

Automatic Bootstrapping of a Morphable Face Model using Multiple Components

Frank B. ter Haar

Electro-Optics, TNO Defence Security and Safety
The Hague, The Netherlands

Frank.terHaar@tno.nl

Remco C. Veltkamp

Information and Computing Sciences
Utrecht University, the Netherlands

Remco.Veltkamp@cs.uu.nl

Abstract

We present a new bootstrapping algorithm to automatically enhance a 3D morphable face model with new face data. Our algorithm is based on a morphable model fitting method that uses a set of predefined face components. This fitting method produces accurate model fits to 3D face data with noise and holes. In the fitting process, the dense point-to-point correspondences between the scan data and the face model may become less reliable at the borders of components. We solve this by introducing a blending technique that improves on the distorted correspondences close to the borders. Afterwards, a new face instance is acquired similar to the 3D scan data and in full correspondence with the face model. These newly generated face instances can then be added to the morphable face model to build a more descriptive one. To avoid our bootstrapping algorithm from needlessly adding redundant face data, we incorporate a redundancy estimation algorithm. We tested our bootstrapping algorithm on a set of scans acquired with different scanning devices, and on the UND data set. Quantitative and qualitative evaluation shows that our algorithm successfully enhances an initial morphable face model with new face data, in a fully automatic manner.

1. Introduction

Statistical models of the human face have proven to be an effective tool for person identification of 3D face scans. To build a statistical model, a set of example faces is required with face features in full correspondence. With such a model, a new face instance can be constructed as a linear combination of the example faces. For 3D face identification, the idea is to use the statistical model to construct a face instance that resembles an input image. The way these example faces are combined linearly to represent an input face, provides both global and local information about the

input face, that can be used to classify and identify different input faces. For a statistical face model to be applicable in face recognition systems all over the world, it is important to include example data on all possible face variations. Since this is an intractable task, a flexible model is required that updates itself in case of new example faces. For that, a system should automatically fit the face model to face data, estimate dense and accurate correspondences beyond the linear combinations of current example data, and measure the redundancy of the new example faces. When full correspondence between the face model and the scan data is established and the new face instance is not redundant, it can be added as a new example to the statistical face model, increasing the descriptiveness of the model. The process of using a statistical model to enhance itself automatically, is referred to as *bootstrapping* the synthesis of the model [14]. The difficulty of bootstrapping is that: (1) If the model (as is) fits a new example well, there is no use of adding the new example to the model. This must be automatically verified. (2) If the model doesn't fit the new example, the correspondences are incorrect and the example cannot be added to the model. (3) It should be fully automatic. Nowadays, several statistical models are available, ready to be used and reused. In this work we present a bootstrapping algorithm based on an initial statistical model, which automatically fits to new scan data with noise and holes, and which is capable of measuring the redundancy of new example faces.

1.1. Related work

The need for bootstrapping statistical models was posed by Vetter et al. [14]. They introduced a bootstrapping algorithm for statistical models, and showed that the use of merely an optic flow algorithm was not enough to establish full correspondence between example faces and a reference face. Instead, they attain an effective bootstrapping algorithm by iteratively fitting the face model, applying the optic flow algorithm, and updating the face model. Blanz and Vetter also used this bootstrapping algorithm in [3] to build a 3D morphable face model.

¹This research was conducted at the Utrecht University

Their bootstrapping algorithm works well in case of input data with constant properties, but fails when input data is incomplete and when the optic flow algorithm fails. To bootstrap the 3D morphable face model with more general face data, Basso et al. [2] added a smoothness term to regularize the positions of the vertices where the optic flow correspondence is unreliable. In case a 3D morphable face model is not yet available, a reference face can be used as an approximation instead, which is a major advantage.

Amberg et al. [1] proposed a non-rigid Iterative Closest Point (ICP) algorithm to establish dense correspondences between a reference face and face scans, but they need an initial rigid transformation for the reference face based on 14 manually selected landmarks. Afterwards, the reference face and the fitted face instances can be used to construct a new morphable face model.

Huang et al. [8] proposed a global to local deformation framework to deform a shape with an arbitrary dimension (2D, 3D or higher) to a new shape of the same class. Their method also operates in the space of implicit surfaces, but uses a non-statistical deformation model. They show their framework’s applicability to 3D faces, for which they deform an incomplete source face to a target face.

The use of multiple components has been used by Blanz et al. to improve the face model fitting [3] and for face recognition purposes [4], but so far the resulting face instances were not accurate enough to be incorporated in the statistic model. The explicit point-to-point correspondences of the fitted face instance and the statistical model had to be established by techniques based on optic flow or non-rigid ICP.

In our previous work [10] a set of predefined face components was used to increase the descriptiveness of a 3D morphable face model. With the use of multiple components, a tighter fit of the face model was obtained and higher recognition rates were achieved. However, by fitting each component individually, components started to intersect, move apart, or move across. So, afterwards the full point-to-point correspondences between the morphable model and the fitted instance were distorted. The post-processing method to blend the borders of the components introduces a new set of surface samples without correspondence to the model either.

1.2. Contribution

We present a new bootstrapping algorithm that can be applied to general 3D face data. Our algorithm automatically detects if a new face scan cannot be sufficiently modeled, establishes full correspondence between the face scan and the model, and enhances the model with this new face data. Compared to previous work, our algorithm is fully automatic, reuses initial face statistics, checks for redundancy, and retains the full correspondence even in case of noisy

scan data with holes.

Without the use of unreliable optic flow [2] or semi-automatic non-rigid ICP [1], we are able to bootstrap the 3D morphable face model with highly accurate face instances. As a proof of concept, we (1) fit the initial morphable face model to several 3D face scans using multiple components, (2) blend the components at the borders such that accurate point-to-point correspondences with the model are established, (3) add the fitted face instances to the morphable model, and (4) fit the enhanced morphable model to the scan data as one single component. In the end, we compare each single component fit obtained with the enhanced morphable model to the single component fit obtained with the initial morphable model. Qualitative and quantitative evaluation shows that the new face instances have accurate point-to-point correspondences that can be added to the initial morphable face model. By comparing the multiple and single component fit, our bootstrapping algorithm automatically distinguishes between new face data to add and redundant data to reject. This is important to keep both the model fitting and the face recognition with model coefficients time-efficient.



Figure 1. Changing weight $w_3 \{-2,0,+2\}$ of the USF model causes an unwanted change in the gaze direction.

2. Morphable face model

In this work we fit the morphable face model of the USF Human ID 3D Database [13] to 3D scan data to obtain a clean model of the face scan, that we use to identify 3D faces. This statistical point distribution model (PDM) was built from 100 cylindrical 3D face scans with neutral expressions from which $n=75,972$ correspondences were selected using an optic flow algorithm. Each face shape S_i was described using the set of correspondences $S = (x_1, y_1, z_1, \dots, x_n, y_n, z_n)^T \in \mathbb{R}^{3n}$ and a mean face \bar{S} was determined. Principal Component Analysis (PCA) was applied to these 100 sets S_i to obtain m eigenvectors of the PDM. Because there are only 100 faces in the n dimensional face space, there are at most $m=99$ meaningful eigenvectors. The mean face \bar{S} , the eigenvectors $s_i = (\Delta x_1, \Delta y_1, \Delta z_1, \dots, \Delta x_n, \Delta y_n, \Delta z_n)^T$, the eigenvalues λ_i ($\sigma_i^2 = \lambda_i$) and weights w_i are used to model new face instances according to $S_{inst} = \bar{S} + \sum_{i=1}^m w_i \sigma_i s_i$. Weight w_i , also referred to as coefficient, represents the number of standard deviations a face instance morphs along eigenvector s_i . Since the connectivity of the n correspondences in the PDM is known, each instance is a triangular

mesh with proper topology and without holes.

Fig. 1 shows the changes of the original face model when the weight of the third eigenvector (w_3) is varied. It can be noticed that the model is tilted upwards and downwards. This variation in one of the first eigenvectors means that the alignment of the 100 sets S_i is not optimal for face identification using model coefficients. Furthermore, the morphable face model has $n=75,972$ vertices that cover the face, neck and ear regions and its resolution in the upward direction is three times higher than in its sideways direction. Because the running time of the model fitting method is dependent on the number of vertices, and frontal face scans cover merely the face region, we recreated the morphable face model such that it contained only the face (data within 110 mm from the tip of the nose) and not the neck and ears. To obtain a more uniform resolution for the model, we reduced the upward resolution to one third of the original model. To adjust the original model, we selected from the mean face \bar{S} the reduced set of $k=12,956$ correspondences $\bar{S}' = (x_1, y_1, z_1, \dots, x_k, y_k, z_k)^T \in \mathbb{R}^{3k}$ that satisfy the above conditions, realigned each reduced face shape S'_i to \bar{S}' using the ICP algorithm, and recomputed the PCA model. Visual inspection of our newly constructed PCA model showed no signs of pose variations.

3. Face scans

We fit the morphable face model to 3D scan data from the UND [6], GAVAB [9], BU-3DFE [15], Dutch CAESAR [5], and our local dataset. From all except the UND set, we randomly select four scans yielding a first test set of 16 scans. These scans vary in pose, facial expression, resolution, accuracy, and coverage. This set of 16 face scans is used to test our bootstrapping algorithm. To test the automatic redundancy check, we used a subset of 277 face scans from the UND dataset, namely the first scan of each new subject.

To segment the 3D face data from these scans, we apply the pose normalization method described in [11]. This method randomly samples the surface mesh, such that every $\approx 2.0 \text{ mm}^2$ of the surface is approximately sampled once. We use these locations in combination with their surface normal as initial placements for a nose tip template. To locations where this template fits well, a second template of global face features is fitted to normalize the face's pose and to select the tip of the nose. The face is then segmented by removing the scan data with a Euclidean distance larger than 110 mm from the nose tip. In the end the face data is centered with the localized tip of the nose in the origin. To remove some of the speckle noise, we applied five iterations of Taubin's $\lambda|\mu$ (0.330|-0.331) smoothing method.

4. Bootstrapping algorithm

The main problem in bootstrapping the 3D morphable face model, is that (1) we only want to add example faces that are not covered by the current model, (2) new example faces suffer from noise and missing data, which makes it hard to establish the required point-to-point correspondences, and (3) it should be fully automatic. In this section, we briefly describe the used model fitting method, then we explain our algorithm to establish dense point-to-point correspondences between the multiple component fits and the morphable face model, and finally, we explain how the bootstrapping algorithm can distinguish between new face data to add to the model and redundant data to reject.

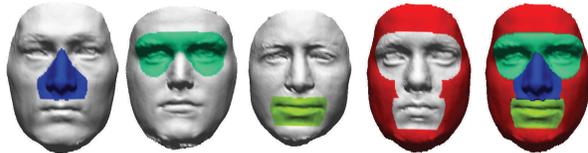


Figure 2. Model Fitting. Four face instances S_{comp} , are constructed during the model fitting process each with the focus on a smaller subset of vertices. The composition of these components provides an accurate fit.

4.1. Model fitting

The morphable face model that we use (see Sect. 2) has $m=99$ coefficients that can be varied to fit the face model to new scan data. To fit the model, we bring a new 3D face scan into alignment with the 3D face model automatically. First, we automatically normalize the pose of the face, detect the tip of the nose, and segment the face as described in Sect. 3. Secondly, we align the face scan to the face model, coarsely by using their nose tips and more accurately using the ICP algorithm. Then we apply a model fitting algorithm similar to [10]. With a set of four face components a more accurate fit is achieved than with a single component. This model fitting algorithm iteratively adjusts the m coefficients w_i for each component, such that the vertices move closer to the vertices of the scan data. After all components are fitted to the scan data individually, an accurate representation of the new face S_{fine} is acquired. Each component can then be described using the set of m coefficients w_i for the eigenvectors of the face model. However, the fitted face instance S_{fine} may show artifacts at the borders of these fitted component. Note that we use the same PCA model for each component, but with a subset of the model's vertices only.

4.2. Correspondence estimation

After the application of the model fitting method, most of the face model's vertices are brought into correspondence with the face scan, but at the component's borders these point-to-point correspondences are inaccurate. Only in highly exceptional cases, borders are good enough to

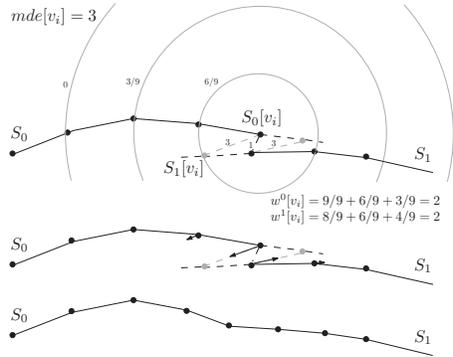


Figure 3. Repairing inconsistencies at the borders of components. Vertex $S_0[v_i]$ is moved towards its corresponding point $S_1[v_i]$ using a weighted voting scheme. Close by vertices in both components S_0 and S_1 are used to compute the appropriate weight.

bootstrap the face model with S_{fine} directly. To resolve the artifacts at the borders of the individually fitted components, we use their sets of m coefficients w_i that were used to obtain each component. In fact, each set of coefficients can be used to acquire a full face instance of which the component is simply a predefined subset of vertices. We refer to such a full face instance as S_{comp} . Because we fitted a set of $c=4$ components, we have $c=4$ face instances S_{comp} . So the face instance S_{fine} is basically a composition of the c intermediate face instances (Fig. 2). To blend the c components, we blend the vertices of the c face instances. In this process, the goal is to determine for each vertex in S_{fine} a new position, such that it has a smooth transition to its neighboring vertices. Once we reach this state, we refer to this final face instance as S_{final} .

Because components can overlap more than several millimeters, Laplacian smoothing of the vertices at the borders would not suffice. The selection of a larger region of triangles and vertices to smooth causes a non-statistical shape deformation and local features may not be preserved. In particular, the places were three or more components overlap, it is hard to regularize the vertex positions using smoothing techniques. Surface stitching techniques as in [12] could be applied to stitch the components, but this would distort the point-to-point correspondences. Mesh fairing using Gaussian curvatures, as in [16] for instance, could smooth some of the border triangles properly. However, these techniques focus on the preservation of sharp features, whereas we want to remove sharp creases caused by overlapping components.

To repair the discontinuities at the borders, we have developed an algorithm, that uses the vertex positions of the intermediate face instances S_{comp} , local neighborhood properties and the PCA model. Since all instances S_{comp} were acquired with the same PCA model, their vertices are all in full correspondence. In case two connected vertices cause a discontinuity in S_{fine} , one can assume that mov-

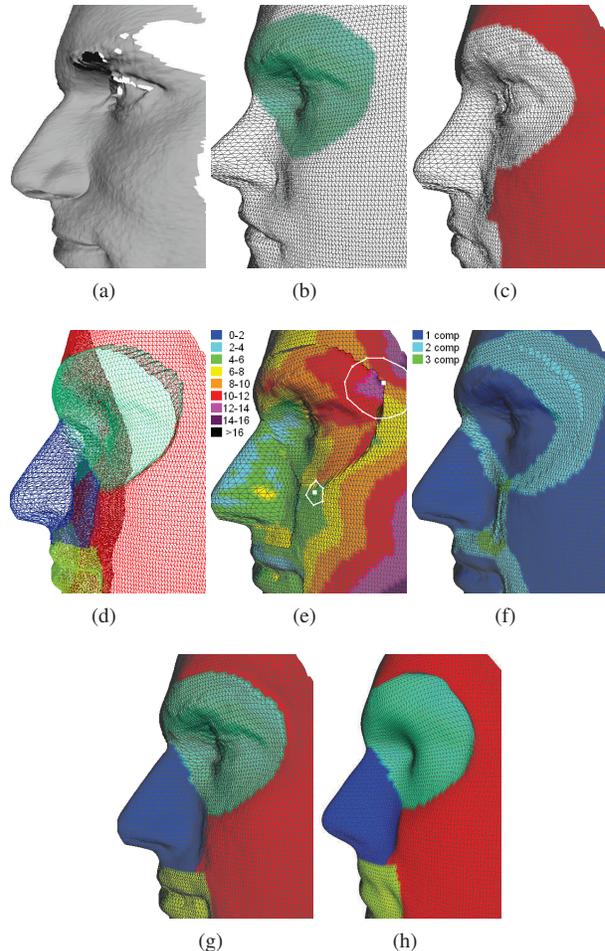


Figure 4. Correspondence estimation. By individually fitting components (b,c) to scan data (a), artifact may occur at the borders of S_{fine} (d). The maximum displacement error (r in mm) for a vertex is an indication of the local mesh distortion (e), towards purple/black means a higher distortion. Using this error, surrounding vertices are selected that all contribute to the voting for a new position of the vertex. After the voting, each vertex is repositioned based on the weighted votes for close by components (f). The final model fit S_{final} (g) presents smooth transitions between the different components. Five iterations of Laplacian smoothing (h) was not enough to repair the artifact close to the eyebrow whereas the face already lost most of its detail.

ing each vertex towards the center of mass of its adjacent vertices (as in Laplacian smoothing) slightly improves on the situation. However, propagating this smoothing locally causes the border discontinuities to attract well positioned vertices instead of solving the discontinuity. Propagating such smoothing globally, changes the statistical shape of the face (Fig. 4h). Instead, we morph each vertex $S_{fine}[v_i]$ towards its c corresponding positions $S_{comp}[v_i]$. For that we need to (1) detect the local distortions, (2) know which components lie close to which vertices, (3) have for each vertex

a weight for each close by component, and (4) recompute the vertex positions.

First, to detect the local distortion, we determine for each vertex $S_{fine}[v_i]$ its *maximum displacement error* to one of its c corresponding positions $S_{comp}[v_i]$, using

$$mde[v_i] = \max_{c \in comp} (e(S_{fine}[v_i], S_c[v_i])),$$

where $e(p, q)$ is the Euclidean distance between two 3D coordinates p and q . When a vertex has approximately the same 3D position in all face instances S_{comp} , its displacement error is small. In that case, we have consensus on its position and thus less need for a repositioning. Second, close by vertices are selected ($S_{fine}[v_j]$) using a sphere of radius $r[v_i]$ around $S_{fine}[v_i]$, where radius $r[v_i]$ equals the displacement error times three, $r[v_i] = 3 \cdot mde[v_i]$. For a vertex $S_{fine}[v_i]$ on a component border, this set of close by vertices include vertices from different components. If all close by vertices belong to the same component as the current vertex nothing will change. Thirdly, each close by vertex $S_{fine}[v_j]$ adds a weighted vote for the component it belongs to. This weight decreases linearly with the distance of vertex $S_{fine}[v_j]$ to $S_{fine}[v_i]$. The maximum weight of one is for the vertex $S_{fine}[v_i]$ itself, and decreases linearly to zero for radius $r[v_i]$. Because components can move away from each other, radius $r[v_i]$ must be at least two times larger than the maximum displacement error, otherwise nothing changes. In the end, a vertex next to the border of the two components, has a number of close by vertices that vote for its own component and a number of close by vertices that vote for the other component. Then, vertex $S_{fine}[v_i]$ is morphed towards its corresponding position $S_{comp}[v_i]$ in the other component according to these weighted votes. For example, if a vertex (weighted vote of 1) has four close by vertices with weighted votes of 0.25 of which two vertices lie on an other component S_{comp} , then we have a 3-to-1 vote and vertex $S_{fine}[v_i]$ if morphed for 25% towards $S_{comp}[v_i]$. Another example in 2D is shown in Fig. 3. In case close by vertices belong to three or more components, the vertex $S_{fine}[v_i]$ is morphed to three or more different positions $S_{comp}[v_i]$.

This *weighted voting scheme* for the local geometry of a face instance S_{fine} , results in a proper blending of the components. Since the overall topology of the face instance is retained, and the problematic vertices and their local neighborhood are assigned to new positions, the adjusted face instance S_{fine} is now in full correspondence with the face model. Fig. 4, shows the multiple component fit S_{fine} based on four components S_{comp} (two are shown), the displacement errors, the blending, and the final result S_{final} .

4.3. Redundancy estimation

After the regularization of the vertex positions, the paired face instance S_{final} can be added to the morphable



Figure 5. Redundancy estimation. From left to right, the face scan, S_{single} , the residual errors of S_{single} , S_{mult} , the residual errors of S_{mult} , and the color map for the point-to-point distances (in mm).

face model. To do so, we can apply the ICP algorithm to finely align S_{final} to \bar{S} and include it in the example set of 3D faces from which the morphable face model was built. Then, we can recompute the PCA model using 100+ k example faces, and keep the $m + k$ principal eigenvectors and eigenvalues of the face (Sect. 2). This way the face properties of a new example face can be added to the statistical face model. With this enhanced model, we should be able to produce accurate model fits with only a single component to face scans similar to those k example scans. In the end, each face scan can be described using $m + k$ model coefficients, and face identification can be performed based on such a $m + k$ feature vector.

The addition of k extra example faces to the current face model, causes an increase of computation costs for both the model fitting and face identification method. So, it is important not to add example faces that are already covered in the current morphable face model. Therefore, we estimate the redundancy of encountered example data. First, the morphable face model is fitted as a single face component to the 3D scan data, which we refer to as S_{single} . Secondly, we fit the morphable face model using multiple face components with the improved correspondence estimation described above, S_{mult} . After the model fitting process, a residual error between the vertices of the model fit and the scan data remains. The difference of the two residual errors of S_{single} and S_{mult} , can be used to estimate the redundancy of the new face scan. In case the residual error of S_{single} is significantly larger than that of S_{mult} , then the face scan is most likely not contained in the current morphable face model, and we should add S_{mult} to the model.

To compute the residual error of these model fits we use the RMS distance of closest point pairs,

$$d_{rms}(S, scan) = \sqrt{\frac{1}{n} \sum_{i=1}^n e_{min}(p_i, scan)^2} \quad (1)$$

using all n vertices of S_{single} and S_{mult} . Closest point pairs (p, p') for which p' belongs to the boundary (including holes) of the face scan, are not used in the distance measure. Fig. 5 shows for one face scan, the two model fits and their residual error maps. For this example the RMS error for S_{single} is 0.89 mm and for S_{mult} 0.68 mm.

Dataset	S_{single}	S_{mult}	S_{single}^+	$S_{si} - S_{mu}$	$S_{si}^+ - S_{mu}$
GAVAB	1.36	1.24	1.27	0.13	0.04
GAVAB	1.34	1.16	1.19	0.18	0.03
GAVAB	2.04	1.59	1.56	0.45	-0.03
GAVAB	1.32	1.16	1.19	0.17	0.03
BU-3DFE	1.18	1.01	1.02	0.18	0.02
BU-3DFE	1.28	1.12	1.15	0.16	0.03
BU-3DFE	1.06	0.96	0.96	0.10	0.00
BU-3DFE	1.61	1.40	1.49	0.21	0.09
local	0.70	0.55	0.58	0.15	0.03
local	0.89	0.68	0.69	0.21	0.01
local	0.79	0.62	0.67	0.17	0.05
local	0.69	0.55	0.58	0.14	0.04
CAESAR	1.86	1.77	1.80	0.09	0.03
CAESAR	1.88	1.77	1.78	0.11	0.01
CAESAR	1.81	1.74	1.77	0.07	0.03
CAESAR	1.80	1.75	1.78	0.05	0.03

Table 1. RMS errors (mm) of output models to input scans. The model fits with the smallest and largest difference in residual errors (bold) are show in Fig. 6.

5. Results

To elaborate on the performance of our bootstrapping algorithm, we applied it to the dataset of 16 different face scans and to the subset of 277 UND scans. The small set is used to evaluate the model fitting and correspondence estimation algorithm. The UND set is used to test the redundancy estimation.

5.1. Correspondence estimation

To evaluate the correspondence estimation, we compare the residual errors of fitted face instances. First, we fit the initial morphable face model as a single component to the segmented face data S_{single} . Secondly, we fit the initial model to the segmented face data using the four components and blend their borders. Thirdly, we add these 16 face instances S_{mult} to the example set and recompute the PCA model, keeping the $m=99$ principal components of the face. Finally, we fit the enhanced morphable face model to the same segmented face data using a single component S_{single}^+ .

In Tab. 1, we report the residual errors of the fitted face instances. We use these errors for quantitative evaluation of the produced fits. For all face scans, S_{mult} has a lower residual error than S_{single} , which means that a higher fitting accuracy is achieved with the use of multiple components in combination with the improved correspondence estimation. After the model was enhanced with the 16 face instances S_{mult} , the model was again fitted as a single component to the 16 face scans. Now, all residual errors are lower for S_{single}^+ than they were for S_{single} , which means that our bootstrapping algorithm successfully enhanced the morphable face model. For one face scan, the face instance S_{single}^+ is even more accurate than S_{mult} . This is possible, because we enhanced the model with the facial variety of 16 faces at once. We expect that the residual errors of S_{single}^+

can be lowered further, by iterating the process of (1) fitting the enhanced model using multiple components, (2) replacing the 16 face instances S_{mult}^+ , and (3) building a new PCA model. In fact, we tried it for the face scan in Fig. 5 and lowered its RMS distance for S_{mult} (0.68 mm) and S_{single}^+ (0.69 mm) to 0.64 mm for S_{mult}^+ . So, iteratively replacing the 16 instances of the enhanced model with their improved instances S_{mult}^+ , will probably help to some extend. Note that the residual errors are lowest for our local set, which contains the highest resolution scans, and the highest errors for the low resolution CAESAR faces. This is due to the RMS distance that measures point-to-point distances. Because we are interested in the difference between residual errors, this works fine, otherwise, one could use a surface mesh comparison tool instead [7].

In Fig. 6, we show some of the resulting model fits and their distance maps acquired with our bootstrapping algorithm. In this figure, we show the two faces per dataset that achieved the smallest and largest difference in residual errors in the same order as in Tab. 1. Visual inspections of the fitted models shows an improved single component fit of the enhanced model to the scan data (S_{single}^+), compared to the single component fit using the initial morphable model (S_{single}). This can be seen in the residual error maps as well. That the bootstrapping algorithm successfully incorporated the 16 face scans in the morphable face model, can be seen by face instances S_{mult} and S_{single}^+ , which are very similar. The initial morphable face model consisted of neutral expression scans only. Nevertheless, the use of multiple components allows for correspondence estimation among some expression data as well.

5.2. Redundancy estimation

To distinguish between new and redundant face data, we computed the residual errors for face instances S_{single} and S_{mult} using the RMS distance measure. In Tab. 1, we reported the RMS errors for our set of 16 faces. These differences in RMS error for $S_{single} - S_{mult}$ vary between 0.05 and 0.45. The maximum difference of 0.45 was achieved for the sad looking person on row four in Fig. 6. With the use of a threshold t for the difference in RMS error, we decide whether a face is redundant or new. Based on the visual inspection of the faces in Fig. 6, we decided to select $t=0.17$ for our experiments. In case the RMS difference is higher than t , we consider a face to be new and otherwise as redundant. With this threshold, we classify only the faces in row two, four, and five and being new.

We applied our bootstrapping algorithm to the 277 UND scans, and let our algorithm automatically select potential faces to add to the model, without actually adding them. This way we can see which face scans (persons) are new to the model. For these persons, we may assume a difficulty in identifying them, because a coefficient based recognition



Figure 6. Automatic correspondence estimation. From left to right, the segmented and pose normalized faces, the single component fit S_{single} , its distance map, the multiple component fit S_{mult} , its distance map, and the single component fit S_{single}^+ with its distance map. Faces in rows two, four, and five were selected as being new to the model.

system may confuse that person with a different person that has those coefficients. Out of the 277 UND scans, 35 scans were found as being new to the system, that is, having a decrease in RMS error of $S_{single} - S_{mult}$ higher than threshold t . Some of these produced fits are shown in Fig. 7. Most of the selected faces have indeed new face features and should

be added to the morphable face model. However, some of the faces that are covered by facial hair produce less reliable fits. To improve on these fits, one could apply a skin detection algorithm and remove the hair beforehand. The total time spend by our algorithm is less than two minutes per face scan on a desktop computer. Since our algorithm fits



Figure 7. UND faces (1st model) that were automatically bootstrapped. S_{single} (2nd and 3rd) and S_{mult} (4th and 5th) are shown.

the face model twice, parallelizing this process can greatly reduce the computation time.

6. Concluding remarks

In this work we presented a method for establishing accurate correspondences among 3D face scans. With the use of an initial morphable face model and a set of predefined components, we are able to produce accurate model fits to 3D face data with noise and holes. Afterwards, the components still have defects on their border, for which we presented a new blending technique that retains the full correspondence between the face instance and the morphable model. These newly generated face instances can be added to the example set from which a new and enhanced morphable model can be built, as we show in this work. We applied our bootstrapping algorithm to 277 UND scans, and it successfully established full correspondence between these scans and the initial morphable model, and our automatic redundancy check approved 35 new scans to be added to the model.

With our new bootstrapping algorithm, we are able to successfully update an initial face model, which we use to produce more accurate fits to new scan data. Compared to previous work, our algorithm is fully automatic, reuses initial face statistics, checks for redundancy, and retains the full correspondence even in case of noisy scan data with holes.

Acknowledgements

This research was partially supported by FOCUS-K3D FP7-ICT-2007-214993. The authors thank the University of South Florida for providing the USF Human ID 3D Database and the morphable face model, University of Notre Dame for the UND set, Binghamton University for the BU-3DFE set, TNO for the Dutch CAESAR set, and GAVAB for their dataset.

References

- [1] B. Amberg, S. Romdhani, and T. Vetter. Optimal Step Non-rigid ICP Algorithms for Surface Registration. In *CVPR*, pages 1–8, 2007. 2
- [2] C. Basso, P. Paysan, and T. Vetter. Registration of Expressions Data using a 3D Morphable Model. In *Automatic Face and Gesture Recognition*, pages 205–210, 2006. 2
- [3] V. Blanz and T. Vetter. A Morphable Model for the Synthesis of 3D Faces. In *SIGGRAPH*, pages 187–194, 1999. 1, 2
- [4] V. Blanz and T. Vetter. Face Recognition Based on Fitting a 3D Morphable Model. *PAMI*, 25(9):1063–1074, 2003. 2
- [5] CAESAR-survey. The Civilian American and European Surface Anthropometry Resource at <http://store.sae.org/caesar>, 2008. 3
- [6] K. I. Chang, K. W. Bowyer, and P. J. Flynn. An Evaluation of Multimodal 2D+3D Face Biometrics. *PAMI*, 27(4):619–624, 2005. 3
- [7] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring Error on Simplified Surfaces. *Computer Graphics Forum*, 17:167–174, 1998. 6
- [8] X. Huang, N. Paragios, and D. N. Metaxas. Shape Registration in Implicit Spaces Using Information Theory and Free Form Deformations. *PAMI*, 28(8):1303–1318, 2006. 2
- [9] A. Moreno and A. Sanchez. GavabDB: a 3D Face Database. In *Workshop on Biometrics on the Internet COST275, Vigo*, pages 77–85, 2004. 3
- [10] F. B. ter Haar and R. C. Veltkamp. 3D Face Model Fitting for Recognition. In *European Conference on Computer Vision (ECCV)*, pages 652–664, 2008. 2, 3
- [11] F. B. ter Haar and R. C. Veltkamp. A 3D Face Matching Framework for Facial Curves. *Graphical Models*, 71(2):77–91, 2009. 3
- [12] G. Turk and M. Levoy. Zippered Polygon Meshes from Range Images. In *SIGGRAPH*, pages 311–318, 1994. 4
- [13] University of South Florida, Sudeep Sarkar. USF HumanID 3D Face Database. 2
- [14] T. Vetter, M. J. Jones, and T. Poggio. A Bootstrapping Algorithm for Learning Linear Models of Object Classes. In *CVPR*, pages 40–46, 1997. 1
- [15] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. A 3D Facial Expression Database For Facial Behavior Research. In *Automatic Face and Gesture Recognition*, pages 211–216, 2006. 3
- [16] H. Zhao and G. Xu. Triangular Surface Mesh Fairing via Gaussian Curvature Flow. *Journal of Computational and Applied Mathematics*, 195(1):300–311, 2006. 4