Interactive Model-based Reconstruction of the Human Head using an RGB-D Sensor

M. Zollhöfer, J. Thies, M. Colaianni, M. Stamminger, G. Greiner
Computer Graphics Group, University Erlangen-Nuremberg, Germany

Abstract

We present a novel method for the interactive markerless reconstruction of human heads using a single commodity RGB-D sensor. Our entire reconstruction pipeline is implemented on the GPU and allows to obtain high-quality reconstructions of the human head using an interactive and intuitive reconstruction paradigm. The core of our method is a fast GPU-based non-linear Quasi-Newton solver that allows us to leverage all information of the RGB-D stream and fit a statistical head model to the observations at interactive frame rates. By jointly solving for shape, albedo and illumination parameters, we are able to reconstruct high-quality models including illumination corrected textures. All obtained reconstructions have a common topology and can be directly used as assets for games, films and various virtual reality applications. We show motion retargeting, retexturing and relighting examples. The accuracy of the presented algorithm is evaluated by a comparison against ground truth data.

Keywords: Virtual Avatars, Model-based Face Reconstruction, 3D Scanning, Non-linear Optimization, GPU, Statistical Head Models

1 Introduction

The release of the Microsoft Kinect made a cheap consumer level RGB-D sensor available for home use. Therefore, 3D scanning technology is no longer restricted to a small group of professionals, but is also accessible to a broad audience. This had a huge impact on research in this field shifting the focus to intuitive and interactive paradigms that are easy-to-use.

Figure 1: Hardware Setup: The RGB-D stream of a single PrimeSense Carmine is used to reconstruct high-quality head models using an interactive and intuitive reconstruction paradigm.

This work presents a model-based reconstruction system for the human head that is intuitive and leverages a commodity sensor's RGB-D stream interactively. Using the presented system a single user is able to capture a high-quality facial avatar (see Figure 1) by moving his head freely in front of the sensor. During a scanning session the user receives interactive feedback from the system showing the current reconstruction state. Involving the user into the reconstruction process makes the system immersive and allows to refine the result. Our system effectively creates a high-quality virtual clone with a known semantical and topological structure that can be used in various applications ranging from virtual try-on to teleconferencing.

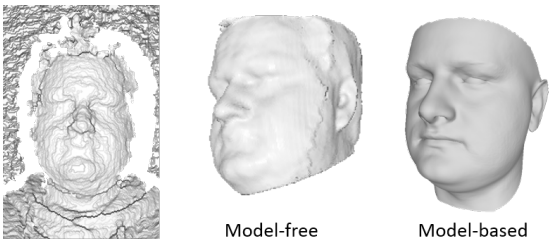

Figure 2: Comparison: Model-free approaches (Kinect Fusion, 256^3 voxel grid) are prone to oversmoothing. In contrast to this, our model-based approach allows to estimate fine-scale surface details.

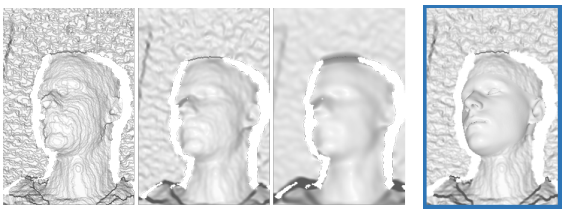

Figure 3: Denoising: Statistical noise removal (right) better deals with noisy input than spatial filtering approaches (left). Fine scale features are retained, while still effectivly dealing with noise.

In the following, we discuss related work (Section 2), present an overview of our reconstruction system (Section 3) based on a fast GPU tracking and fitting pipeline (Section 4-6). We sum up by showing reconstruction results, applications and ground truth comparisons (Section 7) and give ideas for future work (Section 8).

2 Related Work

3D-Reconstruction from RGB-D images and streams is a well studied topic in the geometry, vision and computer graphics communities. Due to the extensive amount of literature in this field, we have to restrict our discussion to approaches closely related to this work. Therefore, we will focus on model-free and model-based algorithms that are suitable for capturing a detailed digital model (shape and albedo) of a human head. We compare these approaches based on their generality and applicability and motivate our decision for a model-based reconstruction method.

2.1 Model-free 3D-Reconstruction

3D-Reconstruction mainly is about the acquisition of a real world object's shape and albedo. This includes capturing and aligning multiple partial scans [1, 2, 3] to obtain one complete reconstruction, data accumulation or fusion [4] and a final surface extraction step [5] to obtain a mesh representation. Systems based on a direct accumulation of the input sample points [6, 7] preserve information, but scale badly with the length of the input stream. In contrast, systems based on volumetric fusion [4] accumulate the data directly into a consistent representation, but do not keep the raw input for further postprocessing steps. The Kinect Fusion framework [8, 9] is such a system and made real-time 3D-Reconstruction with a moving RGB-D camera viable for the first time. Because this approach deals with noise by spatial and temporal filtering, it is prone to oversmoothing (see Figure 2). Model-free approaches allow to digitize arbitrary real world objects with the drawback of the output to be only a polygon soup [5] with no topological and semantical information attached. Therefore, these reconstructions can not be automatically animated or used in virtual reality applications.

2.2 Model-based 3D-Reconstruction

In contrast to model-free approaches, model-based methods heavily rely on statistical priors and are restricted to a certain class of objects (i.e., heads or bodies). This clear disadvantage in generality is compensated by leveraging the class-specific information built into the prior [10, 11, 12, 13]. In general, this leads to higher reconstruction quality, because noise can be statistically regularized (see Figure 3) and information can be propagated to yet unseen and/or unobservable regions. These properties make model-based reconstruction algorithms the first choice for applications that are focused on one specific object class.

Blanz and colleagues reconstruct 3D models from RGB and RGB-D input [10, 11] by fitting a statistical head model. These methods require user input during initialization and registration is performed in a time consuming offline pro-

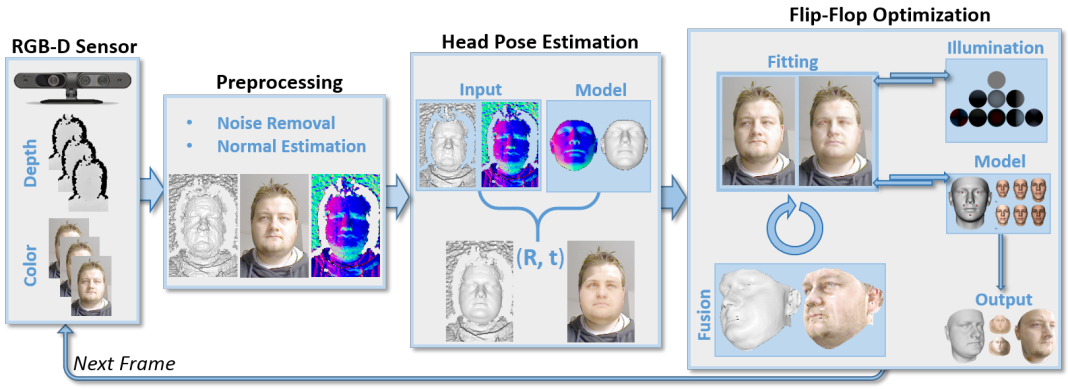

Figure 4: Per-Frame Pipeline (from left to right): The RGB-D input stream is preprocessed, the rigid head pose is estimated, data is fused and a joint optimization problem for shape, albedo and illumination parameters is solved iteratively.

cess. Statistical models have also been extensively used to reconstruct templates for tracking facial animations [14, 15, 16]. While the tracking is real-time, the reconstruction is performed offline. In addition, these methods only use depth and do not consider the RGB channels which allows to jointly estimate the illumination and can be used to improve tracking [17, 18]. Other methods specifically focused on reconstruction are either offline or do not use all the data of the RGB-D stream [19, 20, 21, 22, 23, 24, 25]. In many cases, they only rely on a single input frame.

In contrast, our method utilizes all data provided by the RGB-D stream and gives the user immediate feedback. We specifically decided for a model-based approach because of its superior reconstruction quality and better reusability of the created models. Applications able to use our reconstructions range from animation retargeting [26, 14, 15, 16] to face identification [27, 28], as well as virtual aging [29] and try-on [30]. The main three contributions of this work are:

- An intuitive reconstruction paradigm that is suitable even for unexperienced users
- The first interactive head reconstruction system that leverages all available information of the RGB-D stream
- A fast non-linear GPU-based Quasi-Newton solver that jointly solves for shape, albedo and illumination.

3 Pipeline Overview

Our reconstruction system (Figure 4) has been completely implemented on the GPU with an interactive application in mind. The user sits in front of a single RGB-D sensor (see Figure 1) and can freely move his head to obtain a complete and high-quality reconstruction. In the preprocessing stage, the captured RGB-D stream is bilaterally filtered [31] to remove high-frequency noise. We back-project the depth map to camera space and compute normals at the sample points using finite differences. We track the rigid motion of the head using a dense GPU-based iterative closest point (ICP) algorithm. After the global position and orientation of the head has been determined we use a non-rigid registration method that flip-flops between data fusion and model fitting. We fuse the unfiltered input data into a consistent mesh-based representation that shares its topology with the statistical prior. This step allows for super-resolution reconstructions, closes holes and fairs the data using a fast GPU-based thin-plate regularizer [32]. The resulting faired displacements define the position constraints for non-rigidly fitting the statistical model. After the best fitting model has been computed, we use the solution to initialize the next flip-flop step. This allows us to temporally fair and stabilize the target correspondences.

4 Head Pose Estimation

We compute an initial guess for the global head pose using the Procrustes algorithm [33]. The required feature points are automatically detected using Haar Cascade Classifiers [34] for the mouth, nose and eyes. Corresponding features on the model have been manually preselected and stay constant. Starting from this initialization, we use a fast GPU-based implementation of a dense ICP algorithm in the spirit of [8, 9] to compute the best fitting rigid transformation $\Phi^t(x) = R^t x + t^t$. We rasterize the model under the last rigid transformation Φ^{t-1} to generate synthetic position $p_{i,j}$ and normal images $n_{i,j}$. Projective correspondence association is used between the input and the synthetic images. The registration error between the rendered model positions and the target correspondences $t_{\mathcal{X}(i,j)}$ under a point-to-plane metric is:

$$\arg\min_{\hat{\Phi}} \sum_{i,j} w_{i,j} < n_{i,j}, \hat{\Phi}(p_{i,j}) - t_{\mathcal{X}(i,j)} >^2.$$

The corresponding $6x6$ least squares system is constructed in parallel on the GPU and solved via SVD. We set the correspondence weights $w_{i,j}$ based on distance and normal deviation and prune correspondences ($w_{i,j} = 0$) if they are too far apart ($>$ 2cm), the normals do not match ($>$ 20°) or the pixels are not associated with the head. The valid region for correspondence search is selected by a binary mask that specifies the part of the head that stays almost rigid (red) under motion. The predicted head pose of the current frame $\Phi^t(x) = \hat{\Phi}(x)\Phi^{t-1}(x)$ is used as starting point for the reconstruction of the non-rigid shape.

5 Data Fusion

Depth data of consumer level RGB-D sensors has a low resolution, contains holes and a lot of noise. We use a fusion scheme similar to Kinect Fusion [8, 9] to achieve super-resolution reconstructions and effectivly deal with the noisy input. A per-vertex displacement map is defined on the template model to temporally accumulate the input RGB-D stream. Target scalar displacments are found by ray marching in normal direction, followed by four bisection steps to refine the solution. The resulting displacement map is faired by computing the best fitting thin-plate. We approximate the non-linear thin-plate energy [32] by replacing the fundamental forms with partial derivatives:

$$-\lambda_s \Delta d + \lambda_b \Delta^2 d = 0.$$

The parameters λ_s and λ_b control the stretching and bending resistance of the surface and d are the faired scalar displacements. These displacements are accumulated using an exponential average. A fast GPU-based preconditioned gradient descent with Jacobi preconditioner is used to solve the resulting least squares problem. The preconditioner is constant for a fixed topology and can be precomputed, gradient evaluation is performed on-the-fly by iteratively applying the Laplacian kernel to compute the bi-Laplacian gradient component. This nicely regularizes out noise and fills holes in the data. For RGB, we use a one-frame integration scheme to deal with illumination changes.

6 Estimating Model Parameters

By projecting the accumulated data into the space of statistically plausible heads, noise can be regularized, artifacts can be removed and information can be propagated into yet unseen regions. We pose the estimation of the unknown shape α, albedo β and illumination γ parameters as a joint non-linear optimization problem. Shape and albedo is statistically modeled using the Basel Face Model [10, 27], illumination is approximated using spherical harmonics. In the following, we give details on the used statistical model, the objective function and show how to efficiently compute best fitting parameters using a fast GPU-based non-linear Quasi-Newton solver.

6.1 Statistical Shape Model

The used statistical shape model encodes the shape (albedo) of 200 heads by assuming an underlying Gaussian distribution with mean μ_α (μ_β) and standard deviation σ_α (σ_β). The principal components E_α (E_β) are the directions of highest variance and span the space of plausible heads. New heads can be synthesized by specifying suitable model parameters α (β):

$$\begin{aligned} \text{Shape}: \ \mathcal{M}(\alpha) &= \mu_\alpha + E_\alpha \alpha, \\ \text{Albedo}: \ \mathcal{C}(\beta) &= \mu_\beta + E_\beta \beta. \end{aligned}$$

Synthesis is implemented using compute shaders. We use one warp per vertex and a fast warp reduction to compute the synthesized position (albedo).

6.2 Objective Function

Finding the instance that best explains the accumulated input observations is cast as a joint non-linear optimization problem:

$$E(\mathcal{P}) = \lambda_d E_d(\mathcal{P}) + \lambda_c E_c(\mathcal{P}) + \lambda_r E_r(\mathcal{P}).$$

The individual objectives E_d, E_c and E_r represent the depth, color and statistical regularization constraints. The empirically determined weights $\lambda_d = \lambda_c = 10$ and $\lambda_r = 1$ remain fixed and have been used for all shown examples. The parameter vector $\mathcal{P} = (\alpha, \beta, \gamma)$ encodes the degrees of freedom in the model. In the following, we will discuss the different objectives and their role in the optimization problem in more detail.

6.2.1 Depth Fitting Term

The depth fitting term incorporates the accumulated geometric target positions t_i into the optimization problem:

$$E_d(\mathcal{P}) = \sum_{i=1}^{n} ||\Phi^t(\mathcal{M}_i(\alpha)) - t_i||^2.$$

This functional only depends on the shape parameters α and measures the geometric point-point alignment error for every model vertex (n vertices). Minimizing this objective on its own is a linear least squares problem in the unknowns α due to the linearity of the model $\mathcal{M}$.

6.2.2 Color Fitting Term

The visual similarity of the synthesized model and the input RGB data is modeled by the color fitting term:

$$E_c(\mathcal{P}) = \sum_{i=1}^{n} ||\mathcal{I}(t_i) - \mathcal{R}(v_i, c_i, \gamma)||^2,$$

with $v_i = \Phi^t(\mathcal{M}_i(\alpha))$ being the current vertex position, $c_i = \mathcal{C}_i(\beta)$ the current albedo and $\mathcal{I}(t_i)$ is the RGB color assigned to the target position. This part of the objective function is non-linear in the shape α and linear in the illumination γ and albedo β. Illumination is modeled using spherical harmonics (spherical environment map). We assume a purely diffuse material and no self-shadowing:

$$\mathcal{R}(v_i, c_i, \gamma) = c_i \sum_{j=1}^{k} \gamma_j H_j(v_i).$$

H_j is the projection of the angular cosine falloffs on the spherical harmonics basis. We use 3 spherical harmonics bands ($k = 9$ coefficients per channel) and sample 128 random directions (Hammersley sampler) for the numerical evaluation of H_j.

6.2.3 Statistical Regularization

The heart of this method is a statistical regularizer [10] that takes the probability of the synthesized instances into account. This prevents overfitting the input data. Assuming a Gaussian distribution of the input, approximately 99% of the heads can be reproduced using parameters $x_i \in [-3\sigma_{x_i}, 3\sigma_{x_i}]$. Therefore, the parameters are constrained to be statistically small:

$$E_r(\mathcal{P}) = \sum_{i=1}^{m} \left[\frac{\alpha_i^2}{\sigma_{\alpha_i}^2} + \frac{\beta_i^2}{\sigma_{\beta_i}^2} \right] + \sum_{i=1}^{k} \left(\frac{\gamma_i}{\sigma_{\gamma_i}} \right)^2.$$

The standard deviations σ_{α_i} and σ_{β_i} are known from the shape model, $\sigma_{\gamma_i} = 1$ encodes the variabilty in the illumination and has been empirically determined. m specifies the number of used principal components.

6.3 Parameter Initialization

The objective function E is non-linear in its parameters $\mathcal{P}$. Therefore, a good initial guess

is required to guarantee convergence to a suitable optimum. Current state-of-the-art methods heavily rely on user input in form of sparse marker or silhouette constraints to guide the optimizer through the complex energy landscape. In this work, when tracking is started, we initialize the parameters by decoupling the optimization problem into three separate linear problems that can be solved independently. We start by fitting the model against the detected sparse set of markers to roughly estimate the size of the head. As mentioned earlier, the depth objective on its own is a linear least squares problem in the unknown shape parameters. After searching correspondences, a good approximation for α can be computed by solving the linear system that corresponds to the first 40 principal components. Then, the illumination parameters γ are estimated separately by assuming a constant average albedo. Finally, the albedo parameters β are initialized by assuming the computed shape and illumination to be fixed. Once the parameters have been initialized, a joint non-linear optimizer is used to refine the solutions.

6.4 Joint Non-Linear GPU Optimizer

To refine the solutions of the uncoupled optimization problems, we jointly solve for the parameters in each new input frame. We use a fast GPU-based implementation of a Quasi-Newton method to iteratively compute the best fit:

$$\mathcal{P}_{n+1} = \mathcal{P}_n - \lambda(HE(\mathcal{P}_n))^{-1}\Delta E(\mathcal{P}_n).$$

$HE(\mathcal{P}_n)$ is the Hessian matrix and ΔE the gradient of E, λ controls the step size. The step size is adaptively computed based on the change in the residual. We use a simple approximation of the inverse Hessian for scaling the descend directions. This is similar to preconditioning [35]. Because of the global support of the principal components, the derivatives with respect to α and β are influenced by all constraints. Therefore, we use one block per variable and a fast block reduction in shared memory to evaluate the derivatives. Per flip-flop step we perform 2 Quasi-Newton steps. During optimization we slowly increase the number of used principal directions to avoid local minima in the energy landscape.

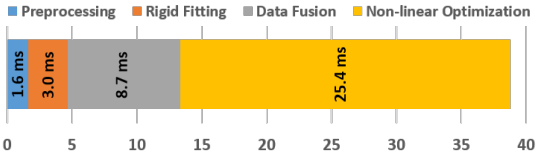

Figure 5: Per-Frame Runtime: Runtime of the different stages in our reconstruction pipeline (in ms).

7 Results

In this section, we discuss the runtime behaviour of our system, compare the reconstruction results to ground truth data obtained by a high-quality structured light scanner and present applications that leverage the known semantical structure of the reconstructed models.

7.1 Runtime Evaluation

Our reconstruction pipeline is completely implemented on the GPU to allow for interactive reconstruction sessions. The average per-frame runtime for the examples in Figure 9 is shown in Figure 5. We used an Intel Core i7-3770 CPU with a Nvidia Geforce GTX 780. Note, that for the person in the fourth row the beard could not be faithfully recovered by the model coefficents, this is due to the fact that facial hair is not contained in the model. But the customly generated texture captures and adds these details to the reconstruction. For all presented examples, we used 2 flip-flop steps (with 2 Quasi-Newton steps each) and 5 iterations of the thin-plate solver. We start with 40 eigenvectors and slowly increase the number to 120. Note, that our system always remains interactive and gives the user direct feedback during the reconstruction process. For the results in Figure 9 the complete scanning sessions took about $3-5$ seconds each. In most cases, moving the head once from right to left is sufficient to compute a high quality model.

7.2 Reconstruction Quality

To evaluate the accuracy of the presented system we compare our reconstructions (PrimeSense Carmine 1.09) with high-quality 3D scans captured by a structured light scanner. The scanning

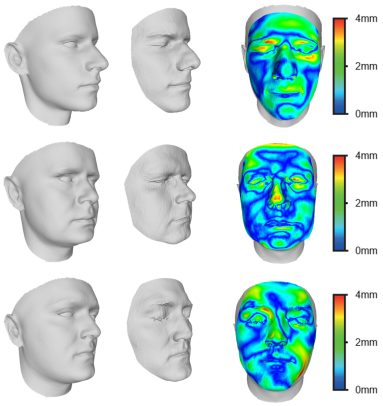

Figure 6: Ground Truth Comparison: Distance (right) between our reconstructions (left) and high-quality structured light scans (middle).

Figure 7: Comparison to [19]: Their reconstruction (left), ours (right).

sessions with the structured light setup took several minutes. As can be seen in Figure 6, the actual shape difference is small and our geometry has comparable quality. Because the eyes had to remain closed during structured light scanning, most of the error is located in the eye region. The mean error was 1.09 mm, 0.81 mm and 1.19 mm respectively (from top to bottom).

We also compare our reconstructions to [19] (see Figure 7). In contrast to their approach, we can reconstruct the complete head, illumination correct the textures and have higher super-resolution geometry.

7.3 Applications

In this section, we discuss some applications that can directly use the reconstructed models, see Figure 10. We compute a complete texture by fusing multiple frames using pyramid blending [36], the required mask is automatically computed using depth and angular thresholds. The illumination parameters allow us to compute illumination corrected textures and relight the head. Because of the known semantical structure, we can place a hat on the model (virtual try-on) and add additional textures. The

Figure 10: Applications: The textured models can be relighted, retextured and used for virtual try-on.

known topological structure of the models allows us to easily retarget animations (Figure 8).

8 Conclusion

We have presented a complete system for the reconstruction of high-quality models of the human head using a single commodity RGB-D sensor. A joint optimization problem is solved interactively to estimate shape, albedo and illumination parameters using a fast GPU-based non-linear solver. We have shown that the obtained quality is comparable to offline structured light scans. Because of the known topological and semantical structure of the models, they can be directly used as input for various virtual reality applications.

In the future, we plane to add motion tracking to our system to animate the reconstructions. We hope that we can leverage the reconstructed albedo to make non-rigid tracking more robust.

Acknowledgements

We thank the reviewers for their insightful feedback. The head model is provided courtesy of Prof. Dr. T. Vetter (Department of Computer Science) and the University of Basel. This work was partly funded by GRK 1773.

References

[1] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, February 1992.

[2] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, June 2001.

Figure 8: Animation Retargeting: The known topology of the reconstructed models allows to easily retarget animation sequences. The input sequence has been taken from [26].

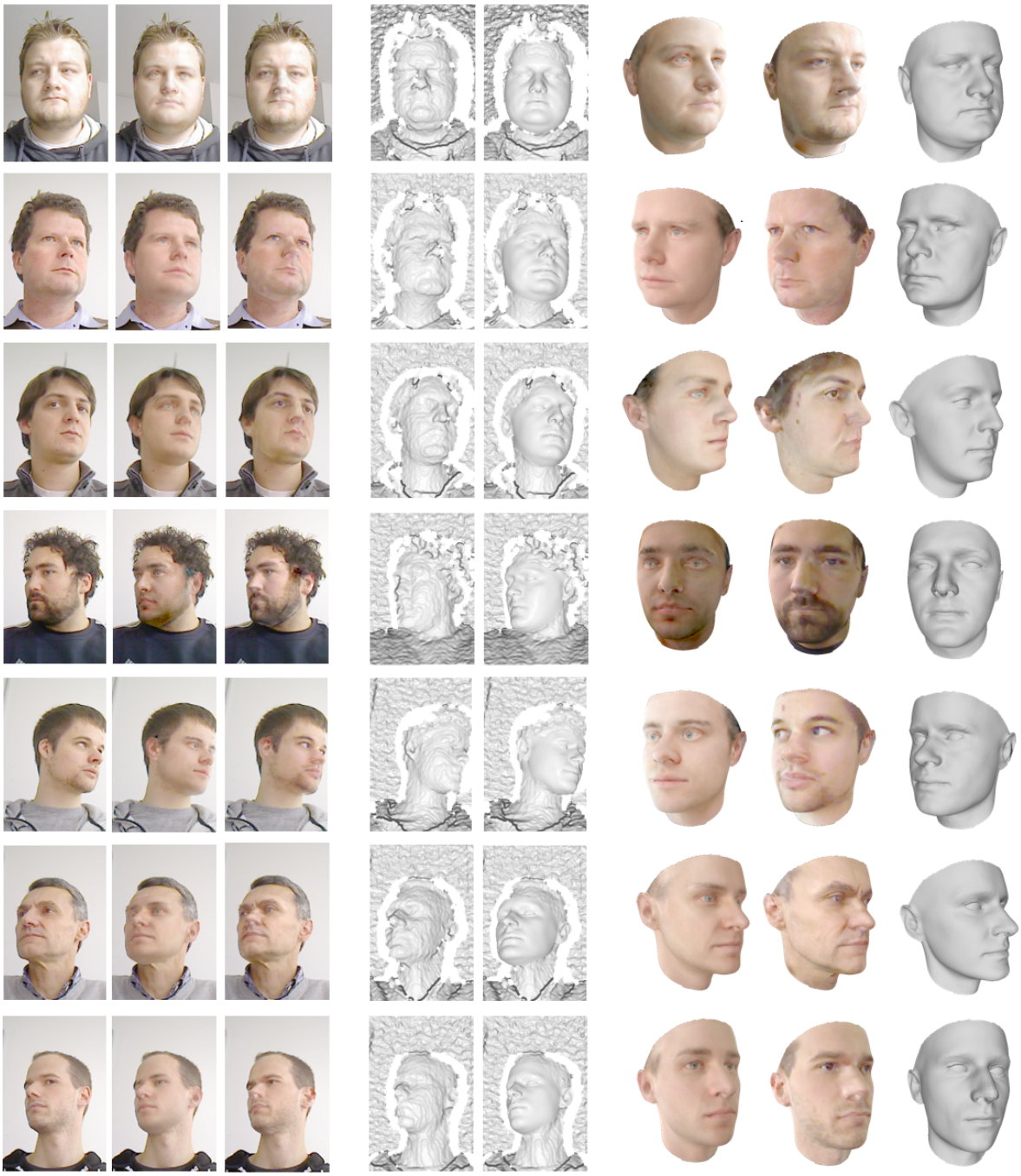

Figure 9: 3D-Reconstruction Results (from left to right): Input color image, overlaid model (fitted color), overlaid model (textured), filtered input positions, overlaid model (phong shaded), reconstructed virtual avatar (fitted color, textured, phong shaded).

[3] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155, April 1992.

[4] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proc. of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996.

[5] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proc. of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 163–169, New York, NY, USA, 1987. ACM.

[6] Thibaut Weise, Bastian Leibe, and Luc J. Van Gool. Accurate and robust registration for in-hand modeling. In *CVPR*. IEEE Computer Society, 2008.

[7] T. Weise, T. Wismer, B. Leibe, , and L. Van Gool. In-hand scanning with online loop closure. In *IEEE International Workshop on 3-D Digital Imaging and Modeling*, October 2009.

[8] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *Proc. ISMAR*, pages 127–136, 2011.

[9] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In *Proc. UIST*, pages 559–568, 2011.

[10] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proc. of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[11] Volker Blanz, Kristina Scherbaum, and Hans peter Seidel. Fitting a morphable model to 3d scans of faces. In *In CVPR*, pages 1–8, 2007.

[12] Sami Romdhani and Thomas Vetter. Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *Edges, Specular Highlights, Texture Constraints and a Prior, Proc. of Computer Vision and Pattern Recognition*, pages 986–993, 2005.

[13] David C. Schneider and Peter Eisert. Fitting a morphable model to pose and shape of a point cloud. In Marcus A. Magnor, Bodo Rosenhahn, and Holger Theisel, editors, *VMV*, pages 93–100. DNB, 2009.

[14] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, pages 77:1–77:10, New York, NY, USA, 2011. ACM.

[15] Hao Li, Jihun Yu, Yuting Ye, and Chris Bregler. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph.*, 32(4):42:1–42:10, July 2013.

[16] Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou. 3d shape regression for real-time facial animation. *ACM Trans. Graph.*, 32(4):41:1–41:10, July 2013.

[17] Thibaut Weise, Hao Li, Luc Van Gool, and Mark Pauly. Face/off: Live facial puppetry. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation (Proc. SCA'09)*, ETH Zurich, August 2009. Eurographics Association.

[18] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Trans Vis Comput Graph*, 20(3):413–25, 2014.

[19] Michael Zollhöfer, Michael Martinek, Günther Greiner, Marc Stamminger, and Jochen Süßmuth. Automatic reconstruction of personalized avatars from 3d face

scans. *Computer Animation and Virtual Worlds (Proceedings of CASA 2011)*, 22(2-3):195–202, 2011.

[20] Li an Tang and Thomas S. Huang. Automatic construction of 3d human face models based on 2d images. In *ICIP (3)*, pages 467–470. IEEE, 1996.

[21] Pia Breuer, Kwang In Kim, Wolf Kienzle, Bernhard Schlkopf, and Volker Blanz. Automatic 3d face reconstruction from single images or video. In *FG*, pages 1–8. IEEE, 2008.

[22] Jinho Lee, Hanspeter Pfister, Baback Moghaddam, and Raghu Machiraju. Estimation of 3d faces and illumination from single photographs using a bilinear illumination model. In Oliver Deussen, Alexander Keller, Kavita Bala, Philip Dutr, Dieter W. Fellner, and Stephen N. Spencer, editors, *Rendering Techniques*, pages 73–82. Eurographics Association, 2005.

[23] Axel Weissenfeld, Nikolce Stefanoski, Shen Qiuqiong, and Joern Ostermann. Adaptation of a generic face model to a 3d scan, berlin, germany. In *ICOB 2005 - Workshop On Immersive Communication And Broadcast Systems*, 2005.

[24] Won-Sook Lee and Nadia Magnenat-Thalmann. Head modeling from pictures and morphing in 3d with image metamorphosis based on triangulation. In *Proc. of the International Workshop on Modelling and Motion Capture Techniques for Virtual Environments*, CAPTECH '98, pages 254–267, London, UK, UK, 1998. Springer-Verlag.

[25] T. Goto, S. Kshirsagar, and N. Magnenat-Thalmann. Automatic face cloning and animation. *IEEE Signal Processing Magazine*, 18(3):17–25, May 2001.

[26] Robert W. Sumner and Jovan Popović. Deformation transfer for triangle meshes. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 399–405, New York, NY, USA, 2004. ACM.

[27] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. Genova, Italy, 2009. IEEE.

[28] Sami Romdhani, Volker Blanz, and Thomas Vetter. Face identification by fitting a 3d morphable model using linear shape and texture error functions. In *in European Conference on Computer Vision*, pages 3–19, 2002.

[29] Kristina Scherbaum, Martin Sunkel, Hans-Peter Seidel, and Volker Blanz. Prediction of individual non-linear aging trajectories of faces. *Comput. Graph. Forum*, 26(3):285–294, 2007.

[30] Kristina Scherbaum, Tobias Ritschel, Matthias B. Hullin, Thorsten Thormählen, Volker Blanz, and Hans-Peter Seidel. Computer-suggested facial makeup. *Comput. Graph. Forum*, 30(2):485–492, 2011.

[31] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. of the Sixth International Conference on Computer Vision*, ICCV '98, pages 839–, Washington, DC, USA, 1998. IEEE Computer Society.

[32] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *IEEE Trans. Vis. Comp. Graph*, 14(1):213–230, 2008.

[33] J. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, March 1975.

[34] G. Bradski. Opencv 2.4.5 (open source computer vision). *Dr. Dobb's Journal of Software Tools*, 2000.

[35] M.Y. Waziri, W.J. Leong, M.A. Hassan, and M. Monsi. A new newton's method with diagonal jacobian approximation for systems of nonlinear equations. *Journal of Mathematics and Statistics*, 6:246–252, 2010.

[36] C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid Methods in Image Processing. 1984.