3D face reconstruction and retargeting from RGBD

Korobov Nikita Zhakshylyk Nurlanov Sirin Fki dit Kaaniche

Hossameldien Abdalaleem

October 16, 2020

Abstract

The aim of this project is the 3D reconstruction of human faces represented as an RGBD video stream and further re-targeting of the estimated face expression to the other mesh. For an initial model fitting, facial landmarks are detected on the image, and together with respective landmarks on the neutral (mean) face, meshes are aligned using the Procrustes algorithm. To estimate the initial identity and expression of target face, the parametric face model fitting is done via non-linear optimization. Then the output is refined using non-rigid ICP. Obtained deformation is transferred to another face mesh using few manually chosen correspondences. The experiments are carried out both on the provided Dr. Justus Thies's face scan as well as on our own recorded RGBD sequence.

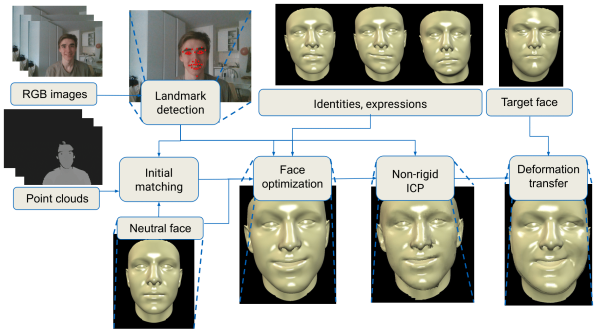

Figure 1: Pipeline overview

1 Overview

The pipeline of the algorithm is shown in Figure 1. Firstly, landmarks are detected on the image and used for initial fitting of the scanned point cloud to the initial face model (IFM). Then the identity and expressions are modeled using optimization proposed by [10] and blendshape basis vectors [8]. The face model provided by fitting is limited by the space dimensionality of blendshapes basis, that is why they cannot perfectly reflect all small details of the face. Non-rigid ICP algorithm is used to perform refinement of the obtained face by applying small deformations to vertices, preserving smoothness of the shape. The total deformation from the IFM to the optimized face is transferred to another target face mesh.

1.1 Landmarks detection and initial fitting

To perform optimization and refinement, scanned face and IFM need to be aligned with each other. For initial alignment, we firstly detect facial landmarks on the RGB image using facial landmarks detector from DLib [1, 11]. It provides 68 landmarks on the mouth, nose, eyes, eyebrows, chin, and cheeks. However, not all the landmarks are detected reliably, that is why we have chosen the set consisting of 36 of them for further usage (Fig. 1). Then, detected landmarks are projected to the scan point cloud. The corresponding landmarks of the face

model are manually chosen as a set of corresponding vertices. Given these correspondences, the scanned point cloud and the IFM are aligned using the Procrustes algorithm. It worth mentioning that the alignment is not required to be perfect, as it will be used as head pose initialization for the further optimization described in Section 1.2.

1.2 Parametric model fitting

The parametric facial model fitting is done as described in [10]. We are using the morphable model ($\mathbf{a_{id}}, E_{id}$) of [8] for identity parameterization, and the expressions blendshapes (E_{exp}) from [5]. We are also modeling model-to-world transformation (R, t) via SE(3) transform parameterization provided by Sophus library [3]. However, we do not model illumination and albedo in this work. Overall, the final model is described as follows:

$$V(\alpha, \delta, R, t)_i = R(\mathbf{a_{id}} + E_{id}\alpha + E_{exp}\delta)_i + t. \tag{1}$$

The optimized energy includes 4 terms: point-to-point, point-to-plane, landmarks, and regularization: $E(\alpha, \delta, R, t) = E_{point} + E_{plane} + E_{land} + E_{reg}$.

For **point-to-point** and **point-to-plane** constraints we are iterating through all 3d points $v_i \in V$ (source) on face model and looking for the closest point $p_j \in T$ (target) on RGBD scan using pre-built kd-tree. As a result, we assign the residual as:

$$r(j) = \begin{cases} \min\limits_i \{dist^2(p_j, v_i)\} & p_j \text{ is closest to } v_i \\ 0 & \text{no corresponding point (distance to closest point is big),} \end{cases} \tag{2}$$

where $dist$ is point-to-point L_2-distance. $E_{point} = w_{point} \sum r(j)$, and similarly $E_{plane} = w_{plane} \sum r(j)$ with symmetrical point-to-plane distance.

Moreover, we are using the provided landmark correspondences $\{(p_k, v_k)\}$ to build **landmarks** constraint:

$$E_{land} = w_{land} \sum \|p_k - v_k\|^2 . \tag{3}$$

The final component is statistical **regularizator**, where the standart deviation for each shape coefficient is precomputed:

$$E_{req} = w_{reg} \left[\sum_{i=1}^{160} \left(\frac{\alpha_i}{\sigma_{id,i}} \right)^2 + \sum_{i=1}^{76} \left(\frac{\delta_i}{\sigma_{exp,i}} \right)^2 \right]. \tag{4}$$

The optimization is formulated as non-linear least squares problem and solved using ceres library [4].

1.3 Non-rigid ICP

The non-rigid ICP algorithm aims to find the optimal set of affine transformations $\mathbf{X}$ of the source mesh vertices [6]. The optimality of the transformations is defined through the optimized functional, consisting of 3 terms: distance, stiffness, and landmark.

Normally we want the **distance** between source V and target T meshes to be small, that is why the following distance term is being used:

$$E_d(\mathbf{X}) = \sum_{v_i \in V} w_i dist^2(p_i, \mathbf{X_i} v_i), \tag{5}$$

where $p_i \in T$ is the closest vertex on the target mesh, obtained with the help of pre-built kd-tree [7]. Weight w_i is set to zero in 3 cases: (1) if the angle between normals of source and target vertices is more than the threshold, (2) if the source point is on the boundary of the mesh, (3) if the distance from the source vertex to the target vertex is bigger than some threshold. This is done to prevent unfeasible transformations. Otherwise, w_i is one.

In order to preserve **stiffness** of the deformed mesh, we penalise the difference between affine transform matrices of neighboring vertices:

$$E_s(\mathbf{X}) = \sum_{(v_i, v_j) \in E} \|(\mathbf{X_i} - \mathbf{X_j})\mathbf{G}\|_F^2 , \tag{6}$$

where $\mathbf{G} = diag(1, 1, 1, \gamma)$. The hyper-parameter γ is used to relate rotational and skew part of the affine transformation.

Additionally, given the landmarks-to-vertices correspondences $\{(v_k, p_k)\}$ we have an explicit **landmarks** term:

$$E_l(\mathbf{X}) = \sum \|\mathbf{X_k} v_k - p_k\|^2 . \tag{7}$$

The total energy is

$$E(\mathbf{X}) = E_d(\mathbf{X}) + \alpha E_s(\mathbf{X}) + \beta E_l(\mathbf{X}). \tag{8}$$

At the beginning we set α very high, so that only almost rigid deformations are allowed. Then we linearly decrease α during the optimization to make smoothness constraint relaxed. Another hyper-parameter β (1 at the beginning) approaches zero at the end of the optimization, as scanned point cloud may be noisy and respective landmarks may be inaccurate.

The set of optimal transformations $\mathbf{X}$ is obtained via solving the following least-squares problem in closed form for each iteration:

$$E(\mathbf{X}) = \left\| \begin{bmatrix} \alpha \mathbf{M} \bigotimes \mathbf{G} \\ \mathbf{WD} \\ \beta \mathbf{D_1} \end{bmatrix} \mathbf{X} - \begin{bmatrix} \mathbf{0} \\ \mathbf{WU} \\ \beta \mathbf{U_1} \end{bmatrix} \right\|_F^2 = \|\mathbf{AX} - \mathbf{B}\|_F^2 . \tag{9}$$

It has minimum at $\mathbf{X}^* = (\mathbf{A^T A})^{-1} \mathbf{A}^T \mathbf{B}$

1.4 Deformation transfer

For further applications we want to transfer the optimized total transformation:

$$g : V_{init} \rightarrow V_{final} \tag{10}$$

from the initial face model V_{init} to the final one V_{final} into another arbitrary target face T_{init}. The aim is to obtain the transformed target face T_{final} using known transformation g:

$$T_{final} = g(T_{init}). \tag{11}$$

The transformation g can be defined as a set of local affine transformations applied to triangles of the mesh. For each triangle Δ_k^{init} of source shape V_{init} one can find an affine transformation X_k to corresponding triangle Δ_k^{final} of the final face V_{final}, that is

$$e_{k,j}^{final} = X_k e_{k,j}^{init} \tag{12}$$

for each of basis and normal vectors of triangle $e_{k,j} \in \Delta_k \forall j = \{1, 2, 3\}$.

To apply the same transformation g to an arbitrary target mesh T_{init} we should know the correspondences between triangles of the target T_{init} and source V_{init} meshes. For example, in our case the face models of source and target are the same (face model of our sequence, and Dr. Thies's optimized face model), so the correspondences are known. We can apply the deformation transfer via solving the least-squares problem for the vertices of T_{final} in a closed-form solution.

In case of absence of triangle correspondences, i.e. when the number of vertices on V_{init} and T_{init} is different, or the face model representations are different (e.g. cat and human's faces) the following approach from [9] can be used. Provided with the set of correspondences of source and target vertices $S = \{(v_i, p_i)\}, v_i \in V_{init}, p_i \in T_{init}$ that define the anchor points of the source shape one can find the correspondences for other vertices. As we are assuming that the expressions of V_{init} and T_{init} are similar, we can find transform from T_{init} to V_{init} in a non-rigid ICP manner with hard constraint for known correspondences, the soft constraint for closest points, and with regularizers of local transformations. Then using obtained triangle-wise correspondences, the deformation transfer is applied as in the previous case.

2 Results

For the experiments, we use a provided scan of our lecturer Dr. Justus Thies and our sequence of RGBD images recorded by Intel Realsense SR300 camera [2]. The video sequence is recorded with 15fps with a resolution of 640x480. Several point clouds are aggregated in a temporal window of 2 using Rigid-ICP to have more robust and dense scans.

It is seen from Figure 2 that error decreases in each step of the optimization. After a non-rigid ICP face model fits provided scan with high accuracy. There are still regions with high error: on the back of the head, where no point cloud is available, near nostrils, and lips, where the topology of the face changes. These are challenging problems in face reconstruction [10] and in 3d reconstruction via parametric models in general. For the deformation transfer, we use the mesh of Dr. Justus Thies's neutral face (with identities, but without expressions term) as a target and our own recorded sequence as a source for deformation transfer. Some results of the final 3d face fitting and deformation transferring for different expressions and identities are shown in Figure 3.

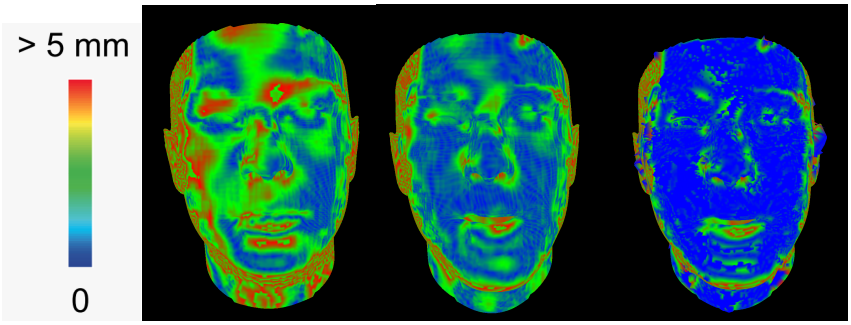

Figure 2: Distance of the face model vertices to the closest points on the scanned mesh. From left to right: distance after initial face fitting using Procrustes, distance after blendshapes fitting, distance after non-rigid ICP.

Figure 3: Top row: Results of face reconstruction. Middle row: Results of re-targeting to Dr. Justus Thies's face. Bottom row: Original RGB images

3 Conclusion and discussions

We have implemented the 3d face reconstruction pipeline and the deformation transfer algorithm. It shows good accuracy and it can capture different identities and emotions with fine details. However, the algorithms are implemented on the CPU and do not work in realtime. It takes $\sim$ 1 min per face on Intel(R) Core(TM) i7-6600U CPU @ 2.60GHz and the most time-consuming part is blendshapes optimization via ceres library [4]. So the next step would be implementing it on GPU.

The current pipeline is not accurate enough to catch wrinkles, and sometimes it has problems with mouth and teeth. So the implementation of algorithms refining these particular tasks would increase the quality of the final model. Also, the albedo and shading modeling are not taken into account, hence it is another way to improve results.

References

[1] Dlib library for image processing. http://dlib.net/imaging.html.

[2] Intel realsense camera sr300. https://ark.intel.com/content/www/us/en/ark/products/92329/intel-realsense-camera-sr300.html.

[3] Sophus library. https://github.com/strasdat/Sophus.

[4] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. http://ceres-solver.org.

[5] Oleg Alexander, Mike Rogers, William Lambeth, Matt Chiang, and Paul Debevec. The digital emily project: Photoreal facial modeling and animation. In *ACM SIGGRAPH 2009 Courses*, SIGGRAPH '09, New York, NY, USA, 2009. Association for Computing Machinery.

[6] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[7] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

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

[9] Robert W Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on graphics (TOG)*, 23(3):399–405, 2004.

[10] Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. Real-time expression transfer for facial reenactment. *ACM Trans. Graph.*, 34(6):183–1, 2015.

[11] Josephine Sullivan Vahid Kazemi. One millisecond face alignment with an ensemble of regression trees. 2014.