Merging Multiple 3D Face Reconstructions

Leonard Thießen, Pascal Laube, Georg Umlauf, Matthias Franz

Institute for Optical Systems, University of Applied Sciences Constance, Germany

***Abstract*—In this paper we present a method to merge multiple 3D face reconstructions into one common reconstruction of higher quality. The individual three-dimensional face reconstructions are computed by a multi-camera stereo-matching system from different perspectives. Using 4-Points Congruent Sets and Iterative Closest Point the individual reconstructions are registered. Then, the registered reconstructions are merged based on point distance and reconstruction tenacity. To optimize the parameters in the merging step a kernel-based point cloud filter is used. Finally, this filter is applied to smooth the merged reconstruction. With this approach we are able to fill holes in the individual reconstruction and improve the overall visual quality.**

I. Introduction

Face recognition is an important problem in biometric applications that is usually based on two-dimensional images. However, it has been shown that the recognition rate can be improved, if the recognition is based on 3D face reconstructions, see Hensler et al. [1]. This requires an accurate and fast reconstruction method, e.g. based on real-time multi-camera stereo-matching as presented in [2]. The algorithm is based on four synchronized cameras (Figure 1) which captures images of a face from different perspectives (Figures 2(a)-2(d)). With this system it is possible to generate high-resolution depth images (Figure 2(e)) in real-time. This reconstruction also contains detailed information about the reconstruction tenacity (Figure 2(f)). A bad matching pixel correspondence in the computation of the depth map will result in a low tenacity value for the reconstructed 3D point. As a result, the algorithm yields one 3D reconstruction of the captured face.

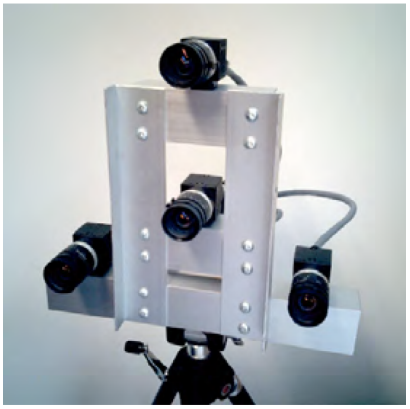


Fig. 1. Multi-camera stereo-matching system.

The main focus of [1] and [2] was on recognition rate and reconstruction speed, reconstruction quality was not the main concern. Thus, the reconstructions may be noisy or have holes. On a single GPU, the reconstruction system can compute up to four 3D reconstructions per second, the reconstruction quality can be improved by merging several 3D reconstructions shown in Figures 2(g)-(j). We propose a process to merge these reconstructions to improve the overall visual reconstruction quality. Due to the lack of a ground truth geometry the noise level is used as an additional quality measure during the merging process.

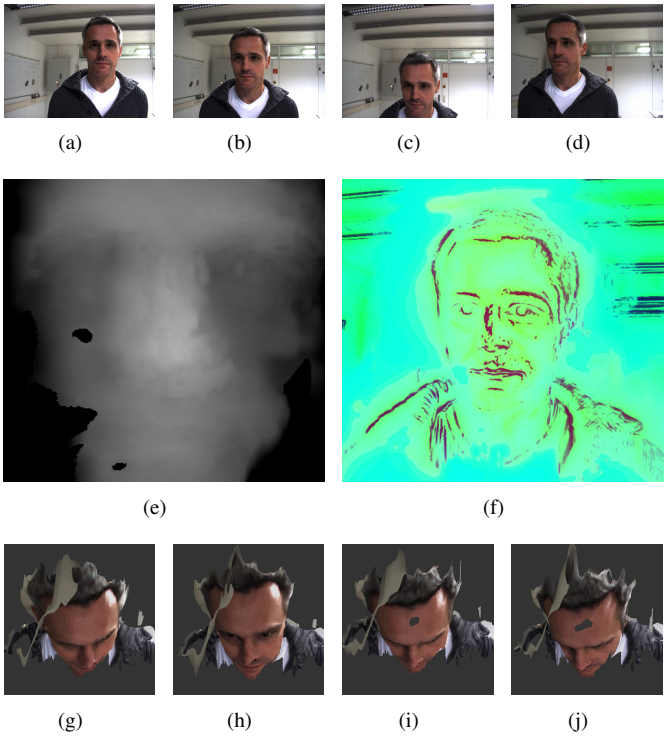


Fig. 2. Output of the multi-camera stereo-matching system. Figures (a)-(d) show the camera images yielding the depth map in Figure (e) with reconstruction tenacity (red: good tenacity, yellow: medium tenacity, cyan: bad tenacity) in Figure (f) and 3D reconstruction in Figure (j). Figures (g)-(j) show different 3D reconstructions from different perspectives.

II. Related Work

Registration and merging of 3D reconstructions are problems relevant to multiple research domains. For the scanning of large objects or terrains, Tang et al. [3] give a general overview of the techniques used to capture buildings. Their approach includes filtering and merging of a large number of scans. Bosse and Zlot [4] use a light detection and ranging sensor mounted on a vehicle to estimate vehicle motion between the

acquired merged scans. Local shape and constraints based on the vehicle motion are used to compute a 3D mapping. A bimodal 3D laser scanner is used in [5] to navigate an autonomous mobile robot. Range as well as reflectance data are combined to generate a navigable map. MacKinnon et al. [6] introduce quality metrics to detect regions which are likely to produce good quality when scanned. These regions are later on merged to generate a region map of optimal quality.

Scanning and reconstruction of smaller scale objects are done in [7]. The objects are scanned from different viewpoints and merged using the VRIP algorithm [8]. Lu et al. [9] use reconstructions from different angles of the human face for face recognition purposes but they do not describe how the different reconstructions are merged.

Thus, existing approaches either do not give a detailed outline of their merging techniques or do not use quality information in the process.

III. Merging 3D Face Reconstructions

The reconstruction system used captures a face from four different angles and uses multi-camera stereo-matching to compute a 3D reconstruction. The reconstructed geometry is represented as point cloud equipped with a measure to the quantify the local tenacity of the reconstruction. For details refer to [2]. Due to the speed of the reconstruction process several 3D reconstructions can be computed per second. These 3D reconstructions are usually from different perspectives since the person moves.

To merge these different 3D reconstructions we propose an approach consisting of four steps. First a coarse registration of the point clouds is done using 4-Points Congruent Set (4PCS) [10], see Section III-A. This is followed by a fine registration using Iterative Closest Point (ICP) [11], see Section III-B. Then the registered point clouds are merged to one 3D reconstruction using tenacity weighted interpolation, see Section III-C. In a last step the merged 3D reconstruction is filtered to erase noise.

A. Coarse Registration

The target of 3D registration is to align a data set $\mathcal{P}$ to a reference data set $\mathcal{Q}$. If $\mathcal{P}$ and $\mathcal{Q}$ are identical point clouds which only differ in position and orientation in space the result are two perfectly aligned point clouds with identical point coordinates. Thus, the actual result of a registration is the affine transformation for the alignment. The registration process is separated into a coarse registration step and the fine registration step. This is due to the fact that algorithms for fine registration are tuned to find local minima. Using these algorithms without initial coarse registration would lead to high computation times or most likely incorrect results.

For coarse registration we use 4PCS, see [10], [12]. This is a RANSAC-based algorithm [13] which performs well even for very noisy data. For RANSAC-based algorithms one has do define appropriate candidates for comparison. In 3D at least three points from each point cloud $\mathcal{P}$ and $\mathcal{Q}$ need to be compared. These randomly selected points define a corresponding pair of local coordinate frames. An affine transformation T can then be determined to align these frames. After applying T to $\mathcal{Q}$ the alignment is evaluated based on the number of points in $T(\mathcal{Q})$ that are within distance d to $\mathcal{P}$. This is the so-called Largest Common Point Set (LCP). This process is repeated until the desired size of the LCP or a certain iteration threshold is reached.

In 4PCS four coplanar points determine the pair of frames. A set of four coplanar points has the advantage that the ratios in the planar congruent set are invariant under affine transformations. A set $B = \{\mathbf{p}_1, \ldots, \mathbf{p}_4\}$ of four points is approximately coplanar, if the distance d_c between the lines $\mathbf{p}_1\mathbf{p}_2$ and $\mathbf{p}_3\mathbf{p}_4$ is small. Then, denote by $\mathbf{p}_E$ the midpoint of the shortest line perpendicular to $\mathbf{p}_1\mathbf{p}_2$ and $\mathbf{p}_3\mathbf{p}_4$, i.e. if $\mathbf{p}_1, \ldots, \mathbf{p}_4$ are coplanar, $\mathbf{p}_E$ is the intersection of these two lines. The ratios characterizing a frame are given by

$$r_1 = \frac{\|\mathbf{p}_1 - \mathbf{p}_E\|}{\|\mathbf{p}_1 - \mathbf{p}_2\|} \qquad \text{and} \qquad r_2 = \frac{\|\mathbf{p}_3 - \mathbf{p}_E\|}{\|\mathbf{p}_3 - \mathbf{p}_4\|}.$$

Finding B in $\mathcal{Q}$ is done by selecting three random points $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and searching for $\mathbf{p}_4$ for which d_c is within a given tolerance. By selecting B with large diameter the algorithm becomes fast and globally robust.

To find a congruent frame in $\mathcal{P}$ the four parameters r_1, r_2 and the point distance $d_1 = \|\mathbf{p}_1 - \mathbf{p}_2\|$ and $d_2 = \|\mathbf{p}_3 - \mathbf{p}_4\|$ are used. After finding all point pairs in $\mathcal{P}$ with distances d_1 and d_2, the ratios r_1 and r_2 are computed, and the corresponding frames in $\mathcal{Q}$ and $\mathcal{P}$ are checked for congruence. The best matching pair of frames is finally selected based on LCP.

Selecting 200 pairs of frames has shown to be a good value for balancing run-time and registration error. Because points near the boundary of the scanned faces are particularly noisy we use only points in the center of the face, i.e. within a certain radius around the tip of the nose. Since the camera system is orthogonal to the captured face the tip of the nose can be found by evaluating z coordinates.

If the overlap of the two data sets is known, search can be limited for faster convergence. We assume an overlap between 50% and 60%.

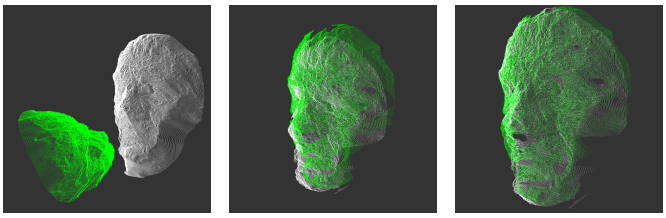
An example for the coarse registration using 4PCS is shown in Figures 3(a) and 3(b). We used the 4PCS implementation of [12].

B. Fine Registration

The fine registration using ICP is based on direct point neighborhoods, see [11], [14]. To register two point clouds $\mathcal{P}$ and $\mathcal{Q}$, for each point in $\mathcal{P}$ the nearest neighbor in $\mathcal{Q}$ is determined. The transformation T to align $\mathcal{P}$ and $\mathcal{Q}$ is computed by minimizing the squared distances between neighbor points. The algorithm is iterated until a specified error threshold is reached. Two error measures are used: The point-to-point distance, which is the Euclidean distance between two points, and the point-to-plane distance, which is the distance between a point and the tangent plane of its neighbor point. First we use the point-to-point distance up to a specified distance threshold. Then, point-to-plain distance is used until

the maximum number of iterations is reached. Repeating this ICP set-up two times yields sufficient registration results.

Figure 3(c) shows an example point cloud after registration with ICP. We used the ICP implementation of the trimesh2 library [15].



(a) (b) (c)

Fig. 3. Registration of two example point clouds (a), after applying 4PCS (b), and after applying ICP (c).

C. Tenacity Weighted Interpolation

Each 3D reconstruction has regions where the point positions are more or less reliable. This is due to lighting effects, relative camera positions and orientations, etc. This reliability of a reconstructed point $\mathbf{p}$ is measured by the tenacity $t(\mathbf{p}) \in [0,1]$, which is given as some weighted normalized cross-correlation of corresponding pixel neighborhoods in the four camera images, see [2]. Smaller values for t indicate higher point tenacity.

Having several 3D reconstructions of the same face, a rather frontal reconstruction is usually reliable for forehead, nose, and mouth regions and unreliable for the cheeks. Thus, we chose a rather frontal reconstruction as reference reconstruction $\mathcal{P}_0$ that is enhanced and enriched be the additional reconstructions $\mathcal{P}_1, \ldots, \mathcal{P}_k$. The merged reconstruction $\mathcal{R} = \{\mathbf{r}_1, \ldots, \mathbf{r}_M\}$ is generated by adding points from one $\mathcal{P}_i, i = 0, \ldots, k$, or from a tenacity weighted interpolation of points from several $\mathcal{P}_i$.

Initially, the merged reconstruction $\mathcal{R}$ is empty. Then, for every point in the reference reconstruction $\mathcal{P}_0 = \{\mathbf{p}_1, \ldots, \mathbf{p}_m\}$ a local merge step is computed. It is based on the point neighborhood N_i of $\mathbf{p}_i$ containing the n nearest neighbors of $\mathbf{p}_i$ from each of the additional reconstructions $\mathcal{P}_1, \ldots, \mathcal{P}_k$. Hence, N_i contains kn points. If $\mathbf{p}_i$ and all its neighbors have tenacity larger than $t_{\min}$, no point is added to $\mathcal{R}$ and further processing of $\mathbf{p}_i$ is skipped. This ensures a minimal overall point quality.

Denote by $\mathbf{p}_t$ a point from N_i with best tenacity and by $\mathbf{p}_d$ a point from N_i with smallest distance to $\mathbf{p}_i$. Furthermore, denote by d_t and d_d thresholds for the maximal distance of $\mathbf{p}_t$ and $\mathbf{p}_d$ to $\mathbf{p}_i$. N_i contains candidate points that represent the geometry better than P_i, if the set

$$\begin{aligned} P_R &= \{\mathbf{p}_t \mid (t(\mathbf{p}_t) < t(\mathbf{p}_i)) \wedge (\|\mathbf{p}_t - \mathbf{p}_i\| < d_t)\} \\ &\cup \{\mathbf{p}_d \mid (t(\mathbf{p}_d) < t(\mathbf{p}_i)) \wedge (\|\mathbf{p}_d - \mathbf{p}_i\| < d_d)\} \end{aligned}$$

is not empty. The point with best tenacity in P_R is added $\mathcal{R}$.

If P_R is empty we define a set of points for interpolation

$$\begin{aligned} P_I &= \{\mathbf{p}_t \mid (|t(\mathbf{p}_t) - t(\mathbf{p}_i)| < \delta_t) \wedge (\|\mathbf{p}_t - \mathbf{p}_i\| < d_t)\} \\ &\cup \{\mathbf{p}_d \mid (|t(\mathbf{p}_d) - t(\mathbf{p}_i)| < \delta_t) \wedge (\|\mathbf{p}_d - \mathbf{p}_i\| < d_d)\} \\ &\cup \{\mathbf{p}_i\}, \end{aligned}$$

where δ_t denotes a tenacity difference threshold. If P_I contains only one point, it is $\mathbf{p}_i$ which is added to $\mathcal{R}$. Otherwise the points in P_I are interpolated:

Linear two-point-interpolation For the set $P_I = \{\mathbf{q}_1, \mathbf{q}_2\}$ add to $\mathcal{R}$ the point $\mathbf{r}$ given by

$$\mathbf{r} = \lambda \mathbf{q}_1 + (1-\lambda)\mathbf{q}_2 \qquad \text{with} \qquad \lambda = \frac{t_{\mathbf{q}_1}}{t_{\mathbf{q}_1} - t_{\mathbf{q}_2}}.$$

Linear multi-point-interpolation For the set $P_I = \{\mathbf{q}_1, \ldots, \mathbf{q}_l\}, l \geq 3$, add to $\mathcal{R}$ the point $\mathbf{r}$ given by

$$\mathbf{r} = \frac{\sum_{i=1}^{l} (1 - t(\mathbf{q}_i))\mathbf{q}_i}{\sum_{j=1}^{l} (1 - t(\mathbf{q}_j))}$$

Overall there are the five parameters n, $t_{\min}$, d_t, d_d, and δ_t in the merge process that need to be optimized to achieve the best possible reconstruction. To compute an error measure for parameter optimization the ground truth geometry is required. However, this ground truth geometry is not available in our application setting. Therefore, a point cloud filter algorithm is used. If $\mathcal{R}_f = \{\mathbf{r}_1^f, \ldots, \mathbf{r}_M^f\}$ denotes the filtered reconstruction, the parameters are chosen such that $\mathcal{R}$ and $\mathcal{R}_f$ are close with respect to the error $e = \sum \|\mathbf{r}_i - \mathbf{r}_i^f\|^2$, i.e. the filtering has minimal effect on $\mathcal{R}$.

As point cloud filter we use the kernel-based filter method of Schall et al. [16]. For a point cloud $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_m\}$ a density function

$$\hat{f}(\mathbf{x}) = \frac{1}{mh^3} \sum_{i=1}^{m} \Phi\left(\frac{\mathbf{x} - \mathbf{p}_i}{h}\right)$$

is used to approximate the actual surface of a noisy point cloud, where Φ is the 3D Gaussian kernel of size h. This means, there is a likelihood function $L(\mathbf{x})$ that gives the probability that a point $\mathbf{x} \in \mathbb{R}^3$ is on the surface. L is an accumulation of local likelihood functions aligned to the local geometry at $\mathbf{p}_i$. This local geometry is represented by an anisotropic 3D Gaussian whose covariance is aligned to the local weighted principal component analysis at $\mathbf{p}_i$. The eigenvectors of the weighted covariance matrix

$$C_i = \sum_{j=1}^{m} (\mathbf{p}_j - \mathbf{c}_i)(\mathbf{p}_j - \mathbf{c}_i)^T \frac{\|\mathbf{p}_j - \mathbf{p}_i\|}{h}$$

approximate the tangent plane and surface normal at $\mathbf{p}_i$, where $\mathbf{c}_i$ is the weighted centroid of points $\mathbf{p}_j$ inside the kernel. The eigenvector corresponding to the smallest eigenvalue of C_i gives the (normalized) normal $\mathbf{n}_i$; the other two span the tangent plane. Thus, L is defined as

$$L(\mathbf{x}) = \sum_{i=1}^{m} \Phi_i(\mathbf{x} - \mathbf{c}_i) \left[h^2 - [(\mathbf{x} - \mathbf{c}_i)\mathbf{n}_i]^2\right].$$

Filtering the point cloud is now done by using the mean-shift method to move all points to positions of high probability. Using gradient-ascent maximization an iterative scheme

$$\mathbf{p}_i^0 = \mathbf{p}_i \qquad \text{and} \qquad \mathbf{p}_i^{k+1} = \mathbf{p}_i^k - \mathbf{m}_i^k$$

with

$$\mathbf{m}_i^k = \frac{\sum_{j=1}^{m} \Phi_j(\mathbf{p}_i^k - \mathbf{c}_j)[(\mathbf{p}_i^k - \mathbf{c}_j)\mathbf{n}_j]\mathbf{n}_j}{\sum_{j=1}^{m} \Phi_j(\mathbf{p}_i^k - \mathbf{c}_j)}$$

is applied. Iteration is stopped if

$$\|\mathbf{p}_i^{k+1} - \mathbf{p}_i^k\| < 10^{-4}h.$$

h is in the interval of one to ten times the average sampling density of the point cloud.

The final step in the merge process is a smoothing step on $\mathcal{R}$ using this kernel-based method.

IV. Results

To demonstrate the effectiveness of the proposed method we compare one reference reconstruction $\mathcal{P}_0$ of a male head to the merging results with three additional reconstructions $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$. Examples are shown in Figures 2(g)-(j). The optimal parameters for the merging process were determined manually. A major influence on the overall visual quality of the reconstruction is the size n of neighborhoods N_i. If n is large the chance to find neighboring points with good tenacity increases. However, these points can be spatially far away, leading to visible holes in the reconstruction. In our tests, smaller neighborhoods led to better values for e. The reconstruction tenacity $t_{\min}$ has the biggest impact on the error e. Smaller values for $t_{\min}$ result in sparser point clouds with higher quality and small e. Distance thresholds d_t and d_d have similar effects on the reconstruction. Large distance thresholds result in holes in the point cloud and reduce the overall appearance. Small distance thresholds lead to merged reconstructions $\mathcal{R}$ mostly consisting of points from $\mathcal{P}_0$. Smaller distance thresholds as well as a tenacity difference threshold δ_t between 5% and 10% have a positive effect on appearance. Applying the kernel-based filter in a last step further smooths the surface and improves the visual quality.

Figure 4(a) shows a reconstruction with non-optimized parameters. It contains visible gaps and cracks that have not been part of the initial reconstructions. Parameters have then been stepwise adjusted to minimize e. For the given example in Figure 4(b), we use $t_{\min} = 0.5$. The smallest error e was achieved for parameters $n = 1, d_t = 0.000005, d_d = 0.000005$ and $\delta_t = 1.0$. Because point coordinates are based on pixel distance of the camera images, d_t and d_d are relative pixel distances. With these parameters almost all gaps and cracks have been removed from the merged reconstruction. By applying the filter to the point cloud, blurred and noisy regions 4(c) are smoothed and the contours become well defined. Evaluation of overall quality improvement is shown in Figures 5 and 6 with $t_{\min} = 0.4$. The reference reconstruction and the merged reconstruction are colored according to tenacity. One can see that the algorithm succeeds in filling the holes in the reference reconstruction and in expanding regions with high tenacity. With $t_{\min} = 0.3$ the merged reconstruction contains up to 25% more points than the reference reconstruction while maintaining or improving the average tenacity.

V. Conclusion

We present a method for merging face reconstructions to improve the overall reconstruction quality for use in face recognition. By endowing the merging process with thresholds and a tenacity-based interpolation as well as with an error measure for optimizing the merging parameters, we could increase the overall visual reconstruction quality.

The initial reconstructions that are merged later on contain color information which at the moment is lost in the interpolation step. For future work, point color has to be included in the process. Color information could be used when deciding point neighborhood as well as for lifelike visualization. The presented method and the resulting parameters have shown to be optimal for reconstructions generated by the stereo-matching approach in [2]. To prove the general applicability of our algorithm data sets of other 3D face reconstruction systems need to be evaluated. The overall algorithm has not been optimized for real-time application yet. Especially the filtering process is computationally expensive and could be improved by splitting the 3D-separable Gaussian function into three 1D functions as described in [17].

References

[1] J. Hensler, K. Denker, M. Franz, and G. Umlauf, "Hybrid face recognition based on real-time multi-camera stereo-matching," in *ISVC 2011*, G. B. et al., Ed. Springer, 2011, pp. 158–167.

[2] K. Denker and G. Umlauf, "An accurate real-time multi-camera matching on the gpu for 3d reconstruction," *Journal of WSCG*, vol. 19, pp. 9–16, 2011.

[3] P. Tang, D. Huber, B. Akinci, R. Lipman, and A. Lytle, "Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques," *Automation in construction*, vol. 19, no. 7, pp. 829–843, 2010.

[4] M. Bosse and R. Zlot, "Continuous 3d scan-matching with a spinning 2d laser," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 4312–4319.

[5] S. Frintrop, E. Rome, A. Nüchter, and H. Surmann, "A bimodal laser-based attention system," *Computer Vision and Image Understanding*, vol. 100, no. 1, pp. 124–151, 2005.

[6] D. MacKinnon, V. Aitken, and F. Blais, "Adaptive laser range scanning using quality metrics," in *Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE*, 2008, pp. 348–353.

[7] D. Huber and M. Hebert, "Fully automatic registration of multiple 3d data sets," *Image and Vision Computing*, vol. 21, no. 7, pp. 637–650, 2003.

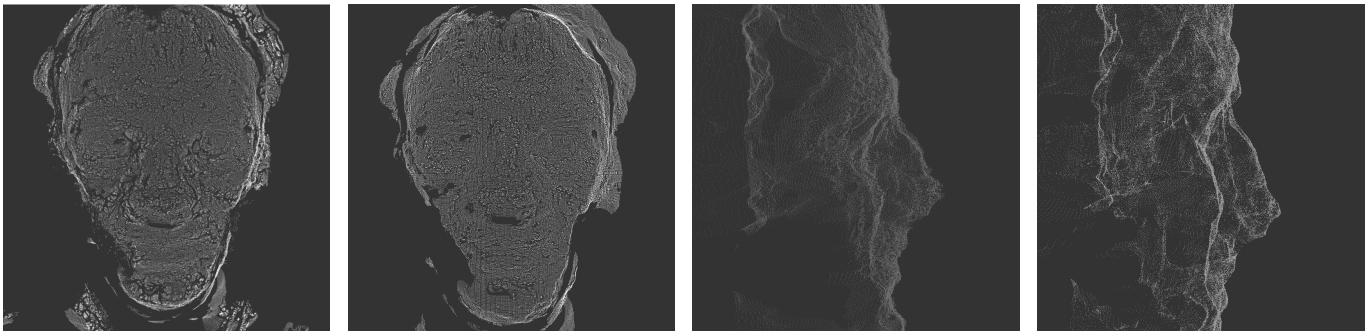
[8] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*. Springer, 1985.

[9] X. Lu, A. K. Jain, and D. Colbry, "Matching 2.5d face scans to 3d models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 31–43, 2006.

[10] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl, "Congruence, similarity, and symmetries of geometric objects," *Discrete & Computational Geometry*, vol. 3, no. 1, pp. 237–256, 1988.

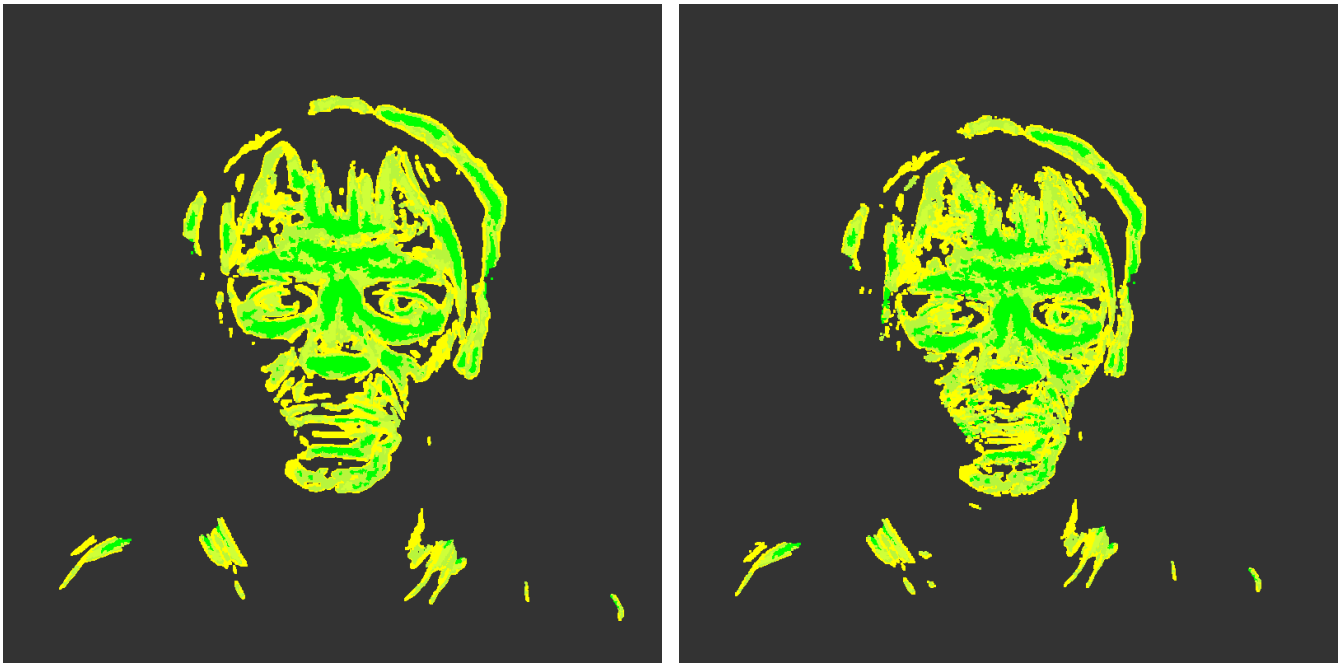
[11] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992, pp. 586–606.

[12] D. Aiger, N. J. Mitra, and D. Cohen-Or, "4-points congruent sets for robust surface registration," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. #85, 1–10, 2008.
[13] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Comm. of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[14] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Third International Conference on 3D Digital Imaging and Modeling*, 2001, pp. 145–152.
[15] S. Rusinkiewicz, "trimesh2," http://gfx.cs.princeton.edu/proj/trimesh2/.
[16] O. Schall, A. Belyaev, and H.-P. Seidel, "Robust filtering of noisy scattered point data," in *Proceedings of the Second Eurographics / IEEE VGTC Conference on Point-Based Graphics*, 2005, pp. 71–77.
[17] C. Lampert and O. Wirjadi, "An optimal nonorthogonal separation of the anisotropic gaussian convolution filter," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3501–3513, 2006.



(a) (b) (c) (d)

Fig. 4. Merged reconstruction with non-optimized (a) and optimized (b) parameters. Closeup of the nose of a joined reconstruction with optimized (c) parameters and after filtering (d).



(a) (b)

Fig. 5. Tenacity colored images of a reference reconstruction (a) and the the respective merged reconstruction (b). The color gradient ranges from bright green (high tenacity) to yellow (bad tenacity).

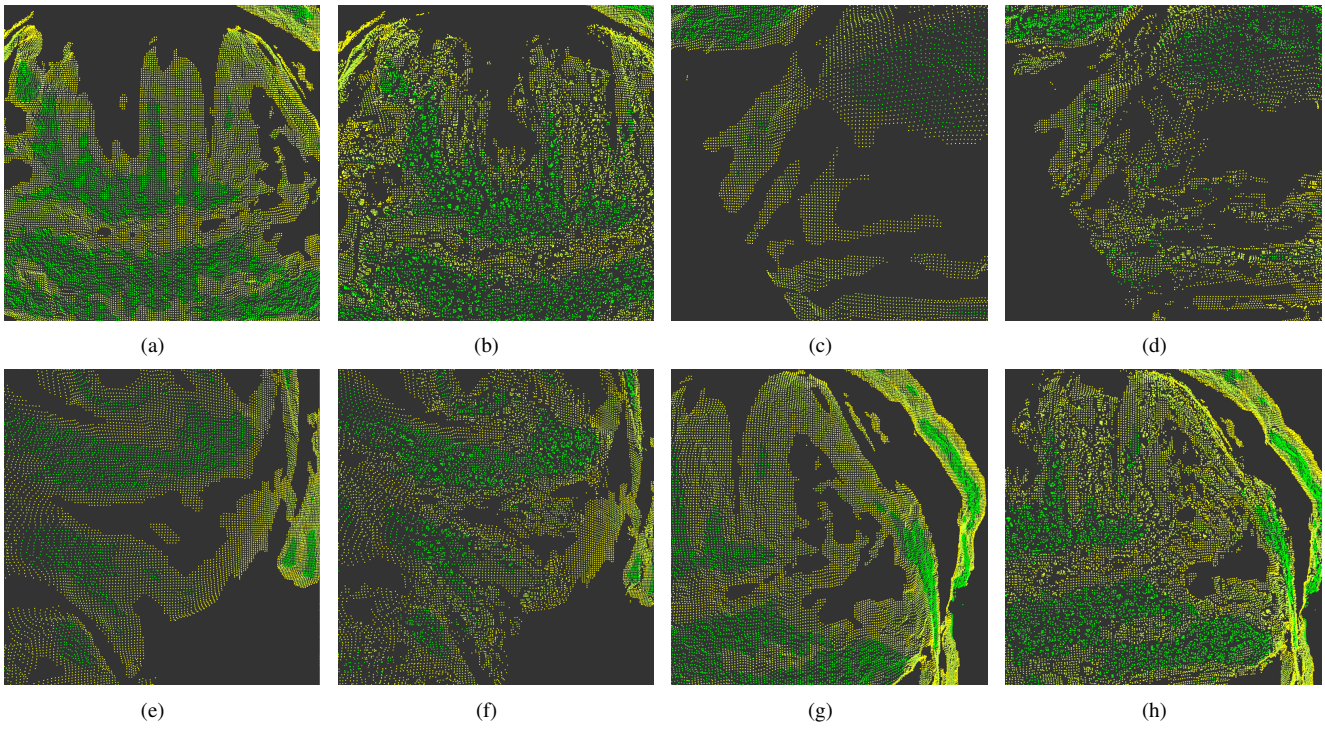


Fig. 6. Closeup on tenacity colored images of a reference reconstructions (a), (c), (e), (g) and the respective merged reconstructions below in (b), (d), (f), (h). The color gradient ranges from bright green (high tenacity) to yellow (bad tenacity).