Simultaneous Incremental Reconstruction of Object Geometry and Appearance for Interactive 3-D Model Acquisition

Hannes Dohrn Hannes Stadler Marco Winter Günther Greiner

University of Erlangen-Nuremberg
Chair of Computer Science 9 (Computer Graphics)
Am Wolfsmantel 33
91058 Erlangen, Germany

{sihadohr, sihastad}@i9.informatik.uni-erlangen.de,
{marco.winter, guenther.greiner}@informatik.uni-erlangen.de

ABSTRACT

While the creation of three-dimensional models from real-life objects is a commonly applied process, the reconstruction of complete datasets from such objects is still a delicate task. In this paper, a simple yet powerful framework is proposed that is able to reconstruct both geometry and appearance of a given object interactively. Working in an incremental mode of operation, it enables the user to reconstruct a given scene with full visual feedback during the progress of the reconstruction process. For geometry reconstruction, an existing surface reconstruction algorithm has been investigated and adjusted to the needs of the framework. Furthermore, a hardware-accelerated surface light field algorithm has been integrated into the framework that performs appearance reconstruction of the object.

The target application of our framework is the reconstruction of real-life objects using mobile acquisition devices. We demonstrate the performance and usefulness of our framework by reconstructing models from previously acquired datasets of real-life objects. Furthermore, we provide results of experiments run in our own model acquisition setup.

Keywords

Image-Based, Acquisition, Light fields, Modeling, Reconstruction, Interaction

1. INTRODUCTION

Today, the creation of three-dimensional models from real-life objects is a commonly applied process. Typical applications include environment modeling, rapid prototyping, virtual museums or the digitalization of scene objects in movie industry. However, the reconstruction of a complete dataset from a real-life object that is ready for rendering or further processing still is a delicate task.

Various methods have been developed to support this process. Structure-from-motion or stereo algorithms are used to reconstruct geometric information using color images of a given scene. Furthermore, hardware devices have been developed that allow for the direct acquisition of partial geometry information of objects, and software algorithms are successively applied to reconstruct the complete surface from this data. While matured, these approaches generally produce imperfect results in the case of insufficient data. Therefore, it is preferable to reconstruct 3-D models incrementally, i. e. to consecutively add new input data to an existing partial reconstruction for further refinement. By visualizing the current results the user has the ability to actively control the reconstruction process. However, this requires the applied reconstruction and visualization methods to work at interactive frame rates.

Today, first hand-held multimodal input devices are available that facilitate the simultaneous real-time acquisition of both color and range images of a scene [BH04, LHWL07]. The resulting huge amount of object information provided by these devices requires processing frameworks that are able to handle this data efficiently for interactive reconstruction. Furthermore, the final reconstruction of the captured object should be of high visual and geometric quality.

In this paper, a framework is presented that addresses the aforementioned issues and allows for simultaneous incremental reconstruction of both geometric and visual properties of a given object. As input it expects color and range images of an object as well as corresponding calibration data. For reconstructing the geometry part of the object, an existing surface reconstruction method has been investigated and modified to be applicable for incremental reconstruction. For optimal reconstruction of the model's visual appearance, a GPU-accelerated online implementation of surface light fields is utilized that is parametrized on the currently reconstructed geometry. Both parts of the framework have been optimized for online reconstruction, thus allowing the interactive visualization of the current status of the reconstructed object.

This paper is organized as follows: previous work related to our paper is summarized in Section 2. Afterwards, Section 3 gives a brief overview of our proposed reconstruction framework. In Section 4, the used surface reconstruction algorithm and its modifications are described. Section 5 then describes the applied surface light field implementation. The results and conclusion of our work are shown in Sections 6 and 7, respectively.

2. RELATED WORK

The acquisition and reconstruction of 3-D models from real-life objects is a heavily investigated research area. For example, different approaches are known that reconstruct surfaces from unordered point clouds or range images. These may be distinguished as *parametric surface methods* [TL94, CL96] that make use of the known topology of the input data, *implicit methods* [HDD$^+$92, OBA$^+$03] which apply implicit distance functions that represent the surfaces, or *Delaunay-based methods* [EM94, DG03]. However, most of these approaches handle surface reconstruction as an offline process only. In the area of incremental surface reconstruction, only few publications are known. Curless and Levoy [CL96] incrementally build a distance volume from range images representing the final surface, however, the successive surface extraction step still is done offline. Turk and Levoy [TL94] directly reconstruct the surface using the provided triangulated range images, which allows for online visualization of the currently reconstructed geometry. Two different frameworks are proposed in [RHHL02] and [BH04] that facilitate the online reconstruction of real-life objects at interactive rates. However, texturing of the reconstructed object is only done in a separate offline step. None of the described approaches supports the incremental reconstruction of visual information of the object.

For reconstructing the visual appearance of real-life objects, *light field* and *lumigraph* models are the preferred choice: by utilizing the provided image information, they are able to synthesize new views of a given object, while also preserving highly view-dependent visual effects (e. g. mirroring, transparency or sub-surface scattering). Different approaches exist [LH96, GGSC96, SVSG01, BBM$^+$01, CBCG02] that differ in the actual light field parametrization used, and whether geometric information of the object is applied to support the synthesizing process. In the latter case, reconstruction results usually are improved significantly, provided that the geometry is of sufficient quality. While some of these approaches are suitable for incremental reconstruction [BBM$^+$01, CHLG05], this only restricts to the visual part. The underlying geometry still has to be provided by using other methods.

3. THE RECONSTRUCTION FRAMEWORK

Our proposed framework aims to provide a complete pipeline for simultaneous incremental reconstruction of geometry and appearance of a given object. Therefore, it makes use of established algorithms which have been extended and adjusted to our needs.

As a basis for the surface reconstruction part, the ridge-based approach by Süßmuth et al. [SG07] has been chosen because of its initial properties. The method aims at providing a very robust reconstruction algorithm that delivers high quality results even in the presence of strong noise, which is a vital property for reconstructing range images coming from scanning devices. Additional post-processing steps have been implemented that complement the main algorithm, like mesh decimation and topology cleaning.

For the appearance reconstruction part, the surface light field model (SLF) [CBCG02] has been chosen as the basic method. Among all light field approaches, it provides the most realistic visual results. Furthermore, efforts have been made to realize an online version for reconstructing surface light fields [CHLG05], which is a major advantage of this approach for our purposes.

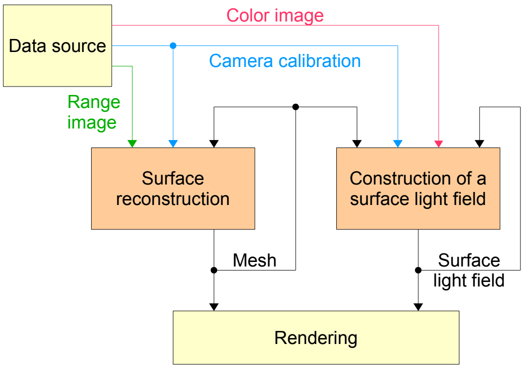


Figure 1: The complete pipeline of our reconstruction framework.

The whole reconstruction process is depicted in Figure 1. Here, a data source provides image information about the object, consisting of color and range images as well as camera calibration data. The latter provides information about the camera's extrinsic and intrinsic parameters. The current range image and calibration data is fed into the surface reconstruction part of our framework, where it is processed for extending the existing reconstructed geometry of the object. The extended geometry is reused in the successive surface reconstruction steps, and is also provided to the light field reconstruction part. Here, together with the current color image and camera calibration data, the existing light field is extended using the new provided information. The progress of the object reconstruction process is visualized at any time using the rendering part of the framework.

4. INCREMENTAL SURFACE RECONSTRUCTION

In this section, the implemented surface reconstruction approach is described briefly, and its modifications for incremental mode of operation are explained. For more detailed information about the original reconstruction process, we would like to address the interested reader to the original paper.

4.1. Overview of the Ridge-based Approach

The ridge-based surface reconstruction approach by Süßmuth et al. assumes that a sample point p does not necessarily interpolate the surface but is rather the center of a Gaussian probability distribution $G_p(x)$ describing its location. By summing up all probability distributions defined by the points from a given point cloud, one receives a global probability distribution $f(x)$ that describes the potential location of the surface.

The algorithm can be subdivided into two main steps. In the first step, the Gaussian kernels centered at the sampled points are summed up and analyzed at the nodes of a regular, discrete grid. The analysis encompasses the computation of the density function's gradient $\nabla f(x)$ and the direction of maximum curvature $K_{max}(f)(x)$, where the latter is calculated by eigenanalysis of the Hessian matrix $H(f)(x)$. In the second step, the scalar product $\langle \nabla f(x), K_{max}(f)(x) \rangle$ is used in a modified marching-cubes [LC87] algorithm in order to identify ridges which are then interpreted as the surface.

Figure 2 illustrates the different stages of the algorithm for the 2-D case. As can be seen in Figure 2d, many "spurious" ridges were extracted beside the actual sine curve. In order to avoid extraction of those ridges, Süßmuth et al. assume that the node with highest density always lies on the main ridge corresponding to the actual surface. Hence they only accept the surface which evolves from this node.

Finally, to lower computational effort, the Gaussion kernel is cut off in regions where its contribution to the density function, gradient and curvature is becoming insignificant. If the value of the Gaussian kernel drops below an user-specified threshold at a grid node, its calculation and analysis is omitted at the respective node. Thus, each grid node receives contributions only from those points which lie within the cut-off radius around that node.

4.2. The Incremental Ridge-based Approach

The original approach was designed to work on a complete point cloud, however, the accumulation of the density function as well as its analysis can be done easily in an incremental way. The Gaussian kernel $G_p(x)$ for point p and its gradient $\nabla G_p(x)$ are evaluated at each grid node n, if the respective point lies within the node's cut-off radius. The density function $f(x)$ and its gradient $\nabla f(x)$ are then evaluated at each node n, by summing up the contributions of all points P_n within the respective node's cut-off radius. For a node n, the evaluation of $f(x)$ would be done like this: $f_n = \sum_{p \in P_n} G_p(\|p - n\|)$. The gradient ∇f_n and the Hessian matrix $H(f_n)$ are evaluated accordingly.

This way, new point clouds can be integrated into the existing sampling of function $f(x)$ and its analysis by simply resuming the summation process. Once all new points have contributed to the density function $f(x)$, its gradient and its Hessian matrix, the eigenanalysis of $H(f)(x)$ is performed and the scalar product $\langle \nabla f(x), K_{max}(f)(x) \rangle$ is evaluated at every updated grid node n.

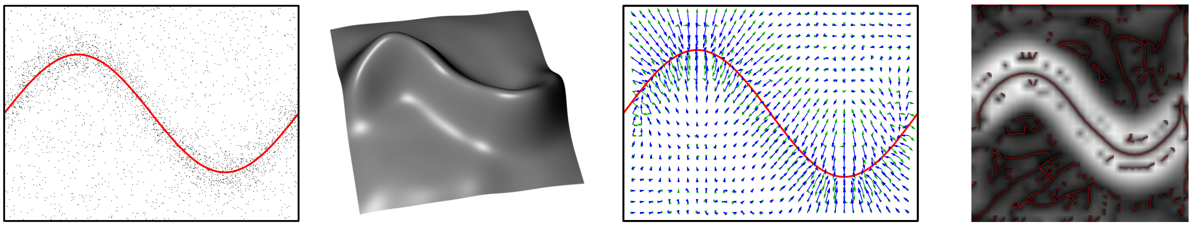
After the discrete grid has been updated, new surface parts are extracted using a modified marching-cubes algorithm. Here, tracing the surface from the node of highest density is not applicable any more: Using an incremental and interactive approach, we need to be able to reconstruct the surface in different parts of the captured object simultaneously, as multiple unconnected parts may occur which all belong to the actual surface.

Therefore, the extraction algorithm considers multiple nodes as starting points for tracing the surface and does so only for nodes where the density has reached a certain user-specified threshold. To further prevent spurious ridges from being extracted, only surface parts with a user-specified minimum number of reconstructed triangles will be accepted for addition to the object's mesh.

Once a new surface part is added to the mesh, all grid cells that contributed to this part are locked, thus preventing the marching-cubes algorithm from considering these cells in upcoming iterations of the reconstruction process. Furthermore, all nodes adjacent to locked cells are also locked, meaning that these nodes' density, gradient and maximum curvature will not be updated any more. This way we make sure that updates will not displace the zero-crossing, which otherwise leads to discontinuities when neighboring cells also extract their part of the surface. Figure 3 illustrates this process.

4.3. Mesh Improvement Techniques

After new mesh parts have been added to the reconstruction in each iteration, we apply different mesh



(a)

(b)

(c)

(d)

Figure 2: (a) A noisy sampling of a sine curve. (b) The density function $f(x)$. (c) The gradient (blue) and maximum curvature direction (green) of the density function. (d) The absolute scalar product as gray value image and the curves (both wanted and spurious) extracted by the marching cubes algorithm.

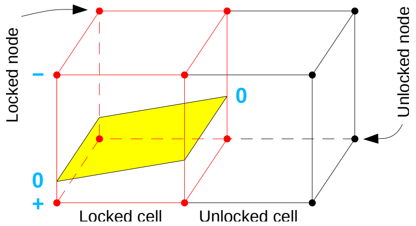


Figure 3: Example for surface extraction inside the discrete grid. Once a surface (yellow) has been extracted within one of the cells, the cell itself is locked as well as all adjacent nodes (red).

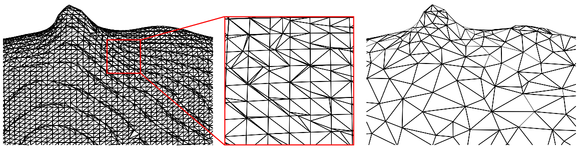


Figure 4: Left: A mesh produced by the marching-cubes algorithm. Middle: An enlarged section from the mesh. Right: The same mesh after improvement.

improvement techniques in order to prepare the mesh for use with surface light field construction. This is necessary for the ridge-based approach, as the used marching-cubes algorithm generates meshes that usually consist of very small triangles, where many of them are showing poor aspect ratios. Furthermore, it is advisable to reduce the number of reconstructed triangles in order to improve reconstruction performance in general. To handle the mentioned issues, the following tasks are performed:

- Small holes are filled using a simplified version of Liepa's method [Lie03], which only triangulates holes, omitting the steps of simplification and fairing.
- Mesh simplification is performed by edge contraction using quadric error metrics as presented by Garland and Heckbert [GH97]. Non-edge contractions are not performed, thus keeping separated parts of the mesh unconnected.
- A simplified version of edge relaxation via mesh edge flipping as proposed by Dyer et al. [DZM07] is applied. The steps of remeshing and mesh decimation are omitted.

Figure 4 shows an example mesh before and after application of the described improvement methods. In order to avoid discontinuities, these steps are not performed at the boundary of the current surface reconstruction. Therefore, the boundary keeps its typical cube-like shape caused by the surface extraction algorithm until new mesh information has been reconstructed. Figure 5 shows the typical topology of a reconstructed mesh at its interior and its boundary, respectively.

5. ONLINE CONSTRUCTION OF SURFACE LIGHT FIELDS

A surface light field approximates the 4-D function $f(r,s,\theta,\phi)$ which describes the outgoing radiance for every point (r,s) on an object's surface into every viewing direction (θ,ϕ). For efficient storage and hardware rendering, Chen et al. [CBCG02] proposed a decomposition of the 4-D light field function into a sum of products of lower-dimensional functions:

$$f(r,s,\theta,\phi) \approx \sum_{k=1}^{K} g_k(r,s)h_k(\theta,\phi)$$

Significant data compression is achieved by limiting the sum to K terms, where K usually lies between 3 and 6 when using an offline principal component analysis (PCA) as decomposition algorithm. Both functions g_k and h_k, which are exclusively parametrized on the surface or the viewing direction respectively, can be stored on graphics hardware as texture maps. Using the vertex-centered light field partitioning scheme introduced by Chen et al., new views of the object can then be rendered at interactive frame rates using programmable graphics hardware.

However, the PCA algorithm used to compress and decompose the 4-D light field function is an expensive offline method that is not suited for online construction of a light field dataset, as it requires a complete recomputation of the decomposition when adding new image information. To this end, Coombe et al. [CHLG05] replaced the offline PCA with an incremental singular value decomposition (Online-SVD), thus making the online construction of surface light fields feasible.

For our framework, we implemented an optimized version of the online construction algorithm from [CHLG05]. In our implementation, the tasks of image resampling and visibility testing were redesigned as GPU algorithms to further speed up the online construction process. Furthermore, the visibility test of the original approach has been modified. Since a complete geometry mesh of the surface is not available for the visibility tests, the current range image is used instead as a representation for the object's geometry.

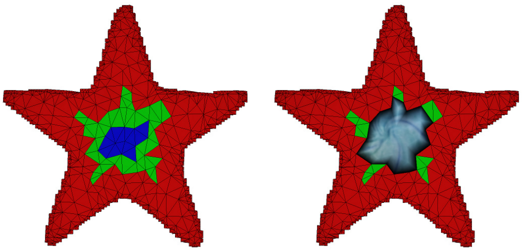


Figure 5: A partial reconstruction of the front of the *star* dataset from the OpenLF toolset [Too09]. The colors indicate the different areas. The right image shows the parts of the mesh for which a light field was already constructed.

The two tasks of surface reconstruction and construction of the surface light field can be executed in an alternating way, such that new light field information may immediately be added to recently reconstructed mesh parts (Figure 1). However, the interaction of both tasks is complicated by the fact that mesh reconstruction and mesh improvement must never change those parts of the surface for which light field information already has been constructed. Therefore, the object's mesh is organized in three areas which are depicted in Figure 5a. Modifications of the mesh, including deletion of faces or vertices, are only possible at the red "border" faces which are near to the mesh boundary where new faces will connect in future iterations. The middle area of green faces consists of so-called "fixed" faces. Neither their vertices nor their topology may change anymore, but since they are adjacent to border faces their vertices' normals may be subject to change. Finally, the blue faces at the center are the "normal-fixed" faces whose vertex normals are fixed as well. In our implementation, light field information is only stored at the triangle fans that contain at least one blue face, as shown in Figure 5b.

6. RESULTS

The proposed framework has been extensively tested against the freely available datasets of the OpenLF toolset [Too09], which contain images, geometry and calibration data from acquired real-life objects. The available information is highly accurate, and therefore well suited as ground-truth data to test our framework's performance and accuracy, without introducing additional errors from various pre-processing steps.

However, the images provided by the OpenLF datasets are photographs taken at discrete camera positions, and the provided geometry is given as a complete mesh instead of several range images. To simulate data coming from a moving hand-held acquisition device that is suited for our framework, a modified light field viewer has been implemented that constructs surface light fields from OpenLF datasets that are freely accessible by the user. Color and range images are generated by the viewer by sampling the currently rendered image and depth information, and are provided via network to our framework for further processing.

Using this setup, the simultaneous reconstruction of geometry and appearance of the OpenLF datasets has been performed. An example for the visual feedback of the reconstruction progress can be seen in Figure 8. A visual comparison between the "original" model and the one reconstructed by our framework is shown in Figure 6. Here, it can be seen that our framework is able to successfully reconstruct the provided models with high geometric and visual accuracy. However, the SLF reconstruction tends to "smooth" visual highlights, which is also a typical result of the original online-SLF approach.

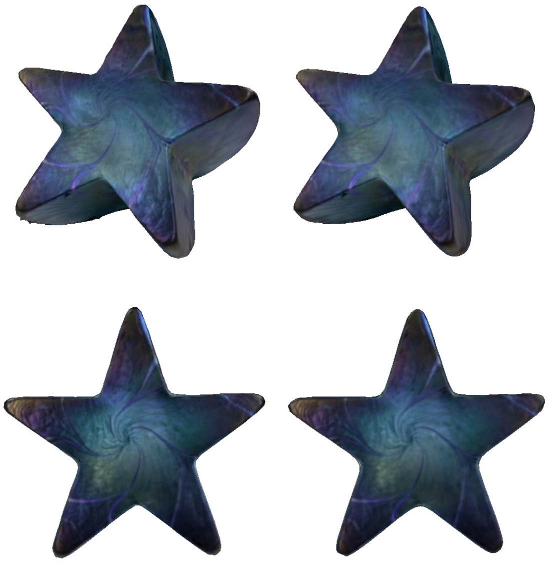


Figure 6: Reconstruction of the *star* dataset: the original model as rendered by our SLF viewer is shown in (a), the model reconstructed by our framework is shown in (b).

Due to the nature of the applied surface reconstruction algorithm, different artifacts may occur during reconstruction. One example is the erroneous extraction of spurious ridges, as shown in Figure 7a. These artifacts cannot be avoided in all cases, however, if identified, they can be successfully discarded by removing all but the largest connected submesh (Figure 7b). Also, the reconstruction algorithm may not be able to close all mesh holes. These reconstruction artifacts are mainly caused by noisy input data and imperfect choice of user-specified reconstruction parameters. As the optimal parameters for a given object cannot be determined analytically, most of them have to be found by trial-and-error.

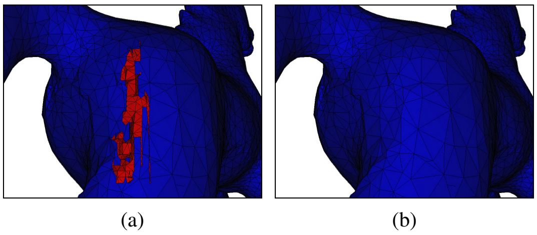


Figure 7: (a) Extraction of spurious ridges at the *horse* dataset. (b) The same mesh after final cleanup.

Tables 1 and 2 show average timings for the integration of new data into an existing reconstruction. The values for “slf update” show the timings of the complete surface light field update process, while the remaining values show timings of different parts of the surface reconstruction and the different improvement algorithms. As can be seen in these tables, the reconstruction speed highly depends on the resolution of the depth images. When using higher resolutions, the most time-consuming part of the reconstruction is the update of the density function, which is also due to the resolution of the function’s grid. Using lower resolutions, the grid’s density may be decreased accordingly, and so the impact of this step vanishes. It can also be noticed that, due to our GPU optimizations, the surface light field update process only takes a fraction of the whole update time. Eventually, it can be seen that the incremental reconstruction process can be performed at interactive frame rates when processing the mentioned datasets.

Using our framework, we also started first experiments with the acquisition of own datasets. As input device, we utilized the 2-D/3-D time-of-flight camera described in [LHWL07], which has been calibrated using the OpenCV library [Lib09]. The provided input data was pre-processed in a simple way in order to reduce noise of the depth information before it was delivered to our framework. Using this setup, we were able to reconstruct information of a silicon liver model, which is shown in Figure 9. Despite the pre-processing, the incoming depth information still exhibited a significant amount of noise. Nevertheless, our modified reconstruction method successfully recreated all parts of the model that were seen by the camera – it can be recognized that holes or gaps in the reconstruction mainly appear in those areas where no information was provided to the framework. Note that the visual reconstruction of the model looks less complete because of our introduced update policy for the surface light field.

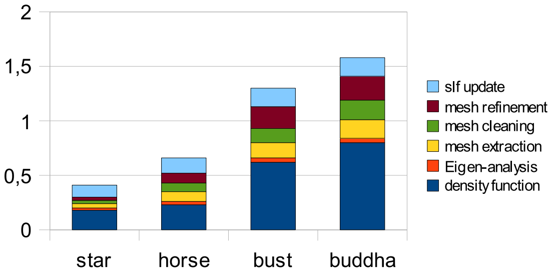


Table 1: Performance of the reconstruction framework for depth map size 176×144 **(timings in s)**

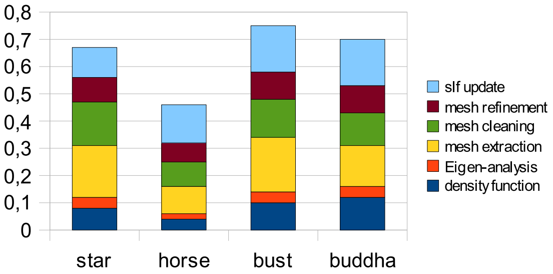


Table 2: Performance of the reconstruction framework for depth map size 64×48 **(timings in s)**

7. CONCLUSION AND FUTURE WORK

We have presented a novel framework for the incremental reconstruction of digital models from real-life objects. The framework features an enhanced incremental approach for interactive surface reconstruction, as well as an online surface light field technique for high-quality reconstruction of both geometry and appearance information.

The surface reconstruction approach has shown good results for data with a moderate level of noise. However, as the noise intensity increases, there is also a gain in the probability for reconstruction artifacts like spurious ridges. Therefore, efficient pre-processing algorithms that smooth data and remove outliers will be necessary to significantly improve the overall quality of reconstructions from “real-life” acquisition devices.

Due to our update policy of the surface light field, the surface reconstruction part of our framework finishes long before the construction of the light field reaches sufficient quality. While the progress of the surface reconstruction is obvious to the user, the quality of the light field is hard to determine and calls out for special

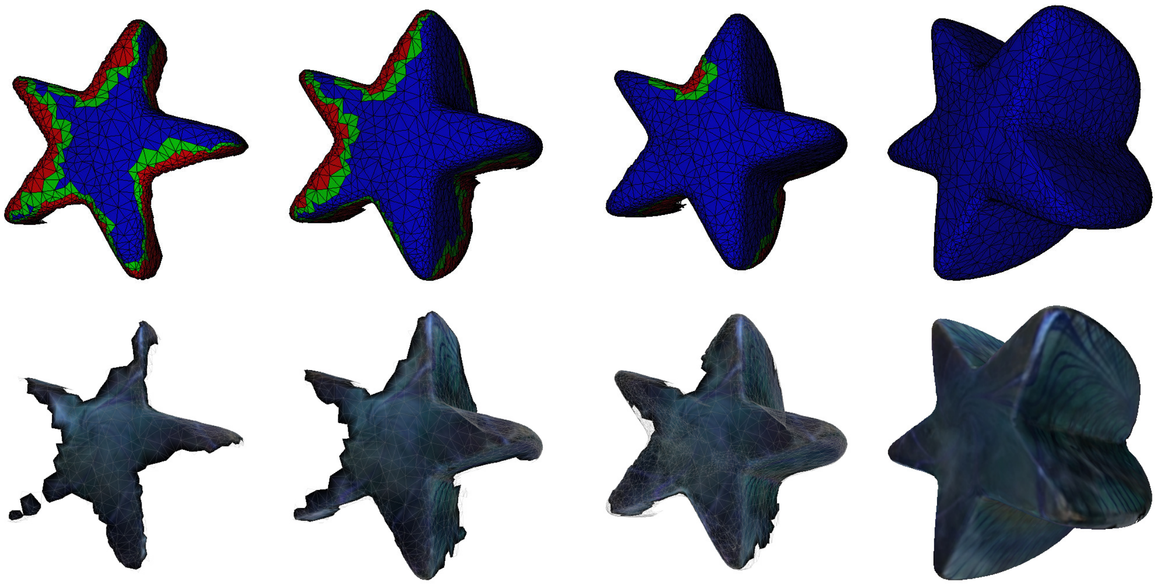


Figure 8: Different iterations of a reconstruction of the *star* dataset from the OpenLF library.

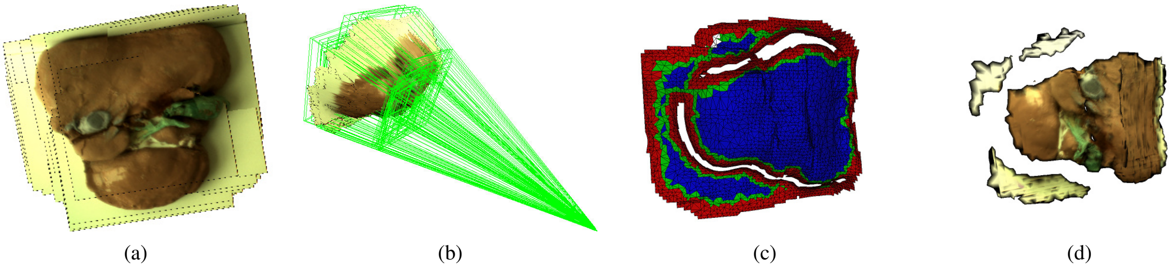


Figure 9: Partial reconstruction of a liver model. Figures (a) and (b) show the source images and calibration parameters, while the reconstructed geometry and appearance is shown in figures (c) and (d), respectively.

visualization. Coombe et al. propose a quality heuristic which directs the user to areas where more data is needed. However, this quality measure does not indicate the viewing direction for which the approximation still lacks information. Therefore, we are interested in investigating different approaches for feedback visualization to further improve acquisition time and final model quality.

Our first experiments with the acquisition and processing of “real” data using a 2-D/3-D time-of-flight camera have been quite promising. For improved reconstruction results, we plan to integrate better pose information of the camera in our acquisition process.

ACKNOWLEDGEMENTS

We would like to thank Jochen Süßmuth for making his implementation available to us.

REFERENCES

[BBM$^+$01] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 425–432, New York, NY, USA, 2001. ACM.

[BH04] T. Bodenmueller and G. Hirzinger. Online surface reconstruction from unorganized 3d-points for the dlr hand-guided scanner system. In *Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium (3DPVT)*, pages 285–292, Washington, DC, USA, 2004. IEEE Computer Society.

[CBCG02] Wei-Chao Chen, Jean-Yves Bouguet, Michael H. Chu, and Radek Grzeszczuk. Light field mapping: Efficient representation and hardware rendering of surface light fields. In *ACM Transactions on Graphics*, pages 447–456, 2002.

[CHLG05] Greg Coombe, Chad Hantak, Anselmo Lastra, and Radek Grzeszczuk. Online construction of surface light fields. In

Proc. Eurographics Symposium on Rendering, 2005.

[CL96] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 303–312, New York, NY, USA, 1996. ACM.

[DG03] Tamal K. Dey and Samrat Goswami. Tight cocone: a water-tight surface reconstructor. In *Proceedings of the eighth ACM symposium on Solid modeling and applications (SMA)*, pages 127–134, New York, NY, USA, 2003. ACM.

[DZM07] Ramsay Dyer, Hao Zhang, and Torsten Möller. Delaunay mesh construction. In *Proceedings of the fifth Eurographics symposium on Geometry processing (SGP)*, pages 273–282, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[EM94] Herbert Edelsbrunner and Ernst P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994.

[GGSC96] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 43–54, New York, NY, USA, 1996. ACM.

[GH97] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 209–216, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[HDD+92] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 71–78, New York, NY, USA, 1992. ACM.

[LC87] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Proceedings of the 14th annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, 21(4):163–169, 1987.

[LH96] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 31–42, New York, NY, USA, 1996. ACM.

[LHWL07] O. Lottner, K. Hartmann, W. Weihs, and O Loffeld. Image registration and calibration aspects for a new 2d / 3d camera. In *EOS Conference on Frontiers in Electronic Imaging*, pages 80–81. European Optical Society, 2007.

[Lib09] Open Computer Vision Library. http://sourceforge.net/projects/opencvlibrary/, 2009.

[Lie03] Peter Liepa. Filling holes in meshes. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing (SGP)*, pages 200–205, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[OBA+03] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. *ACM Transactions on Graphics*, 22:463–470, 2003.

[RHHL02] Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. Real-time 3d model acquisition. *ACM Transactions on Graphics*, 21(3):438–446, 2002.

[SG07] Jochen Süßmuth and Günther Greiner. Ridge based curve and surface reconstruction. In *Proceedings of the fifth Eurographics symposium on Geometry processing (SGP)*, pages 243–251, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[SVSG01] Hartmut Schirmacher, Christian Vogelgsang, Hans-Peter Seidel, and Günther Greiner. Efficient free form light field rendering. In *Proceedings of the Vision, Modeling and Visualization Conference (VMV)*, pages 249–256. Aka GmbH, 2001.

[TL94] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 311–318, New York, NY, USA, 1994. ACM.

[Too09] OpenLF Light Field Mapping Toolset. http://sourceforge.net/projects/openlf/, 2009.