically, multiview images have horizontal disparities and vertical mismatches among views. Horizontal disparities are caused by distances between camera positions, and vertical mismatches can be occurred by the camera misalignment and different camera rotations.

Multiview image rectification transforms each image by applying the rectifying transformations to obtain rectified images which have no vertical mismatches. We use the rectification algorithm by Tanimoto and Fuji (ISO/IEC JTC1/SC29/WG11 M11014, 2004). After camera calibration, we first calculate one baseline that has the minimum distance to each camera center. Then, we assume virtual cameras which are on the perfectly arranged parallel camera array and we can estimate all camera parameters of them. Rectifying transformations can be defined by solving relationships between the original and the estimated camera parameters. By applying these transformations, we finally get rectified images.

C. Synthetic Aperture Image. Figure 5 is a good example of SAI generation (Vaish et al., 2005). As we move the disparity of between two images, we can focus on the image according to the calibration chart or the computer monitor. In our work, we obtain SAI as shown in Figure 6a.

In the digital matting using multiview cameras (Joshi et al., 2006), we overlap identical image pixels in each camera to compute the mean and variance statistics as shown in Figure 6b.

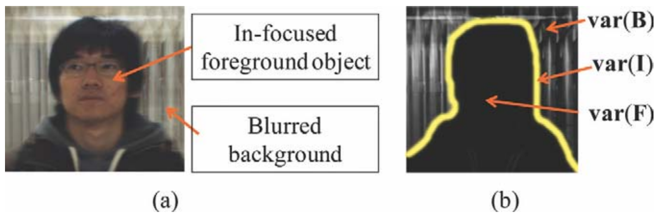


Figure 6. SAI and its pixel variance: (a) SAI, (b) pixel variance of SAI. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Thus, the image of each camera is synthetically refocused according to the foreground object's depth plane (Isaksen et al., 2000). This characteristic is used to make SAI that is well focused on the foreground object but blurred at the background. By shifting the disparity between the foreground object of the reference view and foreground object of the other camera, we can make arbitrary aperture images.

D. Trimap Generation and Sharing. In the previous method (Joshi et al., 2006), the trimap can be generated by taking two threshold values $T1$ and $T2$. Especially, it is difficult to determine $T2$ because it varies in a broad range from 1000 to 5000. Basically, the threshold $T2$ depends on the number of images and the characteristic of images. Thus, $T2$ causes incorrect trimap generation as shown in Figure 7a.

However, the proposed trimap in Figure 7b can depict the boundary area appropriately rather than the previous method. To make the proposed trimap, we adapt automatic threshold selection algorithm to make the binary image from the variance image. Then, we dilate and erode the binary image to outward and inward, respectively. After creating the trimap, we share it for other view images to generate multiview alpha mattes. For efficient trimap sharing, we need to determine the size of structuring element to make a proper trimap for side views. We use the concept of 3D warping techniques to model the size of structuring element.

D.1. Automatic Threshold Selection Algorithm. To select a threshold value automatically for a binary image generation, we first calculate the histogram of the variance pixel values. Then, we set an initial threshold T as 20 because most of pixel intensities of the variance image are skewed around the lower intensity range as shown in Figure 8.

After that, we separate the histogram into two areas, G_1 and G_2, using the initial threshold T and calculate the mean intensities, μ_1

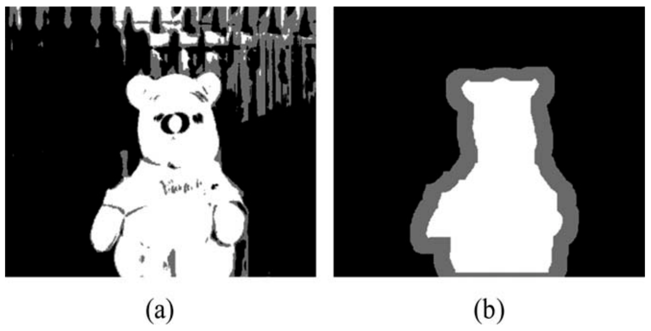


Figure 7. Trimap comparison: (a) previous trimap, (b) proposed trimap.

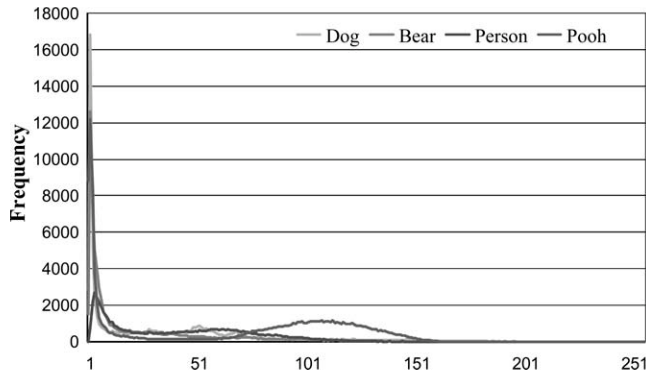


Figure 8. Histogram of variance images.

and μ_2, in both G_1 and G_2 areas. Finally, we update the threshold T as the average of each mean intensity, $T = 0.5 \cdot (\mu_1 + \mu_2)$, and repeat the averaging and updating procedures until T is converged. As shown in Figure 9, we can make the binary image automatically using the threshold value T.

Even though we can make the binary image automatically, it cannot fully specify the foreground object because of the circumferential noise and holes. To reduce the circumferential noise and holes, we perform on 8-connected region-labeling algorithm and use a morphological filter. In constructing a morphological filter, we use erosion and dilation with a flat structuring element as follows (Lim et al., 2006; Gonzalez and Woods, 2002).

$$(f \oplus K)(x) = \max\{f(x-z) + k(z), \quad z \in Z, (x-z) \in F\} \tag{9}$$

$$(f \ominus K)(x) = \min\{f(x+z) - k(z), \quad z \in Z, (x+z) \in F\} \tag{10}$$

where f and k represents an image and a small structuring element, F and K are the domains of f and k. After implementing closing operation by using the composition of the kth order morphological dilation and erosion operations, we perform again on the 8-connected region-labeling algorithm. Finally, the largest labeled region is marked out for the candidate region of the foreground object.

From the binary image in Figure 10, we make a trimap by dilating outward and eroding inward using Eqs. (9) and (10) to specify the unknown area. To this end, the foreground object, background, and unknown areas are represented as a set like Eq. (13).

$$A = \{I_{\text{var}}(x,y) | I_{\text{var}}(x,y) < T, 0 \leq x < T, \quad 0 \leq y < H\} \tag{11}$$

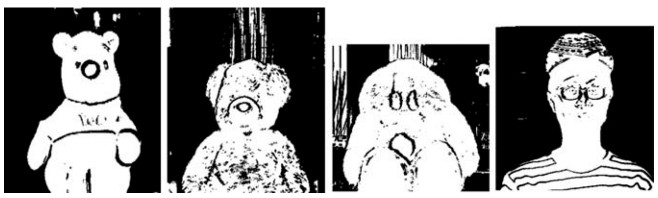


Figure 9. Result of automatic binary images.

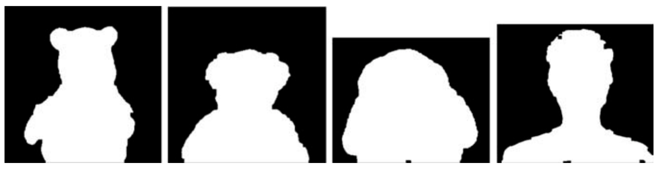


Figure 10. Result of morphological filtering and region-labeling.

$$B = \{I_{\text{var}}(x,y) | I_{\text{var}}(x,y) \geq T, 0 \leq x < T, \quad 0 \leq y < H\} \tag{12}$$

$$C = \{(A \oplus Z) \odot (A \ominus Z)\}^c \tag{13}$$

where set A, set B, and set C contain the foreground, background, and unknown area's pixel values whose intensity are 255, 0, and 128, respectively. The foreground set A specifies the region in which the variance value is less than T, and the background set B specifies the region in which the variance value is greater than T. The Symbols $\oplus$, $\ominus$, and $\odot$ are morphological operations and represent dilation, erosion, and exclusive-or, respectively. $I_{\text{var}}(x, y)$ is the pixel intensity of the location (x, y) in the variance image I_{var}. Figure 11 shows the results of the manual and automatic trimap images. We notice that the automatically generated trimap is appropriate compared with the manually generated trimap.

D.2. Structuring Element Size Modeling. To model the size of structuring element, we use the 3-D image warping technique with the depth and intrinsic camera parameter (McMillan, 1997.). From the geometry of stereo images in Figure 12, we recover the depth value in Eq. (16) by using Eqs. (14) and (15).

$$\frac{Z}{f} = \frac{Y}{nx_1} \tag{14}$$

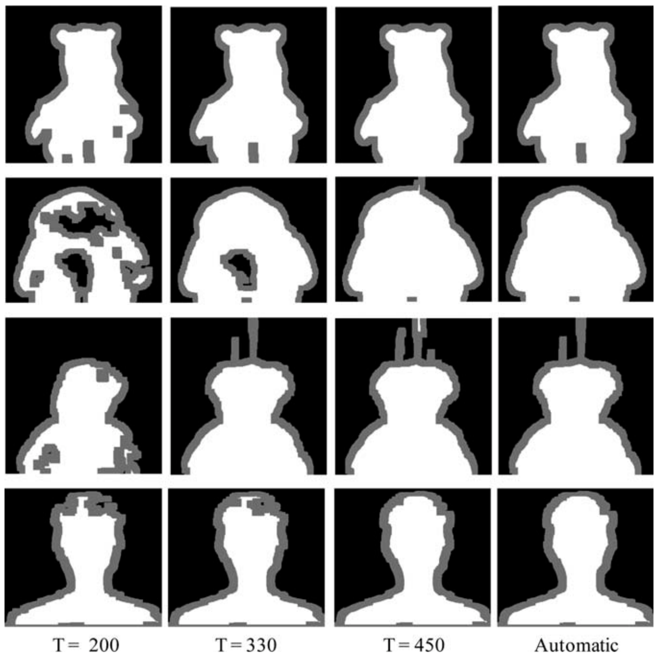


Figure 11. Result of manual and automatic trimap images.

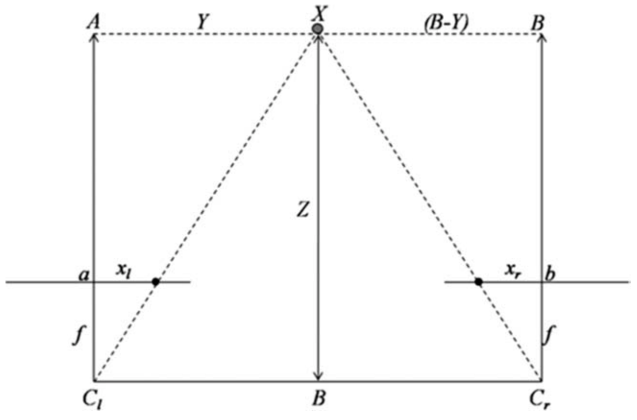


Figure 12. Geometrical representation of the stereo images.

$$\frac{Z}{f} = \frac{(Y - B)}{x_r} \tag{15}$$

$$Z = \frac{f \cdot B}{(x_i - x_r)} = \frac{f \cdot B}{d} \tag{16}$$

where Z, B, f, and d represent the depth, camera shift, focal length and disparity, respectively. Furthermore, the Z should be normalized as a real depth d_r, whose interval is from 0 to 255, in the 3D space by using Eq. (17)

$$d_r = \left\lfloor 255 - \frac{255 \cdot (Z - Z_{\text{ciose}})}{Z_{\text{far}} - Z_{\text{ciose}}} \right\rfloor + 0.5 \tag{17}$$

Here, Z_{far} and Z_{colse} are the maximum and minimum depth value obtained by Eq. (16) and $\lfloor \alpha \rfloor$ means the maximum integer which is less than or equal to α.

For 3D warping, we first map the image pixel location to the real space coordinate system (x, y, z) using Eq. (18),

$$(x, y, z)^T = R_{\text{ref}} A_{\text{ref}}^{-1} (u, v, 1) d_{u,v} + T_{\text{ref}} \tag{18}$$

where A_{ref}, R_{ref}, T_{ref}, and $d_{u,v}$ mean intrinsic parameter, rotation matrix, translation matrix, and the depth value of (u, v), respectively. Second, we obtain the (l_i, m_i, n_i) by projecting the $(x, y, z)^T$ in Eq. (18) to C_i, destination of camera view-point, using Eq. (19)

$$(l_v, m_i, n_i)^T = A_{C_i} R_{C_i}^{-1} \{(x, y, z)^T - T_{C_i}\}. \tag{19}$$

Third, we normalize the (l_i, m_i, n_i) as (l_i/n_i, m_i/n_i, 1), then, it can be represented as (U_i, V_i) through the integer coordinate system. Finally, we map the pixel value located in (u, v) to the location of (U_i, V_i).

After selecting the two points on the foreground in the reference view, whose distance is maximized, we perform 3D warping into the corresponding location of the right most and left most views. Then, we calculate the distance between the warped two points. Likewise, we set the size of the unknown area's width and height as follow

$$\text{Width} = \text{longest distance} - \text{shortest distance} \tag{20}$$

$$\text{Height} = \frac{1}{2} \times \text{Width}. \tag{21}$$

Ideally, we do not care the disparity in the vertical direction because we assume that the height of each camera from the ground is identical. However, we should consider the vertical disparity due to the possibility of vertical height difference in actual application. Here, we set the vertical size of the unknown areas as a half of width.

E. View-dependent Alpha Matte Generation. To generate view-dependent alpha mattes with the shared trimap, we first convert the RGB color space for the multiview images into YCbCr color space, and apply Gaussian filtering into Cb components to reduce the noise around the foreground boundaries. Then, we perform canny edge algorithm. Let P_u is the set of unknown pixels in the trimap and P_e is the set of edge pixel in the multiview images. Thus, boundary edges can be represented as the set $P_b = P_u \cap P_s$. We label N disconnected boundary edge sets, E_1, E_2, . . ., E_N, then, count their pixel number $|E_1|$, $|E_2|$, . . ., $|E_N|$ where $|\cdot|$ denotes the length of the edge. Because short edges can be regarded as a noise, we need to label only edges whose length is bigger than the threshold. The threshold value is decided by analyzing the number of disconnected edges of the boundary edges, and threshold value is experimentally 30. Finally, the labeled edge set is represented by

$$E = \left\{ E_m \middle| \begin{array}{c} E_m > \text{Thershold}, \\ m = 1, 2, \ldots, M < N \end{array} \right\}, \tag{22}$$

Next, we connect the disconnected boundary edges in Eq. (22) by using end points of them. To figure out the end point, we use 3 × 3 block whose element is $B_{ij}(i, j = 1,2,3)$. If the B_{22} contains white pixel 1, then we calculate the

$$\begin{array}{l} Sum = |B_{11} - B_{12}| + |B_{12} - B_{13}| + |B_{13} - B_{23}| + \\ |B_{23} - B_{33}| + |B_{33} - B_{32}| + |B_{32} - B_{31}| + |B_{31} - \\ B_{21}| + |B_{21} - B_{11}| \end{array} .$$

We can regard the point whose Sum is 2 as the end point. We can define the set of end points by $E_m(i, j)$, $0 \le i < W$, $0 \le j < H$, $1 \le m \le M$ where i and j indicate the horizontal and vertical position in edge images, and M is the number of the labeled edges. W and H are the horizontal and vertical resolution of the multiview images, respectively. Finally, we calculate the minimum Euclidean distance $Dist = \min_{i=k}\{E_l(i,j), E_l(i,j)\}, 1 \le l, k \le M$ between the end points. Finally, we make the contour of the foreground by connecting the points $E_l(i, j)$ and $E_k(i, j)$ to make multiview alpha mattes.

However, during view-dependent alpha matte generation, we cannot obtain robust edges from the canny edge algorithm. To overcome the problem in the edge extraction, we apply the histogram equalization operator into the multiview images. Then, we again apply the canny edge operation with Cb components of the histogram equalized multiview images. Finally, we merge the extracted edges of original multiview images with the extracted edges of histogram equalized multiview images.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

For evaluation of the proposed approach, we used a 1D parallel multiview camera system where seven Multi Sync IEEE-1394b cameras were equipped with the camera baseline of 5 cm. Figure 13 shows the multiview camera system. Especially, we have captured

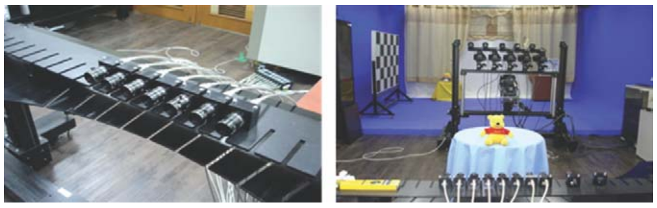
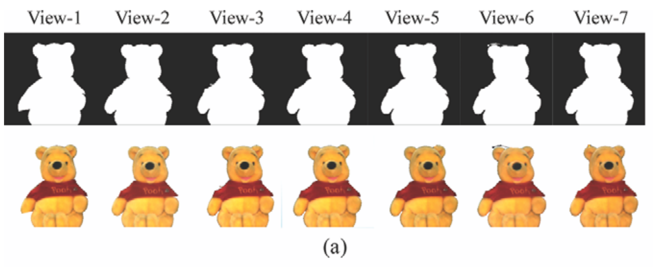


Figure 13. Multiview camera system. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]



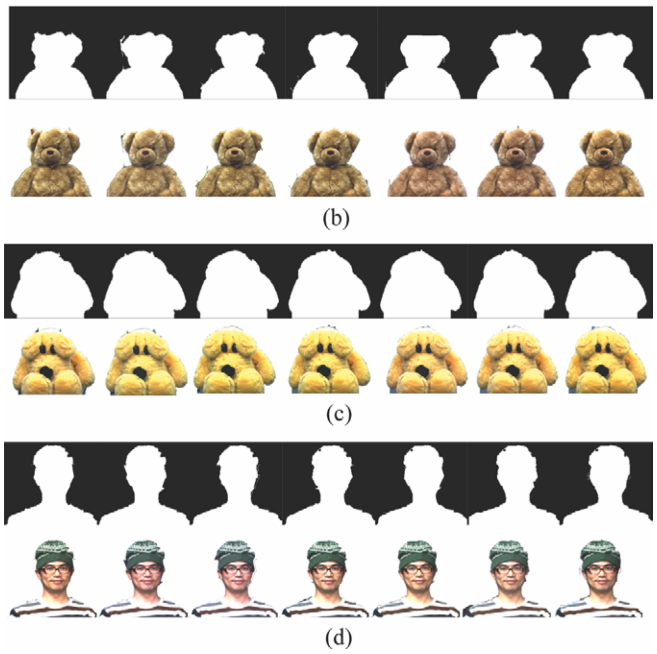


Figure 14. Multiview alpha mattes and foreground objects: (a) pooh sequence, (b) bear sequence, (c) dog sequence, (d) person sequence. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

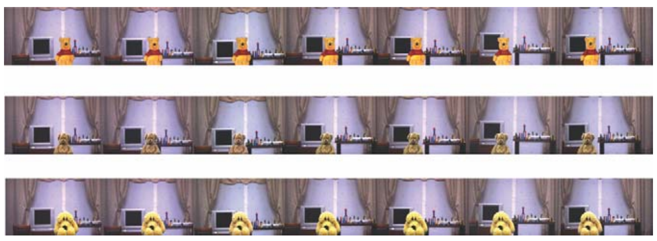


Figure 15. Multiview composite images. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

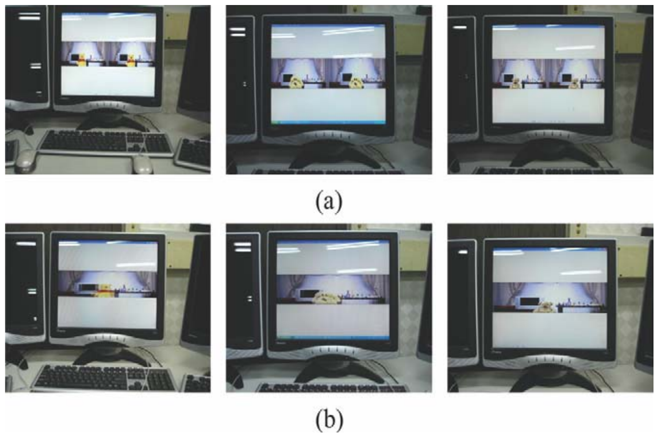


Figure 16. Results of the stereoscopic view: (a) stereo input, (b) stereoscopic view. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

test sequences with narrow camera arrays and a foreground object that is far away from the background but relatively near in camera array because it is efficient to make a SAI (Joshi, et al., 2006) Test data were Pooh, Dog, Bear, and Person sequences and their actual image resolution is 1024 × 768. The experiment was implemented on an Intel based PC (Intel Core 2 Quad CPU 2.41GHz, 2GB DDR2RAM) under Microsoft Windows XP.

A. Visual Evaluation. Figures 14(a), 14(c), 14(e), and 14(h) show the results of multiview alpha mattes. By using the multiview alpha mattes, we extracted foreground objects of each view as shown in Figures 14(b), 14(d), 14(f), and 14(i). As shown in Figure 15, we composited the extracted foreground objects with corresponding multiview background images from the same camera environment. The previous method had no contribution for the 3D scene generation because it only considered the single-view foreground on an arbitrary background image.

However, the proposed method considered not only the multiview foreground objects but also the corresponding multiview

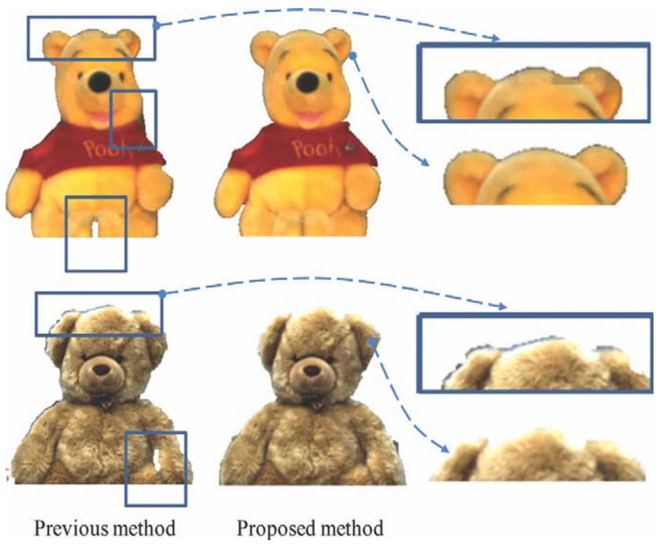
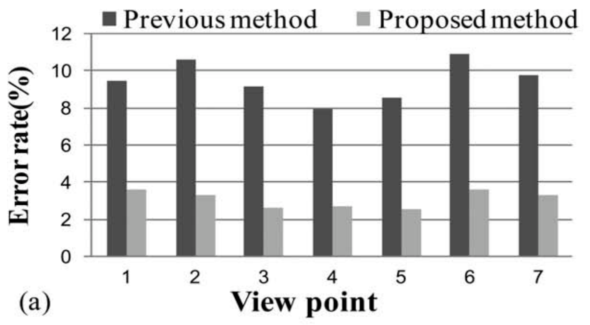
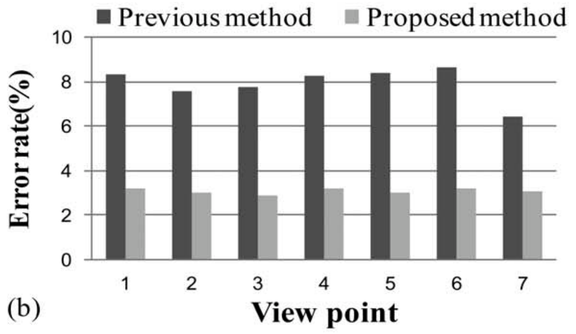


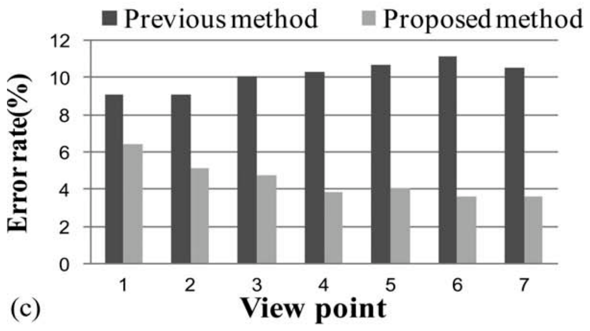
Figure 17. Qualitative comparison. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

background images captured by the same multiview camera system. As a result, as shown in Figure 15, multiview composite images generated by the proposed method could be used as a multiview video.

To prove the multiview composite images can generate the multiview video, we displayed the two adjacent images of the multiview composite images with a stereoscopic monitor as shown in Figure 16. We also displayed the stereoscopic images captured by the two adjacent cameras in the multiview camera system with the stereoscopic monitor. We could experience natural 3D effects by not only the stereoscopic images but also stereoscopic composite images. Thus, the proposed multiview digital matting can be used for a 3D image editing system.







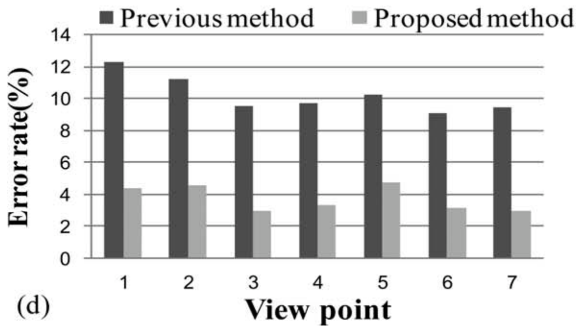


Figure 18. Error rate of the test sequences: (a) pooh sequence, (b) bear sequence, (c) dog sequence, (d) person sequence.

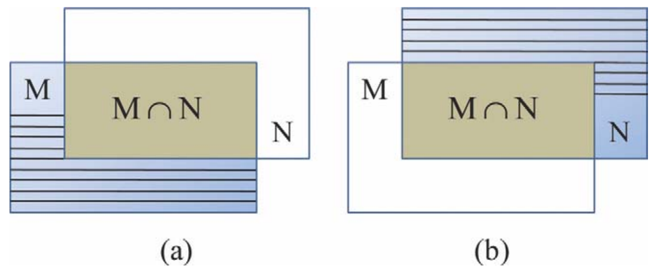


Figure 19. Evaluation system: (a) under-extraction, (b) over-extraction. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

As shown in Figure 17, we compared extracted foregrounds by the previous and proposed methods. The previous method had problems on the foreground boundaries compared to the proposed method. While the proposed method extracted foreground boundaries more exactly than the previous method, the previous method was failed due to the color similarity between foreground and background at times.

B. Quantitative Evaluation. Figure 18 shows the error rate of the test images using the previous and proposed methods. We used error evaluation method as Eq. (23) and Eq. (24) (Kwak et al., 2004),

$$E_{\mathrm{U}} = \frac{M - (M \cap N)}{s_M}, \quad E_{\mathrm{O}} = \frac{N - (M \cap N)}{s_N}, \tag{23}$$

$$\text{Error Rate} = (E_{\mathrm{U}} + E_{\mathrm{O}}) \times 100, \tag{24}$$

where M is pixel number of ground truth image from a image tool and N is pixel number of foreground object from the proposed method. The term S_M and S_N are the numbers of M and N. E_{U} is the error rate of under-extraction and E_{O} is the error rate of over-extraction as shown in Figure 19. With under- and over-extraction, we mean that the corresponding area is and is not to be extracted. As you can see from the quantitative error results in Figure 18, our method provides the lower error rate than the previous method. One reason that the previous method is inferior to the proposed method because the broad unknown area specified by dilation and erosion caused incorrect α. Thus, the incorrect α increased the error rate.

V. CONCLUSIONS

In this article, we have proposed the concept and methodology of multiview foreground extraction and composition to the multiview background technique for the multiview camera system. By using the multiview camera system, we have reduced manual steps for the trimap generation. We have made the trimap from SAI and shared it for multiview images to make alpha mattes by the boundary estimation. Furthermore, we used the motion vector to update the trimap for video matting. Finally, we made alpha mattes by the edge labeling algorithm. Using the view-dependent alpha mattes, we extracted foreground objects of each view and composite them with corresponding multiview backgrounds. In contrast to previous methods, the proposed method enables us to do the multiview matting and compositing by considering foreground objects and their corresponding backgrounds. Finally, we have verified that multiview composite images can generate 3D scenes through the stereoscopic monitor. We, therefore, believe that the proposed approach could be helpful to edit the multiview composite images effectively.

REFERENCES

Y.Y. Chuang, B. Curless, D.H. Salesin, and R. Szeliski, A Bayesian approach to digital matting, Proceeding of the IEEE Computer Vision and Pattern Recognition, (CVPR), 2001, pp. 264–271.

R.C. Gonzalez and R.E. Woods, Digital image processing, Prentice-Hall, Englewood Cliffs, NJ, 2002.

P. Hillman and J. Hannah, Natural image matting, Proc. of Vision, Video, and Graphics, 2005, pp. 213–216.

P. Hillman, J. Hannah, and D. Renshaw, Alpha channel estimation in high resolution images and image sequences, Proceeding of the IEEE Computer Vision and Pattern Recognition (CVPR), 2001, pp. 1063–1068.

A. Isaksen, L. McMillan, and S.J. Gortler, Dynamically reparameterized light fields, Proceeding of ACM SIGGRAPH, 2000, pp. 297–306.

ISO/IEC JTC1/SC29/WG11 M11014, Utilization of inter-view correlation for multiple view video coding, 2004.

N. Joshi, W. Matusik, and S. Avidan, Natural video matting using camera arrays, Proceeding of ACM SIGGRAPH, 2006, pp. 779–786.

N. Joshi, W. Matusik, S. Avidan, and W.T. Freeman, Exploring defocus matting: Nonparametric acceleration, super-resolution, and off-center matting, Comput Graphics Appl, IEEE 27(2007), 43–52.

S.Y. Kwak, B.C. Ko, and H.R. Byun, Automatic salient-object extraction using the contrast map and salient points, Lect Notes Comput Sci 3332 (2004), 138–145.

A. Levin, D. Lischinski, and Y. Weiss, A closed-form solution to natural image matting, IEEE Trans Pattern Anal Mach Intell 30(2008), 228–242.

S.J. Lim, Y.Y. Jeong, and Y.S. Ho, Automatic liver segmentation for volume measurement in CT images, J Visual Commun Image Rep 17(2006), 860–875.

L. McMillan, An image-based approach to three-dimensional computer graphics, Ph.D's thesis, University of North Carolina at Chapel Hill, 1997.

T. Porter and T. Duff, Compositing digital images, Comput Graphics 18 (1984), 253–259.

M.A. Ruzon and C. Tomasi, Alpha estimation in natural images, Proceeding of the IEEE Computer Vision and Pattern Recognition (CVPR), 2000, pp. 18–25.

A.R. Smith and J.F. Blinn, Blue screen matting, SIGGRAPH, 1996, pp. 259–268.

J. Sun, J. Jia, C.K. Tang, and H.Y. Shum, Poisson matting, Proceeding of ACM SIGGRAPH, 2004, pp. 315–321.

V. Vaish, G. Garg, E. Talvala, E. Antunez, B. Wilburn, M. Horowitz, and M. Levoy, Synthetic aperture focusing using a shear-warp factorization of the viewing transform, Proceeding of the IEEE Computer Vision and Pattern Recognition (CVPR), 2005, pp. 129–129.

Y. Wexler, A. Fitzgibbon, and A. Zisserman, Bayesian estimation of layers from multiple images, Proceeding of European Conference on Computer Vision (ECCV), 2002, pp. 487–501.