LASER SCAN QUALITY 3-D FACE MODELING USING A LOW-COST DEPTH CAMERA

Matthias Hernandez, Jongmoo Choi and Gérard Medioni

Institute for Robotics and Intelligent Systems
University of Southern California
3737 Watt Way, PHE 101, Los Angeles, CA, 90089
{hernanad, jongmooc, medioni}@usc.edu

ABSTRACT

We propose a method to produce laser scan quality 3-D face models from a freely moving user with a low-cost, low resolution depth camera. Our approach does not rely on any prior face model and can produce faithful geometric models of star-shaped objects. We represent the object in cylindrical coordinates, which enables us to perform filtering operations very efficiently. We initialize the model with the first depth image, and then register each subsequent cloud of 3-D points to the reference using a GPU (Graphics Processing Unit) implementation of the ICP (Iterative Closest Point) algorithm. This registration is robust in that it rejects poor alignment due to facial expressions, occlusions, or a poor estimation of the transformation. We perform both temporal and spatial smoothing of the successively incremented model. To validate our approach, we quantitatively compare our model to one produced by laser scanning, and show comparable accuracy.

Index Terms— Kinect, face modeling, graphics

1. INTRODUCTION

Since accurate 3-D face modeling with affordable sensors can open exciting applications, many researchers in computer vision and graphics have proposed methods from structured lighting systems, multiple images, videos, and even a single image [1, 2, 3, 4, 5]. We propose a method to produce laser scan quality 3-D face models from a low-cost, low resolution depth camera.

In this paper, we leverage the recent developments of 3-D sensor technologies such as the Primesense [6] camera, which can provide both a standard RGB image and a depth image containing the 3-D information at 30 frames per second in VGA format. It can also provide RGB information is SXGA format at 15 frames per second. The input to our system is a depth video stream acquired by a low-cost 3-D camera, with the user freely moving in front of it.

Getting an accurate 3-D face model from the 3-D camera is a challenging problem. Indeed, the quality of a single frame is not sufficient to generate reasonable 3-D face models. First, the data is of low resolution. Second, the sensor computes a depth map based on the triangulation principle given correspondences between stored pattern and projected pattern. Hence, depth data near boundaries can be very noisy and simple averaging on time is not sufficient.

The main idea to compensate for the noisy depth data is to use several poses, accumulate and refine noisy information through time. This can increase the resolution of the sensor by filtering the provided information.

The first frame, containing a near frontal face is set as a reference, and is used to generate both a 3-D point cloud and an unwrapped cylindrical 2-D image. Given a new frame, we convert the depth map into a 3-D point cloud and register it with respect to the reference point cloud.

To accumulate multiple views from a moving face, we propose to use unwrapped cylindrical 2-D images in canonical form, which is a well-known technique to represent a 3-D face [7, 3]. This method enables us to perform 2-D image-based operations to filter out noisy input, instead of complex 3-D mesh processing. Also, a running mean is performed on every pixel for temporal integration and a bilateral filter [8] is used for spatial smoothing. Figure 1 shows the overview of our approach.

Building a set of unwrapped cylindrical 2-D images enables to add the information on previously occluded parts and refine the bad information on the edges. However, any error in 3-D pose estimation would impact the model. To minimize the added noise, not only an accurate pose estimation process but also a rejection process is necessary. We evaluate the quality of unwrapped cylindrical 2-D images and remove all those which may be due to facial expression changes and partial occlusions.

The contributions of this paper are as follows.

- We infer a very accurate 3-D face model from a single depth camera.
- The use of a set of unwrapped cylindrical 2-D images allows us to use simple 2-D image processing algorithms, making the process computationally efficient.
- A robust registration and a rejection method produce reliable results in the presence of facial expression

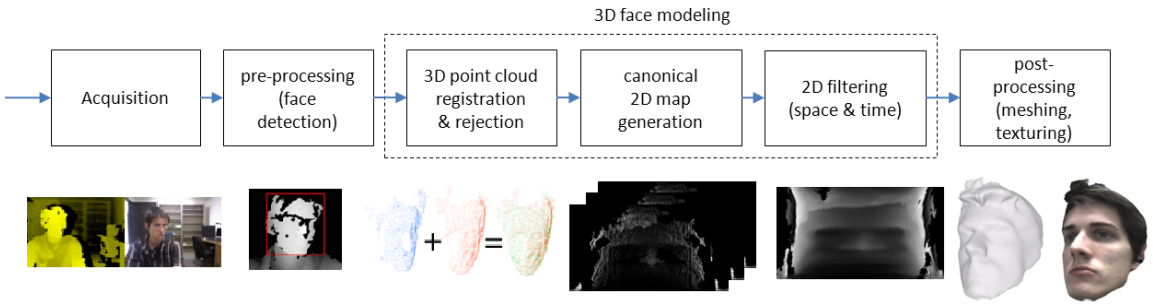


Fig. 1: Overview of the processing pipeline.

changes, partial occlusions and wide head pose angle changes.

- The reconstructed 3-D faces are compared to laser scan ground truth data.
- Our system runs in real-time.

In the following sections, we first review the state of the art, then describe the details of the proposed approach, and provide results.

2. RELATED WORK

3-D face pose estimation and registration. Our canonical 2-D map generation is based on accurate 3-D face pose estimation which is a widely studied problem [9].

In appearance-based methods, head poses are discretized in order to learn pose-related models. Morency [10] presents the AVAM (Adaptive Viewbased Appearance Model) for head tracking from stereo images which integrates differential computing and keyframe tracking paradigms. While 2-D image-based approaches provide limited results in terms of accuracy, recent works on pose estimation use increasingly affordable 3-D sensors, either purely [11] or coupled with RGB information [12]. In [13], ICP is used for pose estimation from range data and shows good results in a small range.

Recently, Fanelli [11] presented real-time head pose estimation without using GPU hardware, which estimates the pose parameters from all surface patches within a regression framework. However, the accuracy (about 5° in each axis) is not sufficient for our registration.

3-D face modeling. Many proposed methods to build a face model start from a generic model which is then deformed to fit the input face. Tang and Huang [4] create face models by locating facial features on the input face and deform the generic face model accordingly. A deformable 3-D face model was introduced to build a 3-D face from a single image in [5]. Le [14] use a linear morphable model on two stereo images to accurately reconstruct the 3D face shape. In [2], Zollhöfer builds a 3-D model using the Microsoft's Kinect sensor. He fits a generic face model to the scans and refines the registration thanks to features detected in the RGB space. All these methods provide decent results but the generic model tends to bias the reconstructed model.

A data-driven face modeling by taking advantage of SfM (Structure-from-Motion) technique with five different views is proposed in [3]. The reconstructed models are very accurate but the process requires high-resolution images.

State of the art methods that provide very high quality face models require a special equipment, such as [15], or a studio environment [1, 16]. In [1], the user has to be scanned in a ball-shaped light stage with 156 LED lights that captures the face's geometry and reflectance. [16] uses a setup of 14 high definition video cameras to capture small patches of the face surface, then applies an iterative binocular stereo method to reconstruct the model.

In [17], KinectFusion system takes live depth data from a moving Kinect camera and creates a high-quality 3-D model for a static scene object. Later, dynamic interaction has been considered in [18] where camera tracking is performed on a static background scene and foreground object is tracked independently of camera tracking. Aligning all depth points with the complete scene model from a large environment (e.g. room) provides very accurate tracking of the camera pose and mapping [17]. However, this approach does not provide accurate results for faces.

Our approach does not rely on any prior face model and can produce faithful geometric models of star-shaped objects with an affordable noisy low-resolution sensor in the presence of free head motion of a subject.

3. 3-D FACE MODELING

3.1. Accurate and robust registration for canonical 2-D map

The main idea is to use registration between the input point cloud and the point cloud from the reference frame. Note that we could register consecutive images and incrementally infer the pose. However, such methods require a really accurate pose computation since any drift would be propagated. Using a reference frame is thus more robust and can recover the pose

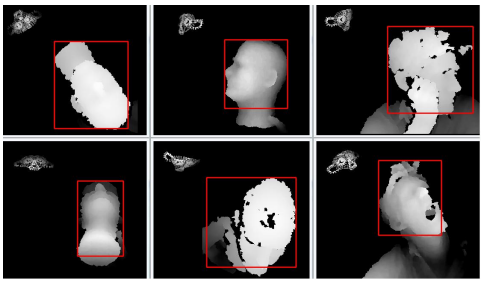


Fig. 2: Head pose estimation examples for high roll value (top-left), high yaw value (top-middle), small occlusion (top right), high positive pitch angle (bottom-left), negative pitch angle (bottom-middle) and expression change(bottom-right).

at any time after an error occurs.

The rigid transformation between the reference frame and the current input is computed by a registration algorithm [19, 20]. We set the first frame as a reference frame. Every new input is registered to that reference frame. At each frame, we segment the face region and sample the points on the face to get an input point cloud.

In our approach, the pose is estimated thanks to EM-ICP on CUDA [21], which enables to obtain real-time performance [20]. The approach does not rely on any specific facial feature which can be challenging to detect depending on the pose.

Note that the initialization step is critical for both accuracy and speed. A wrong initialization would either make the system really slow or converge towards a local minimum and do not provide the desired results. This is handled by initializing the transformation matrix at time t by the value it had at time *(t-1)*. This hypothesis holds since the difference of object position is typically small between two consecutive frames.

Our module provides good pose estimation results for $-40°$ to $70°$ for pitch angles, $-70°$ to $70°$ for yaw angles and $360°$ for roll angle, which is enough for casual behavior in front of a camera. The system can handle some occlusions, some translations along Z and expression changes (Fig. 2). Moreover, it can recover if the person goes out of the field of view and back in.

Our system runs at 6 frames per second on a GeForce GTX460. We use around $1,000$ points for each frame, which provides a good trade-off between speed and accuracy.

3.2. Canonical 2-D map

The key idea for modeling is to accumulate information through time. By processing a video with several poses for the face, we fill in the information missing in a single image. Moreover, the noise of the depth information provided by the sensor can be significantly removed by filtering in both space and time.

The main goal is to add the information provided at each new frame to the model. Two challenges appear. First, the new information should be aligned to the model accurately if we want a consistent result. Second, we want to aggregate the information efficiently.

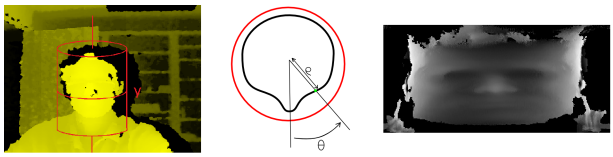


Fig. 3: Cylindrical representation. A cylinder is set around the face (left). The 3-D information is projected onto the cylinder (middle), giving an unwrapped depth map (right).

A cylindrical model is used. This has proven to give good results, as in [7, 3]. Practically, a cylinder is set around the face in the reference frame. The axis of the cylinder is the vertical axis going through the middle of the head, whose location is loosely estimated by taking the middle of the face and setting the z value 10cm deeper than the closest point. Note that an approximate location of the axis is sufficient.

The geometry of a facial surface can be represented using an unwrapped cylindrical depth map D, where the value at $D(\theta, y)$ is the horizontal distance ρ to the cylinder axis (Fig. 3). Figure 3 (right) shows an example of the unwrapped map D generated from one image.

This model enables us to transform easily the 3-D data into a 2-D image. It has several advantages. First of all, it limits the amount of data to a single image, which is suitable for an algorithm where information is endlessly added at each new frame. Second, the 3-D data can be processed as a 2-D image. Processing such as filtering becomes easy to use and can be applied very fast. Third, meshes can be created easily and quickly by creating triangles among the neighboring pixels on the image (Section 4).

The main drawback of this model is that it can handle only star-shaped objects. However, it is suitable for a face and enables fast computation.

3.3. 2-D image-based filtering

To obtain a smooth model, we remove the noise from both the input data and the error in pose estimation by the combination of temporal integration and spatial filtering (Fig. 4).

First, we apply a running mean on the ρ value of each pixel of the unwrapped cylinder map. This temporal integration reduces the intrinsic noise while aggregating the data (Fig. 4).

After aggregating the whole data, the obtained model is still not perfect and has to be refined. For each row in the cylindrical map, we apply a simple linear interpolation method to fill up the remaining holes. Note that linearly interpolated points in the 2-D map fall on a 3-D circular arc in the original space. Then, we process the unwrapped cylindrical

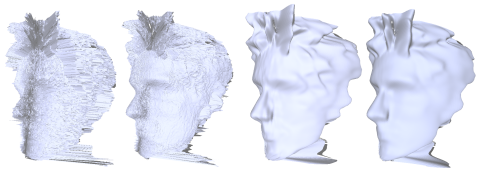


Fig. 4: Noise removal using filtering: Accumulated raw input (a), Running mean only (b), Bilateral filtering only (c), Both filters (d).

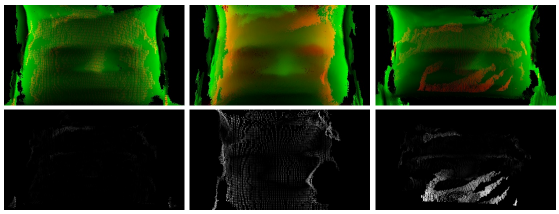


Fig. 5: Overlapping error. The model is displayed in green and the new frame in red (top line). The intensity shows the overlapping error (bottom line) in case of good registration (left), bad registration (middle) and occlusion by a hand (right).

map to remove the remaining noise (Fig. 4).

We choose to apply a bilateral filter [8], which removes noise while keeping the edges.

As mentioned earlier, a good head pose estimation is needed to get a good model. In order to handle pose estimation failure, we reject images in which the pose could not be properly computed. Let us note M and I the unwrapped cylindrical images containing respectively the model and the new image. We simply reject I if the difference between M and I is too large. Rejected images are those in which the registration was poor or where an object was occluding the face (Fig. 5).

4. MESHING AND TEXTURING

To create the mesh, we use neighboring pixels of the unwrapped cylindrical 2-D image containing the model. A mesh is a triangle whose corners are neighboring pixels. Note that our representation is very flexible is terms of resolution. We can easily predefine areas in the unwrapped image where more resolution is needed (i.e. the nose) and reduce resolution elsewhere (cheeks, forehead, etc.).

Adding the color information is another complex problem. Applying a similar algorithm on the RGB input does not provide good results. Indeed, averaging makes every part of the face blurry and the result does not look natural.

In order to get a crisper texture, we use only one image, which is the reference image. When the 3-D information of the model is computed, every point of the model is projected onto this image in order to get the RGB value.

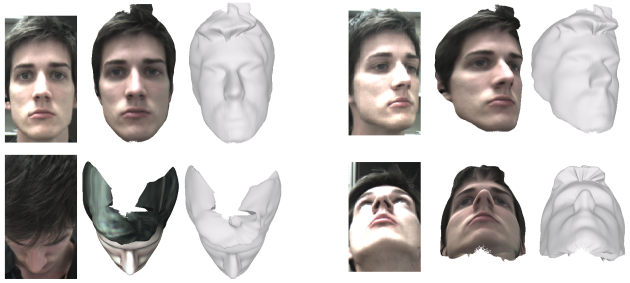


Fig. 6: Reconstructed model of a user (left) rendered in MeshLab [22] with texture (center) and without texture (right).

Fig. 7: Reconstruction of several people and objects.

5. COMPARISON RESULTS

In our experiments, we choose a size of 360×200 for the unwrapped cylindrical map. We work on a single-core Windows 7 ($\times 32$) system with a 2.79 GHz processor. GPU is used for pose estimation and uses a GeForce GTX460. It takes about 10ms to add a new frame and we can get a complete model in about 10 seconds of live video.

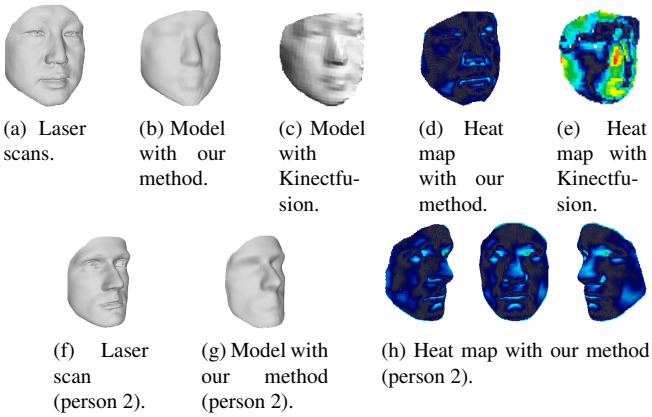
We built many face models. The results are visually accurate, especially the 3-D shapes (Fig. 6 and 7).

To quantify the accuracy of our model, we compare it to a commercial laser scan (Fig. 8a and 8f) from Cyber F/X [23]. The first thing to notice is that our method can get depth for the hair while the laser scanning systems cannot. That is why the error on the hair region needs to be ignored. We can see that our model is very close to a laser scan (Fig. 8). The average error is about 1 mm (Fig. 9).

Besides, we ran Kinectfusion 10 times on the Asian face and collected the best results. The reconstructed 3-D faces did not capture details of the face structure (Fig. 8c).

6. CONCLUSION

We have proposed and implemented an efficient, 3-D face modeling method. The key of our approach is the combination of temporal integration and spatial smoothing to reduce the noise. Reducing the noise in each pixel results in reducing the variance of the data, which enables to increase the precision and get a higher resolution.



(a) Laser scans. (b) Model with our method. (c) Model with Kinectfusion. (d) Heat map with our method. (e) Heat map with Kinectfusion.

(f) Laser scan (person 2). (g) Model with our method (person 2). (h) Heat map with our method (person 2).

Fig. 8: Comparison of our model to laser scans.

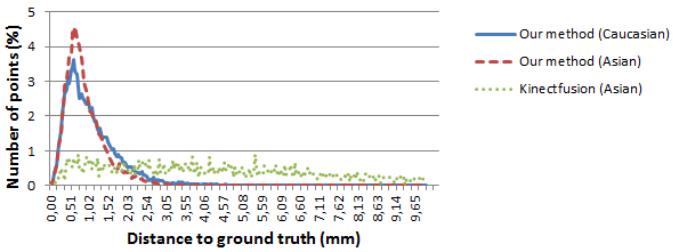


Fig. 9: Error distribution.

Experimental results with laser scan data confirm the accuracy of reconstructed models. Our method also performs as well as other state of the art methods while using a low-cost low-resolution noisy sensor. We will present a live demo, showing excellent face modeling results under large motion, fast movement, occlusion and facial expression variations.

The most obvious thing to improve is the texture information of the model. Lighting modeling and photometric calibration among RGB images is possible, given the reconstructed 3-D surface.

References

[1] O. Alexander, M. Rogers, W. Lamberth, J.-Y. Chiang, W.-C. Ma, C.-C. Wand, and P. Debevec, "The digital emily project: achieving a photorealistic digital actor," *IEEE Computer Graphics and Applications*, 2010.

[2] Michael Zollhöfer, Michael Martinek, Günther Greiner, Marc Stamminger, and Jochen Süßmuth, "Automatic reconstruction of personalized avatars from 3d face scans," *Computer Animation and Virtual Worlds (Proceedings of CASA 2011)*, vol. 22, no. 3-4, 2011.

[3] Y. Lin, G. Medioni, and J. Choi, "Accurate 3D face reconstruction from weakly calibrated wide baseline images with profile contours," *CVPR*, 2010.

[4] L. Tang and T. Huang, "Automatic construction of 3D human face models based on 2D images," *ICIP*, pp. 467–470, 1996.

[5] B. Amberg, A. Blake, A. Fitzgibbon, S. Romdhani, and T. Vetter, "Reconstructing high quality face-surfaces using model based stereo," *ICCV*, 2007.

[6] PrimeSense, "http://www.primesense.com/," .

[7] L. Williams, "Performance-driven facial animation," *Computer Graphics*, vol. 24, no. 4, 1990.

[8] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *IEEE Conference of Computer Vision*, 1998.

[9] E. Murphy-Chutorian and M. Trivedi, "Head pose estimation in computer vision: A survey," *TPAMI*, vol. 31, no. 4, pp. 607–626, 2009.

[10] L.-P. Morency, A. Rahimi, and T. Darrell, "Adaptive viewbased appearance models," *CVPR*, vol. 1, pp. 803–810, 2003.

[11] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," *CVPR*, 2011.

[12] A. Bleiweiss and M. Werman, "Robust head pose estimation by fusing time-of-flight depth and color," *MMSP*, 2010.

[13] D. Simon, M. Hebert, and T. Kanade, "Real-time 3-D pose estimation using a high-speed range sensor," *IEEE International Conference on Robotics and Automation (ICRA '94)*, vol. 3, pp. 2235–2241, 1994.

[14] V. Le, H. Tao, L. Chao, and T.S. Huang, "Accurate and efficient reconstruction of 3d faces from stereo images," *International Conference on Image Processing*, 2010.

[15] 3D3 Solutions, "http://www.3d3solutions.com," *Demonstration at SIGGRAPH*, 2011.

[16] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer, "High resolution passive facial performance capture," *IEEE Computer Graphics and Applications*, vol. 30, no. 4, 2010.

[17] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," *ISMAR 2011*, 2011.

[18] S. Izadi, R. A. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, J. Shotton, A. J. Davison, and A. Fitzgibbon, "Kinectfusion: real-time dynamic 3d surface reconstruction and interaction," *ACM SIGGRAPH 2011*, vol. 23, 2011.

[19] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 119–152, 1994.

[20] D. Luebke and G. Humphreys, "How gpu works," *IEEE Computer*, pp. 126–130, 2007.

[21] T. Tamaki, M. Abe, B. Raytchev, and K. Kaneda, "Softassign and EM-ICP on GPU," *CVPR*, 2010.

[22] Meshlab, "http://meshlab.sourceforge.net/," .

[23] Cyber F/X, "http://www.cyberfx.com," .