Body Part Modeling on the Phone

Ioannis Mariggis, Olivier Saurer, and Petri Tanskanen

Astrivis AG, Zürich, Switzerland

{ioannis.mariggis, olivier.saurer, petri.tanskanen}@astrivis.com

DOI: 10.15221/16.214 http://dx.doi.org/10.15221/16.214

Abstract

In this work we propose a software pipeline which uses the inbuilt RGB camera of an off-the-shelf smartphone, to create a 3D model from a set of ordered images. The motion of the camera is tracked visually and a subset of the images is processed to create depthmaps that are then fused into a single point cloud. Subsequently a textured mesh can be extracted. The software runs entirely on the phone in real-time, transforming the mobile platform into a portable hand-held 3D scanner. The resulting 3D models are compared to 3D models created by a typical structured light sensor for mobile devices.

Keywords: Mobile Vision, Computer Vision, Structure-from-Motion, Stereo, 3D Mobile Scanner, Body Scanning.

1 Introduction

Scanning of 3D objects has gained great attention with the rise of 3D printing, especially in the medical/health sector and the e-commerce domain. Applications reach from orthopedic insoles and shoes, body measurements, custom fitted glasses and custom tailored clothing to name a few. While 3D printers are becoming widely available, 3D data acquisition is still a challenging task and often requires expensive specialized hardware, which is not available to the average consumer.

Today's smart phones provide high computational power and a rich set of sensors (camera, IMU, etc.) which are a fantastic prerequisite to create a mobile 3D scanner. We demonstrate how we can exploit this hardware to create a low cost interactive and mobile 3D scanner. To obtain accurate reconstructions, we use the camera's live image stream to continuously track the camera's position using state-of-the-art structure-from-motion algorithms. A subset of those images are further processed to compute a dense point cloud. The point cloud is then transformed into a triangular mesh and textured using the captured images. The full reconstruction pipeline runs entirely on the phone at interactive frame rate and therefore provides little delay between image acquisition and 3D model generation. The low latency allows for fast visual feedback, helping the user to better understand what has already been scanned and what still needs to be captured. Thus allowing the user to quickly identify missing parts and resume the scanning process at those places.

We investigate the accuracy of our mobile 3D reconstruction application on different datasets and show the effectiveness of the approach on various body part scans. Furthermore, we show that without additional hardware we can obtain comparable results to structured light sensors.

2 Related Work

Reconstructing 3D objects from a set of images has been a hot topic in research for over 20 years. Pollefeys *et al.* proposed in [PVV+01, KPVG00, PKVVG98], algorithms to compute 3D models from a temporarily ordered sequence of images, which is typically obtained from a video camera. Such ordered image sets simplify the reconstruction process, since the images can be processed in a consecutive manner. City wide reconstruction from video streams in real-time has been demonstrated in [PNF+08].

More recently Agrawal *et al.* [ASS+09] proposed algorithms to create large scale 3D models from a internet image collection. To reconstruct the major landmarks in the city of Rom *flickr* images were used and processed in one day on a small 62 node cluster. In [FFGG+10] Frahm *et al.* showed that same scale reconstruction can be achieved on one single powerful computer in a single day, by making use of multiple GPUs.

While the above methods benefit from large image collections with lots of visual overlap by using community image collection, different challenges occur when manually scanning 3D objects. It is often unclear to the person capturing the images if sufficient images have been acquired and if the captured images cover the complete object of interest. In addition it is hard for a user to determine if there is enough visual connectivity between images. To overcome this challenge different live reconstruction methods haven been proposed [ND10, PAR+11, NLD11].

In [TKM+13] Tanskanen *et al.* proposed a real-time 3D reconstruction pipeline that runs on limited mobile phone hardware. The system provides live 3D reconstruction feedback to the operator. The feedback is particularly helpful to the user, since it directly shows where the reconstruction is complete and where it is yet incomplete. The method has further been improved in [KTSP14]. A similar camera based motion stereo approach has also been adapted to run with fisheye images [SSHP15] and has been demonstrated on the Google Tango platform. In [OKI15] a real-time volumetric reconstruction approach was demonstrated on a mobile phone.

While all above approaches rely on passive image sensors to create a compelling 3D model, other approaches exist, which rely on special active sensors such as LiDAR, Time of Flight or Structured light sensors. The advantage of those sensor is that they provide high metric accuracy within a certain range [NIH+11] and can capture large spaces quickly [NZIS13] and even dynamic scenes [NFS15, DKD+16]. The main disadvantage is that special and expensive (compared to image sensors) hardware is required to capture the scene in 3D. The availability of such sensors to the general public is still not given. Devices like the Google Tango are the first step in this direction but due to the high price and energy consumption of mobile 3D sensors, devices with these sensors will not be broadly present in the market in the coming years.

One of the main disadvantages of single camera approaches compared to 3D sensors or stereo cameras is the missing scale information. From the images of a single camera it is not possible to determine the real-world scale of the scanned object. An additional way of estimating the scale is required. In the simplest case this is an artificial object in the scene that can be detected and whose size is known and thus the rest of the scene can be scaled accordingly. The drawback is that such an object needs to be available when the scan is created and the placement in the scene can be difficult. Another approach takes advantage of the fact that today's mobile phones are equipped with inertial sensing units that typically consist of at least a gyroscope and an accelerometer. These sensors with the support of the image stream allow to estimate the metrical motion of the camera directly with an Extended Kalman Filter [MTR+09, HKBR13] or by first tracking with a standard visual odometry and optimizing for the absolute scale in a batch optimization [LS12, FCDS15].

In this paper we show that an off the shelf smartphone can be used as a body part scanner with live visual feedback. The remaining of the paper is organized as follows. In Chapter 3 we will give an overview of our proposed system. Chapter 4 evaluates the proposed method and compares the reconstruction accuracy to the one of an active light sensor. Finally we conclude our work in Chapter 5.

3 Image Based 3D Reconstruction

Fig. 1 shows an overview of the proposed pipeline. In the following subsection each module is shortly described.

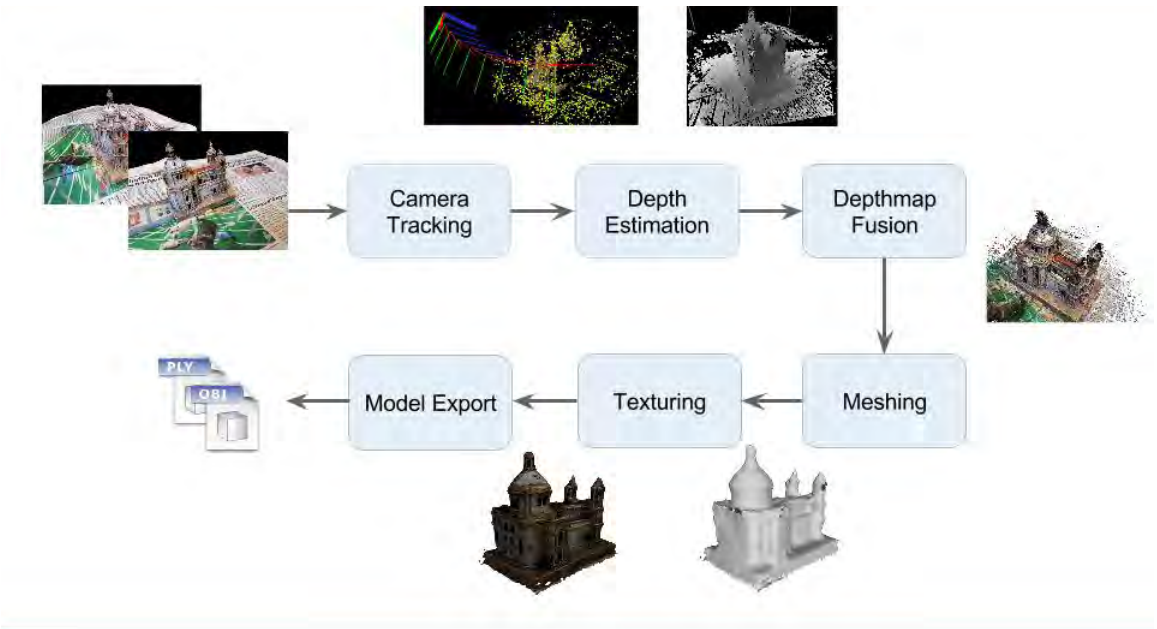

Figure 1: Proposed reconstruction pipeline, to generate a 3D model from a set of images.

3.1 Camera Tracking

The camera position is estimated by a real-time structure from motion approach that consists of two main modules. The first module tracks known scene points in the incoming image stream in real time while in the background the second module is extending and optimizing the scene map. The advantage of this design is that the tracking can run in real-time on every image and more time consuming processing can be offloaded to a separate thread and has been shown to work well in [KM09]. To initialize the camera tracking a short bootstrapping phase is required. During this time the 3D scene is unknown. 2D corners are extracted in the first camera image and are tracked while the camera is moved sideways. In every frame the relative pose between the first and the current camera frame is estimated by using the 5-point algorithm [Nis04] in RANSAC [FB81] and when enough baseline between the camera images is detected such that the scene points can be reliably triangulated the map is initialized. The camera tracker processes every frame of the live video stream. The processing starts similar to [FPS14] by pre-aligning the camera pose of a new image by minimizing the photometric error of small pixel patches around the scene points from the previous image. The optimization is done on multiple image levels to cover larger inter-frame motions. After the pre-alignment known scene points are projected into the current camera image and their exact image location is optimized by a 2D correlation of the pixels around the projected point location. The final camera pose is then computed by a reprojection error minimization. If the number of visible scene points falls below a threshold or the camera was moved far enough a new keyframe is selected and sent to the mapping thread. The mapper integrates the keyframe and all scene point observations into the map and runs an epipolar line search for new scene points in that part of the image where no points where found. This processing is done once for every new keyframe. If nothing else needs to be done, the mapper is running a bundle adjustment optimization that optimizes all map point positions and keyframe camera poses. If absolute scale is required fiducial can be placed in the scene. During tracking these markers are detected and given to the mapper that uses the measurements of the marker poses as constraints in the bundle adjustment to force the map to metric scale.

3.2 Depth Estimation

Once the camera poses of the keyframes have been computed, the next step in the pipeline is to estimate the scene depth. Given two images which are spatially registered, we search for each pixel in one image for the corresponding pixel in the other images. The pairwise matching of pixels is done by evaluating a correlation function within a certain pixel neighborhood. The best

correlation score is assumed to be a correct match, which is not necessarily true. Incorrect pixel correspondences are then removed in a later stage of the pipeline. The pixel correspondences between multiple images, then allow to triangulate the position of the 3D point, resulting in a dense 3D point cloud. The depth of a pixel is jointly estimated with the normal of the 3D point, allowing for higher accuracy when estimating the final surface of the virtual object.

3.3 Depthmap Fusion

Each new depthmap is fused into a global point cloud by projecting the reconstructed oriented points into the camera view of the current depthmap. The points are then fused in 2D by applying geometric and photometric weights. Points that are very close to each other in 3D space are likely to be duplicate observations of the same point and are merged by incremental averaging. This improves the accuracy of the points as well as the accuracy of the point normals and colors, more details about the fusion can be found in [KTSP14].

3.4 Meshing

Once the scanning process has been completed by the user, the pipeline offers a meshing stage that converts the oriented point cloud model into a 3-dimensional triangular mesh. For this, a voxel-grid volumetric representation is generated from the oriented dense points, which allows for a triangular mesh to be extracted based on the occupancy of the voxels in the grid. For the mesh extraction an approach similar to [LC87] is used. This way the water-tightness of the resulting mesh can be enforced.

3.5 Texturing

The last step in the processing pipeline textures the mesh by using the images and the optimized camera poses of the keyframes. The texture for a triangle in the mesh is obtained by the triangle's projection onto a single keyframe, that has been chosen based on some quality functional. Every mesh triangle is textured using a single texture patch from such a projection. The choice of keyframe for any specific mesh triangle is formulated as a graph based multi-label energy optimization problem, where every keyframe is represented by a label. The implementation is a variation of the algorithm from [LI07]. The graph resembles a Markov Random Field, with data costs such that they maximize the triangle's texture patch area on the texture images while having as few transitions of labels between neighboring triangles as possible.

3.6 Model Export

The reconstructed model can be extracted as a Wavefront .obj or as a polygon file format (.ply) file, however within the program the data is available in raw data structures and can be easily converted into any common or specialized format.

4 Evaluation

To evaluate the pipeline, three different devices were used. The Nexus 5X represents a standard mobile phone with a monocular camera. The LG G5 has two cameras with a standard lens and a fisheye lens. For stereo measurements, the images of the normal and fisheye camera were used. To compare the output to results from a typical depth sensor, an iPhone 6 equipped with a *Structure IO*[1] sensor was used. Images of the hardware are shown in Fig. 2. The comparison focuses on three different body parts: ear, hand and foot. The sizes are physically measured and compared to the scans obtained from the different different methods.

In the following, we will discuss two main aspects of the reconstructions. First, we qualitatively evaluate the overall quality of the reconstruction (preservation of details) and in the second paragraph we look at the overall scale recovery of the scanned object.

[1] http://structure.io

Figure 2: Hardware used to capture the data. Left most the monocular camera, a Nexus 5X phone. Center, the LG G5 phone with a stereo camera setup. Right most, the iPhone 6 with the additional *Structure IO* sensor.

Body Part	Structure Sensor (mm)	Monocular (marker) (mm)	Stereo (mm)	Ground Truth (mm)
Hand	158	160	168	157
Ear	59	61	54	60
Foot	265	275	270	273

Table 1: Scale estimation comparison to the ground truth.

Reconstruction: Especially small objects such as ears, toes or finger tips are hard to reconstruct with the depth sensor, due to the low resolution of the sensor. Moving the sensor closer towards the target doesn't help since they typically only operate within a certain range bound. According to the manufacturer of the depth sensor, its operating range is within 40cm to 3m. One of the advantages of 2D imaging sensors is that they don't have such limitations. By moving closer to the target object more details can be captured and reconstructed. Fig. 3, Fig. 4, Fig. 5 and Fig. 6 show reconstructions of various body parts (foot, ear, hand, face), which were obtained by the depth sensor and the image based reconstruction. It is clearly visible that the depth-sensor provides little details, toes, finger tips are almost not visible in the reconstruction, while they are in the image based reconstruction. The depth sensor also provides an over-smooth mesh, which makes structural details to disappear. The lack of texture makes it harder to the observer to recognize different parts of the virtual object. This can be resolved by using the camera of the mobile phone to capture the texture of the object and transfer it onto the captured object. But this is more challenging since it also requires a precise calibration between image sensor and depth sensor and makes the scanning procedure more complicated because the phone needs to be hold still to capture the color images. In contrast, the image based reconstruction method, inherently comes with color information and therefore no special calibration is required.

Object Scale Estimation: In this experiment we evaluate how precise the size of the scanned object can be recovered. Depth sensors directly provide metric size, the same holds for calibrated stereo cameras. But monocular vision systems are unable to recover the metric scale of the scanned scene. This can easily be overcome by placing a reference object of known size into the scene and reconstructing it together with the target object. For this particular experiment we use visual markers to estimate the scale, the scale differences are reported in Table 1. We found that all three methods provide comparable results with a mean error of 5mm. It needs to be noted that the chosen measurement process introduces some inaccuracy in the range of few millimeters (2-5mm).

5 Conclusion

We have shown that standard smartphones can be transformed into an interactive 3D scanner, by exploiting the image sensor and computational power of the device. We have shown that the reconstruction quality can outperform the quality of cheap structured light sensors, especially

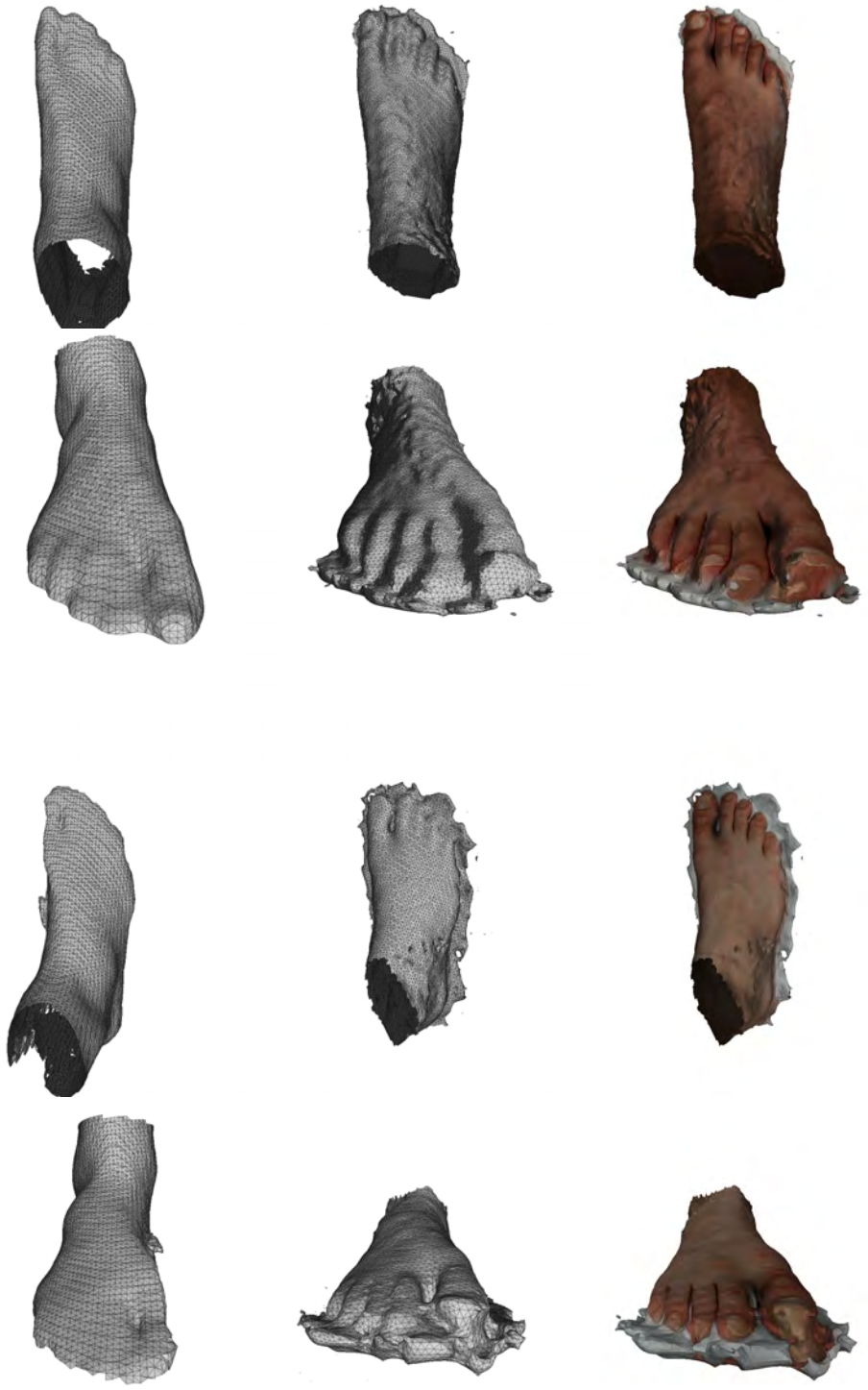

Figure 3: Foot reconstruction. Left most column: mesh obtained using the Structure IO sensor. Center column: mesh reconstructed using image based reconstruction and right most column: textured 3D model, which is the direct output of the proposed pipeline. First and second row show the reconstruction of the same foot from different perspective. Third and forth row show a reconstruction of a second foot.

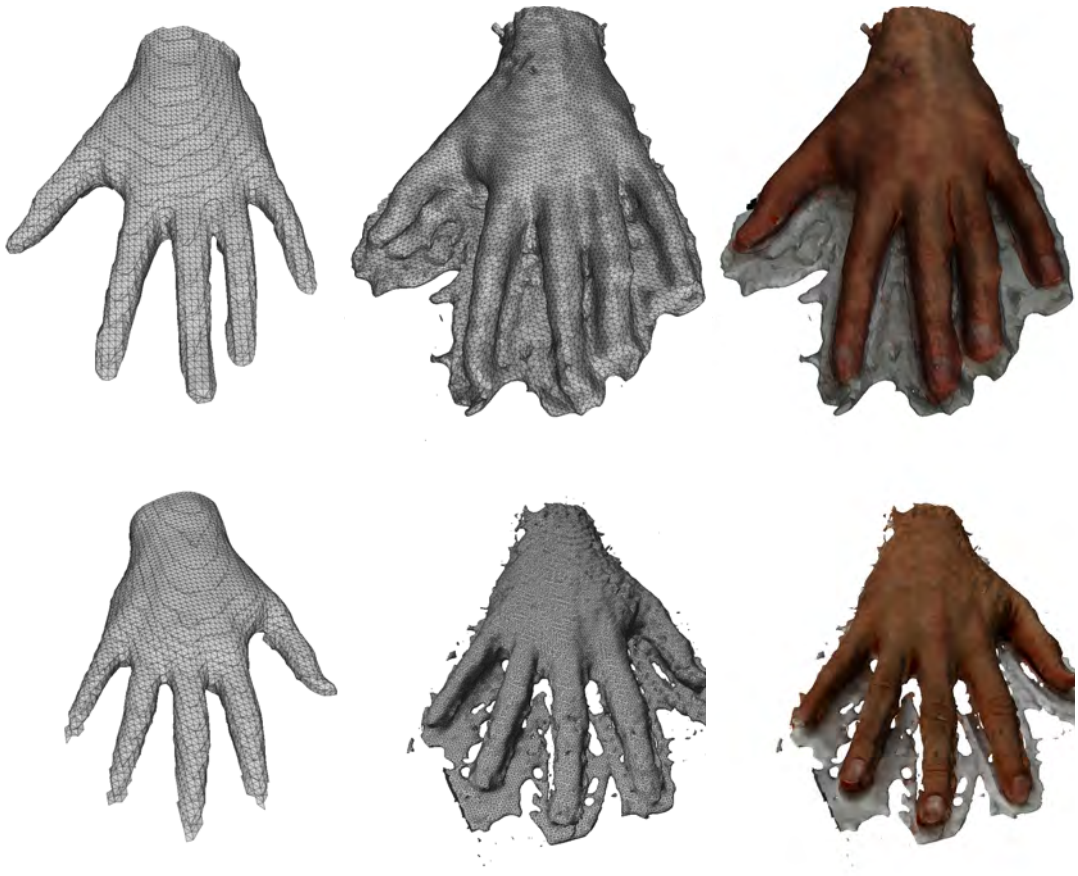

Figure 4: Hand reconstruction. Left most column: mesh obtained using the Structure IO sensor. Center column: mesh reconstructed using image based reconstruction and right most column: textured 3D model, which is the direct output of the proposed pipeline. First and second row, third and forth row show the reconstruction of the same hand from different perspective.

when working with objects that are difficult to capture within the working range of the sensor. These limitations do not exist for camera based approach but the approach requires a certain image quality to work well. In the future we plan to further optimize the algorithms to make the reconstruction process more robust and bring the technology to the end consumer.

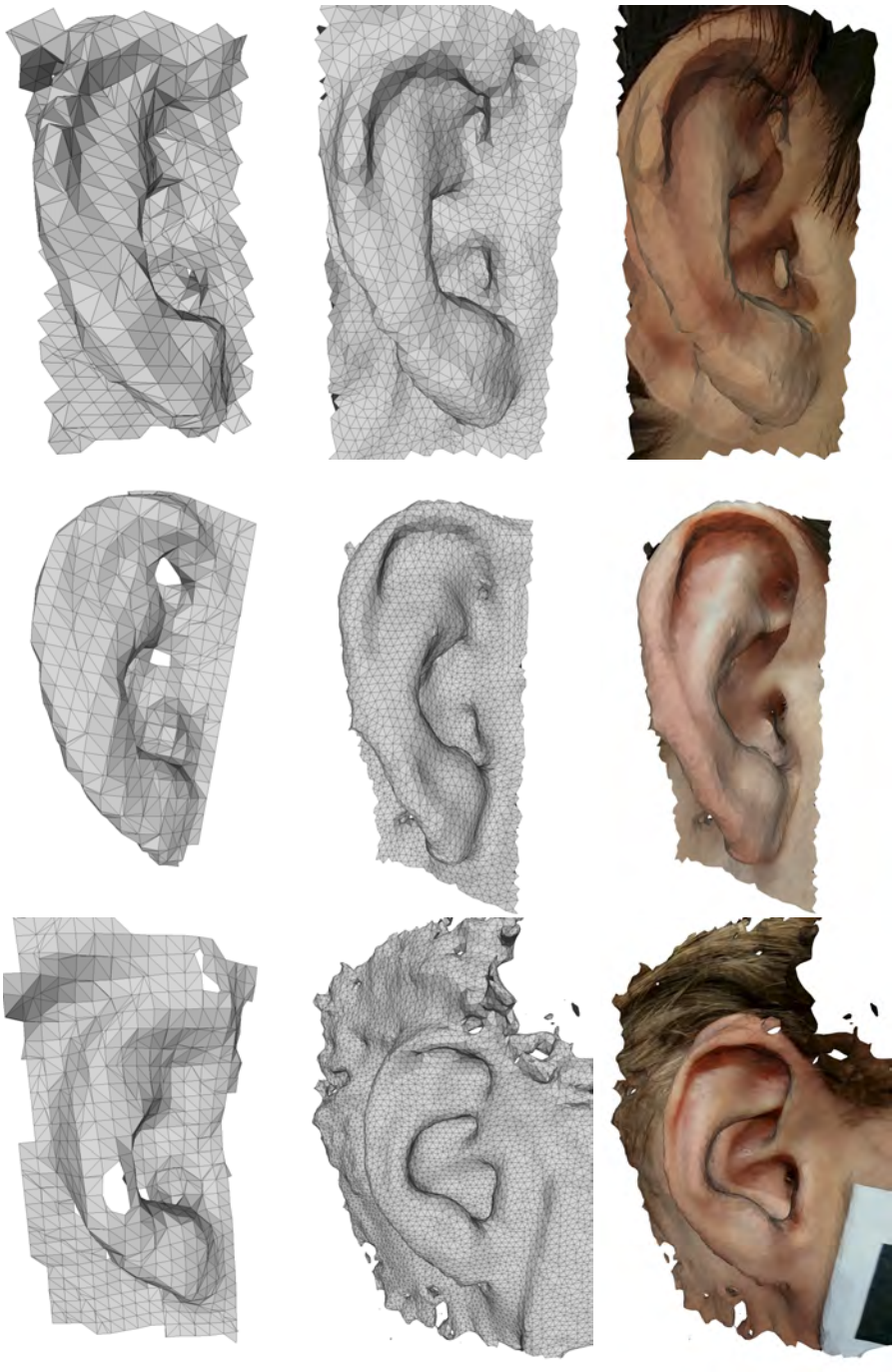

Figure 5: Ear reconstruction. Left most column: mesh obtained using the Structure IO sensor. Center column: mesh reconstructed using image based reconstruction and right most column: textured 3D model, which is the direct output of the proposed pipeline.

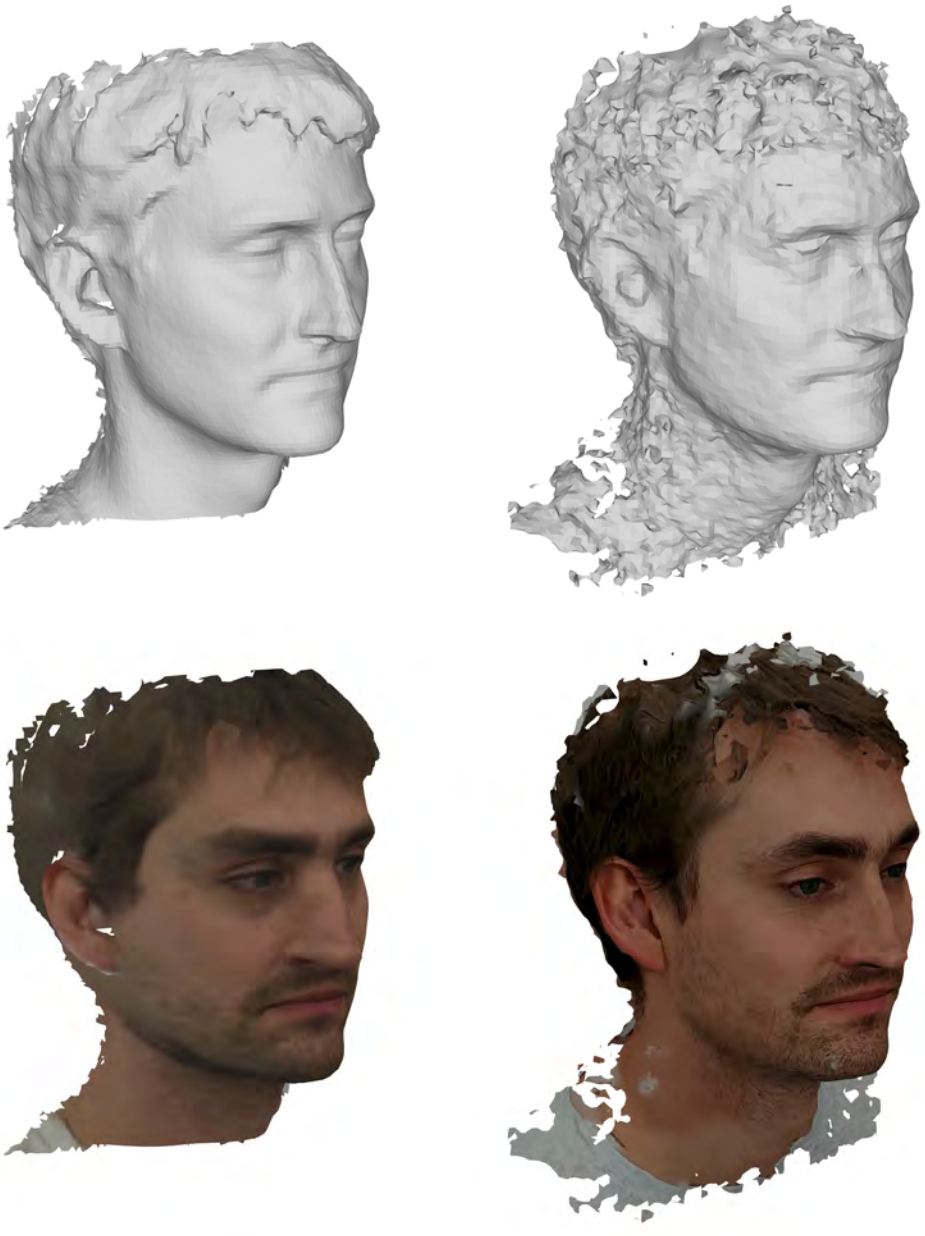

Figure 6: Face reconstruction. Left column: mesh obtained using the Structure IO sensor and right column: 3D model, which is the direct output of the proposed pipeline.

References

[ASS+09] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M Seitz, and Richard Szeliski. Building rome in a day. In *2009 IEEE 12th international conference on computer vision*, pages 72–79. IEEE, 2009.

[DKD+16] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: real-time performance capture of challenging scenes. *ACM Transactions on Graphics (TOG)*, 35(4):114, 2016.

[FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[FCDS15] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In *Robotics: Science and Systems XI*, number EPFL-CONF-214687, 2015.

[FFGG+10] Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, and Marc Pollefeys. Building rome on a cloudless day. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 368–381, Berlin, Heidelberg, 2010. Springer-Verlag.

[FPS14] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22. IEEE, 2014.

[HKBR13] Joel A Hesch, Dimitrios G Kottas, Sean L Bowman, and Stergios I Roumeliotis. Towards consistent vision-aided inertial navigation. In *Algorithmic Foundations of Robotics X*, pages 559–574. Springer, 2013.

[KM09] Georg Klein and David Murray. Parallel tracking and mapping on a camera phone. In *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*, pages 83–86. IEEE, 2009.

[KPVG00] Reinhard Koch, Marc Pollefeys, and Luc Van Gool. Realistic surface reconstruction of 3d scenes from uncalibrated image sequences. *The Journal of Visualization and Computer Animation*, 11(3):115–127, 2000.

[KTSP14] Kalin Kolev, Petri Tanskanen, Pablo Speciale, and Marc Pollefeys. Turning mobile phones into 3d scanners. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3946–3953. IEEE, 2014.

[LC87] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 163–169, New York, NY, USA, 1987. ACM.

[LI07] Victor Lempitsky and Denis Ivanov. Seamless mosaicing of image-based texture maps. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6. IEEE, 2007.

[LS12] Todd Lupton and Salah Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Transactions on Robotics*, 28(1):61–76, 2012.

[MTR+09] Anastasios I Mourikis, Nikolas Trawny, Stergios I Roumeliotis, Andrew E Johnson, Adnan Ansar, and Larry Matthies. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Transactions on Robotics*, 25(2):264–280, 2009.

[ND10] Richard A Newcombe and Andrew J Davison. Live dense reconstruction with a single moving camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1498–1505. IEEE, 2010.

[NFS15] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015.

[NIH+11] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[Nis04] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777, 2004.

[NLD11] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011.

[NZIS13] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 32(6):169, 2013.

[OKI15] Peter Ondrúška, Pushmeet Kohli, and Shahram 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.

[PAR+11] Qi Pan, Clemens Arth, Gerhard Reitmayr, Edward Rosten, and Tom Drummond. Rapid scene reconstruction on mobile phones from panoramic images. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 55–64. IEEE, 2011.

[PKVVG98] Marc Pollefeys, Reinhard Koch, Maarten Vergauwen, and Luc Van Gool. Metric 3d surface reconstruction from uncalibrated image sequences. In *European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, pages 139–154. Springer, 1998.

[PNF+08] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3d reconstruction from video. *Int. J. Comput. Vision*, 78(2-3):143–167, July 2008.

[PVV+01] Marc Pollefeys, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, and Luc Van Gool. From image sequences to 3d models. *Proceedings of Automatic Extraction of Man-Made Objects From Aerial and Space Images*, pages 403–410, 2001.

[SSHP15] Thomas Schöps, Torsten Sattler, Christian Häne, and Marc Pollefeys. 3d modeling on the go: Interactive 3d reconstruction of large-scale scenes on mobile devices. In *3D Vision (3DV), 2015 International Conference on*, pages 291–299. IEEE, 2015.

[TKM+13] Petri Tanskanen, Kalin Kolev, Lorenz Meier, Federico Camposeco, Olivier Saurer, and Marc Pollefeys. Live metric 3d reconstruction on mobile phones. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 65–72, 2013.