High-Definition Texture Reconstruction for 3D Image-based Modeling

Hoang Minh Nguyen
The University of Auckland,
New Zealand
hngu039@aucklanduni.ac.nz

Burkhard Wünsche
The University of Auckland,
New Zealand
burkhard@cs.auckland.ac.nz

Patrice Delmas
The University of Auckland,
New Zealand
p.delmas@cs.auckland.ac.nz

Christof Lutteroth
The University of Auckland,
New Zealand
lutteroth@cs.auckland.ac.nz

Wannes van der Mark
The University of Auckland,
New Zealand
w.vandermark@auckland.ac.nz

Eugene Zhang
Oregon State University,
Oregon, United States
zhange@eecs.oregonstate.edu

ABSTRACT

Image-based modeling is becoming increasingly popular as a means to create realistic 3D digital models of real-world objects. Applications range from games and e-commerce to virtual worlds and 3D printing. Most research in computer vision has concentrated on the precise reconstruction of geometry. However, in order to improve realism and enable use in professional production pipelines digital models need a high-resolution texture map. In this paper we present a novel system for creating detailed texture maps from a set of input images and estimated 3D geometry. The solution uses a mesh segmentation and charting approach in order to create a low-distortion mesh parameterization suitable for objects of arbitrary genus. Texture maps for each mesh segment are created by back-projecting the best-fitting input images onto each surface segment, and smoothly fusing them together using graph-cut techniques. We investigate the effect of different input parameters, and present results obtained for reconstructing a variety of different 3D objects from input images acquired using an unconstrained and uncalibrated camera.

Keywords

Texture reconstruction, Image-based modeling, mesh parameterization, texture mapping

1 INTRODUCTION

Digital 3D models are used in a large number of applications ranging from entertainment (games, movies) to engineering and architecture (design), e-commerce (advertisement) and education (simulation and training). 3D model creation can be made more effective, more affordable, and more accessible to inexperienced users, by using image-based reconstruction methods, which aim to create a high-quality digital model from a set of input photographs [HVC08, REH06].

Most published research has concentrated on the problem of reconstructing 3D geometry from a set of input images, and estimating camera parameters for methods assuming uncalibrated and unconstrained image acquisition. The problem of texture reconstruction for multi-view stereo has also been investigated, however, many authors make assumptions, such as known camera parameters, which can not be guaranteed in practice.

In this paper we present a complete system for texture reconstruction for image-based modeling. The system is fully automatic and input images can be acquired with an unconstrained and uncalibrated camera. The resulting models contain a high-definition texture map and can be integrated into professional production pipelines. Our algorithm automatically estimates the intrinsic and extrinsic parameters of the input cameras using *Structure-from-Motion* and *Bundle Adjustment* techniques. The 3D model is then automatically parameterized using a segmentation and charting technique, which is suitable for surfaces of arbitrary genus [ZMT05]. A texture map is then created by back-projecting the best fitting input images onto each surface segment, and smoothly fusing them together over the corresponding chart by using graph-cut techniques.

The remainder of this paper is organized as follows. Section 2 reviews existing approaches for texture reconstruction in multi-view stereo. Section 3 summarizes our image-based modeling technology, which we use to create 3D geometry and estimate camera parameters. Section 4 describes our texture reconstruction process in detail. Section 5 evaluates our solution and discusses

 ISBN 978-80-86943-74-9

the effect of various parameters and the algorithm's advantages and shortcomings. We conclude this paper and give an outlook on future research in section 6.

2 LITERATURE REVIEW

Image-based texture reconstruction for 3D models requires in general two steps: a surface parameterization of the reconstructed 3D object, and computation of the object's surface texture from a set of input images of the object.

The surface parameterization creates a mapping of a 2D domain (parameter space) to the surface mesh of the reconstructed 3D object. Texture mapping can then be accomplished by creating a 2D texture image over the parameter space. An explicit surface parameterization can be avoided by determining the input image regions best representing the object's surface, blending them together, and storing them in a texture atlas indexed by the mesh vertices [XLL+10]. However, since there is no global parameterization, postprocessing algorithms, such as polygon reduction, can result in unwanted artifacts.

Surface parameterization methods can be classified according to their complexity, whether the resulting mapping is bijective, whether they have a predetermined boundary for the parameter space, and to what extend distortion is minimized [SPR06]. For objects with a non-zero genus or complex geometry the surface must be cut into multiple parts and parameterized individually in order to minimize distortions. The resulting charts can be combined into one single texture atlas using a packaging algorithm.

Most recent image-based texture reconstruction algorithm seem to use a charting approach. Goldluecke and Cremers [GC09] create a planar texture space via an automatically created conformal atlas [LWC06]. The planar texture space is then used to solve a partial differential equation, originally defined over the object's surface, in order to find the surface texture representing the input images best.

Computation of a surface texture from input images is difficult since several images mapping to the same surface region can result in conflicting color information due to geometric errors (camera parameters), limited image resolution, and varying environmental parameters (lighting) during image acquisition. Four classes of solutions are described in the literature:

1. Blend input image information per texel using suitable weights for different source images [BMR01, LH01].
2. Compute texture patches and fuse them seamlessly together by optimizing seam locations [LI07, XLL+10] or warping texture patches [EdDM+08].
3. Compute texture patches and blend them seamlessly together. Chen et al. use multi-band blending in order to minimize seam discontinuities [CZCW12].
4. Use a local optimization step in order to fully utilize the information given by multiple images of the same object region. Goldluecke and Cremers present a technique for computing high-resolution texture maps from lower-resolution photographs [GC09]. The method requires accurate geometry and camera calibration.

Additional optimization steps are possible to take into account texture differences in input images, e.g., due to illumination changes, shadows, and camera parameters such as dynamic range adjustment. Xu et al. [XLL+10] use radiometric correction to adjust color difference between patches. Valkenburg and Alwesh reduce seams resulting from image illumination variations by applying a global optimization to all vertex colors of a 3D mesh [VA12]. Chen et al. remove highlight effects by determining all input images mapping to a surface area [CZCW12]. Image regions which vary too much from the median color of the surface area are removed. Missing or deleted image regions (e.g., highlights) can be filled using Poisson image editing [CZCW12, CAH+13].

3 3D GEOMETRY RECONSTRUCTION

In this section we summarize our image-based modeling algorithm for geometry reconstruction. We concentrate on the algorithm steps effecting texture reconstruction, i.e., camera parameter estimation and surface representation. More details of the algorithm are described in [NWDL13, NWDL12b].

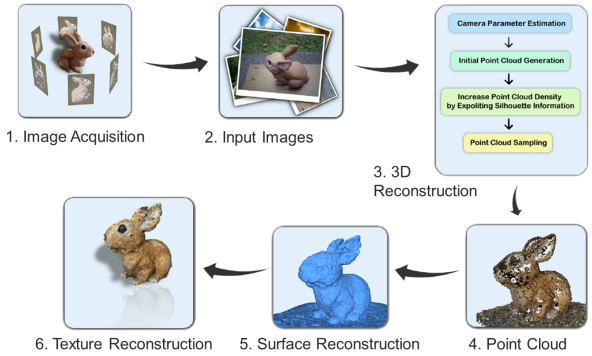

Figure 1: Overview of our algorithm for reconstructing 3D models from a set of unconstrained and uncalibrated images.

An overview of our image-based modeling technology is given in Figure 1. The algorithm uses a coarse-to-fine strategy where a rough model is first reconstructed and then sequentially refined through a series of steps.

 ISBN 978-80-86943-74-9

The first step of the geometry reconstruction consists of estimating the camera parameters for each view. This is accomplished by detecting and extracting distinctive features using a SIFT feature detector [Low99, Low04]. We then isolate all matching images, selecting those that contain a common subset of 3D points [HQZH08].

Given a set of matching images, a scene geometry (point cloud) and camera pose can be estimated simultaneously by using a *Structure from Motion* algorithm and subsequently refining the solution using *Bundle Adjustment*. The last step is critical for the accuracy of the reconstruction, as concentration of pairwise homographies would accumulate errors and disregard constrains between images. The method minimizes the reprojection error, which is defined by the distance between the projections of each point and its observations.

Due to the sparseness of the point cloud representing the scene geometry, artifacts can arise during the surface and texture reconstruction processes. We overcome this problem by integrating a shape-from-silhouette approach. Silhouette data is obtained by using the rough depth estimation from the previous step for a foreground segmentation and applying the Marching Squares algorithm [Lor95]. The complexity of each silhouette line is reduced using the Douglas-Peucker algorithm [VW90]. The 3D positions of silhouette points are estimated by forming cone lines from silhouette contour points and the camera's estimated optical center, projecting the lines onto the other silhouettes, computing the intersection points, and lifting them to 3D [MBR+00].

Adding silhouette points and using them in the bundle adjustment step results in a better camera parameter estimation and smoother surface reconstruction.

Finally the object's surface is reconstructed. We tested the α-shape algorithm, the power crust algorithm, and the ball pivot algorithm. In the end we decided to use the Poisson surface reconstruction algorithm [KBH06]. The technique gives a smoother reconstruction than other tested techniques, is more stable towards noise, and always creates a watertight surface.

A perceived weakness of the algorithm is that it requires oriented normals at the input points. However, we can obtain them from the image and silhouette information. Furthermore, it has been shown hat the approach is quite resilient to inaccuracies in the directions of the normals [Kaz05].

A surface texture is created by projecting each vertex of the mesh onto all input images containing the point (i.e., the surface point is visible from the images' estimate camera location). The mesh vertex color is the weighted average of the corresponding image pixels. The resulting triangle mesh with vertex colors is rendered using Gouraud shading. An example is shown in Figure 2.

Color interpolation suffers from two major shortcomings: (1) detailed input image textures appear blurred (see bottom row of Figure 2), and (2) texture resolution is lost if a mesh reduction method is applied.

Figure 2: Photograph of a rooster statue (left) and the reconstructed model using vertex colors and Gouraud shading (right). The images at the bottom show an enlargement of the neck region of the object.

4 TEXTURE RECONSTRUCTION

We create a high quality texture map for our 3D model in two steps: The 3D mesh model is first parameterized yielding a one-to-one triangle mapping from the 3D model to a 2D planar surface. Input images are then projected onto the surface and suitable texture regions are identified, cut, and fused together to form a 2D texture atlas.

4.1 Surface Parameterization

The objective is to segment the resulting meshes into patches and unwrap them onto a 2D planar surface. We evaluated different surface parameterization techniques, but found that existing libraries, such as Blender, either create a very disjoint map of triangle patches, or create a single parameter patch with large distortions. We hence use a *Feature-based Surface Parameterization*, which consists of three stages [ZMT05]: Genus reduction, feature identification, and patch creation.

Genus reduction In order to identify non-zero genus surfaces, a surface-based *Reeb* graph [Ree46] induced

 ISBN 978-80-86943-74-9

by the *average geodesic distance* [HSKK01] is constructed. The leaf nodes of this graph reveal the tips of the protrusions of the meshes, while loops in the graph signify the existence of handles. The principle behind genus reduction is to identify loops that do not separate the surface into two disjoint connected components and cut the surface open along the cycle, which reduces the combined genus of the surface segments by one. This process is repeated until there are no more handles.

Feature identification From the *Reeb* graph the tips of protrusions are identified and the features are separated from the rest of the surface by constructing a closed curve γ as follows: We separate the region R that corresponds to the tip of the protrusion by first computing the function $f_{\mathbf{p}}(\mathbf{q}) = g(\mathbf{p},\mathbf{q})$, where $f_{\mathbf{p}}(\mathbf{q})$ is the *geodesic distance function* [HSKK01] with respect to $\mathbf{p}$. The value of $f_{\mathbf{p}}$ is normalized to fit in the interval $[0,1]$. Regions which are bounded by a given *isovalue* are examined. Specifically, the interval $[0,1]$ is partitioned into k equal sections. The surface is then divided into levelset bands by performing region-growing from the tip of the protrusion $\mathbf{p}$ based on the values of $f_{\mathbf{p}}$ in these intervals [ZMT05].

Variation in the *area* of this sequence of bands tends to be small along a protrusion slope, and large where the feature connects to the remaining section of the surface. The separating region R can be extracted by examining these areas, which are considered as a continuous function $\mathbf{A}(x)$. To remove any small undulations, $\mathbf{A}(x)$ is passed through a *Gaussian filter* function N times.

Three parameters (*isovalue*, k, and N) influence the effectiveness and efficiency of the region separation process. The larger the *isovalue* is, the further the region-growing process continues. This leads to fewer surface patches being generated. Higher k values result in more samples being used to discretize $\mathbf{A}(x)$, increasing the probability of small noise being considered as potential candidate places for the separating region. Large N values tend to cause the location of the separating region to shift or it being lost, while too small values often result in false separations.

Once the separating region R has been identified, a closed curve γ separating the surface into segments is constructed as follows: A collection of edges in the surface separating the feature from the rest of the surface (the skeleton) of R is found. During this process dangling edges are rejected. A separating cycle ρ from this skeleton is then extracted. Finally, a shorter and smoother separating cycle γ is constructed based on ρ.

Patch creation Patches are created by unwrapping them using a *discrete conformal mapping* [EDD+95]. The method creates first texture coordinates of the boundary vertices, and then determines texture coordinates of the interior vertices through solving a closed form system. The main problem with this mapping technique is that regions can be stretched or compressed during the process leading to areas of the meshes not being preserved. This in turn results in uneven sampling rates across the surface.

Interior vertices' texture coordinates are optimized to reduce the geometric distortion by first computing an initial harmonic parameterization [Flo97]. A *square virtual boundary* enclosing the patch is constructed. The exact coordinates of the boundary are not important as long as they do not coincide with those of the patch boundary. We then perform triangulation of the regions between the virtual boundary and the original boundary using Scaffold triangles. The patch optimization technique proposed by Sandle *et al.* [SGSH02] is then applied to the enlarged patch.

4.2 Texture Map Generation

At this stage, we have successfully generated a parameterization of the 3D model. The next task is to construct a complete texture map using the computed parameterization. This is accomplished in three steps:

1. Identify images and regions of input images to be mapped onto each patch of the parameterization.
2. Cut these patches and paste them over the parameterized surface.
3. Merge overlapping regions using a *graph cut* technique [KSE+03a, CFW+12].

Texture region identification: For each patch of the surface parameterization we need to identify the image regions mapping onto it. We project all triangles of a patch onto all input images where it is visible, i.e.: (1) the triangle normal forms an angle of less than 90° with the vector to the estimated camera position; (2) the triangle is not occluded by other surface regions. The resulting image regions and the one-to-one correspondence between projected triangles and original triangles of the patch is saved for the next stage of the algorithm.

Texture map computation: At this stage for each patch we have a set of texture regions. The goal is to process these texture regions to produce a new texture that will cover the patch. We perform the mapping of a texture region from an input image to a patch for each triangle separately. Given two arbitrary triangles $\triangle_1$ and $\triangle_2$, an affine transformation that transforms triangle $\triangle_1(P,Q,R)$ to $\triangle_2(P^\circ,Q^\circ,R^\circ)$ is defined as follows: Let Φ_1 be the affine transformation that maps the unit triangle to $\triangle_1$, and Φ_2 be the affine transformation that maps the unit triangle to $\triangle_2$. The affine equivalence of these two triangles is $\Phi_2 \circ \Phi_1^{-1}$.

The procedure is repeated for each texture region yielding a set of overlapping textures covering the face of the

 ISBN 978-80-86943-74-9

processed patch. We use a greedy technique to assemble these textures. We start with the least fitting texture and project it onto the input image. We then use the next least fitting texture and add as much as possible of it while minimizing the seam between the two textures using a *graphcut technique* [KSE+03a]. This process is repeated until all input images have been considered. The effect of this strategy is that artifacts which occur only in one input image, such as highlights, are reduced since frequently they result in a visible seam with the current partial texture map. Furthermore the last texture added is the one from the best fitting input image, so most of the final texture results from this image unless it creates inconsistencies with the other input images. Note that the current method does not guarantee removal of artifacts. For example, if a surface region is only visible in one input image and it contains a highlight, then this highlight is part of the final texture map. We have tested this algorithm with more than 40 data sets and did not encounter any problems apart from the shading inconsistencies explained in subsection 5.2.3.

Seam Minimization: Seams between overlapping input image texture regions are minimized by using a *graphcut technique* [KSE+03a]. Given two overlapping images A and B, we want to find the cut within the overlap region, which creates the best transition between these images. The overlap region is represented as directed graph, where each node represents a pixel position p in the overlap region, which is denoted $A(p)$ and $B(p)$ for the two images A and B, respectively. Nodes are connected by edges representing 4-connectivity between pixels. Each edge is given a cost encoding the pixel differences between the two source images at that position.

We have investigated the effect of different parameters for image fusion applications [CFW+12] and tested them with various 3D models. Based on this we use the following parameters: Image pixels are represented in the *RGB* color space. Color distances are computed using the L_2 norm. The cost function w corresponds to the gradient weighted color difference between the images A and B at the neighboring pixels p and q, i.e.,

$$w_{\nabla} = \frac{||A(p)-B(p)||+||A(q)-B(q)||}{||G_A^{pq}(p)||+||G_A^{pq}(q)||+||G_B^{pq}(p)||+||G_B^{pq}(q)||}$$

where $G_A^{pq}(p)$ is the image gradient in the direction of the edge pq at pixel p. This cost function has been originally devised by Kwatra et al. [KSE+03b] based on the observation that seams are more noticeable in low-frequency regions, and a visually more pleasing cut is computed by increasing the cost of an edge with a decreasing image gradient.

Figure 3 illustrates an example in which two texture patches of our *Rooster* model are fused together to form a larger and more complete texture patch. The newly merged texture patch is then fused together with the next available texture patch in the list. The process terminates when all texture patches have been successfully merged.

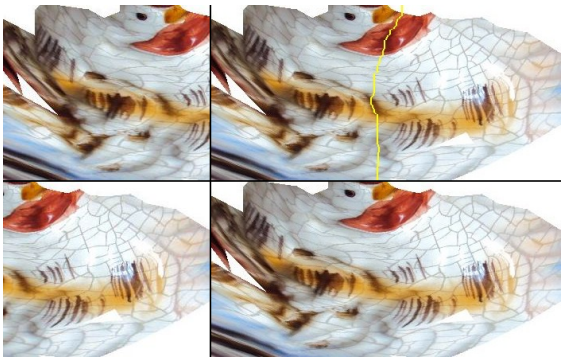

Figure 3: Seam minimization. Source texture patches are shown in the the left column, while the merged texture patch is shown in the right column.

Figure 4 shows the texture map obtained by back-projection surface patches onto the input images (right) and the resulting textured 3D model (left). In many instances the input images do not cover the entire surface of the object. For example, in many of our experiments users did not take photos of the underside of objects. In this case the 3D point cloud contains large gaps. The Poisson surface reconstruction will create a smooth watertight surface interpolating the gaps, but the corresponding regions of the texture map have no color information (red color regions in the top-right image of Figure 4). The accuracy of our new texture reconstruction process is illustrated by comparing the bottom-left image of Figure 2 and the bottom-right image of Figure 4.

5 RESULTS

5.1 Effect of Parameters

We have investigated the effect of different algorithm parameters on the quality of the surface parameterization and texture reconstruction.

5.1.1 Isovalue

The larger the *isovalue* is, the farther the region-growing process continues, and the fewer surface patches are generated. Figure 5 illustrates the surface segmentation and Figure 6 the resulting texture patches. If the isovalue is too large the resulting texture map suffers from large distortions. However, having a single texture patch simplifies some operations such as image inpainting to fill surface regions without matching input images.

 ISBN 978-80-86943-74-9

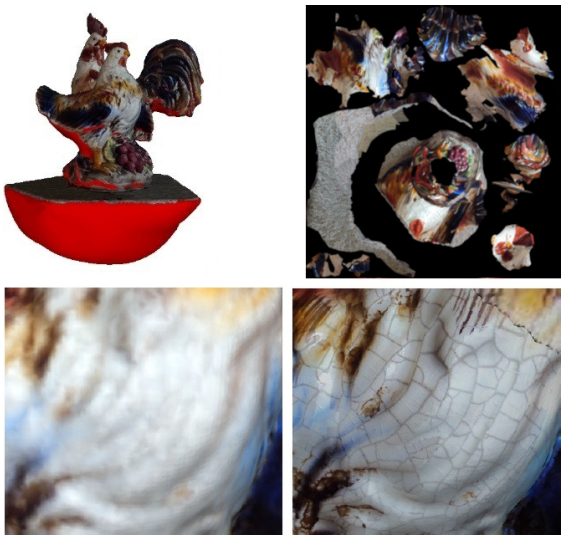

Figure 4: Top row: Reconstruction of the *Rooster* in Figure 2 (left) and the surface parameterization after texture map computation (right). Regions that were not visible in any of the input images are colored red. Bottom row: Surface appearance of the rooster's neck region using vertex color interpolation (left) and our new texture reconstruction process (right).

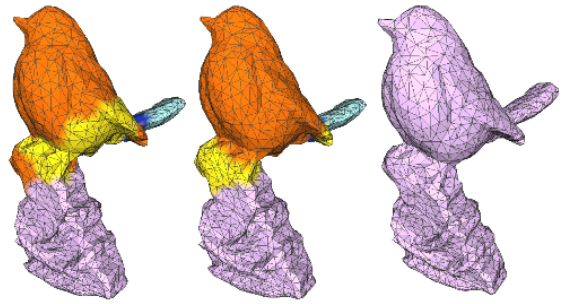

Figure 5: Parameterization of our *Bird* model with the isovalues of 1.0, 2.0, and 5.0, respectively.

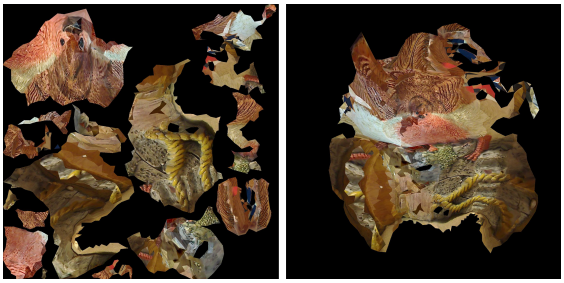

Figure 6: Texture map for the surface parameterization obtained using an isovalue of 2.0 (left) and 5.0 (right).

5.1.2 Number of Gaussian Iteration Steps

Increasing the number of times the *Gaussian filter* function is applied during the parameterization process, effects how sensitive the segmentation process is towards differently sized features. Figure 7 demonstrates that small values result in unnecessarily many segments, whereas large values result in too few patches and hence larger texture distortions.

Figure 8 shows that the resulting texture maps look very similar. However, the texture map generated using 10 Gaussian steps contains falsely oriented texture features in the neck region of the bird model. This seems to be due to aliasing effects caused by a high distortion of the corresponding parameter space region. Contributing causes are the relatively low resolution of the webcam images, and the fact that we currently use a nearest neighbor interpolation for the texture reconstruction.

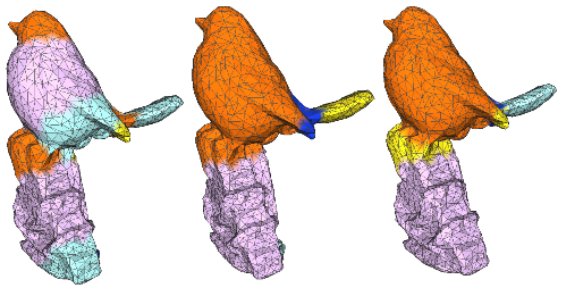

Figure 7: Parameterization of the *Bird* model with (from left to right) 10, 30, and 50 Gaussian steps, respectively.

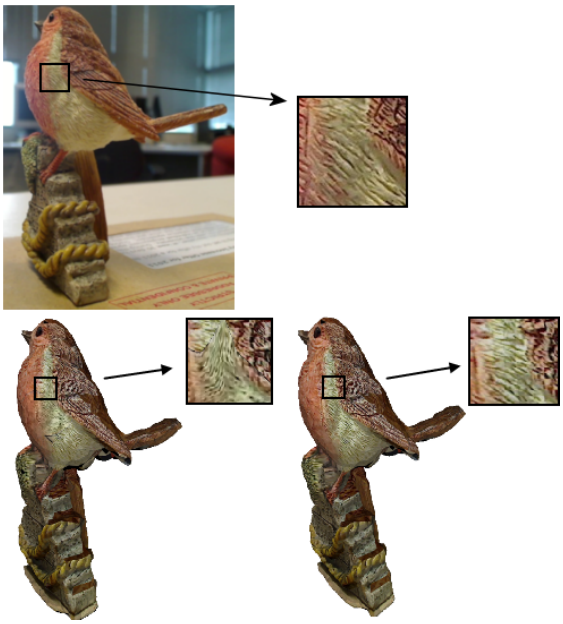

Figure 8: Top: an input image of the *bird* data set. Bottom: the texture map created using 10 (left) and 30 (right) Gaussian steps.

5.2 Reconstruction Results

We have evaluated our system using a variety of datasets of objects at different scales acquired under different weather and lighting conditions. In general, our system produces qualitatively good results with high resolution textures for both uniformly colored and feature-poor objects, and for objects with concave

 ISBN 978-80-86943-74-9

regions and moderately complex geometries. The size of our test datasets varied from as few as 6 images to hundreds of images. All input images were acquired with simple consumer-level handheld cameras, including a Smartphone camera. Our systems fails for objects which have viewpoint dependent surface appearance, e.g., refractive and reflective materials within complex environments. This section contains a summary of different experiments that we performed to evaluate our texture reconstruction method.

5.2.1 *Rooster Dataset*

The first dataset contains 35 images of a *White Rooster* with a resolution of 2592 $\times$ 1944 pixels. Figure 9 shows some of the input images. The original object has a complex surface geometry with many bumps and wrinkles. Notice that most of the surface of the model contains few visual features.

Figure 9: Two out of 35 input images of the *White Rooster* datasets.

The resulting reconstructed model, shown in the left of Figure 13, is of good quality and bears a high resemblance to the original object. The overall shape, along with details such as feathers of the original model are reconstructed well. The resulting model consists of 298,187 polygons. There are a few regions (underneath the model) where no texture has been generated (colored in red) due to missing input images showing these regions.

5.2.2 *General Dataset*

This data set contains 18 images (2592 $\times$ 1944 pixels resolution) of a *General* figurine. The original model has a very smooth, reflective and shiny surface. The reconstruction, shown on the right-hand side of Figure 10, is of good quality and the final model has a high resemblance to the original object. The resulting model consist of 101,778 polygons. The texture is very realistic, but contains some visible seams along patch boundaries

Figure 10: Input image of the *General* dataset (left) and the resulting reconstruction (right).

5.2.3 *Vase Dataset*

This dataset contains 26 images (2592 $\times$ 1944 pixels resolution) of a vase. The original object has a very smooth, reflective and shiny surface with repetitive textures. The reconstructed model has 215,918 polygons. The geometry of the reconstruction is very realistic. However, the texture reconstruction shows some visible illumination differences due to some input images having been taken with flash and some without. In future we plan to overcome these problems by using multiband blending techniques [APK08] and global optimization of luminance values in CIELUV color space along seam boundaries.

Figure 11: Image of a vase (left) and the resulting 3D reconstruction (right). The enlargement shows brightness variations due to some input images taken with flash.

5.2.4 *Objects with High Genus*

Section 2 reviewed previously presented techniques for texture reconstruction. Despite some seemingly impressive results, we did not find any examples in the literature for objects with high genus, for which geometry and texture reconstruction are notoriously difficult. Figure 12 illustrates that our image-based modeling system

 ISBN 978-80-86943-74-9

and texture reconstruction method handles such cases without problems.

Figure 12: Two examples of models with a high genus: input image (top), 3D reconstruction (middle), and the surface parameterization (bottom).

5.3 Running Time

The presented algorithm has not been optimized yet and the running time varies between approximately 10 minutes for the reconstruction of an apple from 6 photographs, to many hours for more complex models. For example, the reconstruction of the rooster data set in subsection 5.2.1 takes 6 hours and 19 minutes on a PC with Intel Quad Core i7 CPU and 6GB RAM. The time requirements of the various stages of the algorithm are:

1. Camera Parameter Estimation: 18.6% = 71 minutes (feature detection and matching are implemented in parallel and use all four cores of the CPU)
2. Point Cloud Generation: 33.0% = 125 mins
3. Mesh Processing: 9.8% = 37 mins
4. Texture Reconstruction: 38.6% = 146 minutes

Initial tests indicate that a GPU implementation would be 50-100 times faster. Alternatively a compute cloud could be used to speed up computation.

5.4 Comparison

The combination of "Bundler" [SSS08] and CMVS & PMVS [FCSS10] is a well-known and open-source image-based modeling system. However, the output of these research tools is a dense point cloud. While we can easily obtain a closed surface from this data, we were unable to find published software for texture reconstruction. We hence compared our system with the only complete systems we could find. We identified thirteen companies working in this field and compared the best four algorithms [NWDL12a]. We showed that our solution and "123D Catch" achieved the best geometry reconstruction. The system presented in this paper achieves even higher quality reconstructions due to the integration of silhouette information and the novel texture reconstruction algorithm. Figure 13 demonstrates that these improvements make a significant difference when dealing with data sets containing few distinct visual features. For such data sets "123D Catch" struggles both with reconstructing a correct geometry and appropriate texture map.

Figure 13: 3D reconstruction from the "white rooster data set" using our method (left) and "123D Catch" (right).

6 CONCLUSION AND FUTURE WORK

We have described a texture reconstruction technique for image-based modeling systems. In contrast to previously presented methods we integrate shape-from-silhouette and correspondence-based methods, which gives us very reliable camera parameter estimates and excellent geometry reconstruction. This enables us to fuse together texture regions obtained from input images without requiring excessive blending and deformations. Textures are combined using a greedy algorithm and a graph-cut technique minimizing gradient weighted color differences. The texture reconstruction uses an advanced surface parameterization method which takes into account the genus and geometric features of an object We have demonstrated the quality of the reconstruction process using objects with different geometries, genus, colors and surface properties. In all cases we achieved an excellent reconstruction and realistic texture. In contrast to laser scanners our system also works for shiny and dark objects, and is easily scalable.

 ISBN 978-80-86943-74-9

Some problems still exist with seams along texture patches, and discontinuities due to color inconsistencies created during the image acquisition process. The current system does not generate a texture for surface regions not visible in the input images. We currently work on texture inpainting techniques and exemplar-based texture synthesis to fill such regions [PGB03, CPT04].

7 REFERENCES

[APK08] Cedric Allene, Jean-Philippe Pons, and Renaud Keriven. Seamless image-based texture atlases using multi-band blending. *19th International Conference on Pattern Recognition*, pages 1–4, 2008.

[BMR01] Fausto Bernardini, Ioana M. Martin, and Holly Rushmeier. High-quality texture reconstruction from multiple scans. *IEEE Trans. on Visualization and Computer Graphics*, 7(4):318–332, October 2001.

[CAH+13] A. Colburn, A. Agarwala, A. Hertzmann, B. Curless, and M.F. Cohen. Image-based remodeling. *IEEE Transactions on Visualization and Computer Graphics*, 19(1):56–66, 2013.

[CFW+12] Xiao Bao Clark, Jackson Finlay, Andrew Wilson, Keith Milburn, Minh Hoang Nguyen, Christof Lutteroth, and Burkhard C. Wünsche. An investigation into graphcut parameter optimisation for image-fusion applications. In *Proceedings of Image and Vision Computing New Zealand (IVCNZ 2012)*, pages 480–485, Dunedin, New Zealand, 2012.

[CPT04] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *Trans. Img. Proc.*, 13(9):1200–1212, September 2004.

[CZCW12] Zhaolin Chen, Jun Zhou, Yisong Chen, and Guoping Wang. 3d texture mapping in multi-view reconstruction. In *Advances in Visual Computing*, volume 7431 of *Lecture Notes in Computer Science*, pages 359–371. Springer Berlin Heidelberg, 2012.

[EDD+95] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppey, Michael Lounsberyz, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. *Computer Graphics Proceedings (SIGGRAPH 1995)*, pages 173–182, 1995.

[EdDM+08] Martin Eisemann, Bert de Decker, Marcus A. Magnor, Philippe Bekaert, Edilson de Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. Floating textures. *Computer Graphics Forum*, 27(2):409–418, 2008.

[FCSS10] Y. Furukawa, B. Curless, S.M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *Proceedings of Computer Vision and Pattern Recognition (CVPR 2010)*, pages 1434–1441, 2010.

[Flo97] Michael S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14(3):231–250, April 1997.

[GC09] B. Goldluecke and D. Cremers. Super-resolution texture maps for multiview reconstruction. In *Proceedings of the 12th International Conference on Computer Vision (ICCV 2009)*, pages 1677–1684, 2009.

[HQZH08] Shaoxing Hu, Jingwei Qiao, Aiwu Zhang, and Qiaozhen Huang. 3d reconstruction from image sequence taken with a hand-held camera. *International Archives of the Photogrammetry*, 37(91):559–563, 2008.

[HSKK01] Masaki Hilaga, Yoshihisa Shinagawa, Taku Komura, and Tosiyasu L Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. *Computer Graphics Proceedings (SIGGRAPH 2001)*, pages 203–212, 2001.

[HVC08] Carlos Hernandez, George Vogiatzis, and Roberto Cipolla. Multi-view photometric stereo. *IEEE Transaction on Pattern Recognition and Machine Intelligence*, 30(3):548–554, 2008.

[Kaz05] Michael Kazhdan. Reconstruction of solid models from oriented point sets. In *Proc. of the 3rd Eurographics symposium on Geometry processing*, pages 73–82, 2005.

[KBH06] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the 4th Eurographics symposium on Geometry processing*, pages 61–70, 2006.

[KSE+03a] Vivek Kwata, Arno Schodl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transaction Graphics*, 22(3):277–286, 2003.

[KSE+03b] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.*,

 ISBN 978-80-86943-74-9

22(3):277–286, July 2003.

[LH01] Hendrik P. A. Lensch and Wolfgang Heidrich. A silhouette-based algorithm for texture registration and stitching. *Graphical Models*, 63(4):245–262, 2001.

[LI07] V. Lempitsky and D. Ivanov. Seamless mosaicing of image-based texture maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pages 1–6, 2007.

[Lor95] W. E. Lorensen. Marching through the visible man. In *Proceedings of IEEE Visualization '95*, pages 368–373, 1995.

[Low99] David G. Lowe. Object recognition from local scale-invariant features. *International Conference on Computer Vision*, 2:1150–1157, 1999.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[LWC06] L. M. Lui, Y. Wang, and T. F. Chan. Solving PDEs on manifold using global conformal parameterization. In *Proceedings of the Third International Workshop on Variational, Geometric, and Level Set Methods in Computer Vision (VLSM 2005)*, pages 309–319, 2006.

[MBR+00] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J. Gortler, and Leonard McMillan. Image-based visual hulls. In *Computer Graphics Proceedings (SIGGRAPH 2000)*, pages 369–374, 2000.

[NWDL12a] Minh Hoang Nguyen, Burkhard C. Wünsche, Patrice Delmas, and Christof Lutteroth. 3d models from the black box: Investigating the current state of image-based modeling. In *WSCG 2012 Communication Proceedings*, pages 249–258, Pilsen, Czech Republic, June 2012.

[NWDL12b] Minh Hoang Nguyen, Burkhard C. Wünsche, Patrice Delmas, and Christof Lutteroth. Modelling of 3d objects using unconstrained and uncalibrated images taken with a handheld camera. In *Computer Vision, Imaging and Computer Graphics - Theory and Applications*, pages 1–16. Springer Verlag, 2012.

[NWDL13] Hoang Minh Nguyen, Burkhard Wünsche, Patrice Delmas, and Christof Lutteroth. A hybrid image-based modelling algorithm. In *Proc. of the 36th Australasian Computer Science Conference (ACSC 2013)*, pages 115–123, Adelaide, Australia, 2013.

[PGB03] Patrick Perez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transaction Graphics*, 22(3):313–318, 2003.

[Ree46] Georges Reeb. Sur les points singuliers díune forme de pfaff completement integrable ou diune fonction numerique [on the (singular points of a completely integrable pfaff form or of a numerical function). *Comptes Randus Acad. Sciences Paris 222*, pages 847–849, 1946.

[REH06] Fabio Remondino and Sabry El-Hakim. Image-based 3d modelling: A review. *The Photogrammetric Record*, 21:269–291, 2006.

[SGSH02] Pedro V Sander, Steven J Gortler, John Snyder, and Hugues Hoppe. Signal-specialized parameterization. *Proceedings of the 13th Eurographics Workshop on Rendering*, pages 87–100, 2002.

[SPR06] Alla Sheffer, Emil Praun, and Kenneth Rose. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.*, 2(2):105–171, January 2006.

[SSS08] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *Int. J. Comput. Vision*, 80(2):189–210, November 2008.

[VA12] Robert Valkenburg and Nawar Alwesh. Seamless texture map generation from multiple images. In *Proc. of the 27th Conference on Image and Vision Computing New Zealand*, IVCNZ '12, pages 7–12, New York, NY, USA, 2012. ACM.

[VW90] M. Visvalingam and J. D. Whyatt. The Douglas-Peucker algorithm for line simplification: re-evaluation through visualization. *Computer Graphics Forum*, 9(3):213–228, September 1990.

[XLL+10] Lin Xu, E. Li, Jianguo Li, Yurong Chen, and Yimin Zhang. A general texture mapping framework for image-based 3d modeling. In *Proc. of the 17th IEEE International Conference on Image Processing (ICIP 2010)*, pages 2713–2716, 2010.

[ZMT05] Eugene Zhang, Kobstantin Mischaikow, and Greg Turk. Feature-based surface parameterization and texture mapping. *ACM Trans. Graph.*, 24(1):1–27, 2005.

 ISBN 978-80-86943-74-9