Seeing Eye to Eye: Perspective Adjustment for Videoconferencing

Ben Cheng **Jessica Forde** **Hirotaka Miura**
Benjamin Rapaport **Frank Wood** **Xinyue Zhang**
Columbia University, New York, NY 10027, USA
bc2490@columbia.edu,jzf2101@columbia.edu,hm2528@columbia.edu,
bar2150@columbia.edu,fw2196@columbia.edu,xz2275@columbia.edu

Abstract

The quality of communication in videoconferencing is compromised by the fact that participants cannot maintain eye contact. We propose an original solution to this problem. Using color and depth information from a Kinect camera, we will use a Conditional Random Field to model each participant's eye position in 3D space, which will allow us to reorient the image to create the illusion of looking through a window. Finally, we will use a Fields of Experts model to fill in unobserved parts of the transformed image.

1 Introduction

The quality of communication in videoconferencing is compromised by the fact that participants cannot maintain eye contact. As each user naturally looks at his screen rather than his camera, both users appear to be looking away from one another. Proposed remedies to this problem include inferring a modified view using stereo cameras [2] and super-imposing images of eyes facing the viewer [15]. In this paper, we propose a new approach, which incorporates depth information available from the infrared sensor on a Kinect camera. This approach has already proved useful for human tracking [16], hand gesture recognition [8], and 3D image creation [11]. Using color and depth images, we will model the location of each participant in 3D space using a Conditional Random Field (CRF). The 3D location of the viewer will allow us to infer a new image of the scene displayed relative to his location. As a result, we will create a virtual "window" through which the users can face each other naturally and make direct eye contact. As a final post-processing step, we will use a Fields of Experts (FoE) model to denoise the image and fill in occluded areas [12].

2 Methods

2.1 Problem Description

An inherent problem of one-on-one web conferencing is the inability for users to make eye contact, due to the disparity between where the eyes are displayed, the monitor, and the eyes point of view, the camera. As a result, users tend to appear to one another as looking downwards towards their respective screens. Our approach simulates an environment in which the position and cameras of the two users are fully calibrated, such that each persons screen can be interpreted as a viewing window.

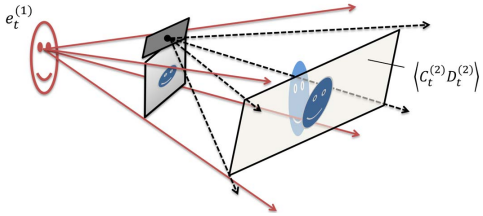

Figure 1a - Problem setup

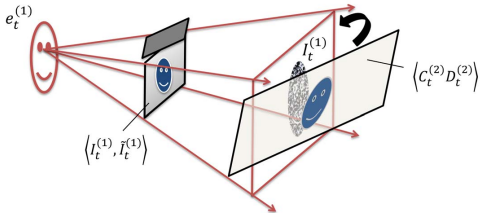

Figure 1b - Proposed solution

Figure 1a illustrates the problem setup. User 2's (blue) Kinect camera records both RGB and depth images $\langle C_t^{(2)}, D_t^{(2)} \rangle$. The orientation of user 2, as observed by user 1's (red) viewing frustum through the screen, is inconsistent with the image captured by $\langle C_t^{(2)}, D_t^{(2)} \rangle$. Figure 1b shows our proposed solution. We will infer the location of user 1, $e_t^{(1)}$, in 3D space. We will then apply a projective mapping from $\langle C_t^{(2)}, D_t^{(2)} \rangle$ to a plane characterized by user 1's viewing frustum to infer a new image, $I_t^{(1)}$. $I_t^{(1)}$ will contain empty pixels where information was unavailable due to occlusion in $\langle C_t^{(2)}, D_t^{(2)} \rangle$. We will then infer the missing pixels, $\tilde{I}_t^{(1)}$, using FoE, and effectively correct $\langle C_t^{(2)}, D_t^{(2)} \rangle$ to obtain $\langle I_t^{(1)}, \tilde{I}_t^{(1)} \rangle$.

2.2 Graphical Model

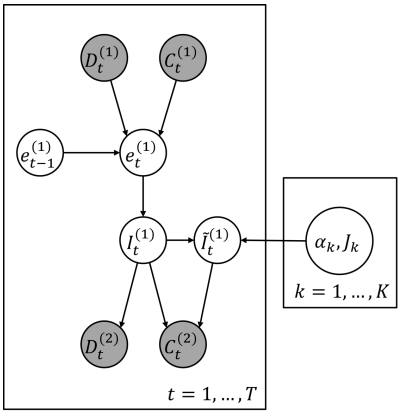

Figure 2: Graphical Model of User-dependent Image Inferences

Figure 2 describes the model we will use to infer the new images. The model can be divided into two parts. In part 1, we will use $\langle C_t^{(1)}, D_t^{(1)} \rangle$ and $e_{t-1}^{(1)}$ to infer $e_t^{(1)}$. We propose $e_t^{(1)} | e_{t-1}^{(1)} \sim N(\mu, \Sigma)$. Part 1, $P(e_t^{(1)}, e_{t-1}^{(1)}, C_t^{(1)}, D_t^{(1)}) = \prod_{i=1}^{T} P(e_t^{(1)} | C_t^{(1)}, D_t^{(1)}) P(e_t^{(1)} | e_{t-1}^{(1)})$, is therefore a CRF. Part 2 infers $\langle I_t^{(1)}, \tilde{I}_t^{(1)} \rangle$ relative to $e_t^{(1)}$. $e_t^{(1)}$ allows us to calculate the angle for which we must rotate $\langle C_t^{(2)}, D_t^{(2)} \rangle$ to infer $I_t^{(1)}$. Using our FoE prior, J_k, α_k, $C_t^{(2)}$, and $I_t^{(1)}$, we will infer the values of $\tilde{I}_t^{(1)}$. As a result, we can write the joint distribution for parts 1 and 2 as

$$
\begin{aligned}
&P(C_t^{(1)}, D_t^{(1)}, e_t^{(1)} C_t^{(2)}, D_t^{(2)}, I_t^{(1)}, e_{t-1}^{(1)}, \tilde{I}_t^{\{1)}, \alpha_k, J_k) \\
&= \prod_{i=1}^{T} [P(e_t^{(1)} | C_t^{(1)}, D_t^{(1)}) P(I_t^{(1)} | e_t^{(1)}) P(D_t^{(2)} | I_t^{(1)}) P(C_t^{(2)} | I_t^{(1)}, \tilde{I}_t^{(1)}) P(e_t^{(1)} | e_{t-1}^{(1)}) P(\tilde{I}_t^{(1)} | I_t^{(1)})] \\
&\quad \prod_{k=1}^{K} [P(\tilde{I}_t^{1} | \alpha_k, J_k) P(\alpha_k, J_k)]
\end{aligned}
$$

We will repeat this process to create the image seen by user 2, $\langle I_t^{(2)}, \tilde{I}_t^{(2)} \rangle$, exchanging the variables belonging to user 1 with the variables belonging to user 2 and vice versa.

2.3 Eye Tracking

To detect and track the eyes in 3D space through successive video frames, we model the relationship between the eye position and the RGB and depth images as the CRF described in the graphical model. In order to reduce the initial search space for the user's eyes, we will implement the Viola Jones object detection algorithm to isolate the face of the user [14]. We will train the model using RGB-only videos of moving heads from the Boston University Head Tracking database [1] with ground truth eye locations provided by the UvAEyes annotations [13]. Once the eyes have been located within the RGB image, we will integrate the depth information from the Kinect to estimate the eyes' locations in 3D. Other potential approaches can be found in the appendix.

2.4 Rotation and Projection of Images into 2D

Using the methods described in Herrera et al. [6] to calibrate the RGB and depth images, we will map each pixel to a 3D coordinate to obtain a 3D image of the scene. Using the camera transform [9], in conjunction with calibration measurements taken of the physical layout of the camera and monitor, we will project those 3D coordinates onto a 2D plane as seen from the shifted point of view. An initial milestone will be to rotate 3D objects in MATLAB and project them into 2D.

2.5 Fields of Experts

FoE is described as follows. A neighborhood system is a rectangular region of dimension m that connects all nodes within the area. In this case, the nodes are contiguous sets of pixels in an image. Every neighborhood centered on a node k, $k \in [1:K]$, defines a maximal clique $x_{(k)}$. FoE extends Markov Random Fields to higher order using a neighborhood system that connects all nodes in the entire image. An Expert refers to a probabilistic model, ϕ, defined over a linear one-dimensional subspace or direction. The FoE model is defined as

$$P_{FoE}(x, \Theta) = \frac{1}{Z(\Theta)} \prod_{k=1}^{K} \prod_{i=1}^{N} \phi(J_i^T x_{(k)}, \alpha_i)$$

where J_i is the linear filter that defines the direction modeled by ϕ, α_i is a parameter greater than zero, $\Theta = \{J_i, \alpha_i | i = 1, ..., N\}$ is the set of the model parameters, and Z is the partition function. The parameters, Θ, are learned by maximizing their likelihood. Since no closed form solution exists, we perform an approximation using contrastive divergence by running N samplers in conjunction with MCMC sampling over a number of iterations to reach the target distribution [7].

2.6 Data Sources

We will use a number of data sets to train and test our model. We will utilize Boston University's Head Tracking database [1], which contains videos of individuals in various head poses. The database hosts 45 videos taken under uniform lighting, each roughly 7 seconds each. These videos will be used in conjunction with the UvAEyes [13] eye annotations to effectively train our model to track eye locations in streams of video. To approximate the modified image of the users, we will use data collected by the Computer Vision Laboratory at ETH Zurich. The dataset contains Kinect video recordings of people in head poses ranging from -75° to 75° yaw and -60° to 60° pitch. There are 15,000 frames, each containing a depth image and an RGB image [4]. Additionally, we will place the Kinect for Windows cameras on opposing sides of a room on two identical monitors and record conversations between two individuals communicating as if by videoconferencing. We will use this data to generate the modified images based on the eye position.

3 Expected Conclusions

At the conclusion of this project, we expect to develop a model that infers the perspective of user 1 looking into the space of user 2 and its corresponding image. Applications, however, are dependent on developments that allow for improved computational speed. With the dissemination of cameras that include technology similar to the Kinect, we believe that the results of this project will be useful in allowing users to interact more naturally via webcam.

References

[1] Marco La Cascia, Stan Sclaroff, and Vassilis Athitsos. Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3d models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):322–336, 2000. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=845375.

[2] A. Criminisi, J. Shotton, A. Blake, C. Rother, and P. H. S. Torr. Efficient dense-stereo and novel-view synthesis for gaze manipulation in one-to-one teleconferencing. Technical report, 2003.

[3] M. Everingham and A. Zisserman. Regression and classification approaches to eye localization in face images. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 441 –446, april 2006. doi: 10.1109/FGR.2006.90.

[4] G. Fanelli, T. Weise, J. Gall, and L. Van Gool. Real time head pose estimation from consumer depth cameras. In *33rd Annual Symposium of the German Association for Pattern Recognition (DAGM'11)*, September 2011.

[5] Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):478–500, March 2010.

[6] C. Daniel Herrera, Juho Kannala, and Janne Heikkilä. Accurate and practical calibration of a depth and color camera pair. In *Proceedings of the 14th international conference on Computer analysis of images and patterns - Volume Part II*, CAIP'11, pages 437–445, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-23677-8. URL http://dl.acm.org/citation.cfm?id=2044575.2044638.

[7] Geoffrey Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14, 2002.

[8] Nikolaos Kyriazis Iason Oikonomidis and Antonis Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *Proceedings of the British Machine Vision Conference*, pages 101.1–101.11. BMVA Press, 2011. ISBN 1-901725-43-X. http://dx.doi.org/10.5244/C.25.101.

[9] Kenneth I. Joy. The camera transform, 1999. URL http://www.idav.ucdavis.edu/education/GraphicsNotes.

[10] Shinjiro Kawato and Jun Ohya. Detection and tracking of eyes for gaze-camera control, 2002.

[11] Oliver Kreylos. Kinect hacking, 2010. URL http://idav.ucdavis.edu/~okreylos/ResDev/Kinect/.

[12] Stefan Roth and Michael J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2): 205–229, April 2009.

[13] R. Valenti, Z. Yucel, and T. Gevers. Robustifying eye center localization by head pose cues. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:612–618, 2009. doi: http://doi.ieeecomputersociety.org/10.1109/CVPRW.2009.5206640.

[14] Paul Viola and Michael Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

[15] Lior Wolf, Ziv Freund, and Shai Avidan. An eye for an eye: A single camera gaze-replacement method. Technical report, The Blavatnik School of Computer Science and Department of Electrical Engineering-Systems, Tel-Aviv University, 2010.

[16] Lu Xia, Chia-Chih Chen, and J. K. Aggarwal. Human detection using depth information by kinect. Technical report, Department of Electrical and Computer Engineering, University of Texas at Austin, 2012.

[17] Alan L. Yuille, Peter W. Hallinan, and David S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.

4 Appendix

4.1 Equipment

The Kinect for Windows hardware includes an RGB camera, an infrared (IR) camera, and an IR light source. The RGB camera returns a stream of standard RGB images. The IR light source projects dots of IR light throughout the room, which can only be seen by the IR camera. The IR camera, in turn, returns a matrix of depth measurements for each pixel, which can be converted to real-world depth values. In order to map these depth values onto the RGB pixels, we must calibrate for the physical distance between the two cameras. Once mapped, we can use the combined data to map each RBG/depth pixel to coordinates in 3D space, which allows us to create a 3D image.

As opposed to the standard Kinect, the new Kinect for Windows device provides more accurate depth measurements at close range. This will be beneficial for the problem domain of video conferencing. The Kinect for Windows will be available February 1, 2012. We would like to pre-order the cameras in anticipation of the release date The retail price for each is $250. We also require two identical monitors to simulate a video conferencing environment. We are looking into used monitors for around $60 each. Thus, the total estimated cost of our project, which includes the price of the two Kinect cameras and two monitors is $620.

4.2 Eye Tracking Literature Review

A number of potential methods have been proposed for locating the observer's eyes based on images of the observer. Hansen and Ji [5] categorizes methods of eye detection into several general approaches.

Shape-based approaches use prior geometric models of the human eye, as well as a similarity measure, to map a region of the image to the prior model. Some shape-based methods use a simple ellipse as a prior eye model. Other shape-based approaches use more complex shapes that are more flexible. For example, Yuille et al. [17] uses a deformable eye model, which is characterized by a set of learned parameters that describe a prior model of the eye. These parameters are adjusted based on the image and a similarity function to account for changes in shape, scale, orientation and other factors. Deformable shape methods such as this have the benefit of detecting eyes under varying conditions. However, they are often limited due to computational complexity and restrictive image requirements, and are generally ill-suited for handling eye occlusions.

Feature-based methods attempt to distinguish features either surrounding the eye (e.g. eye corners, eyebrows, nose), or within the eye (e.g. pupil, iris, sclera, eye glint), as landmarks for locating the eyes themselves. For example, Kawato and Ohya [10] determines the area between the eyes by recognizing that regions to the left and right (eyebrows and eyes) are darker and regions above and below (forehead and nose) are brighter. Once the center between the eyes is identified, the location of the eyes are estimated based on the geometric relationship between the center and the eyes. Methods such as these are potentially advantageous when the chosen features are less sensitive to different lighting and orientation conditions.

Appearance-based methods attempt to identify the eye by its color/intensity distribution based on large sets of training data. In contrast to shape and feature-based methods, the eye is not explicitly modeled, but rather parameters describing the color distributions of an eye are learned. Due to its generality, these methods can be used to locate a variety of object classes. For example, the Viola and Jones [14] object-detection framework has been widely implemented to detect faces. This method has been extended in conjunction with a number of other approaches to locate the eyes within a face. Everingham and Zisserman [3], for example, found a simple Bayesian approach which determines the log likelihood ratio between the probabilities of eye and non-eye appearance classifications of image patches to be effective in eye detection, outperforming comparable methods such as regression and discriminative eye detector approaches.

Many state-of-the-art eye tracking techniques utilize various active infrared (IR) illumination schemes to detect eyes via eye-glint and dark pupil/bright pupil effects. However, the Kinect's depth measurements rely on perceiving irregular dotted patterns of emitted IR light, which precludes the use of these methods, which require consistent IR reflection by the eyes.