BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 67, No. 1, 2019
DOI: 10.24425/bpas.2019.127341

3D face reconstruction with region based best fit blending using mobile phone for virtual reality based social media

G. ANBARJAFARI[1, 2], R.E. HAAMER[1], I. LÜSI[1], T. TIKK[1], and L. VALGMA[1]*

[1] iCV Lab, Institute of Technology, University of Tartu, Tartu 50411, Estonia
[2] Department of Electrical and Electronic Engineering, Hasan Kalyoncu University, Gaziantep, Turkey

Abstract. The use of virtual reality (VR) has been exponentially increasing and due to that many researchers have started to work on developing new VR based social media. For this purpose it is important to have an avatar of the user which look like them to be easily generated by the devices which are accessible, such as mobile phones. In this paper, we propose a novel method of recreating a 3D human face model captured with a phone camera image or video data. The method focuses more on model shape than texture in order to make the face recognizable. We detect 68 facial feature points and use them to separate a face into four regions. For each area the best fitting models are found and are further morphed combined to find the best fitting models for each area. These are then combined and further morphed in order to restore the original facial proportions. We also present a method of texturing the resulting model, where the aforementioned feature points are used to generate a texture for the resulting model.

Key words: facial modeling, morphing, stretching, computer vision, similarity calculation, deformable model, 3D faces, virtual reality, blendshape.

1. Introduction

Reconstruction 3D face models has been a challenging task for the last two decades, because even a small amount of changes can have a huge effect on the recognizability and accuracy of the generated model. Therefore, perfectly modelling a 3D human face still remains one of the quintessential problems of computer vision. Also with the rapid increase of applications of virtual reality (VR) in gaming as well use use of VR application for making VR based social media, the task of making real-time 3D model of head which can be later mounted on avatar bodies have become very important [1–7]. So far the easiest, fastest and most accurate methods have benefited from depth information that can be captured with recording devices like RGB-D cameras [8–15] or 3D scanners [16–20]. However, nowadays development has been directed towards mobile devices, which limited to only using RGB information.

As an alternative commodity mobile phones that have their own on-device stereo cameras [21–24] can be used to recreate the depth data. Models created with passive multi-view stereo [25, 26] have distinct features but very rough surfaces. Due to the noisiness of the input data heavy smoothing needs to be applied, resulting models that do not have finer details around more complex features like the eyes and nostrils.

For 3D face reconstruction based on regular input images or videos, a large variety of methods and algorithms have been developed. The most conventional approach has been to use a 3D morphable head model [27, 28]. Even though each researcher has used different metrics and methods, the main idea has been to minimize the error between input image and the 2D render of the model. The models features are iteratively morphed to minimize the difference between the input image and the image with the render of the model, using suitable lighting and orientation, overlaid on the original face. A huge downside to these methods is that they require a lot of processing power and time [27, 29]. Another popular method is silhouette extraction, where the outer silhouette of a face is detected in video frames and the base model (without texture) is then constructed iteratively [30, 31]. In [32] and [33] landmark based shape reconstruction of 3D faces, which is very similar to RGB-D reconstruction, is used. In these cases features extracted using SIFT, SURF etc. were utilized. In some cases, the missing depth info was approximated using shading and lighting source estimation [34]. All of the above mentioned methods are of high complexity, which can result in longer computational time.

More efficient methods for face reconstruction generally rely on the detection of facial feature points or sparse landmarks and then stretching of a generic base model to produce a realistic output [35–38]. Among those are some that have even been specifically designed for mobile applications, but unfortunately the produced models tend to lack distinct features and the end result looks generic [39]. Even though recognizable result can be achieved using excellent texturing techniques, the base model itself looks nothing like the real person. The algorithm behind Landmark-based model-free 3D face shape reconstruction [40] tries to address the problem of generic looking outputs by avoiding statistical models as the base structure. They managed to produce genuine-looking models, but as the output is untextured and not a simple quadrilateral mesh structure, it is unusable for consumer applications.

*e-mail: shb@icv.tuit.ut.ee

Manuscript submitted 2017-12-01, revised 2018-03-19, initially accepted for publication 2018-04-21, published in February 2019.

In this paper we present a computationally light method for creating a 3D face model from video or image input, by constructing models for 4 key regions and later blending them. The method benefits from a high quality 3D scanned model database consisting of 200+ male head. As a pre-processing step all 4 key areas are individually rendered for each entry. As a first step of the method, the detected face is separated from the frame and 68 feature points are extracted. The face is divided into the 4 key regions, which are matched up against corresponding rendered areas of the database. These comparisons yield weights that are then used to combine a model from the predefined face regions. A stretching method similar to the ones described in [35–38] is the applied. The model is then further morphed using the 68 feature points and their corresponding vertexes on the model in order to restore facial proportions.

A texture is created by aligning the input images using piece-wise affine transformations to fit the UV map of the model. As an output our proposed method creates a 3D model with accurate facial features and fixed vertexes. This enables rigging, simple integration into existing systems or reintegration into the database while using minimal computational power. As a prerequisite, our method places some constraints on the input images and video, namely full visibility of the face with minimal facial hair and somewhat fixed lighting conditions.

2. Proposed method

In this section a novel methodology for making 3D models from videos or image sequences is proposed. This approach benefits from a unique reference database of 3D head scans. As illustrated in Fig. 1, the process can be divided into four main parts:

- feature extraction
- selection closest models
- model creation
- texture creation.

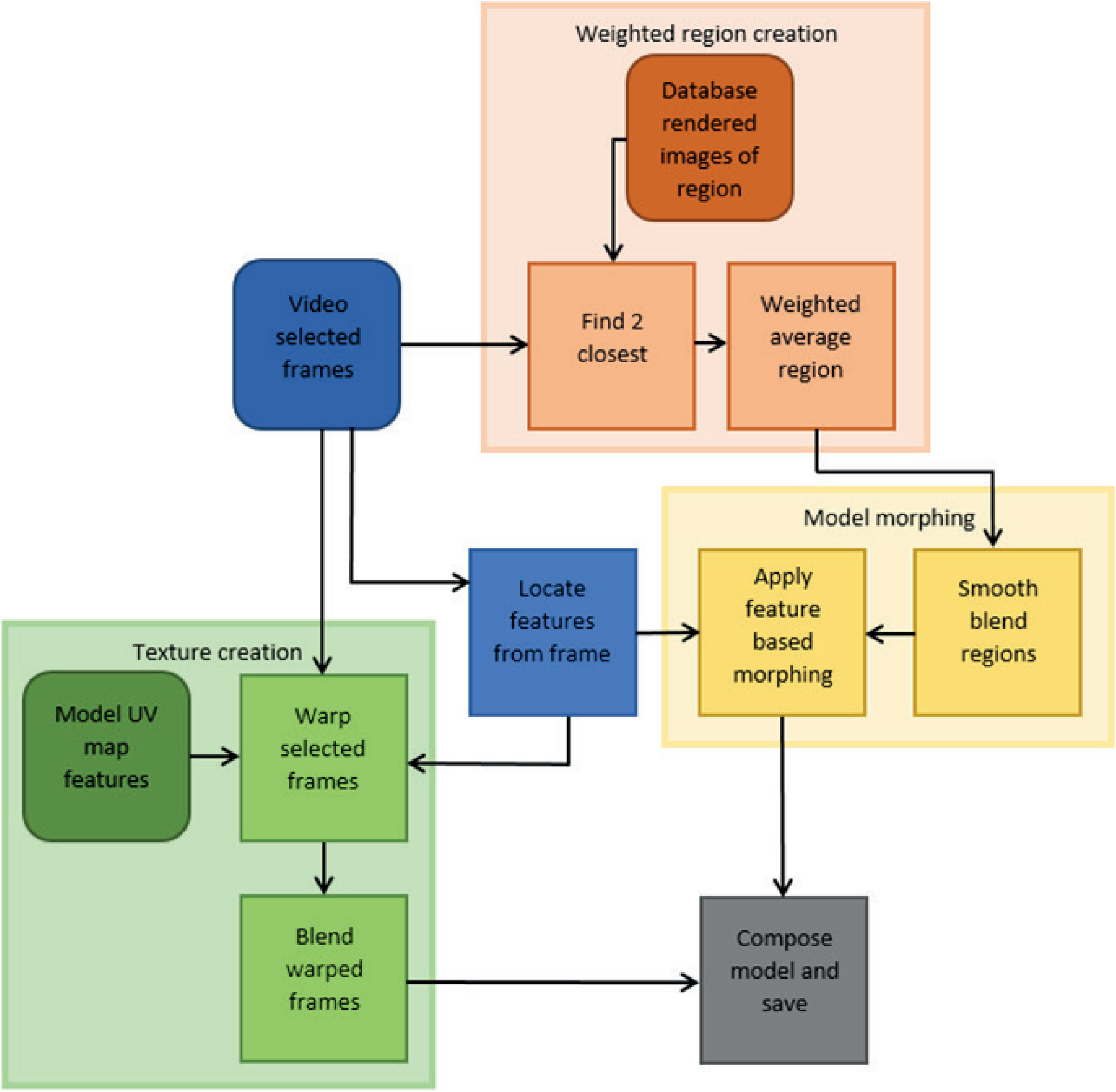

Fig. 1. Block diagram describing the overall process

2.1. Database

2.1.1. Data description. The preliminary database consisted of textured 3D scans of 217 mostly Caucasian male head area, that were morphed and simplified so all shared the same vertex mapping and count. All of the textures were also warped to fit the same UV map. In this step everything behind the head from the original scans was discarded, including the hair and the ears. In the original data some of the scans contained facial hair or severe noise/deformations. However the rest of the regions were usually not affected.

The pre-processed model consists of a mesh that contains 6000 quadrilateral faces and a corresponding 2048×2048 RGB texture. The indexes of the vertexes in all of the models are ordered and do not vary from model to model.

2.1.1. Rendering regions. An auxiliary database of rendered face regions was made in order to find the weights of each model based on the input image. All of the models were rendered in frontal orientation using perspective cameras with 5 directional light sources from the front, sides and top – all aimed towards the center of the head. The rendered images were then cut into 4 sections which correspond to the eyes, nose, mouth regions and rest of the face as shown in Fig. 2.

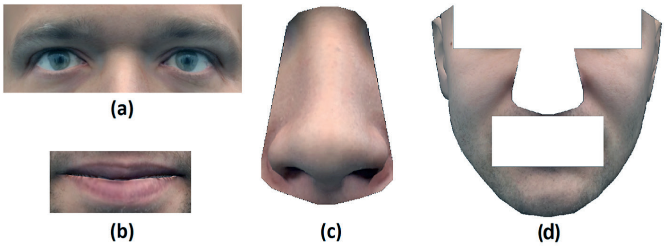

Fig. 2. Cut out sections of a rendered model from the model database. The 4 sections represent the eyes (a), mouth (b), nose (c) and the rest of the face (d) regions

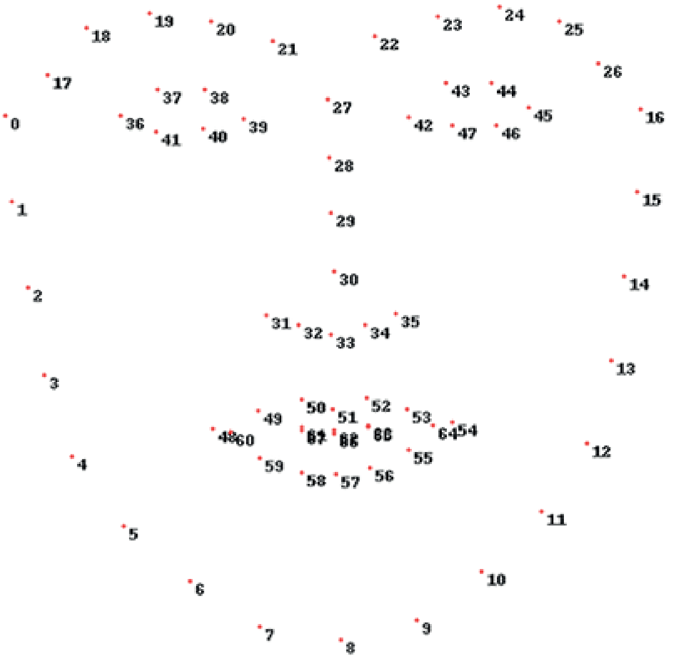

Fig. 3. The 68 feature points that are extracted

2.2. Feature extraction. For the facial feature extraction we use facial analysis toolkit [41] to estimate and extract 68 facial feature points from the input video sequence. Due to limitations of the toolkit, the input face must have a near neutral face, with no eye-wear or thick facial hair which obstruct the feature detection. These points are illustrated in Fig. 3. The algorithm also outputs the rotation parameters, so this can be used to pick the frontally best aligned frame as the main reference for creating the model. This also allows alignment of the face to a frontal one in case of slight tilt or rotation.

As the next step the area containing feature points is extracted and the feature points from that area are normalized to fit between 0 and 1.

2.3. Region-based selection and weight calculation. In this step we calculate similarities between database regions and the input grayscale regions. There are altogether three different similarity indexes used: PCA based measure, SSIM [42] (structural similarity index) and a LBP(local binary pattern) histograms difference measure [43].

In the PCA based approach the database images are first vectorized and the covariance matrix of these vectors is found. The principal components and score vectors are stored for future use. When looking for the closest match the corresponding image region is also vectorized and its score vector on the principal components is found. We treat the Eucleidean difference between the score vectors as the error and its' inverse as the similarity measure.

Since the SSIM finds the similarity between two images, the maximum being one, we use the dissimilarity $(1 - ssuim)$ as the error. In case of LBP features we also apply PCA to the output feature and find the distances as described earlier.

For picking the closest mouth and nose regions, SSIM was used. For eyes we used LBP as they have the most high frequency data and for face shape we used PCA based approach as the placement zero and non-zero values has high impact on this measure.

For each region a similarity function between database rendered regions and the regions extracted from the frontal face is applied. After which the corresponding similarity function, an error vector is obtained for each region. Let us denote it by $E = (E_1, E_2, \ldots, E_n)$, where n is the number of reference heads in the database. Let I denote the set of indexes of the database models corresponding to smallest error. The weights W_i, $i \in I$ are calculated according to the formula:

$$W_i = \frac{E_i^{-1}}{\sum_{j \in I} E_j^{-1}}. \tag{1}$$

The rest of the weights are set to zero: $W_i = 0, \forall i \notin I$.

Based on the errors of the similarity three (or in some cases 1) closest matches are picked for each region, and weights that are inversely proportional to the errors are assigned. The weights are normalized to they sum to 1.

A single match was picked for some facial features when the other closest matches produced unfavourable similarity values. This was purely caused by the sparsity of the underlying model database and only affected unique facial features. There is no specific threshold specified for determining when to only select a single match, as this was not an intended feature.

2.4. Blended model creation. The model is separated into 5 primary regions as shown in Fig. 4, out of these regions, only the eyes, nose, mouth regions and rest of the face area are used. For each region 3D models AM are created as blendshapes [30] from weighted combinations of an array of models $M = (M_0, M_1, M_2, \ldots, M_n)$ and their corresponding weights $W = (W_0, W_1, W_2, \ldots, W_n)$, as in formula 2. Each model (AM) has its corresponding region of interest signified with an array of indexes called I_A.

Out of the four models created, the one representing the rest of the face will be called the base model BM. The BM will be the starting model, on which the other weighted models are added one-by-one. The base model BM also has its own corresponding index array called I_B.

$$BM = M_i \cdot W_i^T. \tag{2}$$

Then regions from these composite 3D models are combined together to form the overall base shape of the 3D model as shown in Fig. 5a.

To minimize the effect of large transitions between regions, mean locations are calculated for the overlapping vertexes of 2 combined regions. The vertex indexes for this new region will be denoted as $I_{AB} = I_A \cap I_B$.

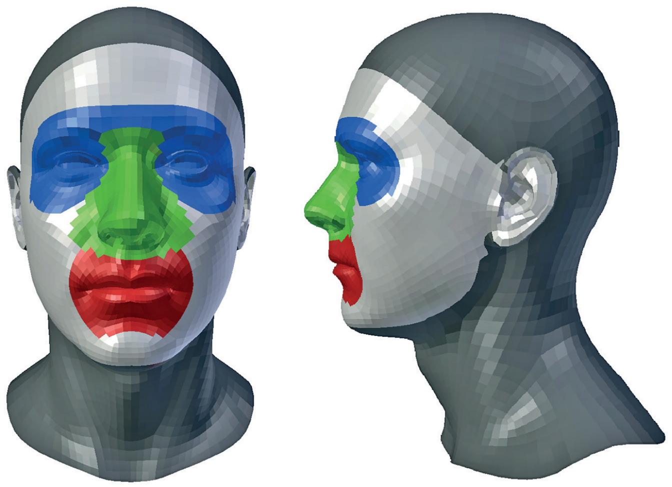

Fig. 4. The 5 primary regions as shown in colors are used for linear weighted combinations to generate the base model. The eye region is shown in blue, the nose region is shown in green, the mouth region is shown in red, the rest of the face region is shown in white and the unused parts of the model are shown in gray

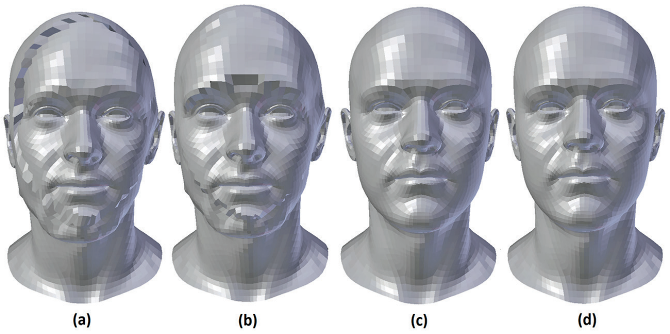

Fig. 5. Processing steps for blending and morphing the 3D model regions to create an output model. The weighted model regions are combined to form the base shape of the output model (a). To ease the transitions between different regions the regions are first aligned with one another (b) and then the overlapping areas are blended together using a blending function (c). The final combined model is then further morphed (d) to fit the input image

The vector originating from the mean point of the added region to the mean of the existing region is found using formula 3.

$$\vec{s}_{BA} = \frac{1}{|I_{AB}|} \cdot \sum_{i \in I_{AB}} \left(v_i^B - v_i^A\right). \tag{3}$$

To reduce the large effect of small locational variations that have later a big effect on the smoothing process, the vector $^1\!/_2 s_{BA}$ is added to the points in the added region using the formula 4.

$$v_i^A = v_i^A + \frac{\vec{s}_{BA}}{2} \quad i \in I. \tag{4}$$

To further smooth the transition between different regions, a blending function is applied when calculating the model weights for each vertex. This can be seen as the transition from model (b) to (c) in Fig. 5. In order to scale the blending function for the added regions area, the maximum translations along the X and Y axes of the added region are calculated.

$$\Delta_x = \max_{i \in I_A}\left\{x_i^A\right\} - \min_{i \in I_A}\left\{x_i^A\right\}, \tag{5}$$

where x_i^A is the X axis component of v_i^A. Δ_y is found using a similar function as 5 and they will be denoted as $\Delta = (\Delta_x, \Delta_y)$.

The distances of all of the added region points from the their mean can then be calculated and normalized using the function 6. From now on j will denote the index of a vertex from the model.

$$\delta_j = \left\| \frac{\mu_{AM_{I_A}} - v_i^A}{\Delta} \right\| \quad i \in I. \tag{6}$$

As all of the distances will be in reference to the mean of the added region, similar calculations will not be applied to the base region.

A transition weight function $twf(x)$ is then applied to each of the point distances to get the weights for each vertex. The two regions are then combined using:

$$twf(\delta_j) = 1.013 - \frac{1.019}{1 + \left(\frac{\delta_i}{0.264}\right)^{3.244}} \tag{7}$$

$$BM_j = BM_j \cdot (1 - twf(\delta_j)) + AM_j \cdot twf(\delta_j). \tag{8}$$

Since the nose region has some overlapping areas with the eye and the mouth region, it is added to the base model first. Next the eye and mouth regions are added, though here the order plays no role as they don't share any vertexes.

The function for $twf(x)$ and later $awf(d_{ji})$ were created by reverse generating smooth transition functions for the facial regions. The constant coefficients in both functions were selected based on a combination of the the working model's scale and visual feedback.

2.5. Model morphing. The generated blended model BM has features which resemble the desired face, but some of the proportions are off. Due to that the model is further morphed so key features match their actual locations. The model has 68 vertexes mapped to 68 feature points $F = \{F_i\}\ i \in \{1, 2, \ldots, 68\}$ where value of i will remain the same for all of the following functions. For each mapped vertex a movement vector to the feature location is calculated as:

$$\vec{f_i} = F_i - BM_i. \tag{9}$$

For each of the points in the model, a distance is calculated from each mapped vertex and a adjustment weight function $awf(x)$ is applied to get the weight of each vector for each vertex.

$$d_{ji} = |F_i - BM_j| \tag{10}$$

$$awf(d_{ji}) = 1 - \left(1 + e^{\left(\frac{-d_{ji}}{\sigma_i^2 \cdot k} + 0.5\right) \cdot 7}\right)^{-1} \tag{11}$$

Where σ_i defines the drop-off rate for the adjustment weight function, which is separately defined for each feature point, k is a similar multiplier but it is independent from the feature point values. The vectors $\vec{f_i}$ are then multiplied by the weights for each point and summed together:

$$\vec{s_j} = \sum_{i=1}^{n} \left(awf(d_{ji}) \cdot \vec{f_i}\right). \tag{12}$$

Then the sum of these weights r_j is calculated by:

$$r_j = \sum_{i=1}^{n} awf(d_{ji}). \tag{13}$$

The resulting vector s_j is further divided by the sum of the weights, if $r_j > 1$. This is done to keep densely packed features from overwhelming vertexes with their respective changes. The final vectors are added to every point in the model.

$$BM_j = \begin{cases} BM_j + \vec{s_j}, & \text{if} \quad r_j \le 1 \\ BM_j + \frac{\vec{s_j}}{r_j}, & \text{if} \quad r_j > 1 \end{cases} \tag{14}$$

Since some feature vertexes define larger areas like the eyebrows and chin, while others define finer details like the eyes and nostrils, the features also have their own weights called σ_i. These weights allow for different features to have different adjustment weight functions as shown in the formula 11. The morphing is applied 3 times with 3 different k values, starting with more general features and ending with smaller details, where the adjustment weight function sets the weights of points that have larger σ_i values to zero.

2.6. Texture creation. The texture of any 3D model is strongly bound by the UV map. By hand it is difficult to morph images into a form that would correspond to this map. In this work an average texture was first created based on the database textures Fig. 7a. Since all of the models in the database have a corresponding UV map, the input image had to be transformed to fit the same map. In order to warp the input image appropriately, 68 vertexes on the UV map were marked corresponding to the 68 feature points on the input image.

For texture creation we assume that there are no distinctive shadows present in the video sequence. Based on the extracted feature points and rotations, three frames are picked from the sequence based on yaw at approximately 30 degrees on both sides and the frontal frame, as in Fig. 6. From the face area also the median face color was extracted.

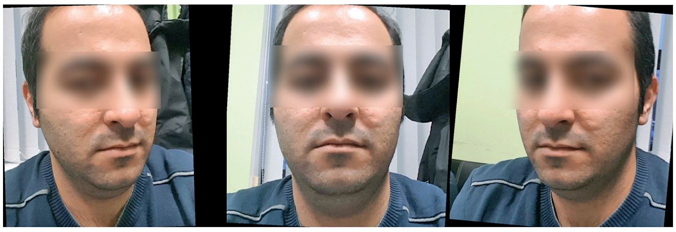

Fig. 6. An example of images picked to combine

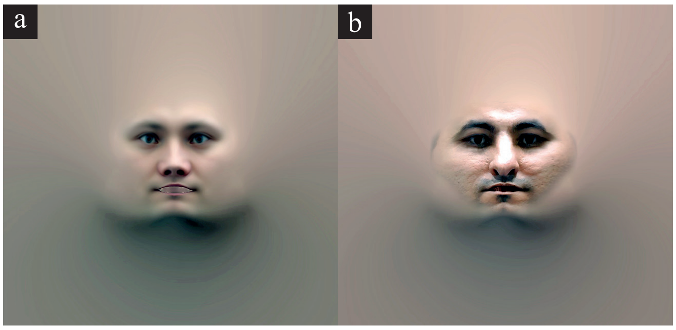

Fig. 7. The average texture created from the existing textures in the database on the left and the input image fitted to the UV map on the left

As the next step the average texture colors are shifted to match the median skin tone to assure a smooth and natural transition on the edges of the skin. Then the frames are piece-wise affinely transformed into an image of size 2048×2048 and blended together into a partial texture. This however has very sharp edges and lots of black pixels, so it is then blended only on the edges with the underlying and already shifted average texture. The average texture and the resulting texture can be seen in Fig. 7b.

3. Experimental results

We evaluated our method with several faces, some of which were included in the original scanned database. This was done in order to test the method's model selection and reconstruction proficiency.

For the faces of people included in the 3D scanned database, the system selected most of the facial features that matched the person. In some cases a different mouth or a set of eyes were chosen when the scanned emotion and the emotion in the captured image were different. This is illustrated by Fig. 8, where the participant had a more surprised facial expressions during the scan and a neutral face during the recording. Regardless, the resulting models were nearly identical to the original scanned versions of those faces.

In case of subjects that were not in the database, the system managed to choose facial features that were remarkably similar to the desired face. On occasions where no corresponding feature existed in the database, a blended version from 2 nearest features was created. The result of this blending were not always very similar to the original. This shortcoming can be seen Fig. 9, for the nose in the first row had no structurally similar nose in the entire database.

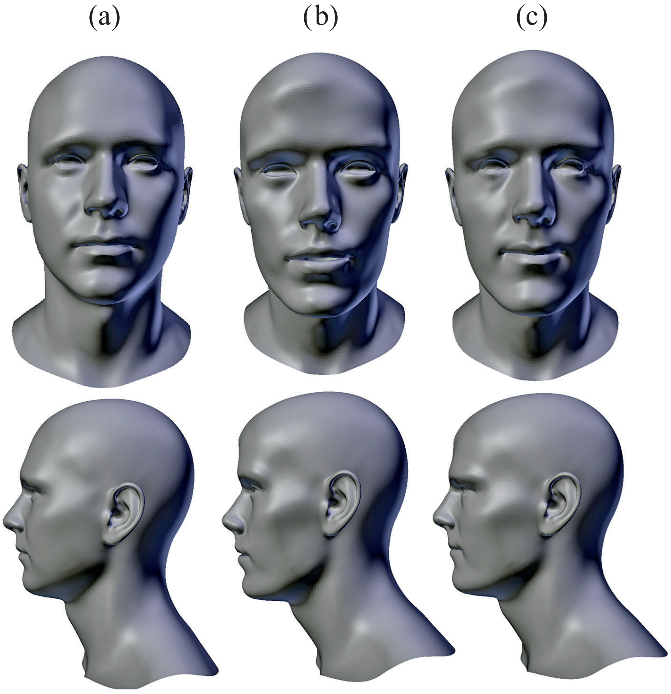

Fig. 8. An average model from the scanned database (a), a scanned model (b) and the reconstructed version using a mobile phones camera (c)

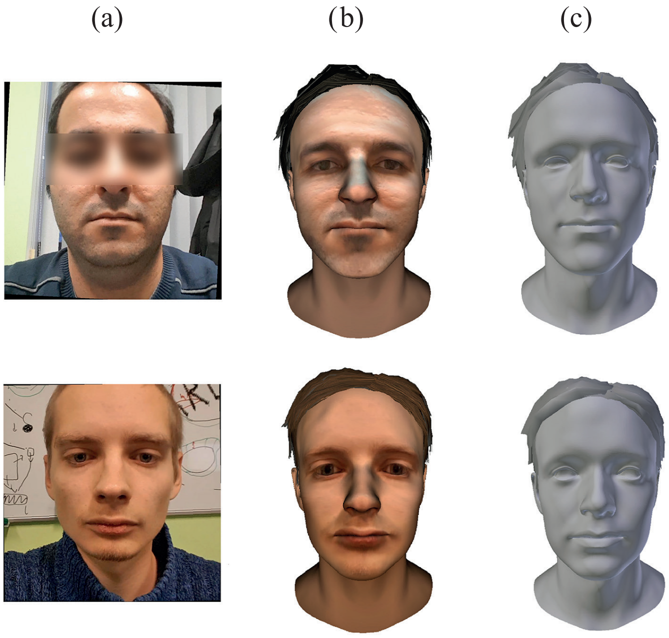

Fig. 9. Textured (b) and untextured (c) models generated from the example input images (a) using the developed method. Both models have a manually created hair mesh. Neither participants had scanned counterparts in the model database

The goodness of the method was largely limited by the feature detection and the nature of the database. The database contained very little information on the general shape of the head and the feature detection method was very unreliable when it came to detecting the shape of larger areas like the chin. Because of that the resulting heads tended to have the same overall structure. This is very visible in case of the model in the first row in Fig. 9 that has a jawline which is smaller and sharper jawline than expected.

The recordings were conducted with a Samsung Galaxy S7 and the model creation process was run on an i7-4790@3.6GHz CPU with MATLAB 2015a. The whole process, including the recording of a 10 second clip, took about 3 minutes per model. For an i7@4.2GHz with MATLAB 2016b the model generation took less than a minute.

4. Conclusion

In this paper, a method of recreating 3D facial models from portrait data, which uses very little computational power in order to produce a recognizable facial model, was proposed. The main idea behind our system is finding the closest matching models for different facial regions and then combining them into a single coherent model. We have also presented a novel and simple method of texturing the resulting model by piecewise affinely transforming input images to fit a desired UV map based on detected feature points.

Our experimental evaluation has shown that our method is able to select the best corresponding facial features within a reasonable time frame. Our method has also partly overcome the problem of average looking models, which is a huge obstacle in methods that only use stretching to recreate shape. Unfortunately our method is still susceptible to problems relating to the overall shape of the head. This can be fixed with better facial feature extraction methods and with a database that includes a good selection of head shapes. The method can easily be made more robust by rendering the regions with more refined parameters.

Acknowledgements. This work has been partially supported by Estonian Research Council Grants (PUT638), The Scientific and Technological Research Council of Turkey (TÜBİTAK) (Proje 1001–116E097), the Estonian Centre of Excellence in IT (EXCITE) funded by the European Regional Development Fund and the European Network on Integrating Vision and Language (iV&L Net) ICT COST Action IC1307.

References

[1] J.L. Olson, D.M. Krum, E.A. Suma, and M. Bolas, "A design for a smartphone-based head mounted display," in *Virtual Reality Conference (VR), 2011 IEEE*, pp. 233–234.

[2] B.S. Santos, P. Dias, A. Pimentel, J.-W. Baggerman, C. Ferreira, S. Silva, and J. Madeira, "Head-mounted display versus desktop for 3d navigation in virtual reality: a user study," *Multimedia Tools and Applications*, 41(1), p. 161 (2009).

[3] J.-S. Kim and S.-M. Choi, "A virtual environment for 3d facial makeup," *Virtual Reality*, pp. 488–496 (2007).

[4] G. Anbarjafari, "An objective no-reference measure of illumination assessment," *Measurement Science Review*, 15(6), 319–322 (2015).

[5] B.J. Fernández-Palacios, D. Morabito, and F. Remondino, "Access to complex reality-based 3d models using virtual reality solutions," *Journal of Cultural Heritage*, 23, 40–48 (2017).

[6] D. Zeng, H. Chen, R. Lusch, and S.-H. Li, "Social media analytics and intelligence," *IEEE Intelligent Systems*, 25(6), 13–16, (2010).

[7] D. Trenholme and S.P. Smith, "Computer game engines for developing first-person virtual environments," *Virtual reality*, 12(3), 181–187 (2008).

[8] E. Avots, M. Daneshmand, A. Traumann, S. Escalera, and G. Anbarjafari, "Automatic garment retexturing based on infrared information," *Computers & Graphics*, 59, 28–38 (2016).

[9] T. Yamasaki, I. Nakamura, and K. Aizawa, "Fast face model reconstruction and synthesis using an rgb-d camera and its subjective evaluation," in Multimedia (ISM), *IEEE International Symposium on*, pp. 53–56 (2015).

[10] S. Ding, Y. Li, S. Cao, Y.F. Zheng, and R.L. Ewing, "From rgbd image to hologram," in *Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS), 2016 IEEE National*, pp. 387–390.

[11] X. Huang, J. Cheng, and X. Ji, "Human contour extraction from rgbd camera for action recognition," in Information and Automation (ICIA), *IEEE International Conference on*, pp. 1822–1827 (2016).

[12] L. Valgma, M. Daneshmand, and G. Anbarjafari, "Iterative closest point based 3d object reconstruction using rgb-d acquisition devices," in *Signal Processing and Communication Application Conference (SIU), 2016 24th*, pp. 457–460.

[13] C. Ding and L. Liu, "A survey of sketch based modeling systems," *Frontiers of Computer Science*, 10(6), 985–999 (2016).

[14] I. Lüsi and G. Anbarjafari, "Mimicking speaker's lip movement on a 3d head model using cosine function fitting," *Bul. Pol. Ac.: Tech.*, 65(5), 733–739 (2017).

[15] M. Daneshmand, E. Avots, and G. Anbarjafari, "Proportional error back-propagation (peb): Real-time automatic loop closure correction for maintaining global consistency in 3d reconstruction with minimal computational cost," *Measurement Science Review*, 18(3), 86–93 (2018).

[16] V. Blanz, K. Scherbaum, and H.-P. Seidel, "Fitting a morphable model to 3d scans of faces," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8.

[17] A. Traumann, M. Daneshmand, S. Escalera, and G. Anbarjafari, "Accurate 3d measurement using optical depth information," *Electronics Letters*, 51(18), 1420–1422 (2015).

[18] M. Daneshmand, A. Aabloo, C. Ozcinar, and G. Anbarjafari, "Real-time, automatic shape-changing robot adjustment and gender classification," Signal, *Image and Video Processing*, 10(4), 753–760 (2016).

[19] I. Fateeva, M.A. Rodriguez, S.R. Royo, and C. Stiller, "Applying 3d least squares matching technique for registration of data taken with an 3d scanner of human body," in *Sensors and Measuring Systems 2014; 17. ITG/GMA Symposium; Proceedings of*, VDE, pp. 1–5 (2014).

[20] M. Daneshmand, A. Helmi, E. Avots, F. Noroozi, F. Alisinanoglu, H.S. Arslan, J. Gorbova, R.E. Haamer, C. Ozcinar, and G. Anbarjafari, "3d scanning: A comprehensive survey," *arXiv preprint arXiv:1801.08863*, 2018.

[21] K. Kolev, P. Tanskanen, P. Speciale, and M. Pollefeys, "Turning mobile phones into 3d scanners," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3946–3953 (2014).

[22] P. Ondrúška, P. Kohli, and S. Izadi, "Mobilefusion: Real-time volumetric surface reconstruction and dense tracking on mobile phones," *IEEE transactions on visualization and computer graphics*, 21(11), 1251–1258 (2015).

[23] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys, "Live metric 3d reconstruction on mobile phones," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, 65–72 (2013).

[24] H. Zhu, Y. Nie, T. Yue, and X. Cao, "The role of prior in image based 3d modeling: a survey," *Frontiers of Computer Science*, 11(2), 175–191 (2017).

[25] F. Maninchedda, C. Häne, M.R. Oswald, and M. Pollefeys, "Face reconstruction on mobile devices using a height map shape model and fast regularization," in *3D Vision (3DV), 2016 Fourth International Conference on*, IEEE, pp. 489–498 (2016).

[26] H. Jain, O. Hellwich, and R. Anand, "Improving 3d face geometry by adapting reconstruction from stereo image pair to generic morphable model," in *Information Fusion (FUSION), 2016 19th International Conference on*, IEEE, pp. 1720–1727 (2016).

[27] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., pp. 187–194 (1999).

[28] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling, "A 3d morphable eye region model for gaze estimation," in *European Conference on Computer Vision*, Springer, pp. 297–313 (2016).

[29] V. Blanz and T. Vetter, "Face recognition based on fitting a 3d morphable model," *IEEE Transactions on pattern analysis and machine intelligence*, 25(9), 1063–1074 (2003).

[30] J.P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F.H. Pighin, and Z. Deng, "Practice and theory of blendshape facial models," *Eurographics (State of the Art Reports)*, 1, 8 (2014).

[31] C. Baumberger, M. Reyes, M. Constantinescu, R. Olariu, E. de Aguiar, and T.O. Santos, "3d face reconstruction from video using 3d morphable model and silhouette," in *Graphics, Patterns and Images (SIBGRAPI), 2014 27th SIBGRAPI Conference on*, IEEE, pp. 1–8 (2014).

[32] P. Dou, Y.Wu, S.K. Shah, and I.A. Kakadiaris, "Robust 3d face shape reconstruction from single images via two-fold coupled structure learning," in *Proc. British Machine Vision Conference*, pp. 1–13 (2014).

[33] J. Choi, G. Medioni, Y. Lin, L. Silva, O. Regina, M. Pamplona, and T.C. Faltemier, "3d face reconstruction using a single or multiple views," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, IEEE, pp. 3959–3962 (2010).

[34] I. Kemelmacher-Shlizerman and R. Basri, "3d face reconstruction from a single image using a single reference face shape," *IEEE transactions on pattern analysis and machine intelligence*, 33(2), 394–405 (2011).

[35] Q. Zhang and L. Shi, "3d face model reconstruction based on stretching algorithm," in *Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on*, IEEE, 1, 197–200 (2012).

[36] W. Lin, H. Weijun, C. Rui, and W. Xiaoxi, "Three-dimensional reconstruction of face model based on single photo," in *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, IEEE, 3, V3–674 (2010).

[37] X. Fan, Q. Peng, and M. Zhong, "3d face reconstruction from single 2d image based on robust facial feature points extraction and generic wire frame model," in *Communications and Mobile Computing (CMC), 2010 International Conference on*, IEEE, 3, 396–400 (2010).

[38] T. Wu, F. Zhou, and Q. Liao, "A fast 3d face reconstruction method from a single image using adjustable model," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 1656–1660 (2016).

[39] C. Qu, E. Monari, T. Schuchert, and J. Beyerer, "Fast, robust and automatic 3d face model reconstruction from videos," in *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*, pp. 113–118 (2014).

[40] C. van Dam, R. Veldhuis, and L. Spreeuwers, "Landmark-based model-free 3d face shape reconstruction from video sequences," in *Biometrics Special Interest Group (BIOSIG), 2013 International Conference of the*, IEEE, pp. 1–5 (2013).

[41] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Openface: an open source facial behavior analysis toolkit," in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pp. 1–10.

[42] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, 13(4), 600–612 (2004).

[43] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 971–987 (2002).