GAZE CORRECTION WITH A SINGLE WEBCAM

Dominik Giger[1]*, Jean-Charles Bazin*[1]*, Claudia Kuster*[1]*, Tiberiu Popa*[2] *and Markus Gross*[1]

[1]Department of Computer Science, ETH Zurich, Switzerland
[2]Department of Computer Science and Software Engineering, Concordia University, Canada

ABSTRACT

Eye contact is a critical aspect of human communication. However, when talking over a video conferencing system, such as Skype, it is not possible for users to have eye contact when looking at the conversation partner's face displayed on the screen. This is due to the location disparity between the video conferencing window and the camera. This issue has been tackled by expensive high-end systems or hybrid depth+color cameras, but such equipment is still largely unavailable at the consumer level and on platforms such as laptops or tablets. In contrast, we propose a gaze correction method that needs just a single webcam. We apply recent shape deformation techniques to generate a 3D face model that matches the user's face. We then render a gaze-corrected version of this face model and seamlessly insert it into the original image. Experiments on real data and various platforms confirm the validity of the approach and demonstrate that the visual quality of our results is at least equivalent to those obtained by state-of-the-art methods requiring additional equipment.

Index Terms— Gaze correction, video conferencing

1. INTRODUCTION

With the wide availability of broadband Internet, video conferencing is becoming more and more popular both for professional and private use, and gradually replacing traditional audio calls. However when talking over a traditional video conferencing system such as Skype or Apple's FaceTime, conversation partners do not have eye contact. Concretely, when the camera is at the top of the screen, the users have the impression that the conversation partner is looking down. Beyond the pure aesthetic aspect, gaze awareness (eye contact) is a key and indispensable aspect of human communication [1, 2, 3] and thus it is important to be preserved when communicating over an electronic link. The problem of missing eye contact is simply due to the location disparity between the video conferencing window and the camera. Eye contact could be achieved by placing a camera right behind the video window, but this would require a transparent screen or a special mechanism involving mirrors. This has been addressed at the high-end level of teleconferencing systems using special hardware [4, 5, 6, 7] but has not yet been convincingly solved at the consumer level.

In this work, we present a practical gaze correction system that relies on only a single camera, and thus can be used on a variety of platforms ranging from desktop computers to laptops and from professional video-conferencing systems to tablets. In the absence of any geometric information and with only one view available, our method fits a generic template to the image in real-time, preserving the facial expression of the participant, and uses this geometric proxy to synthesize a gaze corrected version of the head that is then transferred seamlessly into the original image.

2. RELATED WORK

As many have noted [8, 2, 9, 10, 6, 11] the gaze correction problem can be cast as a novel view synthesis problem: re-render the scene from a virtual view point along the expected viewing path. Some high-end teleconferencing systems achieve this using expensive and cumbersome physical devices that create the illusion that the camera is along the viewing path [4, 5, 6, 7]. Other systems employ several cameras and render the scene from a virtual camera location by view interpolation [12, 13, 14, 15, 16]. Unfortunately, these solutions are too expensive and cumbersome for the consumer level. The release of inexpensive hybrid (depth+color) cameras, such as the Kinect, recently triggered the first convincing gaze correction method using a single off-the-shelf device [11]. However, despite the increasing popularity of hybrid cameras, this market segment still represents a tiny fraction of the communication devices equipped with standard webcams, such as desktop computers, laptops and tablets. Therefore a gaze correction system using a single video camera is highly desirable.

A few methods requiring only one color camera have been developed to perform gaze correction. Some of them operate purely in image space [17], trying to find an optimal warp of the image. However, only small corrections are possible since it is difficult to compute an appropriate warp without prior knowledge of the face geometry. Therefore, some methods use a proxy geometry to approximate or estimate the face shape. For example, a family of methods [18, 9, 19] start with a geometric face template and then deform it consistently with the facial features that can be extracted in the image by a face tracker. These methods can return a visually appealing textured 3D face model of the user. Then, the model can be rotated

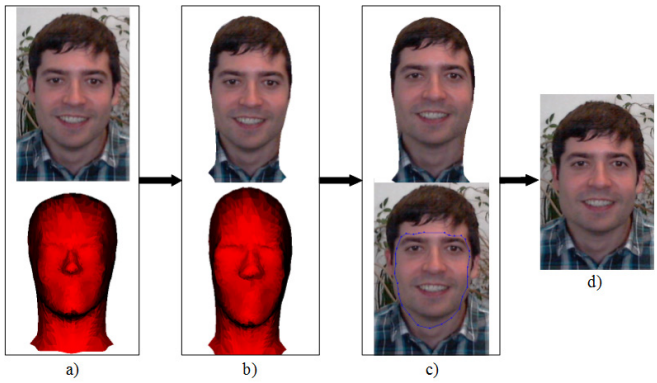

Fig. 1. Our pipeline: a) Input: color image acquired by a single camera and a generic 3D head mesh, b) deformation of the head mesh according to the extracted facial features. Bottom: deformed head mesh. Top: its textured version. c) gaze correction. Top: gaze-corrected rendering of the textured deformed head mesh. Bottom: transfer into the original image with seam optimization (blue curve), d) final result.

to achieve eye contact. However they consider only the inner part of the face, which can lead to unwanted modifications of the face proportions (as will be seen later), and/or render only the gaze-corrected face without the background scene. In contrast, our method considers the entire head, which better preserves the face proportions, and outputs the gaze-corrected view along with the original background of the scene.

Another category of methods to obtain a 3D face model is to apply Principal Component Analysis (PCA) on a set of pre-scanned face models, and then compute the coefficients that match the input 2D image. The reconstructed face model can be further optimized with deformation techniques [20, 21]. Impressive results can be obtained, however these methods generally cannot run in real-time, which is a key requirement for video-conferencing.

3. PROPOSED APPROACH

Similarly to Kuster et al. [11], our method follows a multi-perspective rendering approach: it synthesizes a gaze corrected version of the face and then transfers it into the input image. However, our method does not require the geometry from a depth camera (i.e. works with a single webcam), better preserves the proportions of the face and is more robust to the location of the camera (e.g. top of the screen). Our method uses a generic head mesh fitted closely to the facial features extracted from the input image. The mesh is textured and rotated to correct the gaze. The resulting image is inserted seamlessly into the original image. Figure 1 illustrates our pipeline and the following sections describe the method in more detail.

3.1. Face template fitting

We employ a smooth generic 3D head mesh composed of face and neck (see Figure 1-a) and deform it in every frame to fit the video of the participant. The deformation follows the facial feature points extracted by a state-of-the-art live face tracker [22] in the input image. An initial correspondence between the feature points and the 3D mesh vertices is manually performed only once. This correspondence is used for all the sequences (i.e. all the persons and devices) shown throughout the paper. Since the face tracker captures facial features in image space and the mesh is in world space, it is necessary to convert between the two coordinate systems. The image is embedded in 3D space as a rectangle aligned with the xy-plane. The mesh is then deformed to match the tracked facial features. Although many advanced algorithms for mesh deformation exist, we chose the simple, but fast Laplacian deformation technique [23], as real-time performance is critical for a video-conferencing application. In general, the solution of the Laplacian deformation can be obtained by solving a linear system. However, since in our case the constraint points never change, it is possible to precompute the matrix decomposition and find the solution at every frame via simple back substitution, which is computationally efficient. The downside is that the Laplacian deformation is neither rotational nor scale invariant. Therefore, the global scale and rotation of the mesh have to be computed first, as explained in the following.

The width of the head in the input image is estimated from the tracked facial features that represent the sides of the head. The mesh is then scaled such that the vertices matched to these facial features correspond with the input image embedded in 3D space. This scaling is applied for every frame, as from the perspective of the camera the size of the head may change as the user moves closer or further away from the camera. The global orientation of the face is obtained directly from the face tracker. Once the mesh has been correctly scaled and oriented, we deform it to closely match the facial features of the user. The Laplacian deformation is performed using the tracked points as constraints along the x and y directions. The z-coordinates of the template mesh vertices remain unchanged. Figure 2 shows a typical example of a deformed face mesh and its registration (alignment, scale and rotation) with respect to the input color image. Note that we do not aim to compute a perfectly faithful face model, but rather obtain a facial geometry (i) in real-time and (ii) whose accuracy is reasonable enough to provide a convincing eye contact.

3.2. Occlusion and Texture Stretching

After the template mesh has been deformed, the input image is projected onto the deformed mesh for texturing. However, this procedure may lead to stretched or missing texture in occluded areas. Especially if the camera is placed at the top of the screen, a large part of the neck is not visible from the camera's perspective and thus not textured. To address this,

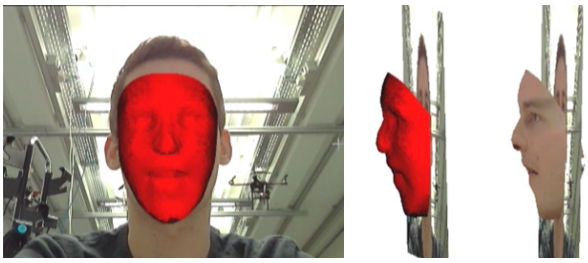

Fig. 2. Left: registration of the deformed face mesh with respect to the input image. Middle: tilted view to show the geometry. Right: same view with the texture.

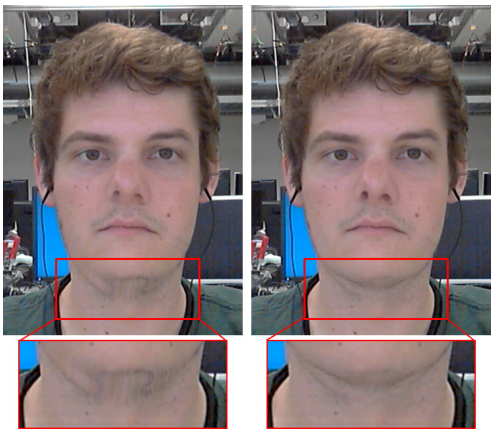

Fig. 3. Left: due to occlusion of the neck when the camera is above the head (typically at the top of the screen), the neck texture of the face mesh is highly stretched. Right: our result.

we parameterize the template in the 2D domain and create a complete, albeit static texture of the user's face with the correct gaze direction, that we can use for the occluded vertices. This is performed at the beginning of the session when the user is asked to look straight at the camera for just a brief instant. The vertices at the border of the live texture are blended with the static texture to avoid any noticeable seam. A representative example is shown in Figure 3.

3.3. Seam optimization

To correct the gaze direction, the textured deformed head mesh now needs to be rotated to achieve the feeling of eye contact. This rotation is automatically computed by having the user looking at the video conferencing window and at the camera, and then estimating the 3D rotation that aligns the 3D points corresponding to the facial features extracted in the two views projected onto the face geometry.

To seamlessly transfer the resulting gaze-corrected version of the textured deformed face mesh into the original image, the least visually noticeable seam must be computed. Our approach for seam optimization is inspired by [11] with an important change (to be detailed later). Essentially, the optimized seam algorithm (Fig 1-c-bottom) provides a polygonal shape whose boundary is as similar as possible to the source image (Fig 1-a-top) and the gaze-corrected view (Fig 1-c-top, i.e. after the rotation correction). The similarity measure is the sum of the intensity differences in the two images along the seam. To reduce the potential visual discontinuities, the two images are blended together along the seam, which provides the final gaze corrected result (Fig 1-d).

The method of Kuster et al. limits the seam optimization to the upper face while fixing the lower feature points to the chin. On one hand, this avoids the occlusion problem in the neck region, and additionally the lower part of the chin does not have to be blended as it is aligned by construction to the natural depth discontinuity of the chin. However, on the other hand this has the unintended consequence of slightly changing the proportions of the face. Although this is a small geometric aberration, it can be clearly noticeable since the human visual system is very sensitive to faces. This was confirmed by most of our subjects, especially when the face of the participant is familiar. This phenomenon is illustrated in Figure 5. To avoid this shortcoming, our approach allows the optimization process to place the lower part of the seam anywhere below the chin (see Figure 1-c). This is possible because our template mesh also includes the geometry of the neck. The occlusion problems that emerge are addressed using the method presented in Section 3.2. In practice, to ensure that the lower part of the computed seam is below the chin, we initialize it at the extracted facial feature points and only let it move away from the center of the face.

3.4. Discussion

Our gaze correction system provides an effective and practical solution to an important and challenging problem. Our contributions include the combination and efficient implementation of several non-trivial algorithms to build an efficient real-time gaze correction system using a single webcam. The real-time constraint is of key importance for live video conferencing, and most previous work on realistic face manipulation does not fulfill this requirement. Extensive experiments available in the next section show that our results are convincing both in terms of eye contact and scene/background preservation, and favorably compare to state-of-the-art methods, without the requirement of any particular hardware.

4. RESULTS

Our system is fully automatic and runs in real-time, namely at 25fps for 800x600 input videos and 30fps for 640x480 input videos on a standard consumer computer equipped with a CPU Intel Core i7 2.93GHz, 8GB RAM and a NVIDIA GeForce GTX 260. About 45% of the execution time is spent for the facial feature extraction, and the processing delay is less than 30ms, which is unnoticeable. Note that we do not aim to impose continuous eye contact: similarly to real-life face-to-face communication, our system provides eye contact only

Fig. 4. Left: original image from the color camera (i.e. without correct gaze). Middle: gaze correction results obtained by the Kinect-based approach of Kuster et al. [11], using both the color image and the depth map. Right: results obtained by our approach, using only the color image.

when the user is looking at the communication partner, i.e. at the video conferencing window.

To evaluate the validity of our approach we conducted several experiments. First, we compared our approach to the recent state-of-the-art method [11]. This system uses a depth map and a color image acquired by a hybrid camera (Kinect) to correct the gaze. To apply our method, we simply considered the color images. A comparison is available in Figures 4 and 5 and in the accompanying video. The quality of our output is similar to [11] in terms of eye contact and visual appeal, with the advantage that no hybrid camera is needed.

One might observe in Figure 5 that the results obtained by the two methods can look different for certain regions of the face, especially the shape of the chin. Therefore we conducted a second set of experiments in which we captured several sequences as well as a ground truth view. For this, we created an experimental setup composed of two cameras, one at the top of the screen and one in the middle of the screen, which we refer to as the ground truth camera. We asked the user to look at the ground truth camera while talking. To compare against [11], depth information must be available. Therefore we used two Kinects for our setup, but using only the color stream for our algorithm. Results are shown in Figure 5. In the top row, the lower and upper limits of the head are displayed. Since [11] modify only the inner part of the face, the length of the head is the same in the original image and in their results, while not matching the ground truth view. In contrast, the head length in our results correctly corresponds to the ground truth view. In addition, the bottom row analyzes the shape of the chin. For the same reasons as above, the chin of [11] exactly corresponds to the chin of the original image, but not to the ground truth view. The shape of the chin obtained by our method is almost indistinguishable from the ground truth view. Thus, this experiment indicates that our method better preserves the proportion of the user's face. This is possible because our complete 3D head mesh as well

Fig. 5. Comparison with ground truth view (2 sequences). a) original color image, b) result obtained by [11], c) result obtained by our approach and d) ground truth view. Top row: emphasize the lower and upper limits of the head. Bottom row: emphasize the chin with close up views overlayed in d) for easier comparison. The head limits and the chin shape are shown in green for the original image and the result of [11], cyan for our results and red for the ground truth views.

as our seam incorporate both the chin and the neck regions. Sequences showing this important improvement are available in the supplementary video.

We conducted a preliminary user study to quantitatively measure our results. We uniformly sampled 50 images from 10 different video sequences, and prepared the results obtained by [11], our method and the ground truth camera. We asked Amazon Mechanical Turk users to grade the quality of the facial appearance between 1 for "very distorted" and 5 for "very natural (unmodified)". Each image was graded by 6 participants, and the total number of participants was 103. The average grades for [11], our method and the ground truth camera are respectively 3.46, 3.58 and 3.87. It indicates that our method slightly outperforms the state-of-the-art method, without the need of RGBD cameras.

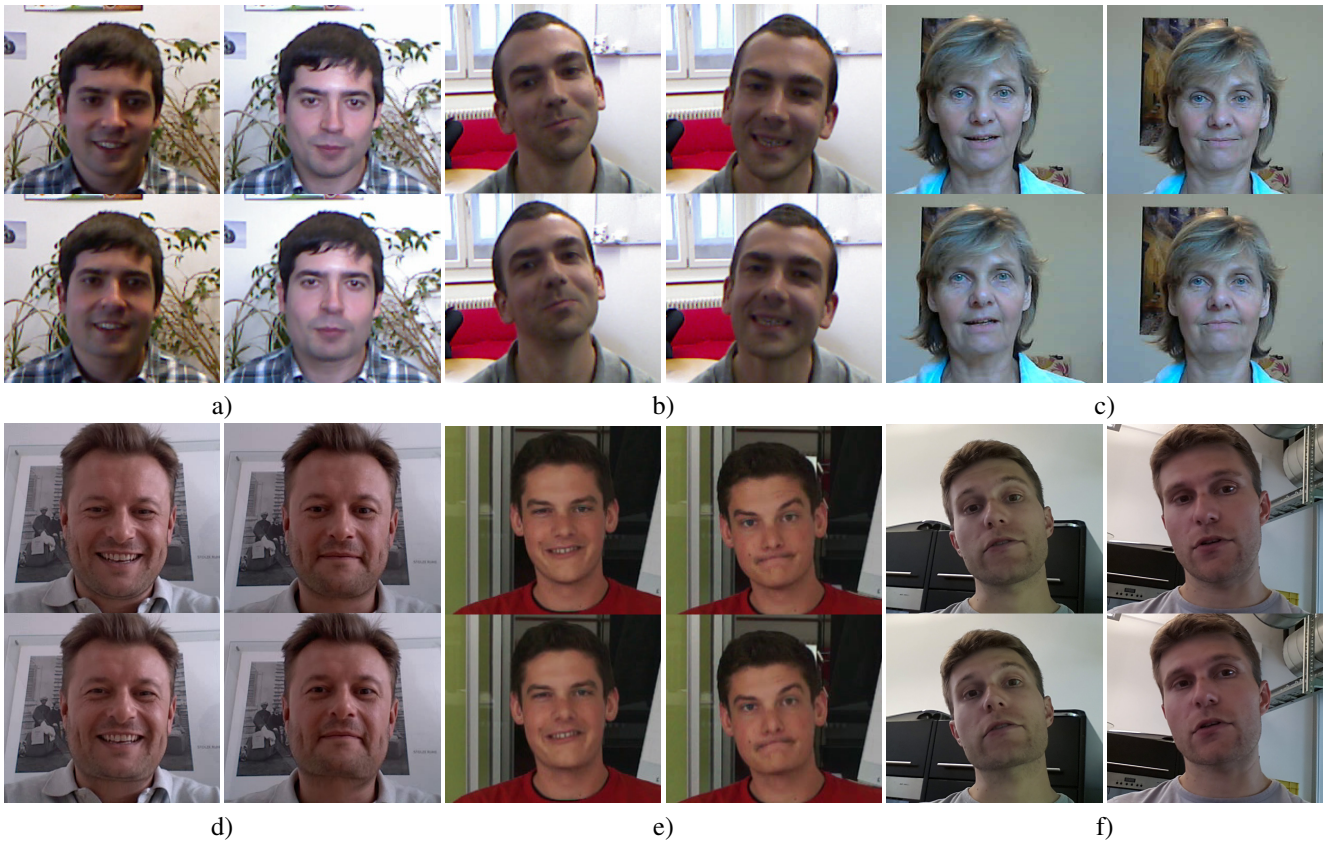

Fig. 6. A representative selection from our results. Top rows: original images from a real camera. Bottom: gaze corrected images obtained with our system. Please refer to the accompanying video for the complete sequences and additional results.

In further experiments we tested the temporal consistency and robustness of our system. We applied our approach to all the sequences processed in [11] and experiments have shown that our system has an equivalent level of robustness with respect to accessories (headphones, earrings, light beard and hats) and user's motion (standing up, sideway motions, going back and forth). Some of the sequences are included in the supplementary video. Figure 6 shows frames issued from several additional sequences. In the sequence of Figure 6-a, the lights of the room have been turned on and off several times. Note for example, the modifications of the skin color and shadows on the face in the input video. The outputs show that our method is robust against such sudden lighting changes. Figure 6-b illustrates how our system handles top-down and partial left-right rotations of the user's head while talking. The sequence of Figure 6-c has been acquired on a desktop computer equipped with a standard external webcam located at the top of the screen, showing that our approach can be applied for standard home video conferencing. The sequence of Figure 6-d has been acquired with the built-in webcam of a laptop. The user was close to the screen and this camera has a limited field of view so that the user's face is covering almost the entire image (the original non-cropped views are available in the accompanying video). This shows that our method also works at close range ($<$ 50cm), where a Kinect (as used in [11]) would not provide geometric data. On the other end of the range spectrum, Figure 6-e was acquired on a professional Cisco MX300 teleconferencing system where the user was seated in front of an office desk about 1.5 meters away from the 55-inch display. The sequence of Figure 6-f was acquired on a tablet (Samsung Galaxy Note 10.1) hand-held by a user walking in a room. One may note the large changes of the background. It shows that our approach can also be applied in the context of mobile devices.

Limitations and Future Work. Our current system is not adapted for users wearing glasses. The glasses are captured by the webcam, but not represented in the geometric head template. Therefore they become distorted when correcting the user's gaze. A possible solution would be to add a 3D model of the glasses to the head template or to treat the glasses and the head with independent geometric templates.

To detect the facial features our system is dependent on the robustness of the face tracker. In case the face tracker fails (e.g. when the face is occluded by more than 30% or is rotated close to sideways), the system cannot properly correct the gaze. In these situations we revert to the original, uncorrected video stream. When the face is picked up again by the tracker, the correction is automatically reapplied. To avoid sudden transitions, we smoothly blend the transition between the corrected

and the uncorrected output, via rotation interpolation over a few frames (typically about 5 frames).

All the sequences shown in this paper have been processed on a desktop computer. However, in a future version we would like to port our application to run directly on a variety of devices such as smart TVs, tablets and smartphones.

5. CONCLUSION

In this paper we demonstrate a practical, user-friendly solution for a very common problem in video conferencing. Our real-time gaze correction system is robust, computationally inexpensive and provides visually appealing and convincing results while relying only on a single webcam, making it available on a wide range of computing devices, such as desktop computer, laptop, tablet or smartphone. As a consumer level product our software could potentially enhance online communication for millions of users around the world.

Acknowledgement. This research, which has been partially carried out at BeingThere Centre, is in part supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

6. REFERENCES

[1] M. Argyle and M. Cook, *Gaze and mutual gaze*, Cambridge University Press, 1976.

[2] M. Chen, "Leveraging the asymmetric sensitivity of eye contact for videoconference," in *CHI*, 2002.

[3] C. N. Macrae, B. Hood, A. B. Milne, A. C. Rowe, and M. F. Mason, "Are you looking at me? eye gaze and person perception," in *Psychological Science*, 2002.

[4] H. Ishii and M. Kobayashi, "Clearboard: a seamless medium for shared drawing and conversation with eye contact," in *CHI*, 1992.

[5] K. Okada, F. Maeda, Y. Ichikawaa, and Y. Matsushita, "Multiparty videoconferencing at virtual social distance: Majic design," in *Proc. Conference on Computer supported cooperative work (CSW)*, 1994.

[6] A. Jones, M. Lang, G. Fyffe, X. Yu, J. Busch, I. McDowall, M. Bolas, and P. Debevec, "Achieving eye contact in a one-to-many 3D video teleconferencing system," in *SIGGRAPH*, 2009.

[7] D. Nguyen and J. Canny, "Multiview: spatially faithful group video conferencing," in *CHI*, 2005.

[8] A. F. Monk and C. Gale, "A look is worth a thousand words: Full gaze awareness in video-mediated conversation," *Discourse Processes*, 2002.

[9] J. Gemmell and D. Zhu, "Implementing gaze-corrected videoconferencing," in *Communications, Internet, and Information Technology*, 2002.

[10] B. Yip and Jesse S. Jin, "Face re-orientation in video conference using ellipsoid model," in *OZCHI*, 2003.

[11] C. Kuster, T. Popa, J.-C. Bazin, C. Gotsman, and M. Gross, "Gaze correction for home video conferencing," *ACM TOG (SIGGRAPH Asia)*, 2012.

[12] M. Gross, S. Würmlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. Van Gool, S. Lang, K. Strehlke, A. V. Moere, and O. Staadt, "Blue-C: a spatially immersive display and 3D video portal for telepresence," in *SIGGRAPH*, 2003.

[13] A. Criminisi, J. Shotton, A. Blake, and P. H. S. Torr, "Gaze manipulation for one-to-one teleconferencing," in *ICCV*, 2003.

[14] R. Yang and Z. Zhang, "Eye gaze correction with stereovision for video-teleconferencing," in *ECCV*, 2002.

[15] J. Zhu, R. Yang, and X. Xiang, "Eye contact in video conference via fusion of time-of-flight depth sensor and stereo," *3D Research*, 2011.

[16] M. Dumont, S. Rogmans, S. Maesen, and P. Bekaert, "Optimized two-party video chat with restored eye contact using graphics hardware," in *e-Business and Telecommunications*, vol. 48 of *Communications in Computer and Information Science*. 2009.

[17] T.-J. Cham, S. Krishnamoorthy, and M. Jones, "Analogous view transfer for gaze correction in video sequences," in *International Conference on Control, Automation, Robotics and Vision*, 2002.

[18] J. Gemmell, K. Toyama, C. L. Zitnick, T. Kang, and S. Seitz, "Gaze awareness for video-conferencing: A software approach," *IEEE MultiMedia*, 2000.

[19] A. C. A. del Valle and J. Ostermann, "3D talking head customization by adapting a generic model to one uncalibrated picture," in *IEEE International Symposium on Circuits and Systems*, 2001.

[20] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *SIGGRAPH*, 1999.

[21] D. Jiang, Y. Hu, S. Yan, L. Zhang, H. Zhang, and W. Gao, "Efficient 3D reconstruction for face recognition," *Pattern Recognition*, 2005.

[22] J. Saragih, S. Lucey, and J. Cohn, "Deformable model fitting by regularized landmark mean-shift," *IJCV*, 2011.

[23] O. Sorkine, "Laplacian mesh processing," in *STAR Proceedings of Eurographics*, 2005.