Photorealistic Texturing of Human Busts Reconstructions

Christoph HEINDL*, Sharath Chandra AKKALADEVI, Harald BAUER
PROFACTOR GmbH, Steyr, Austria

DOI: 10.15221/16.225 http://dx.doi.org/10.15221/16.225

Abstract

Recent developments in the RGBD sensors systems, lead to a new generation of sensors providing high quality color images along with moderate depth resolution images. Most state-of-the-art real-time reconstruction algorithms support colorized reconstruction by methods that rely on vertex colorization. However, the resolution of vertex colorization is directly linked to mesh resolution. High quality color images therefore require an enormous amount of vertices in order to capture image details. This leads to unattractive large model files, large memory consumption and long processing times. In this work we propose an automated pipeline that is able to texture low-resolution geometric meshes with high quality color images. Experiments evaluate the proposed method in challenging environments such as varying illumination and non-rigid reconstructions.

Keywords: 3d body scanning, texture mapping, HD texture, low-cost;

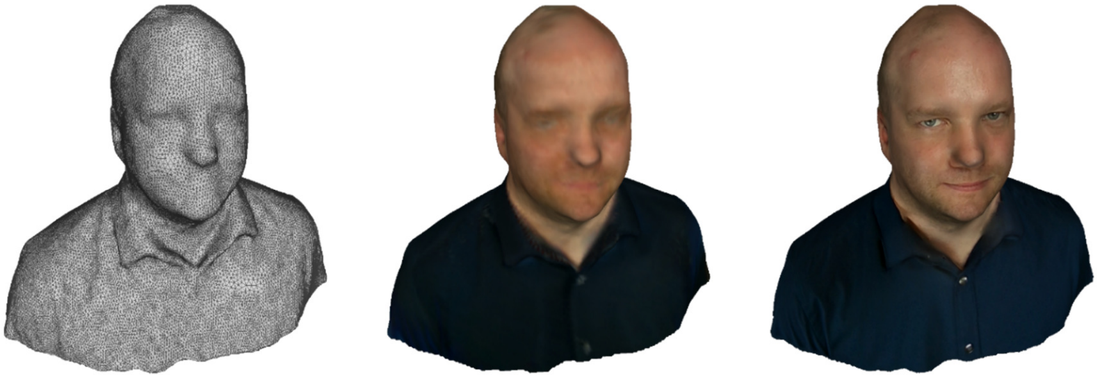

Figure 1 Comparison of vertex coloring and texture mapping. Left: Wireframe of model consisting of 70k vertices. Middle: Vertex colored version using 70k vertices. Right: Texture mapped version using 10k vertices.

1. Introduction

With the availability of low-cost RGBD sensors [1] there is growing interest in creating 3D models of objects, workspaces and human busts etc., on the fly. Until recently the intrinsic resolution of color and depth streams have been on par. Modern RGBD sensors, such as the Intel SR300 and Kinect One, provide color images that outperform previous sensor generations in the number of color pixels offered.

Many real-time reconstruction algorithms, such as KinectFusion [3] and variants [5], support colored model outputs. The colorization approach often utilized, assigns a unique color to every vertex of the model (vertex coloring). When rendered, pixels in-between vertices are interpolated from surrounding vertices. This generally leads to a blurred visual appearance. To increase the model’s color resolution, one therefore needs to increase the number of vertices of the model. However, capturing all image details often leads to a vertex resolution that is beyond the limit of current systems and is computationally very expensive.

Texture-mapping has proven to be a sensible replacement for vertex coloring in 3D model reconstruction [4][6][7]. In texture mapping, vertices are assigned a unique texture coordinate that refers to a specific location in the color image. When rendered, texture coordinates for the pixels in-between vertex locations are interpolated from surrounding ones and colors are directly read from those locations from the associated texture map. One of the biggest advantages of texture-mapping over vertex coloring is the possibility to increase color resolution independently of the geometric resolution and hence a reduced computational cost. Figure 1 compares the both methods applied to the same input.

In the following sections we propose fully automatic texturing mapping framework

- for generating visually appealing photorealistic (colored) 3D reconstructions of human busts
- using low-cost RGBD sensors
- that is computationally attractive

The remainder of this paper is organized as follows: In section 2 we introduce the hardware setup used to capture the input data and detail our reconstruction and texture mapping approach. In section 3 we present the experimental approach and evaluation. Section 4 concludes the paper with outlook of future work.

2. Automatic Texture Mapping

2.1. Setup

For geometric and texture reconstruction, pairs of color and depth images are required. In order to acquire these in a consistent way, our mechanical setup consists of a single fixed RGBD sensor mounted on a vertical bar, a turntable for the human to step on, and an additional light source that is optional. The sensor height is manually adjustable for support of different sized users. The distance between the sensor and the human head is roughly 60 centimeters. More details of the setup can be found in [12].

2.2. Data Capture

Each full turn of the turntable takes around 20 seconds. During movement the RGBD sensor captures roughly 600 pairs of depth images (resolution 640x480) and color images (resolution 1920x1080). Additionally, we obtain the intrinsic camera parameters of both the color and depth camera and the relative transformation between both cameras using stereo calibration.

2.3. Geometric Reconstruction

For geometric reconstruction, we utilize a voxel volume based reconstruction approach that runs in real-time. We use ReconstructMe SDK [2] for this step. During reconstruction of depth images, we take note of the tracked camera position in each frame and compute the relative transformation between geometric model and each color image. Once the reconstruction is complete, we fill in any topological holes using Poisson reconstruction [8] and perform a horizontal cut at the bottom of the bust. For performance reasons we execute this operation using CSG [9][2] operators defined directly over the voxel volume. Figure 2 illustrates the result and intermediate steps of this operation.

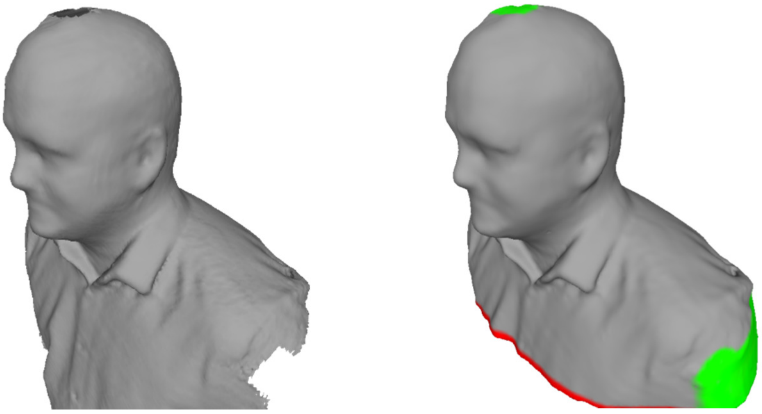

Figure 2 Left: Raw volume reconstruction. Right: Hole-Filled (green) and cut (red) model

2.4. Texture Mapping

The way data capturing is performed leads to varying visual appearances of the object between individual views. Additionally, the acceleration of the turntable induces non-rigid body motion that results in inaccurate geometric reconstructions. An automatic texture mapping system needs to account for these factors in order to generate a visually appealing reconstruction. As outlined above, our method takes as input a triangulated watertight surface and a set of color images and their viewpoints computed during reconstruction.

2.4.1. Bundle Adjustment

Viewpoints associated with color images tend to be erroneous for specific reasons: a) the commodity depth and color cameras are rarely synchronized well in time. b) our system needs to estimate the camera pose during turntable registration from depth maps. These factors induce additional frame-to-frame tracking errors [10]. Both errors tend to accumulate in the viewpoint estimation and lead to visible seams between neighboring viewpoints.

In order to minimize the error we apply a variation of bundle adjustment [11] approach, in which correspondences between color images are found via 3D model matching. Given the set of viewpoints *R* and the mesh *S,* the objective of our bundle adjustment approach is to maximize the color intensity at each vertex location between of all registered views.

$$E(\mathcal{R},\mathcal{S}) = \sum_{i=1}^{N} \sum_{v \in \mathcal{S}_{vertices}} k_{v,i}(A(v) - I_i(\pi T_i^{-1} v))^2$$

Here *A(v)* denotes the average intensity at vertex *v, and* I_i is the image intensity at the vertex location projected into the i-th registered view. A detailed description of this method is given in [12]. The equation is minimized using an iterative Gauss-Newton solver scheme.

2.4.2. Chart Creation

Given the surface and the updated set of registered viewpoints, texture mapping requires a unique image assignment for each surface triangle. In order to reduce the possible candidates, we first purge all images in which a specific triangle is unseen using ray-tracing. Next, an energy measure for each triangle - viewpoint pair that reflects how well a triangle is seen in the given viewpoint is computed. For computing the data energy term, a gradient based measurement [13] is computed that captures the distance from the surface, sharpness of the image and orthogonality between view rays and triangle orientations. In order to avoid over fragmentation, another energy term that reflects how well neighboring triangles agree with the current label in question is introduced.

The minimization of both energy terms is formulated as Markov Random Field and is solved using loopy belief propagation [14]. All topological connected regions carrying the same viewpoint level are extracted and each of these regions is called a chart.

2.4.3. Color Correction

Illumination and exposure differences viewpoints influence the visual appearance of the textured model. Especially across chart boundaries large color gradients are easily spotted. Leveling of these inequalities in done in a twofold process:

First, a global color correction is carried out over all viewpoints as described in [15]. The knowledge of the 3D model to estimate color correspondences required in the optimization procedure is also used. After global color correction, hard visual seams on chart boundaries (usually a result of micro movements during data capture) are identified. Poisson image editing [16] methods is applied to in-paint these narrow spots within a local support region around the visual edge.

2.4.4. Area In-painting

The manner in which data is captured often results in specific parts of the body not being perceived. This is especially true to the scalp region and regions under the chin in case of human busts. As explained in the geometric reconstruction section, unseen geometry is filled using a Poisson solver over 3 dimensional space. Similarly, one can infill unseen regions in color space in the following way:

First, each unseen geometric region is unfolded to a plane using conformal maps optimization [17] in order to obtain texture coordinates for each vertex. Next, an empty image is created that is able to hold the unfolded chart in desired resolution. Then the chart border colors from corresponding colors of associated viewpoints are transferred. Finally, it is solved over a Poisson equation

$$\Delta\delta(u,v) = 0$$

letting the border color values act as Dirichlet boundary conditions to constrain the system. Linearizing the Laplacian leads to a linear system of equations that can be solved efficiently. The resulting image shows a smooth color transition from border to image center. This approach usually works well for moderate sized regions. For large areas this method leads to noticeable blurring and texture based in-painting methods can be used [20].

2.4.5. Chart Stitching

For convenience and compatibility with post-processing products, it makes sense to pack all charts into one larger unit. This process is usually called texture-atlas or simply texture-map.

When each chart in texture space by its axis bounding box is parameterized, it becomes a bin-packing type of problem. Bin-packing has been widely studied in literature and is known to be NP-hard [18]. Various heuristics and approximations have been proposed to tackle the problem in an acceptable manner. The proposed algorithm is based on a modified MAXRECT [19], that additionally enforces certain constraints on texture map sizes.

After texture chart packing and adapting the vertex texture coordinates accordingly, chart colors are extrapolated into texture-map background regions. This helps in avoiding edge bleeding artifacts when rendering the model. These artifacts stem from numerical inaccuracies when down sampling textures and coordinates.

3. Experiments

The framework is validated by scanning several hundred subjects during a national research event including many challenging poses. Sample results are shown in Figure 3.

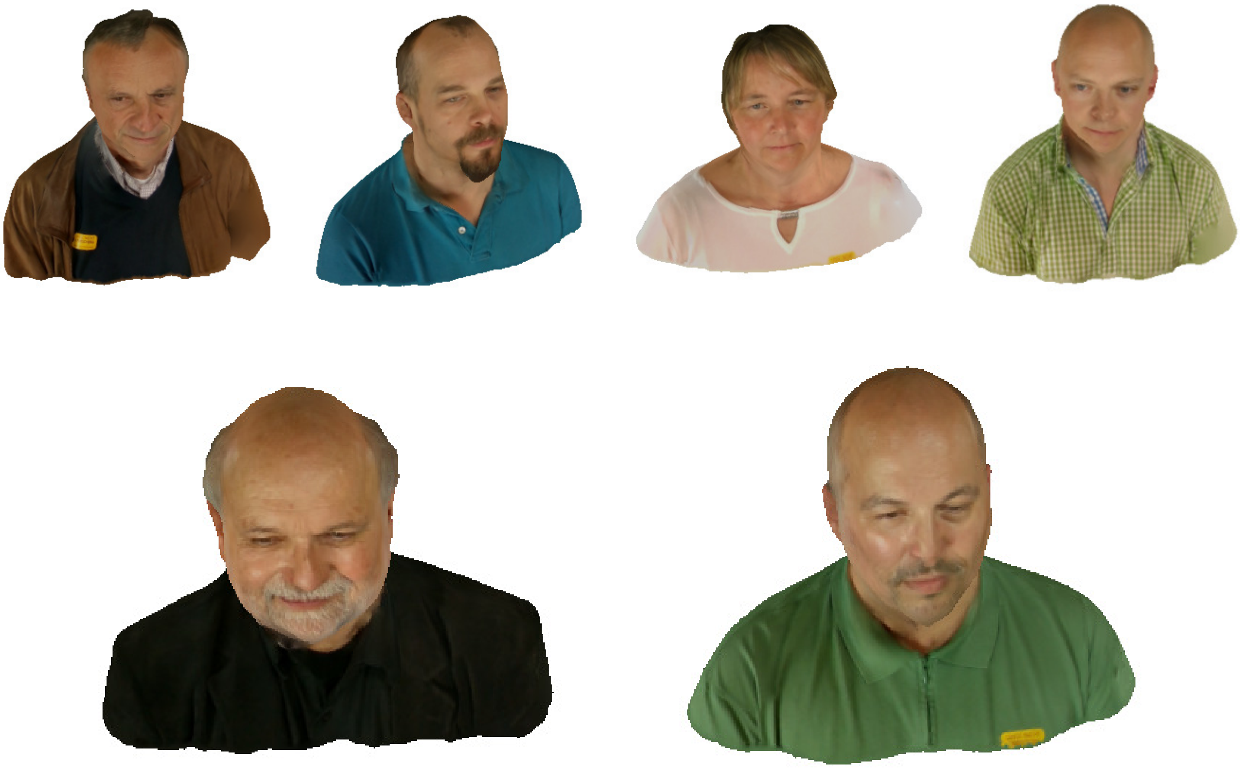

Figure 3 Sample results from our capture database

Reconstruction and texture quality is assessed using synthetic depth and color images. Ground truth data is extracted from a virtual camera movement around a hand-crafted 3D model. As described in [12], the geometric error fluctuates around 0.9mm +/- 0.6mm and the texture intensity deviates 5 percent on average from ideal intensity values. Geometric deviations are as shown in Figure 4.

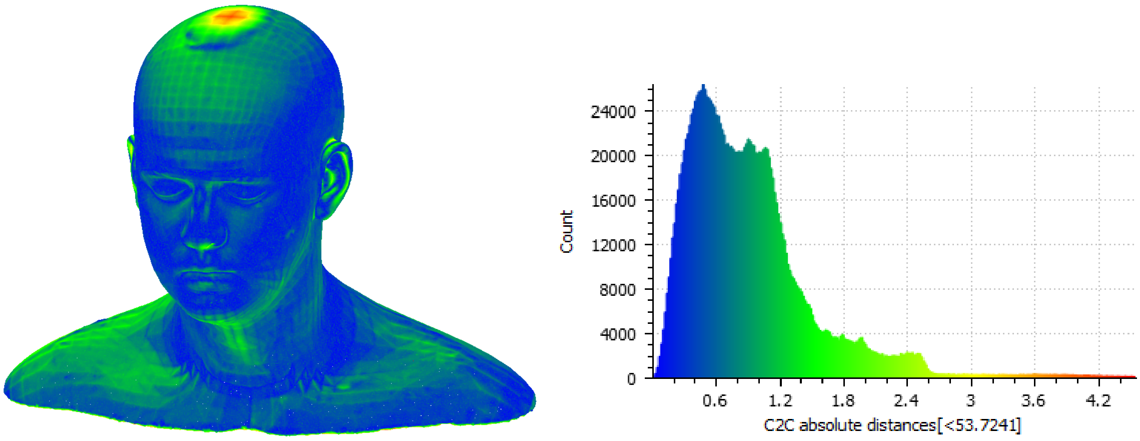

Figure 4 Geometric deviation Left: Color coded reconstruction errors of synthetic input data. Right: Histogram of error distribution in mm.

For benchmarking the runtime, the proposed method was implemented in C++ and all experiments were carried out on a commodity desktop pc running Intel Core i5-2000 CPU with an NVIDIA Geforce GTX 560 and 16GB of RAM. Figure 5 compares the total runtime of the proposed method with varying color image resolutions.

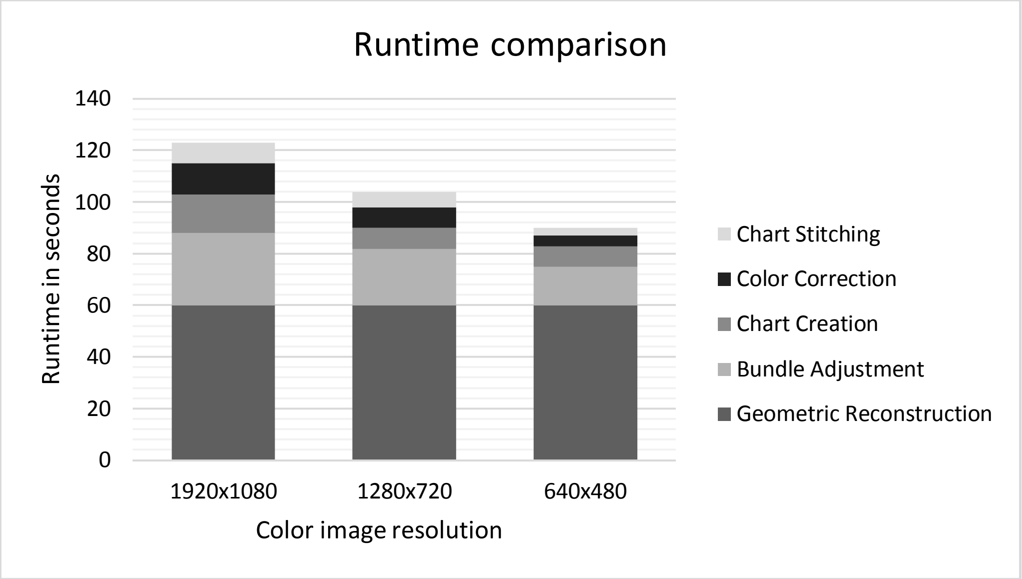

Figure 5 Runtime results with varying color image resolutions. The geometric reconstruction time is similar for all experiments as the depth image resolution has not changed during experiments.

4. Conclusion and Future Work

In this paper, a method that automatically produces photorealistic 3D models from low-cost RGBD sensors is proposed. While the system works well in most cases, points for improvement are also identified. One future step would be to reformulate bundle adjustment to allow for geometric deformations. This should improve visual appearance when non-rigid body motions are present in the captured data. Another field of action concerns the retrieval of the underlying albedo from color images.

A semantic differential questionnaire to evaluate the influence of parameter changes on aesthetic appearance of the model is also conducted. In the future this should help to gain insights on modelling aesthetic parameters from this data and a learning based approach can be used to classify and tune visual appearance of 3D reconstructions.

5. Acknowledgment

This research is carried out within the "FTI-Projekt ProTechLab" project funded by the State of Upper Austria through the Strategic Economic and Research Program "Innovatives OÖ 2020"

References

[1] Cruz, Leandro, Djalma Lucio, and Luiz Velho. "Kinect and rgbd images: Challenges and applications." *Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 2012 25th SIBGRAPI Conference on*. IEEE, 2012.

[2] Heindl, Bauer, "ReconstructMe SDK: a C API for Real-time 3D Scanning", 6th International Conference and Exhibition on 3D Body Scanning Technologies, 2015

[3] Newcombe, Richard A., et al. "KinectFusion: Real-time dense surface mapping and tracking." *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE, 2011.

[4] Soucy, Marc, Guy Godin, and Marc Rioux. "A texture-mapping approach for the compression of colored 3D triangulations." *The Visual Computer* 12.10 (1996): 503-514.

[5] Li, Hao, et al. "3D self-portraits." *ACM Transactions on Graphics (TOG)* 32.6 (2013): 187.

[6] Rocchini, Claudio, et al. "Multiple textures stitching and blending on 3D objects." *Rendering Techniques' 99*. Springer Vienna, 1999. 119-130.

[7] Gal, Ran, et al. "Seamless montage for texturing models." *Computer Graphics Forum*. Vol. 29. No. 2. Blackwell Publishing Ltd, 2010.

[8] Kazhdan, Michael, Matthew Bolitho, and Hugues Hoppe. "Poisson surface reconstruction." *Proceedings of the fourth Eurographics symposium on Geometry processing*. Vol. 7. 2006.

[9] Requicha, Aristides AG, and Herbert B. Voelcker. "Constructive solid geometry." (1977).

[10] Rusinkiewicz, Szymon, and Marc Levoy. "Efficient variants of the ICP algorithm." *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. IEEE, 2001.

[11] Triggs, Bill, et al. "Bundle adjustment—a modern synthesis." *International workshop on vision algorithms*. Springer Berlin Heidelberg, 1999.

[12] Christoph Heindl, Sharath Chandra Akkaladevi, Harald Bauer. "Capturing Photorealistic and Printable 3D Models Using Low-Cost Hardware.*" (to appear) the 12th International Symposium on Visual Computing*. Springer, 2016

[13] Lempitsky, Victor, and Denis Ivanov. "Seamless mosaicing of image-based texture maps." *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007.

[14] Szeliski, Richard, et al. "A comparative study of energy minimization methods for markov random fields with smoothness-based priors." *IEEE transactions on pattern analysis and machine intelligence* 30.6 (2008): 1068-1080.

[15] Brown, Matthew, and David G. Lowe. "Automatic panoramic image stitching using invariant features." *International journal of computer vision* 74.1 (2007): 59-73.

[16] Pérez, Patrick, Michel Gangnet, and Andrew Blake. "Poisson image editing."*ACM Transactions on Graphics (TOG)*. Vol. 22. No. 3. ACM, 2003.

[17] Lévy, Bruno, et al. "Least squares conformal maps for automatic texture atlas generation." *ACM Transactions on Graphics (TOG)*. Vol. 21. No. 3. ACM, 2002.

[18] Coffman Jr, Edward G., Michael R. Garey, and David S. Johnson. "Approximation algorithms for bin packing: a survey." *Approximation algorithms for NP-hard problems*. PWS Publishing Co., 1996.

[19] Jylänki, Jukka. "A thousand ways to pack the bin-a practical approach to two-dimensional rectangle bin packing." *retrived from http://clb. demon. fi/files/RectangleBinPack. pdf* (2010).

[20] Criminisi, Antonio, Patrick Pérez, and Kentaro Toyama. "Region filling and object removal by exemplar-based image inpainting." *IEEE Transactions on image processing* 13.9 (2004): 1200-1212.