GAZE CORRECTION FOR 3D TELE-IMMERSIVE COMMUNICATION SYSTEM

Wei Yong Eng[†]*, Dongbo Min*[†]*, Viet-Anh Nguyen*[†]*, Jiangbo Lu*[†]*, and Minh N. Do*[§]

Advanced Digital Sciences Center (ADSC), Singapore[†]
University of Illinois at Urbana-Champaign, IL, USA[§]

ABSTRACT

The lack of eye contact between participants in a tele-conferencing makes nonverbal communication unnatural and ineffective. A lot of research has focused on correcting the user gaze for a natural communication. Most of prior solutions require expensive and bulky hardware, or incorporate a complicated algorithm causing inefficiency and deployment. In this paper, we propose an effective and efficient gaze correction solution for a 3D tele-conferencing system in a single color/depth camera set-up. A raw depth map is first refined using the corresponding color image. Then, both color and depth data of the participant are accurately segmented. A novel view is synthesized in the location of the display screen which coincides with the user gaze. Stereoscopic views, i.e. virtual left and right images, can also be generated for 3D immersive conferencing, and are displayed in a 3D monitor with 3D virtual background scenes. Finally, to handle large hole regions that often occur in the view synthesized with a single color camera, we propose a simple yet robust hole filling technique that works in real-time. This novel inpainting method can effectively reconstruct missing parts of the synthesized image under various challenging situations. Our proposed system works in real-time on a single core CPU without requiring dedicated hardware, including data acquisition, post-processing, rendering, and so on.

Index Terms— Gaze correction, tele-conferencing, depth camera, foreground segmentation, depth image based rendering (DIBR)

1. INTRODUCTION

In a computer-based video conferencing system, a user usually sits at a certain distance in front of a screen. Since the user usually tends to look into the screen during tele-conferencing, the line-of-sight between the user and the color/depth camera does not coincide with the user's gaze, leading to lack of eye contact. To tackle such a well-known problem, a number of gaze correction methods have been proposed, e.g. by generating virtual view corresponding to the viewpoint of the user in the stereo camera set-up [1][2]. However, the unstable quality of stereo matching algorithms discourages the use of such stereo based gaze correction methods. In contrast, we propose a solution using minimal equipment requirement targeting home consumer with a single color/depth camera set-up.

Recently, rapid advances in active depth sensing technologies enable us to acquire a depth map at a video rate. These techniques further provide new opportunity to develop reliable tele-immersive conferencing system with the gaze correction capability [3]. In contrast to [3], which generates a novel view in the face region only, we propose a solution which renders a virtual view of the segmented foreground. Also, our solution provides the flexibility to generate a stereoscopic view for 3D display. However, in 3D tele-conferencing where stereoscopic virtual views are rendered with associated depth maps, there still exist challenges dealing with missing or inaccurate depth data due to inherent limitations of active depth sensors.

2. PROPOSED GAZE CORRECTION SOLUTION

To perform the gaze correction, we employ a single color-plus-depth camera often mounted on top of a display screen as illustrated in Fig. 1. Our goal is to provide a real-time solution effectively dealing with several algorithmic and implementation issues of 3D immersive tele-conferencing system with the gaze correction capability. We integrate this solution with the multi-party tele-conferencing framework, called ITEM (Immersive Telepresence for Entertainment and Meetings) [4]. In this paper, we focus on explaining our solution for gaze correction using single color-plus-depth camera.

2.1. Overview

In this system, a virtual camera with corrected gaze is assumed to be located in the center of a screen at the eye-level of a user so that the user looks into the virtual camera, as shown in Fig. 1. The camera system used in our work provides both color and depth data from RGB and infrared depth-sensing cameras. Initial raw depth maps at depth camera coordinate are first aligned to color images. The warped depth map is refined with the associated color image for achieving better segmentation and view rendering quality. Then, the segmented color image and its corresponding *refined* depth map are employed to synthesize a novel view corresponding to the virtual camera specified above. Finally, hole regions which often occur in the synthesized view are completed for seamless view generation.

In our system where a single color image is utilized, such hole completion (extrapolation) is very challenging, since there is no stereo cue available to fill the holes. Typical hole filling methods, designed to deal with general type of scenes, often employ computationally heavy iterative algorithms, making it intractable to be applied to a real-time system [5][6]. We propose an efficient hole completion method tailored to our communication system, especially under a real-time constraint. In our tele-conferencing scenario, the holes often occur when a foreground object is rendered in the virtual view, e.g. farther regions inside the foreground object. Note that the view rendering is performed for the segmented foreground. Such holes are filled by stretching texture information of neighboring visible pixels. Also, considering a relative location of the color camera and the screen (e.g. Fig. 1), we assume the visible pixels used in the filling process exist in the vertical line. Fig. 2 shows the overall scheme of the proposed system. The details of foreground extraction and refinement are further illustrated in the block diagram on the right hand side of Fig. 2.

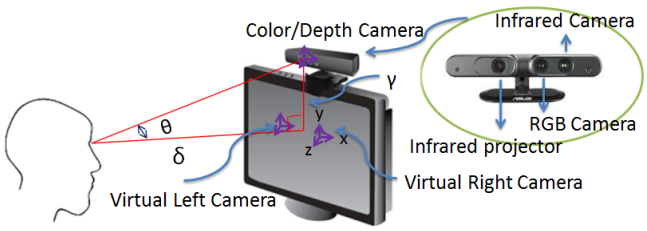

Fig. 1. Gaze correction system set up with a user, screen and depth camera.

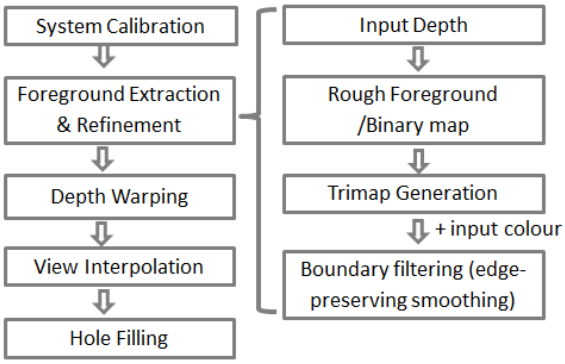

Fig. 2. Block diagram of the system set up

2.2. System Calibration

Our method uses the color and depth cameras (e.g. ASUS Xtion [7]), so calibration for two sensors are first performed. A built-in OpenNI function is then utilized to warp a raw depth map onto a color image coordinate. Next, we define a projection matrix of a virtual view from the virtual camera directly beneath the color camera in the set-up of Fig. 1. We assume that an intrinsic parameter of the virtual centre camera is the same as that of the color camera (K_c). A rotation matrix of the virtual centre camera with respect to the reference color coordinate is assumed to contain only a tilt angle (θ) with respect to the color camera. This angle θ is determined with two distance values:

a) distance between the virtual and color cameras, γ

b) distance between the virtual camera and the user, σ

$$\theta = tan^{-1}\left(\frac{\gamma}{\sigma}\right). \tag{1}$$

A translation vector (T_{cv}) also exists in a vertical direction (y) only, which is measured by γ. The virtual left and virtual right camera can be defined in the same way which includes an displacement in the horizontal axis, x. Thus, the extrinsic matrix E_v of the virtual camera is defined as

$$E_v = [R_{cv}|T_{cv}] = [R_{vc}^T| - R_{vc}^T T_{vc}] \tag{2}$$

$$R_{vc} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix}, \quad T_{vc} = \begin{bmatrix} 0 \\ -\gamma \\ 0 \end{bmatrix} \tag{3}$$

2.3. Foreground Segmentation on Color/Depth Images

First, color and depth images are aligned, as explained above. An initial foreground object is then obtained using a user tracking function of the OpenNI library, as shown in Fig. 3. This foreground binary map is then refined with the corresponding color image.

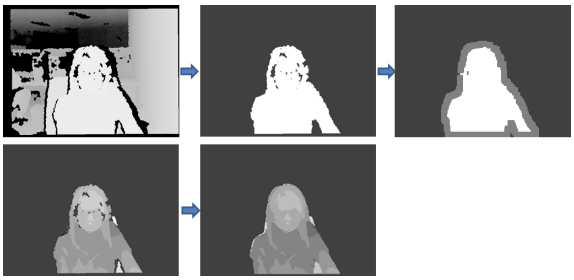

Fig. 3. Foreground segmentation and depth refinement are performed. Top: Input depth, initial foreground/binary map and trimap generation. Bottom: Initial depth map and refined depth image. Note that the contrast of the foreground depth map was enhanced for illustration purpose.

As the initial foreground object is not perfect especially along the depth boundary, a post-processing is required to handle these problematic foreground boundaries due to wrong or missing depth data. First, a trimap which consists of definite foreground, definite background and unknown region is created using the initial binary map, as illustrated in Fig. 3. The label of regions around the initial foreground boundary is assigned unknown region as the ownership of belonging to foreground or background of this region is uncertain. With the assumption that the boundaries of color image and depth map are likely to be co-located, the color image is utilized as a guidance image for joint image filtering. With color information, a joint image filtering is performed on the generated trimap. Specifically, efficient edge-aware smoothing filters [8][9] can be utilized to refine the foreground boundary on the trimap using a color cue. The smoothed result is thresholded to provide a final *refined* binary map. The refined foreground map is used to segment both color and depth images. Also, to handle inaccurate or missing parts in the segmented depth map, it is further filtered using the estimated binary map, as shown in Fig. 3.

Also, for a real-time processing, we perform the foreground extraction on low resolution color, depth and trimap images (a coarse-to-fine scheme). Then, the *refined* binary map is upsampled into original resolution with linear interpolation and adaptive refinement is performed around object boundaries only. More specifically, the refined binary map B_L on the low resolution is mapped into the grid of the original resolution, producing an boundary map B_I. Then, an adaptive voting is performed for each pixel p in a vicinity of the boundaries by using a color distance measured with the neighboring pixels in the pre-defined window $N(p)$ as

$$V(p,l) = \sum_{q \in N(p)} exp[-\sum_{c=r,g,b} (I_{c,p} - I_{c,q})^2 / 2\sigma^2] \delta(l - B_I(q)) \tag{4}$$

where $V(p,l)$ represents a likelihood belonging to a foreground ($l = 1$) or background ($l = 0$). $\delta(a)$ represents a delta function (1 for $a = 0$, 0 otherwise). Finally, a refined boundary map $B(p)$ on the original resolution is obtained as follows:

$$B(p) = \arg\max_{l} V(p,l). \tag{5}$$

The adaptive voting scheme used in the multiscale approach requires computing the voting function $V(p,l)$, whose complexity depends

on the window size N used, but it is relatively marginal since the refinement is performed on the object boundaries only.

2.4. Virtual View Synthesis

2.4.1. Depth warping

Given the segmented color and depth data, the virtual view is synthesized using a depth image based rendering (DIBR) technique [10]. The segmented depth map is first projected into 3D space and reprojected back onto the virtual camera coordinate system. A well-known pinhole camera model is assumed to project a pixel location $[u_c, v_c]^T$ of the color camera into the world coordinate $[x_c, y_c, z_c]^T$ and so on [11]. The obtained depth map $D(u_c, v_c)$, intrinsic K_c of the color camera are used as in (6).

$$[x_c, y_c, z_c]^T = K_c^{-1}[u_c, v_c, 1]^T D(u_c, v_c) \tag{6}$$

Then, the world coordinate $[x_c, y_c, z_c]^T$ is reprojected back into target coordinate of the virtual camera $[u_v, v_v, w_v]^T$ as in (7).

$$[u_v, v_v, w_v]^T = K_c R_{vc}^T \{[x_c, y_c, z_c]^T - T_{vc}\} \tag{7}$$

Lastly, the target coordinate $[u_v, v_v, w_v]^T$ is changed to homogeneous form $[u_v/w_v, v_v/w_v, 1]^T$ to get a pixel location in the virtual camera so that $\widehat{D}(u_v/w_v, v_v/w_v, 1) = D(u_c, v_c)$ where $\widehat{D}$ denote the warped depth in virtual camera. In the case of multiple depth data are warped into the same pixel location in virtual camera, the further depth data with respect to virtual camera is discarded as it is occluded by a closer object. In order to detect occlusion, the world coordinate $[x_c, y_c, z_c]^T$ which originally defines with reference to the color camera is transformed so that it is with respect to the virtual camera $[x_v, y_v, z_v]^T$ in a similar manner of 3D coordinate system transformation as in (8).

$$[x_v, y_v, z_v]^T = R_{vc}^T.([x_c, y_c, z_c]^T - T_{vc}) \tag{8}$$

2.4.2. View interpolation

Then, the warped depth $\widehat{D}(u_v, v_v)$ is used to project the pixel location for the RGB image in the virtual camera $[u_v, v_v]^T$ into the world coordinate with T_v, R_v, and K_v of the virtual camera as in (6). Next, the world coordinate $[x_c, y_c, z_c]^T$ is reprojected back into the color camera target coordinate $[u_c, v_c, w_c]^T$ with T_c, R_c, and K_c of color camera as in (7). As the obtained color camera pixel location which is acquired by homogeneous form transformation $[u_c/w_c, v_c/w_c, 1]^T$ might not fall onto an exact integer location, interpolation among neighboring pixels of the RGB image in color camera is performed.

2.5. Uniform Stretching for Hole Filling

By using a single color camera, large hole regions often appear in the virtual view when a view is visible in the virtual camera but not the color camera. Image completion technique [6] can be employed for filling the large hole, but the quality is not guaranteed as the foreground constraint is not taken into account. Even further, most image inpainting methods are too slow to be applied to our system under a real-time constraint. Most of the hole filling techniques [6][12] use fixed visible neighboring regions surrounding a hole regardless of the hole size. In contrast, our solution is based on a texture stretching method in which the size of the visible reference regions used is proportional to the hole size. This adaptive manipulation is useful

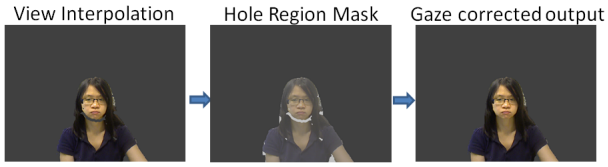

Fig. 4. Morphological closing operation is used to obtain the hole region within foreground.

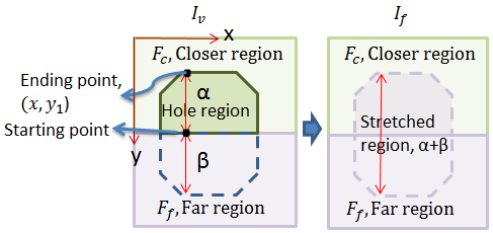

Fig. 5. Hole filling process with the texture from far region within the extracted foregound, F_f. (from bottom to top)

especially for a large hole region where using fixed (small) visible regions may be insufficient to infer the texture of the hole region.

In this paper, a simple yet robust hole filling technique is implemented, especially tailored to our tele-conferencing scenario, with the information from warped depth, texture image, and hole region mask indicating the regions to be filled. Note that the view synthesis is performed for the segmented object, so such a hole mask should be estimated and distinguished from non-segmented region as illustrated in Fig. 4. An underlying assumption for hole filling is that the texture of the hole is highly likely to be similar to that of neighboring farther regions inside the segmented object, F_f. (see white mask of Fig. 4.) First, the morphological closing operation is performed on the segmented binary map to obtain a hole region within *refined* foreground object, as illustrated in Fig. 4. Then, the hole region mask is scanned vertically to determine the starting and ending points for each vertical line of the hole region. The direction (e.g. from top or bottom) of filling a hole is determined by comparing two depth values on two points, and the point belonging to the closer region F_c is set to the ending point, (x, y_1) as shown in Fig. 5. The width of a hole line, α is defined as from the starting to ending point. Here, we explain only the case where the filling starts from bottom, as the opposite case is performed similarly.

A key observation is that the hole usually appears in the vertical direction due to the location geometry of the virtual and color cameras. This leads to the idea of hole filling along the vertical direction which is consistent during different challenging disocclusion, as will be in the experiments.

After estimating the filling direction and its width α for all the vertical lines, a far region line of length β is defined to be used for the filling process. The far region line is extended until it meets another hole or its length β is not larger than α. The entire stretched line $(\alpha + \beta)$ is defined as shown in the Fig. 5. Then, an texture of the far region line of length β is stretched in order to fill an entire stretched line. Hence, for every pixel $c \,\epsilon\, [0, \alpha+\beta]$ in the stretch line, a relative position ξ_f inside far region line β is defined as:

$$\xi_f = c\frac{\beta}{\alpha + \beta}. \tag{9}$$

Finally, the interpolated pixel of intensity I_v is obtained at c pixel

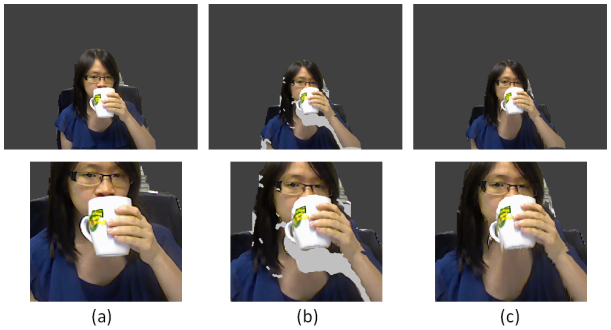

Fig. 6. Foreground segmentation, virtual view synthesis (with hole) and hole filling are performed: (a) Refined foreground object from streaming color and depth video of ASUS Xtion. Note the lack of eye contact due to the color camera direction not coinciding with the user gaze. (b) Virtual view generated in the virtual camera location. Note the hole (white mask) appears due to disocclusion. (c) Result image obtained with the proposed hole filling method.

location of the stretched line in (10).

$$I_f(x, y_1 + c) = I_v(x, y_1 + \alpha + \xi_f) \tag{10}$$

3. EXPERIMENTS

In the experiment, the tilt angle, θ is calculated as 15° and the translation in y-axis, γ is set as 15cm. The color and depth images are both of 640x480 pixels. During the foreground extraction, they are downsampled to half of its original resolution, 320x240 pixels. The boundary refinement on the processed upsampled color image is performed with a window size of 5x5 for each pixel in the vicinity of foreground boundary. In the boundary refinement process, σ is set to 30 and these parameters remain the same for all results.

We have tested our system for different scenarios including hand movement, occlusion, pose changes and changes in facial expressions. Although large holes appear due to the disocclusion, our proposed hole filling method works well and the rendered images shows visually consistent results. The current system runs at about 9 Hz (110 ms) on a computer containing an Intel Xeon 2.8-GHz CPU (using a single core) and a 6-GB RAM. It includes all processing for trimap generation (15 ms), non-linear edge-aware smoothing (48ms), virtual view synthesis (30ms), hole filling (17ms) from a single virtual camera.

In Fig. 6, usual hand movements during a video conference call often lead to disocclusion along the depth boundary where the proposed hole filling method is applied. It is shown that the proposed method works well regardless of the hole dimension, in real time. Also, a snapshot of stereoscopic rendered views with 3D background scene for 3D monitor is generated as in Fig. 7.

4. CONCLUSION

In this paper, we have presented a gaze correction technique which can be employed in 3D tele-conferencing systems. We have shown a real-time system which generates a novel view using only a single color-plus-depth setup. To tackle the hole-filling problem without the stereo cues, we proposed a robust hole-filling method which detects and fills the holes automatically, in real-time. The system is shown to work under various challenging situations consistently. It demonstrates the potential of a future 3D tele-conferencing system to the typical consumer level with its low cost equipment, simplicity and effectiveness. In further research, we plan to enhance the quality of the novel view generation by integrating the temporal cues.

Fig. 7. Stereoscopic rendered views with 3D background scene

5. REFERENCES

[1] R. Yang and Z. Zhang, "Eye gaze correction with stereovision for video-teleconferencing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 956–960, 2004.

[2] A. Criminisi, A. Blake, C. Rother, J. Shotton, and P. Torr, "Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming," *International Journal of Computer Vision*, vol. 71, no. 1, pp. 89–110, 2007.

[3] C. Kuster, T. Popa, J. Bazin, C. Gotsman, and M. Gross, "Gaze correction for home video conferencing," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 174, 2012.

[4] V. A. Nguyen, T. D. Vu, H. Yang, J. Lu, and M. N. Do, "ITEM: immersive telepresence for entertainment and meetings with commodity setup," in *ACM Multimedia*, 2012, pp. 1021–1024.

[5] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[6] K.-J. Oh, S. Yea, and Y.-S. Ho, "Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-D video," in *Picture Coding Symposium*, 2009, pp. 233–236.

[7] "Xtion pro live," http://www.asus.com/Multimedia/Xtion_PRO_LIVE/.

[8] E. Gastal and M. Oliveira, "Domain transform for edge-aware image and video processing," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4, p. 69, 2011.

[9] J. Lu, K. Shi, D. Min, L. Lin, and M. N. Do, "Cross-based local multipoint filtering," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012, pp. 430–437.

[10] C. Fehn, R. D. L. Barre, and S. Pastoor, "Interactive 3-dtv: Concepts and key technologies," *Proceedings of the IEEE*, vol. 94, no. 3, pp. 524–538, 2006.

[11] E. Martinian, A. Behrens, J. Xin, and A. Vetro, "View synthesis for multiview video compression," in *Picture Coding Symposium*, vol. 37, 2006, pp. 38–39.

[12] L.-M. Po, S. Zhang, X. Xu, and Y. Zhu, "A new multidirectional extrapolation hole-filling method for depth-image-based rendering," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 2589–2592.