High Detail Flexible Viewpoint Facial Video from Monocular Input using Static Geometric Proxies

Markus Kettern
Fraunhofer Heinrich Hertz Institute
Einsteinufer 37
10587 Berlin, Germany
markus.kettern@hhi.fraunhofer.de

David Blumenthal-Barby
Fraunhofer Heinrich Hertz Institute
Einsteinufer 37
10587 Berlin, Germany
david.blumenthal@hhi.fraunhofer.de

Peter Eisert
Fraunhofer Heinrich Hertz Institute
Einsteinufer 37
10587 Berlin, Germany
peter.eisert@hhi.fraunhofer.de

ABSTRACT

We propose a method for creating flexible-viewpoint facial video from monocular input employing highly detailed static 3D reconstructions of an actor's head. We use the term *flexible-viewpoint* to indicate that the viewpoint can be arbitrarily chosen (as in free-viewpoint video), but from a restricted set of viewing directions[1]. Our method enables dynamic changes of the viewpoint without requiring estimation of the head geometry form the video sequences which is hard with the methods typically used for creating free-viewpoint video. Alongside video capture of a certain actor, we record static high resolution stereo images of the actor's head and face. From these images, we create a detailed 3D model of the head by image-based reconstruction methods. We propose two methods to register this 3D model to the a starting frame of the video stream following the actors head and facial action. Furthermore, we show how model-based tracking over the whole video sequence provides precise head pose estimates for each video frame. Once the registration is complete, the 3D model serves as geometric proxy for image-based rendering techniques in order to create novel viewpoints using the video stream as texture.

1. INTRODUCTION AND OVERVIEW

Free-viewpoint video allows to watch a scene recorded with multiple cameras from an arbitrary viewpoint, not just from the positions of the cameras. It has evolved rapidly through the last decade and has been successfully demonstrated even in live broadcasts of large sports events like the 2012 Olympics, where the required rendering had to happen very close to real-time. However, most scenes for which free-viewpoint video is available are quite similar. They typically contain one actor in a full-body shot and show smooth camera movement around the actor with their motion often freezed or played back with altered speed. For this application, a work-flow of extracting time-consistent visual hulls and multiview textures of the actor from the video streams and using image-based rendering techniques for display has been established and is used by many protagonists free-viewpoint video today. In this paper, we propose a method for flexibly changing the viewpoint of a video showing the face and head of a person recorded with just one video camera. It is obvious that from this input, the person cannot be viewed from an arbitrary viewing position since there is no information contained in the video about how the person looks from strongly rotated viewpoints (thus we termed the application *flexible viewpoint* video). However, our approach offers sufficient flexibility for applications such as correction of the viewing direction towards the observer or towards another character while providing very realistic results.

Changing the viewpoint of close-up facial video is delicate because, different from the full-body case, movement of the actor as well as the viewpoint will be more subtle to allow viewers to still "read" the face they see - a task the human vision is exceptionally good at and will spot almost any artifacts introduced by the rendering. Therefore, applications that employ changing the viewpoint in facial video usually are restricted to small changes like in head pose adjustment for video-conferencing [14, 18]. Moreover, in a setup where the actor is likely to move around (like in nearly any movie or interactive content production), cameras which show the face closely will have to follow the actor by panning, tilting and zooming and therefore may be much harder to calibrate thoroughly than a static setup. In addition to the difficult calibration, visual hulls of the face are typically very coarse since the face is not strongly geometrically articulated and contains major concavities. For a typical application of

[1]The term *free-viewpoint video* is used frequently for applications with a highly restricted choice of novel viewpoints. We propose to term these applications *flexible-viewpoint* to indicate this difference.

MIRAGE '13 June 06 - 07 2013, Berlin, Germany

free-viewpoint video where the face is not shown close up, a coarse geometric approximation as obtained by a visual hull may suffice. For a close view of the face however, more geometric detail is certainly required.

We propose to compute the 3D information used for rendering from static multiple-view stereo capture of the actor as opposed to computing it from the video stream. We use state-of-the-art image-based stereo reconstruction methods to create a highly detailed textured 3D model of the actor's head offline. Once generated, the 3D model is matched to a single key-frame of the video stream. This can be achieved by using feature matches if the imaging properties of the still and video cameras as well as the lighting conditions of both recording stages are suffeciently similar. If not, we propose a semi-automatic method minimizing mutual information between the texture from the static capture and the video frame in an analysis-by-synthesis framework. Depth information can now be added to the whole video sequence by rigidly tracking the matched 3D model over the frames following the key-frame. For this purpose, the original texture of the 3D model is exchanged for the key-frame to allow for direct minimization of the optical flow between subsequent frames. This depth information is exploited for creating novel views of the head with flexibly changed viewpoints by rendering the 3D model from the desired viewpoint using the original video frames as the texture.

2. RELATED WORK

The first occurrences of the term **free-viewpoint video** in context with recordings of humans may be found at the beginning of our millennium when Carranza et al set the focus of what was introduced more generally as 3D video by [11, 25] to the rendering of captured data from arbitrary viewpoints in their seminal paper [3]. Since then, free-viewpoint video for human actors has received great attention and finally a complete system has been released as an open-source tool in 2009 [21]. Several techniques for free-viewpoint video are lined out in [9]. However, the techniques typically used in these setups heavily depend on silhouettes of the recorded actor from which a visual hull is extracted and thus are not appropriate to capture detailed facial geometry. In coding of facial video, an approach similar to ours was taken by [4] where a coarse, hand-crafted 3D model of the face was used to obtain a parametric representation of a moving face along with its expressions for very low bitrate transmission of video streams. One main application of rendering novel views of captured face and head video is gaze correction for video like proposed in [14, 18]. However, we argue that this application can hardly be termed *free*-viewpoint. Our system consists of three main steps, for which we shortly review the most relevant literature in the following.

Image-based **3D reconstruction** of heads and faces has been thoroughly researched in the last decades as lined out by several surveys [24, 22]. The method we use to generate our geometric proxies employs a deformable mesh to create dense correspondences between a non-wide-baseline pair of images using estimation techniques similar to optical flow. Approaches similar to ours have been proposed by [1, 2, 16]

There are numerous approaches to **head tracking** and head pose estimation, as displayed by the survey [12]. Since we have an accurate geometry model, which we want to use in the rendering stage as well, we resort to a model-based approach for head tracking. Because this technique requires a 3D model to be fit to the image, head pose estimation is always performed jointly. Model-based approaches have been shown to be robust under occlusion, varying expressions and low resolution as well as motion blur [10, 23, 5].

To display the face from different viewpoints, we use a simple form of hybrid or geometry-assisted **image-based rendering** as termed by pertinent surveys [7, 13] where a geometric proxy is used to deform the image of an object and thus make it appear as being viewed from a different point in space.

3. DATA ACQUISITION AND GENERATION OF 3D MODELS

The texture data is acquired using one or more dynamic video cameras capturing the face and head of the actor. Since the footage should allow for rendering of close-up views, the cameras should follow the head as close up as possible while they are performing the actions to be displayed and thus will be panning, tilting and zooming. If the full body of the actor is to be shown as well, this will happen in a studio where multiple static cameras are placed around the borders of a recording volume which restricts the face cameras to keep a certain minimal distance to the actor in order not to interfere with full-body capture. The face cameras do neither have to be calibrated towards each other nor towards the full-body cameras in order to enable registration of the individual video streams since this can be done using the head / face model of the actor. However, we assume the intrinsic camera parameters to be known for each frame in order to render out geometric proxy into the camera view.

The geometric data is generated using a multiview stereo setup consisting of several pairs of synchronized D-SLR cameras and adequate lighting. Calibration of these cameras may be provided by utilizing an appropriate calibration target (e.g. [8]) or directly from the recorded facial images by bundle-adjustment [19, 20]. From each stereo pair, we create a high detail textured 3D model using the reconstruction method we presented in [17] where a mesh-based warping function between images is computed that describes a dense, piecewise affine correspondence map between these images. We initialize this warping function by feature matches and use a Laplacian smoothness term. From these correspondences, a 3D reconstruction is computed by triangulation of the stereo disparities. An image pair along with the corresponding reconstruction result is shown in figure 1. With some minor improvements of the optimization procedure (e.g. reuse of matrix reordering in subsequent iterations) a high-detail reconstruction is computed in about 10 seconds.

4. REGISTRATION OF STATIC 3D MODELS TO VIDEO FRAMES

In order to use the created 3D models as geometric proxies for free-viewpoint rendering, we first have to precisely register them to a single video frame (a *key-frame*) as an initialization to our tracking routine. If the imaging properties of the video cameras are comparable to those of the D-SLRs (e.g. when using D-SLRs for the video capture as well) and the lighting situation is very similar, this can be done automatically using feature correspondences like SIFT in a RANSAC fashion using 3 feature correspondences as a set of landmarks (the sample) to compute the inlier set in

Figure 1: Example reconstruction result: stereo input image pair and result of our 3D reconstruction method.

each instance of the algorithm. If the imaging properties of the cameras are too dissimilar (which is not unlikely when using video cameras or recording in different rooms), feature matching may not provide enough useable correspondences (if any). In this case we propose to label three landmarks by hand in the texture of the geometric proxy as well as in the key-frame (e.g. eye corners and nose tip). In both cases the registration is performed by finding the optimal rotation (3 parameters) of the 3D model where the translation (another 3 parameters) is always chosen to minimize the distance of the landmarks between the rendered model and the key-frame and perform the registration using an analysis-by-synthesis approach.

For both cases, we first normalize the model to have its centroid at the coordinate origin by subtracting vector $\mathbf{D}$. Then, we create 3D points from the landmark positions in the texture image using the known relation between the texture camera and the 3D model. Each landmark correspondence is now given by a tuple $(\mathbf{x}, \mathbf{X})$ of a 2D point $\mathbf{x}$ in the key-frame and a 3D point $\mathbf{X}$ in space, lying on the 3D model. We use a non-skewed pinhole camera model [6] where the projection function is completely specified by the principal point $\mathbf{c}$ and the vector of focal lengths $\mathbf{f} = [f_x\, f_y]^T$, measured in sensor pixels [4]. The projection of a 3D point $\mathbf{X}$ into the image plane of the camera is given by

$$\mathbf{x} = \mathbf{c} - \mathbf{f} \circ \begin{bmatrix} \frac{X_1}{X_3} \\ \frac{X_2}{X_3} \end{bmatrix} \tag{1}$$

where $\circ$ denotes the Hadamard product. We will denote this operation by $\text{proj}(\mathbf{X})$ and omit the camera parameters since we refer to the same video camera throughout the whole paper.

4.1 Automatic case

If there are enough feature correspondences to use the automatic approach, three landmark correspondences are chosen at random from the set of feature matches for each instance of the RANSAC procedure. Be L the set of these landmark correspondences and F the set of all feature correspondences. For a rotation matrix R, which can be parametrized completely by 3 values, the error to be minimized is given by

$$\mathcal{E}_f(R) = \sum_{(\mathbf{x},\mathbf{X})\in F} \left\|\text{proj}(R\mathbf{X} + \mathbf{T}^* + \mathbf{D}) - \mathbf{x}\right\|^2 \tag{2}$$

with

$$\mathbf{T}^* = \arg\min_{\mathbf{T}\in\mathbb{R}^3} \sum_{(\mathbf{x},\mathbf{X})\in L} \left\|\text{proj}(R\mathbf{X} + \mathbf{T} + \mathbf{D}) - \mathbf{x}\right\|^2 \tag{3}$$

being the translation which minimizes the distance between the landmark positions on the rendered 3D model. This translation can be found very quickly using numerical optimization, e.g. Newton's method. $\mathcal{E}_f$ is minimized using the same type of optimization for every choice of L. As typical in RANSAC, we count the feature matches for which the distance in the calculation of $\mathcal{E}_f$ is below a certain threshold as inliers, select from all instances that have been run the one that yields the highest inlier count and re-calculate $\mathbf{R}$ using the full inlier set of the chosen instance. The running time of this registration process is usually a few seconds, depending on the settings for the RANSAC procedure.

4.2 Semi-automatic case

With the landmarks manually set a selection procedure as in the automatic case is not necessary. However, the error has to be computed directly from the difference between the rendered images of the 3D model and the key-frame which strongly increases the non-linearity of the error function. Since the texture of the rendered images has been captured under lighting conditions different from those during video capture, we use the mutual information ($\mathcal{M}$) between rendered image and key-frame to compensate for low-frequency lighting discrepancies [15]. The error function now becomes

$$\mathcal{E}_i(R) = \mathcal{M}\left(J(R, T^*), I\right) \tag{4}$$

with $J(R, T^*)$ being the rendered image of the 3D model using rotation and translation of the model R and T^* as defined above, and I being the key-frame. Since $\mathcal{E}_i$ is much less smooth than $\mathcal{E}_f$, we choose to optimize it with a hierarchical approach with the top layer being an exhaustive search over a coarse set of parameters using blurred versions of the images and the layers beneath being either numerical optimization or another exhaustive search. An optimization using 3 layers of exhaustive search can be performed with less than 500 evaluations of the error function and has been used to create the results shown in this paper. The performance of this kind of optimization depends heavily on the speed of rendering $J(P, R, T^*)$ for a given set parameters. Using OpenGL for rendering and downloading only the relevant portion of the rendered image for comparison allows to complete the optimization in less than two seconds. Results of the registration are shown in figure 2.

Figure 2: Results of the initial matching procedure for two different movie key-frames of a medieval monologue sequence. Top left: original frame, top right: overlay of high-resolution 3D model after matching landmarks by translation, bottom left: overlay after matching rotation, bottom right: absolute intensity differences between key-frame and matched 3D model.

5. MODEL-BASED HEAD TRACKING AND POSE ESTIMATION

Once the 3D model has been registered to the video key-frame, obtaining the pose of the geometric proxy for the subsequent frames is an instance of model-based tracking. To prevent the texture from drifting over the geometry, we use the texture of the key-frame throughout the whole sequence to be tracked. If the difference between the key-frame texture and the video frame is too high, track may be lost. In this case, the tracker has to be re-initialized either by the process described in section 4 or by simply stepping back a few frames and exchanging the texture used for tracking. It is obvious that the latter approach may introduce drift as above, just slower. However, in our experiments sequences with up to 10 seconds of highly expressive facial action could be tracked before track was lost so re-initialization will occur rarely in most applications.

Similar to [5], we use a linearized relation between euclidean transformations applied to the 3D model and the 2D pixel displacements they impose on the rendered image. If we assume the rotational component of the transformation to be reasonably small, the rotation matrix to be applied to each vertex of the model is

$$R_{lin} = \begin{bmatrix} 1 & -r_z & r_y \\ r_z & 1 & -r_x \\ -r_y & r_x & 1 \end{bmatrix} \tag{5}$$

with r_x being the angle of rotation around the x-axis etc. The position of a vertex $\mathbf{X}$ of a mesh rotated around the coordinate origin by angles r_x, r_y, r_z (remember that we normalized the mesh to have its centroid in the origin) and then translated by $\mathbf{T}$ is thus linearly approximated by

$$\mathbf{X}' \approx R_{lin}\mathbf{X} + \mathbf{T} \tag{6}$$

A first order approximation of the projection of this displaced vertex into the image is given by

$$\mathbf{x}' = \mathbf{x} + \delta_{\mathbf{x}} \tag{7}$$

$$\delta_{\mathbf{x}} \approx J_{\mathbf{X}} \left(\mathbf{X}' - \mathbf{X}\right) \tag{8}$$

where $J_{\mathbf{X}}$ is the Jacobian of the projection function and is given for a point $\mathbf{X} = [x\, y\, z]^T$ by

$$J_{\mathbf{X}} = \begin{bmatrix} -f_x \frac{1}{z} & 0 & f_x \frac{x}{z^2} \\ 0 & -f_y \frac{1}{z} & f_y \frac{y}{z^2} \end{bmatrix} \tag{9}$$

After some arithmetic manipulations, substituting (6) into (8) yields the image point displacement

$$\delta_{\mathbf{x}} \approx \frac{1}{z}\mathbf{f} \circ H \tag{10}$$

$$H = \begin{bmatrix} -r_y z + r_z y - t_x + r_x \frac{xy}{z} + r_y \frac{x^2}{z} + t_z \frac{x}{z} \\ -r_z x + r_x z - t_y + r_y \frac{xy}{z} - r_x \frac{y^2}{z} - t_z \frac{y}{z} \end{bmatrix} \tag{11}$$

$$= \frac{1}{z}\mathbf{f} \circ C_{\mathbf{X}} \begin{bmatrix} \mathbf{R} \\ \mathbf{T} \end{bmatrix} \tag{12}$$

with

$$C_{\mathbf{X}} = \begin{bmatrix} \frac{xy}{z} & -z + \frac{x^2}{z} & y & -1 & 0 & \frac{x}{z} \\ z - \frac{y^2}{z} & \frac{xy}{z} & -x & 0 & -1 & -\frac{y}{z} \end{bmatrix} \tag{13}$$

which is linear in all unknowns, namely $\mathbf{R} = [r_x\, r_y\, r_z]^T$ and $\mathbf{T}$. The offset $\delta_{\mathbf{x}}$ can be used to minimize the optical flow error between frame I and its successor J defined by

$$\mathcal{E}_m = \sum_{\mathbf{x} \in I} \left(\left(\nabla I_{\mathbf{x}}\right)^T \delta_{\mathbf{x}} - \left(J_{\mathbf{x}} - I_{\mathbf{x}}\right) \right)^2 \tag{14}$$

with $\nabla I_{\mathbf{x}}$ being the image gradient of I at pixel $\mathbf{x}$. Defined over the area Ω of the rendered 3D model, this error can be minimized directly by solving on overdetermined system of linear equations of the form

$$A \begin{bmatrix} \mathbf{R} \\ \mathbf{T} \end{bmatrix} = \left(J_\Omega - I_\Omega\right) \tag{15}$$

where I_Ω is the vector containing all pixel intensities of I in Ω. From equation (12) we can calculate each pair of rows of A depending on a point $\mathbf{X}$ as

$$A_{\mathbf{X}} = \frac{1}{z} \nabla I_\Omega \left(\mathbf{f} \circ C_{\mathbf{X}}\right) \tag{16}$$

where ∇I_Ω is the $2 \times |\Omega|$ matrix containing the x and y gradients of I in region Ω and the rows of A being dependent on the 3D points projected to the individual pixels in Ω.

Since our simple optical flow error assumes the image to be locally smooth (for details, refer to [4]), we perform the

optimization iteratively in a hierarchical, coarse to fine manner.

Figure 3 shows some results of our head-tracking method in a highly expressive monologue sequence. It is obvious that although the expression, and thus the appearance, varies strongly, the head is accurately tracked.

6. IMAGE-BASED RENDERING OF NOVEL VIEWPOINTS

Once the 3D model is attached to each video frame, it can be used to render the head from a different viewpoint, using the original video frame and the depth information obtained from the 3D model. Since we do not assume the calibration of the video camera to be known, we can describe a novel viewpoint only in relation to the original video camera. Since we want to render viewpoint changes, the intrinsic parameters of the camera used to create the novel view will be identical to those of the original video camera and only the extrinsic parameters, i.e. rotation and translation will be different.

To synthesize a view, we render the depth of the 3D model as seen from the novel viewpoint. This depth image allows to trivially compute the 3D position $\mathbf{X}$ associated with each pixel $\mathbf{x}'$ of the image to be rendered. This 3D point can be projected into the current video frame using equation (1), yielding an image position $\mathbf{x}$ at which the image is sampled to obtain the color of pixel $\mathbf{x}'$. Since rendering the depth images can be done in real-time, the speed of this procedure is either real-time or slower due to the method of interpolation used when sampling the video image.

7. RESULTS AND CONCLUSION

Figure 4 shows some results of synthesizing several new viewpoints for frames of a monologue sequence which has been tracked using one proxy and one key-frame. The tracking was stable for more than 8 seconds (200 frames) of material before a reset was necessary. All results shown in figure 4 are taken from these first 8 seconds of this dynamic scene. We find that for viewpoint changes of up to 20 degrees the head is reproduced realistically with artifacts only arising at the edges of the model and texture. These artifacts will be reduced once we use integrated 3D models which are derived from more than one stereo pair of static cameras and correctly represent the head geometry from more than one perspective.

One limitation of the presented approach is that the proposed way of creating novel views does not allow for large changes in the viewing position since texture is only available for the portion of the head covered by the video camera.

We have proposed a method for changing the viewpoint of monocular facial video sequences using static geometric proxies. We have shown how to create these geometric proxies using state-of-the-art mesh-based dense 3D reconstruction methods on stereo pairs of high-resolution still images taken alongside a video production. We have proposed a procedure to robustly match geometric proxies to a single video key-frame either automatically if enough feature correspondences can be found between one still image and the key-frame or semi-automatically by manually annotating a small set of correspondences and optimizing the pose of the 3D model to minimize mutual information between the rendered model and the key-frame. Furthermore, we have presented a model-based scheme to track the model throughout a video sequence accurately enough to allow for consistently rendering the video stream from a novel viewpoint. We have shown this approach to provide visually pleasing results for viewpoint changes of up to 20°.

Future work on this topic will include:

- Integration of multiple, non-calibrated monocular video streams for higher texture coverage
- Registration and calibration of multiple video cameras using estimated head model poses in each individual video stream
- Simultaneous use of several 3D models created from different viewpoints for improved geometric accuracy and coverage
- Use of several 3D models representing different facial expressions to better fit the expressions in the video stream
- Estimation and compensation of lighting, re-lighting

Acknowledgement

The work presented in this paper has been funded by the Seventh Framework Programme EU project RE@CT (grant agreement no. 288369)

8. REFERENCES

[1] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross. High-quality single-shot capture of facial geometry. In *ACM SIGGRAPH 2010 papers*, SIGGRAPH '10, pages 40:1–40:9, New York, NY, USA, 2010. ACM.

[2] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer. High resolution passive facial performance capture. *ACM Transactions on Graphics*, 29:41:1–41:10, 2010.

[3] J. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. In *ACM Transactions on Graphics*, pages 569–577, 2003.

[4] P. Eisert. *Very Low Bit-Rate Video Coding Using 3-D Models*. Shaker, 2000.

[5] P. Eisert and B. Girod. Model-based 3d-motion estimation with illumination compensation. In *Image Processing and Its Applications, 1997*, volume 1, pages 194–198 vol.1, Jul.

[6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[7] S. B. Kang. *A survey of image-based rendering techniques*. Number January. Spie, 1999.

[8] M. Kettern, D.C. Schneider, B. Prestele, F. Zilly, and P. Eisert. Automatic acquisition of time-slice image sequences. In *Conference onVisual Media Production (CVMP)*, pages 40–48, 2010.

[9] J. Kilner, J. Starck, and A. Hilton. A comparative study of free-viewpoint video techniques for sports events. In *Conference on Visual Media Production (CVMP)*, 2006.

[10] L. Lu, Z. Zhang, H.-Y. Shum, Z. Liu, and H. Chen. Model- and exemplar-based robust head pose tracking under occlusion and varying expression.

[11] T. Matsuyama and T. Takai. Generation, visualization, and editing of 3d video. In *3D Data Processing Visualization and Transmission*, pages 234–245.

[12] E. Murphy-Chutorian and M. Manubhai Trivedi. Head pose estimation in computer vision: A survey, 2008.

[13] M. M. Oliveira. Image-based modeling and rendering techniques: A survey. *RITA*, 9(2):37–66, 2002.

[14] C.-H. Pan, S.-C. Huang, Y.-L. Chang, Chung-Jr L., and L.-G. Chen. Real-time free viewpoint rendering system for face-to-face video conference. In *International Conference on Consumer Electronics (ICCE)*, 2008.

[15] J.P.W. Pluim, J.B.A. Maintz, and M.A. Viergever. Image registration by maximization of combined mutual information and gradient information. *IEEE Transactions on Medical Imaging*, 19(8):809–814.

[16] D.C. Schneider, M. Kettern, A. Hilsmann, and P. Eisert. Deformable image alignment as a source of stereo correspondences on portraits. In *NORDIA workshop, Computer Vision and Pattern Recognition (CVPR)*, 2011.

[17] D.C. Schneider, M. Kettern, A. Hilsmann, and P. Eisert. A global optimization approach to high-detail reconstruction of the head. In *International Workshop on Vision, Modeling and Visualization (VMV)*, pages 9–15, 2011.

[18] O. Schreer, I. Feldmann, N. Atzpadin, P. Eisert, P. Kauff, and H.J.W. Belt. 3dpresence -a system concept for multi-user and multi-party immersive 3d videoconferencing. In *Conference on Visual Media Production (CVMP)*, pages 1–8, 2008.

[19] N. Snavely, S.M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Transactions on Graphics*, 25(3):835–846, July 2006.

[20] N. Snavely, S.M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal on Comput. Vision*, 80(2):189–210, November 2008.

[21] J. Starck, J. Kilner, and A. Hilton. A free-viewpoint video renderer. *Journal of Graphics, GPU, and Game Tools*, 14(3):57–72, 2009.

[22] C.Y. Suen, A.Z. Langaroudi, Chunghua F., and Yuxing M. A survey of techniques for face reconstruction. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pages 3554–3560, Oct.

[23] J. Tu, T. Huang, and H. Tao. Accurate head pose tracking in low resolution video. In *International Conference on Automatic Face and Gesture Recognition (FGR)*, pages 573–578, 2006.

[24] W. N. Widanagamaachchi and A. T. Dharmaratne. 3d face reconstruction from 2d images. In *Proceedings of the 2008 Digital Image Computing: Techniques and Applications*, DICTA '08, pages 365–371, Washington, DC, USA, 2008. IEEE Computer Society.

[25] S. Würmlin, E. Lamboray, O.G. Staadt, and M.H. Gross. 3d video recorder. In *Pacific Conference on Computer Graphics and Applications*, pages 325–334, 2002.

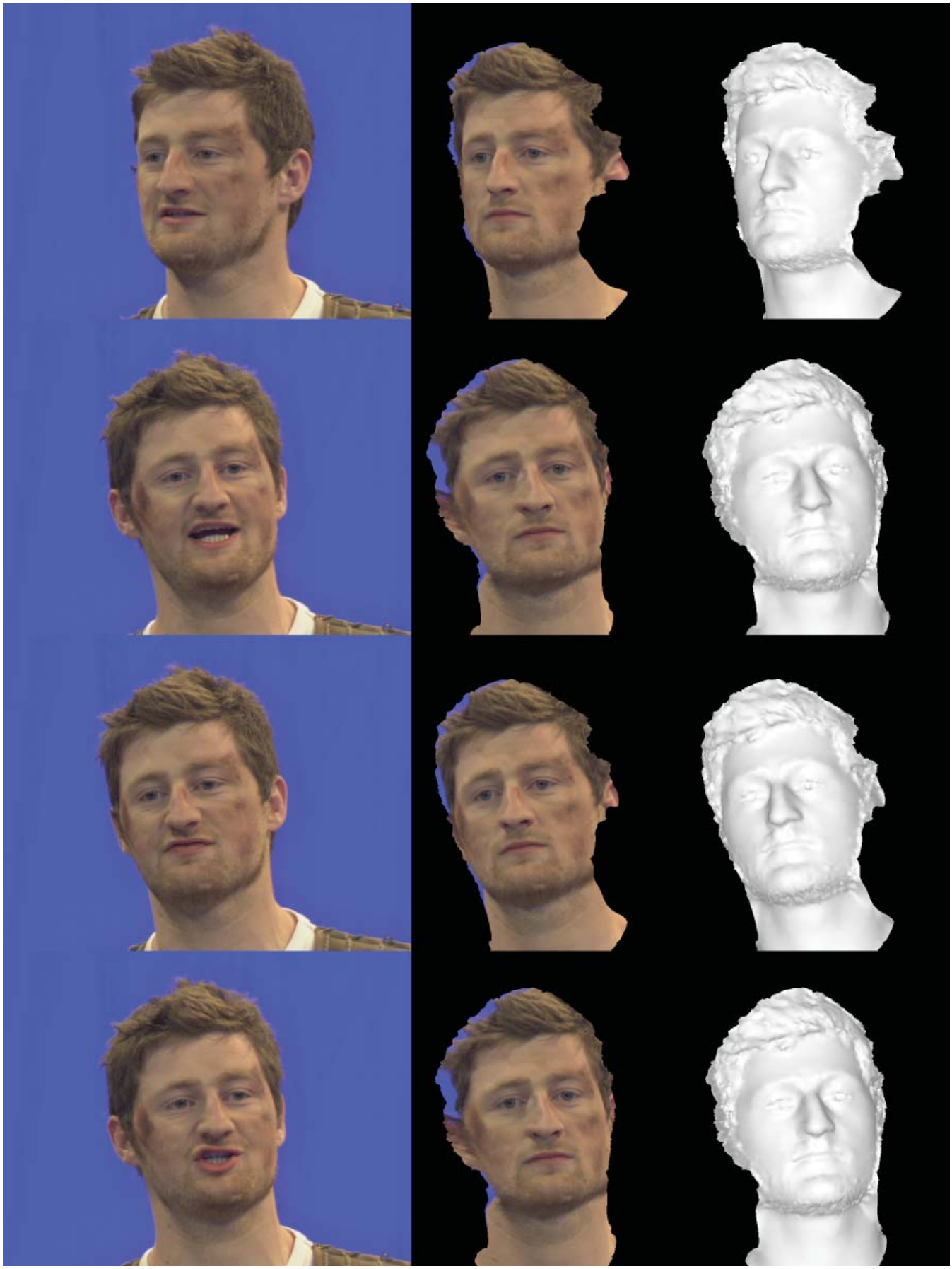

Figure 3: Tracking Results: Each row shows a movie frame and the textured / untextured geometric proxy that has been tracked in the sequence starting at the key-frame.

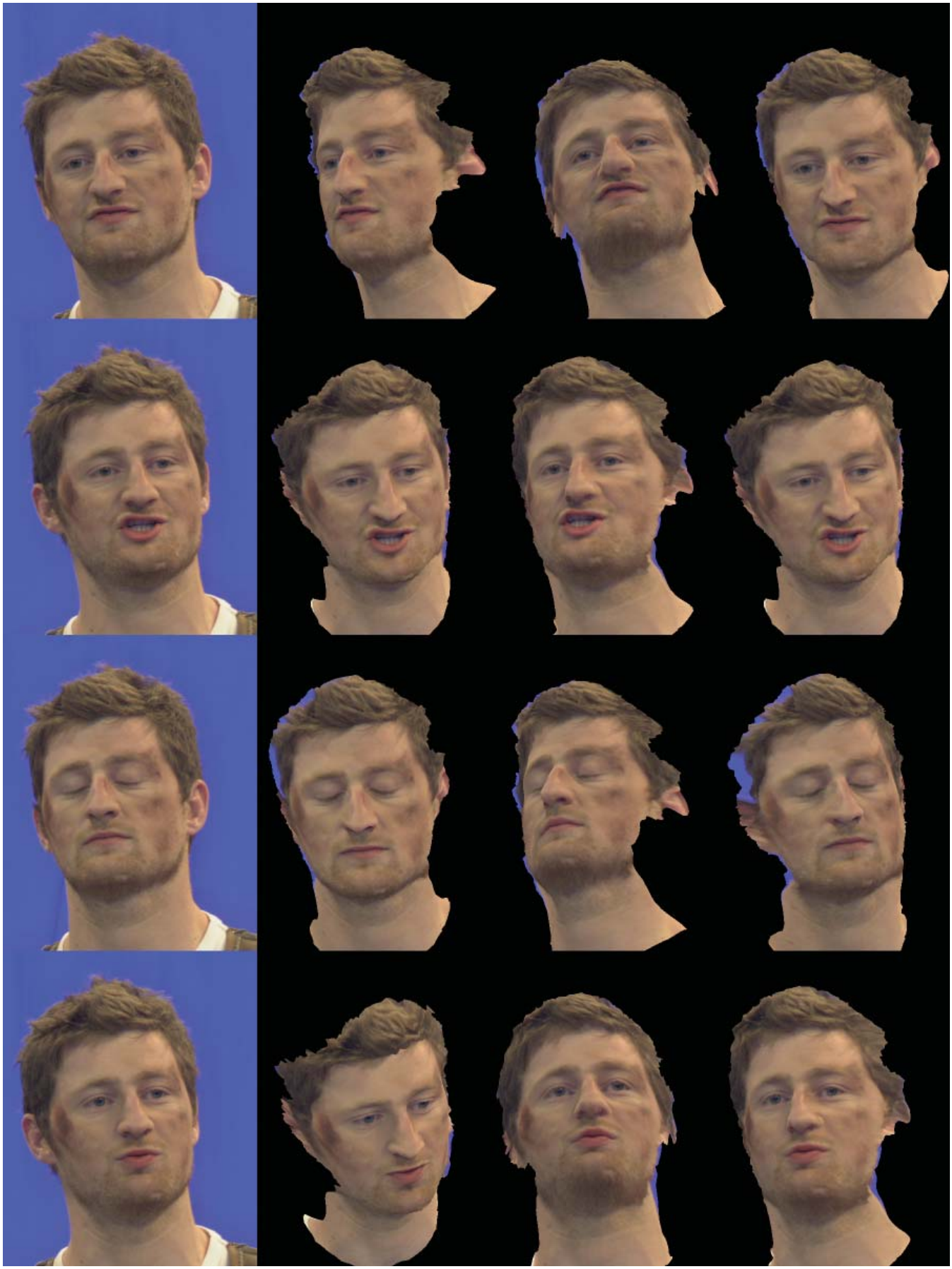

Figure 4: Rendering results: each row contains the original frame and three synthesized views from novel viewpoints.