User-guided 3D reconstruction using multi-view stereo

Sverker Rasmuson
sverker.rasmuson@chalmers.se
Chalmers University of Technology

Erik Sintorn
erik.sintorn@chalmers.se
Chalmers University of Technology

Ulf Assarsson
uffe@chalmers.se
Chalmers University of Technology

Figure 1: An object reconstructed using our interactive tool. From left to right: preserved topology based on user input, the same model with projected textures, a reference image.

CCS CONCEPTS

• **Mathematics of computing** → *Mathematical optimization*; • **Computing methodologies** → *Computer graphics*; *Graphics systems and interfaces.*

KEYWORDS

modeling, multi-view stereo, interactive systems

ACM Reference Format:
Sverker Rasmuson, Erik Sintorn, and Ulf Assarsson. 2020. User-guided 3D reconstruction using multi-view stereo. In *Symposium on Interactive 3D Graphics and Games (I3D '20), May 5–7, 2020, San Francisco, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3384382.3384530

ABSTRACT

We present a user-guided system for accessible 3D reconstruction and modeling of real-world objects using multi-view stereo. The system is an interactive tool where the user models the object on top of multiple selected photographs. Our tool helps the user place quads correctly aligned to the photographs using a multi-view stereo algorithm. This algorithm in combination with user-provided information about topology, visibility, and how to separate foreground from background, creates favorable conditions in successfully reconstructing the object.

The user only needs to manually specify a coarse topology which, followed by subdivision and a global optimization algorithm, creates an accurate model with the desired mesh density. This global optimization algorithm has a higher probability of converging to an accurate result than a fully automatic system.

With our proposed tool, we lower the barrier of entry for creating high-quality 3D reconstructions of real-world objects with a desirable topology. Our interactive tool separates the most tedious and difficult parts of modeling to the computer, while giving the user control over the most common robustness issues in automatic 3D reconstruction.

The provided workflow can be a preferable alternative to using automatic scanning techniques followed by re-topologization.

I3D '20, May 5–7, 2020, San Francisco, CA, USA

ACM ISBN 978-1-4503-7589-4/20/05…$15.00
https://doi.org/10.1145/3384382.3384530

1 INTRODUCTION

There is a high demand for photo-realistic assets for use in computer games, movies, and industrial and architectural applications. Often, a large number of assets are used in the pursuit of detailed and highly realistic environments. These realistic assets can be modeled by hand. However, efforts have increasingly been made to use 3D-scanning to automatically produce content from real-world objects.

Automatic scanning of static objects is viable under certain circumstances. However, the process is far from trivial. Fully automatic solutions that only take images as input, are usually dependent on favorable lighting conditions and high-quality photographs to be successful. Other systems use elaborate setups of cameras in a studio setting and typically require expensive equipment to allow for high-quality reconstructions.

Even with a high-quality reconstruction algorithm, the resulting geometry will be of very high density, and with arbitrary topology. While tools exist that allow for semi-automatic simplification of these results, achieving a topology that is sufficiently good for, e.g., animation will still require significant user input. The main contribution in this paper is a novel system that allows the user to specify topology *prior* to reconstruction, which allows the reconstruction algorithm to utilize the user's knowledge of visibility and surface curvature to achieve better results.

With our method, the user draws a coarse mesh on top of a photograph, ensuring the desired topology and resolving visibility, while the tool automatically aligns the geometry with the input photographs.

The user can easily infer where a quad can reasonably approximate a surface and align it with discontinuities, which in turn simplifies the automatic optimization since it can rely on the assumption of a locally flat surface patch. The user can also easily distinguish foreground from background. Since the matching windows of our reconstruction algorithm correspond to the quads of the coarse geometry, it is much easier to capture regions with little or no texture. Additionally, the user can easily work around, or avoid, regions of high specularity that would only introduce noise in an automatic algorithm.

The system is designed to be easy to use for novice users. However, some prior knowledge about quad modeling and how to construct a favorable topology is recommended for optimal results.

In the next section, we will give an overview of related work. Then, in Section 3, we will describe our novel modeling system which allows a user to easily draw a coarse mesh on top of photographs. In Section 4 we will describe an optimization algorithm that interactively aligns this mesh to the input photographs. Finally, in Section 5 we evaluate the quality of our method by comparing the results achieved by a novice modeler using our tool to the results obtained by an automatic reconstruction software.

2 PREVIOUS WORK

2.1 Automatic algorithms

Multi-view reconstruction is a well-studied problem. For an overview, see Hartley and Zisserman [Hartley and Zisserman 2004] and Seitz et al. [Seitz et al. 2006].

In one type of systems, the input to the system is a collection of photographs, generally taken by the same camera, with the camera locations and orientations computed by Structure from Motion (SfM) [Snavely et al. 2006]. This calibration often works provided that there is enough overlap between images, since the algorithm uses image-space features to find matching points between them. With this camera calibration and sparse reconstruction from the SfM algorithm, a dense reconstruction can be performed to create 3D models, e.g. using Patch Match Stereo [Bleyer et al. 2011][Schönberger and Frahm 2016].

Such automatic scanning procedures are compelling due to their simplicity, since only a set of photographs have to be provided. However, to achieve high quality, they are dependent of the quality of the input photographs, especially concerning the lighting conditions under which they are captured. This can be challenging due to the necessary experience in photography and 3D reconstruction techniques.

Due to their ease of use, there are several open-source, as well as commercial, programs that perform this kind of reconstruction. Examples of open-source software are Meshroom, COLMAP and Bundler. Commercial variants include RealityCapture and Photoscan [Bianco et al. 2018].

Another common variant of systems uses a multi-camera setup capturing a fixed-size volume. This setup could capture the volume from all sides, typically in a dome, or could have a more limited set of views from a circle or just one side of the object [Seitz et al. 2006][Beeler et al. 2010][Collet et al. 2015]. Such setups are typically calibrated with some form of calibration target to allow for higher accuracy and wider angle between cameras [Zhang 2000].

Multi-camera setups are typically not accessible to the average user due to the special equipment and resources (cameras, means of synchronization, calibration, green screen, studio, special light sources, and reflectors) required. Professional studios that provide these facilities are starting to become an alternative, although they are still in limited numbers and availability.

Fixed setups like these also have the downside of not being able to capture immobile objects, such as statues, houses or other objects outdoors.

2.2 Interactive algorithms

One popular approach is to use Kinect Fusion, where a depth camera is used as a handheld camera for interactively scanning geometry [Izadi et al. 2011][Newcombe et al. 2011]. This method is compelling due to its ease of use but has a tendency to create too smooth meshes by the use of averaging and cannot handle non-watertight topologies. Kinect Fusion uses a volumetric signed distance field as its internal data representation, which must be converted into triangles with e.g. marching cubes, with no guarantees given on a reasonable topology. The scanning might also be affected by inherent limitations of the depth cameras, such as not being able to scan outdoors, or multi-path interference in the case of time-of-flight cameras.

In certain cases, a predefined template mesh can be used, and automatically fitted to the input images. This has been successfully achieved in real time by, for instance, Zollhöfer et al. [2014]. Unlike our method, the final mesh is obtained through optimizing the mesh vertices against previously obtained depth maps.

Another method is to use user-provided annotations directly in a global multi-view stereo algorithm. These annotations are added directly to the variational energy functions as constraints on smoothness, discontinuity and ordering. This method does not consider object topology. However, it achieves good results for the reconstructed depth map [Doron et al. 2015].

There are also several methods that use SfM in an interactive approach. One successful application of interactive modeling with SfM is for architectural modeling. Architectural motifs often have a geometry which lends itself to piecewise planar approximations that can be aligned with the sparse geometry from SfM and can also make use of constraints from inferred vanishing points [Sinha et al. 2008].

In VideoTrace [van den Hengel et al. 2007], the sparse geometry acquired from SfM is used to interactively model objects in videos. First, a preprocessing step uses an SfM algorithm to create a sparse geometric representation of the filmed sequence. Polygons are then traced out manually in one or several frames and are aligned to the sparse geometry by use of plane fitting. Given a prospective model made up of such polygon faces, further model validation is made with 2D information, comprised of the difference of color histograms for each face [van den Hengel et al. 2007].

SfM has also been used successfully for modeling in combination with a sketch based modeling technique called 3-sweep modeling,

where two strokes represent the profile and one stroke is used to extrude the profile [Chen et al. 2013]. The sparse geometry acquired from SfM is used to compute depth values for stroke endpoints, which creates accurate 3D models in accordance with the input images [Xu et al. 2016].

Another sketch-based approach is due to Habbecke and Kobbelt [Habbecke and Kobbelt 2009], who use an interactive method to piecewise build up a model using dense multi-view stereo. Similarly to our approach, they use a variational framework to find an optimal mesh given a set of input images. However, they use the user input only for visibility and foreground segmentation, whereas our algorithm also takes topology and mesh density into consideration. Their method is more focused on interface simplicity compared to our method, which concentrates on obtaining accurate geometry and topology [Habbecke and Kobbelt 2009]. To the best of our knowledge, no previous method exists in which the multi-view stereo algorithm itself is guided and refined by a user-provided topology.

3 INTERACTIVE MODELING SYSTEM

In this section, we present the user interface and overview the design of our application. We will also elaborate on some of the non-obvious choices we have made building this system.

3.1 Preprocessing

The only preprocessing step necessary is to calibrate the set of images to obtain intrinsic and extrinsic camera parameters, for which we use an off-the-shelf SfM package called COLMAP [Schönberger and Frahm 2016]. The scale of the scene given by the calibration can be adjusted manually if needed.

3.2 View Selection Interface

The proof-of-concept system is implemented as an interactive tool for 3D reconstruction. The input to the system is a set of calibrated images of the object that is to be reconstructed. The user photographs the subject from the view directions that are most relevant to cover. Additional images can be added as needed. The input images are shown as a list at the top of the screen.

To start the reconstruction, the user chooses a set of images (two or more) that have a clear view of the part of the object of interest (see Figure 2). One of the selected images is chosen to be the *reference* image. Modeling will take place in this reference image, and later optimization will not alter the topology that the user specifies in this view. The selected images are saved in a view selection list, allowing the user to later go back and forth between different sets of views.

3.3 Modeling Interface

The user constructs the geometry by creating quads on top of the reference image with the intention of creating a suitable topology. Every time a quad has been created, an underlying multi-view stereo algorithm automatically finds an approximate depth for each of its vertices in the reference image, thus not moving the quad from the position in the image plane where the user constructed it. When the user is satisfied with this particular area of the object, the next place of interest can be chosen, and along with it a new set of views (see Figure 2). For each quad, the corresponding views that were used to create it are stored, along with the chosen reference view.

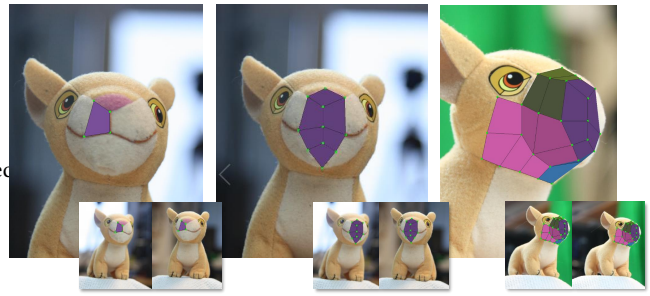

Figure 2: In the left-most image, the user starts modeling by placing a quad just at the front of the model, which is automatically aligned given two views. In the middle image, more quads have been placed in the same views. In the right-most image, other views with better visibility have been chosen, signified by a unique color.

To help with editing, we employ a number of different visualization modes for quads. The default mode shows which quads belong to which views, by using a unique color. Secondly, there is a mode that visualizes the optimization score provided the quad's current vertex positions. Finally there is a mode that, for each view *not* being a reference view, projects the quad and samples from the reference image. This is useful for evaluation, since it makes it easy to spot poorly aligned quads, or quads with incorrect visibility.

3.4 Subdivision And Optimization

When the first coarse topology of the object is finished, the user can choose to subdivide the whole or parts of the mesh where it is deemed necessary and also manually adjust vertices or reapplying optimization to specific quads if required (see Figure 3). If satisfied with the topology, a global optimization can be applied, optimizing the positions of all vertices at once, using a joint photogrammatical and smoothness energy function (see Section 4). In this algorithm, each quad uses its remembered views from construction to ensure correct visibility. Prior to this, the user can also mark quads belonging to particularly troublesome parts of the geometry that are to be excluded from the photometric term of the optimization.

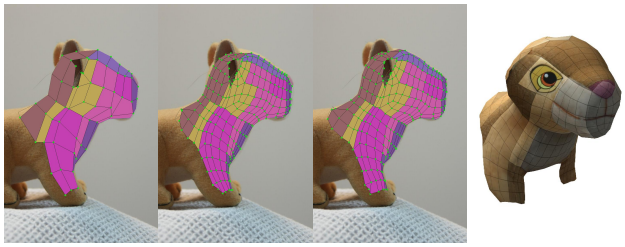

Figure 3: Left: the user has finished a coarse topology. Middle left: after subdivision. Middle right: after global optimization the geometry can be seen to follow the silhouettes closely. Right: a rendered view of the optimized model.

4 MULTI-VIEW RECONSTRUCTION

This section will cover the underlying reconstruction engine of our application, which uses a multi-view stereo algorithm to align the geometry to the set of input photographs.

We use two optimization algorithms in our application: one exhaustive search for coarse grained alignment of quads while modeling, and one global mesh optimization that optimizes all vertices of the mesh at once. Both algorithms use quads as the base primitive, to retain the information provided by the user.

4.1 Photo-consistency score

Most previous work on multi-view stereo attempts to match each pixel in one image pairwise to pixels in other images. To be able to classify two pixels as describing the same point in space, a small matching window around the pixel is used. Our method works similarly but, instead of a pixel, a whole quad of the mesh is to be matched between images, and thus the matching window is the projection of the quad on each camera's image plane. If the vertices of the quad are in the correct position, the quads can be sampled uniformly in 3D, and projected samples will be similar in the images.

To compare projected samples, a suitable photometric consistency score is needed. Common such scores include sum of absolute differences (SAD), sum of squared differences (SSD), and normalized cross-correlation (NCC). Often, these cost functions are augmented by subtracting the mean value over the filter that is used, which increases the robustness with respect to local lighting variations [Szeliski 2010].

To be able to have a consistent photometric score regardless of the number of views that are currently in use, we propose a measure that is an extension to the common mean-subtracted SAD score, which we select because of its simplicity and robustness.

The mean-subtracted SAD is defined as

$$D_{SAD} = \frac{1}{n}\sum_{i}^{n} |(p_i - \mu_p) - (q_i - \mu_q)|, \tag{1}$$

where n is the number of samples, p is each sample from the first image with corresponding mean value μ_p, and q is each sample from the second image with corresponding mean value μ_q. Our proposed measure is analogous to the variance but instead uses the L1-norm such that

$$P = \frac{1}{n}\frac{1}{m}\sum_{i}^{n}\sum_{j}^{m} |(p_{i,j} - \mu_j) - \mu_i|, \tag{2}$$

where m is the number of views used, $p_{i,j}$ is the i:th sample at view, j, and

$$\mu_i = \frac{1}{m}\sum_{j}^{m} p_{i,j}. \tag{3}$$

μ_j is the same mean value as in Equation 1.

This measure also collapses to the regular mean-subtracted SAD when $m = 2$.

4.2 Exhaustive Search

The exhaustive search algorithm starts automatically whenever the user has specified the four vertices of a quad, trying to orient the quad in world space given the current views. This is achieved by generating a number of prospective quad positions by taking N steps in the view-ray direction of the reference view for each free vertex (i.e., each vertex not connected to any other quad), from $-d$ to $+d$ around the current position. Hence, N^V number of quads will be evaluated, where V is the number of free vertices.

This exhaustive approach independently aligns each quad with respect to the input images. The quads can, in this stage, lie quite far from the actual geometry and thus far from a minimum in our photometric cost function (Equation 2). This method robustly finds approximately correct configurations that can be used for further refinement at a later stage (see Section 4.4).

For the exhaustive search, two constraints are added that enforce the assumption that the user has chosen views with full visibility of the current quad.

The first constraint ensures that the quad is not back facing in any of the views in the current view set. This can easily be tested by evaluating the winding order of the projected vertices in each view compared to the reference view. The other constraint is to ensure that the current quad is not occluded by any other geometry in the mesh from the current views. To evaluate this, depth maps of the mesh are rendered from each view in the view set. A depth test is then made for each of the quad's vertices to see that they do not lie behind any current geometry.

4.3 Energy Function

The goal of our application is to align the geometry in accordance to the input photographs while preserving the topology that the user has specified. To achieve this using a variational approach, a suitable energy function must be specified.

The basic energy that we want to minimize is a function over the mesh given the photometric consistency score in Equation 2. Since this is computed per quad the energy functions simply becomes

$$E_1 = \sum_{q \in M} P_q \tag{4}$$

for every quad, q, in model M, where P_q is evaluated with respect to the images contained in the view set for each quad q.

To Equation 4, we also add a smoothness term to counteract inevitable image noise and imperfect calibration. This also leads to a wanted coupling between vertices and generally improves the rate of convergence. It is also important for areas with little or no texture information.

For this smoothness term, we look at the normals for the 1-ring neighbors of triangles around a given vertex. For each quad connected to the vertex, a face point, f, is computed, which is the average of the quad's vertices. Two triangles are then formed for each quad which both have f and the current vertex as their first two vertices, and the third one being the two connected vertices in that quad respectively. For each such triangle, two per connected quad, a normal n_i is computed. Firstly, an average n_v of these normals is computed as the normal of the current vertex. Then, the dot product

between all n_i:s and n_v are accumulated to produce a smoothness score. Expressed as an energy function, this becomes

$$E_2 = \sum_{v \in M} (1.0 - \frac{1}{N} \sum_{i}^{N} \mathbf{n}_v \cdot \mathbf{n}_i) \tag{5}$$

where N is two times the number of connected quads for vertex v.

Finally, we add a term that enforces the quads to be as flat as possible. This works in a similar way as the smoothness term by making use of the normals n_i corresponding to the four triangles produced by the two possible triangulations of a quad. These normals are averaged into n_q, and then the dot product is used to compute the average deviation from this:

$$E_3 = \sum_{q \in M} F_q = \sum_{q \in M} (1.0 - \frac{1}{4} \sum_{i}^{4} \mathbf{n}_q \cdot \mathbf{n}_i) \tag{6}$$

for every quad, q, in model M.

The final energy function is the sum of these three terms, weighted with three scalars a, b, c such that

$$\begin{aligned} E_f = & aE_1 + \\ & bE_2 + \\ & cE_3 \end{aligned} \tag{7}$$

with $a + b + c = 1$. Typical values for these weights are $a = 0.98, b = 0.01, c = 0.01$.

4.4 Global optimization

When the user has built (a part of) a coarse model, a more fine-grained type of optimization can be applied. The information built up during construction provides favorable conditions compared to when the input data comes from a scanning procedure. These conditions include:

- correct visibility,
- quads align with approximately flat areas,
- quads are oriented correctly in world space,
- quads are aligned to discontinuities,
- and correct foreground/background segmentation has been performed.

The final Equation 7 is a function of the mesh vertices. Vertices can only be moved along a ray: for most vertices, this ray is the same as the view direction of the owning quads reference view. For vertices connected to quads with different view sets, however, the average of the corresponding view directions is used.

Equation 7 is thus a function of a set of ray parameters $\mathbf{t}$. This equation needs to be minimized globally for the whole model with respect to these parameters. For this, we need to compute the gradient

$$\begin{aligned} \nabla E_f(\mathbf{t}) = & a\nabla E_1(\mathbf{t}) + \\ & b\nabla E_2(\mathbf{t}) + \\ & c\nabla E_3(\mathbf{t}). \end{aligned} \tag{8}$$

Since the photometric term and the flatness term are computed per quad, we need to compute the partial derivatives per quad. These partial derivatives can be expressed as

$$\frac{\partial E_1}{\partial t_i} = \sum_{q \in M} \frac{\partial P_q}{\partial t_i} = \sum_{q \in Q_i} \frac{\partial P_q}{\partial t_i} \tag{9}$$

where Q_i is the quads connected to the vertex corresponding to the ray parameter t_i. This is done in a similar way for Equation 6 so that

$$\frac{\partial E_3}{\partial t_i} = \sum_{q \in M} \frac{\partial F_q}{\partial t_i} = \sum_{q \in Q_i} \frac{\partial F_q}{\partial t_i}. \tag{10}$$

Given the partial derivatives $\frac{\partial E1}{\partial t_i}$ and $\frac{\partial E3}{\partial t_i}$ along with $\frac{\partial E_2}{\partial t_i}$, which is computed per vertex, all terms for equation 8 have been assembled.

Described in words, our global optimization algorithm works like this: optimize the position of all vertices in the mesh, with the constraint that each vertex lies somewhere along the view-ray of its reference view (or the average of view-rays if the vertex is shared). The positions are optimized so that the score for each *quad* is minimized, where the score is a sum of a photometric, smoothness, and flatness score.

Equation 7 is an error function that can be minimized efficiently using e.g. the Gauss-Newton or Levenberg–Marquardt method. We have opted for a simpler approach using Gradient Descent, which empirically has shown to converge fast enough for our purposes. The partial derivatives are computed using central differences [Nocedal and Wright 2006].

4.5 Implementation

When modeling, it is important that the underlying optimization is unobstructive to the workflow, i.e, that it is fast enough to not stall the user. The, by far, most expensive part of both the exhaustive search and the global optimization is estimating the photometric consistency score for a quad (Equation 2).

This is therefore implemented on the GPU using CUDA, and several such comparisons can be performed efficiently and in parallel by launching kernels with a list of quads as input.

The work is divided into three different kernels, where each kernel processes one quad per thread. The problem is embarrassingly parallel with respect to each quad, and as long as a sufficient amount of quads are processed simultaneously the utilization of the GPU is high. Only for a small amount of quads other strategies such as parallelizing over each quad sample could be considered.

The first kernel samples the quad uniformly in 3D space, and then project these positions to and sample from the relevant images. These image samples are then used as input to a second kernel that computes the mean value (corresponding to μ_j in Equation 2) over the projected quad. The last kernel computes the actual score of Equation 2 by using the sampled values and the computed mean values. The output of this last kernel is a list of scores per quad.

For the exhaustive search, Section 4.2, we send the list of quads corresponding to all prospective configurations to the GPU. The

resulting list of scores is traversed and the lowest score corresponds to the best quad configuration.

The global optimization uses the same kernels but instead sends the quad configurations corresponding to the central difference of vertex positions for evaluation (see Section 4.4).

We use Catmull-Clark subdivision to create a higher resolution mesh [Catmull and Clark 1978]. Since we use quads, lines are preserved, and we are guaranteed not to get any singularities.

5 RESULTS

We have evaluated our tool for four different scenes (see Figure 5). The results from the reconstructions will primarily be compared to the output of the automatic reconstruction software Meshroom [AliceVision 2018]. Meshroom is a free, open-source framework for 3D reconstruction based on photogrammetry. It uses a standard pipeline based on feature matching and SfM in combination with dense stereo reconstruction.

In Figure 5, the resulting topology is compared between our method at different levels of subdivison and an automatic reconstruction with Meshroom. In the left-most image in these figures, the coarse topology modeled by the user is shown. The next two images, from left to right, show the evolving mesh when applying subdivison and global optimization, which are automatic procedures. The right-most images show the automatic reconstruction produced by MeshLab. These models are highly tessellated (the inset shows individual triangles when zooming in). This triangulation is somewhat arbitrary and does not follow any particular topology. In most cases, using these meshes would require a manual re-topologizing pass, an amount of work similar in magnitude to what our approach requires. However, due to noise in the scanned result, the re-topologization would cause much of the detailed geometry to be lost. In contrast, we decide upon the topology first and then optimize with respect to that, which does not lead to the same information loss.

In Figure 6, the quality between the reconstructed models are compared. From left to right they show the scanned model with original tessellation, our model after two subdivisions, and an automatically re-sampled version of the scanned model.

The original scanned model has a high mesh density, and while it captures much of the detail in, e.g., Figure 6d, it also exhibits a lot of noise. When re-sampled, Figure 6f, most of the noise is removed. However, important details such as creases and folds are lost. Due to, in our method, leveraging the provided information from the user regarding visibility, silhouettes, and topology, the problem of noise is mitigated while keeping the detail level high. In several cases, as in Figure 6h and Figure 6k, our examples show more detail than the scanned counterparts.

At the silhouette of the ear in Figure 6k we can see how sharp features are handled using our method. While, for our method, manual work is required to align the coarse topology along the edge of the ear, it is difficult for a fully automatic scanning procedure to capture such features accurately (see Figure 6j).

In Figure 6g, a clear example can be seen where the automatic scanning fails. In the middle of the plastic tunnel and to the sides there are uniform green areas that cannot be handled by a typical photogrammetry pipeline (see Figure 4 for a clearer view of of these areas). In our reconstruction, Figure 6h, large quads (see Figure 5i) are placed in this region that allows them to cover at least some texture and anchor them into place. For a higher level of subdivison (Figure 5k), quads are small enough to only cover uniformly colored regions also in our case. However, our global regularization term keeps noise and distortion from being introduced to the geometry.

Figure 4: One of the photographs of the tunnel scene. The uniform green areas are hard to reconstruct for an automatic scanning procedure since they locally lack texture information. Due to placing large quads with correct visibility and with partial coverage of textured areas this is not a problem for our method.

6 LIMITATIONS

The example application in this paper is a proof-of-concept and as such lacks the ease-of-use and robustness expected from professional modeling applications. It has mainly been developed as a way to demonstrate the possibilities of using an underlying optimization engine while modeling real-world objects, while retaining all the benefits of having an accurate topology, and it more closely resembles modern modeling practices than previous methods.

Since our optimization routine is based on multi-view stereo, and even though we try to mitigate its inherent problems by using as much user input as possible, we still inherit the native problems of this technique. For example, large untextured regions will be problematic, and surfaces with highlights and other non-lambertian effects will be hard to reconstruct accurately.

Objects with much self-occlusion will also be challenging due to the number of vantage points required to fully cover its surface. This is, however, not unique to our method, and since our method is incremental, we can add new images to an ongoing reconstruction if needed without having to start over from scratch.

In general our method has an advantage over automatic scanning methods when it comes to handling problematic areas, since the user, at all stages of the reconstruction, has the possibility to manually adjust the geometry and lock such parts of the geometry when considered correct.

7 CONCLUSIONS AND FUTURE WORK

We present a proof-of-concept application for user-assisted 3D reconstruction using multi-view stereo. Our novel approach takes topology into account and incorporates information that the user

Figure 5: The difference in topology when using our tool (left three, from coarse to fine subdivison levels) and when using Meshroom (right-most images). Notice the high frequency, arbitrary topology shown in the insets of these images.

provides, to get accurate visibility and geometrical constraints. This enables our final global multi-view stereo optimization to work under more ideal conditions not easily achieved through a more automatic procedure.

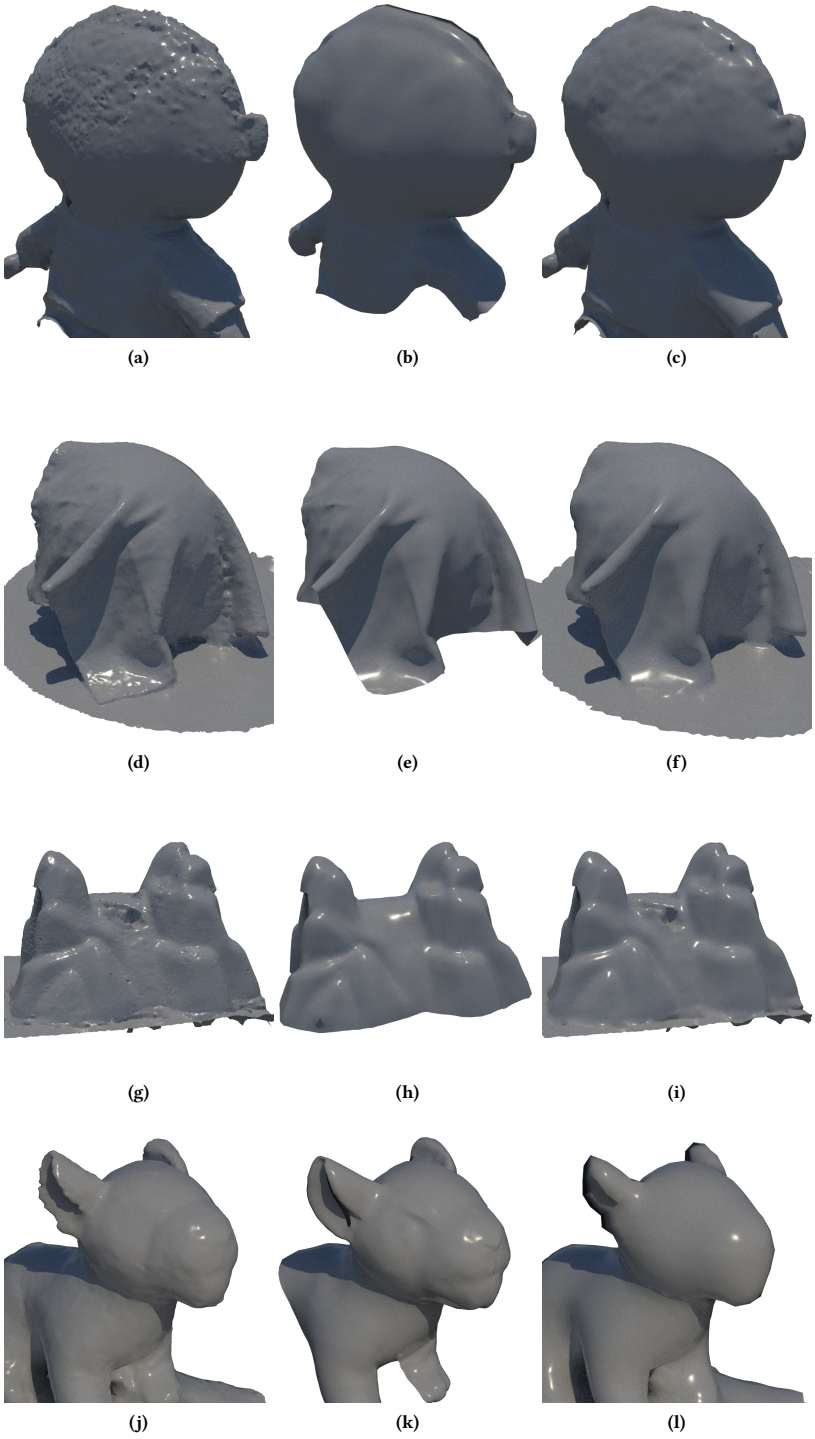

Figure 6: Comparison of quality between our reconstruction and the reconstruction by Meshroom. The left-most images show the original Meshroom reconstruction. The middle images show our reconstruction after two subdivisions. The right-most images show the reconstruction from Meshroom after resampling.

ACKNOWLEDGMENTS

This work was supported by the Swedish Research Council under Grant 2014-4559.

REFERENCES

AliceVision. 2018. Meshroom: A 3D reconstruction software. https://github.com/alicevision/meshroom

Thabo Beeler, Bernd Bickel, Paul Beardsley, Bob Sumner, and Markus Gross. 2010. High-quality Single-shot Capture of Facial Geometry. *ACM Trans. Graph.* 29, 4, Article 40 (July 2010), 9 pages. https://doi.org/10.1145/1778765.1778777

Simone Bianco, Gianluigi Ciocca, and Davide Marelli. 2018. Evaluating the Performance of Structure from Motion Pipelines. *Journal of Imaging* 4, 8 (2018). https://doi.org/10.3390/jimaging4080098

Michael Bleyer, Christoph Rhemann, and Carsten Rother. 2011. PatchMatch Stereo - Stereo Matching with Slanted Support Windows, In BMVC. https://www.microsoft.com/en-us/research/publication/patchmatch-stereo-stereo-matching-with-slanted-support-windows/

E. Catmull and J. Clark. 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design* 10, 6 (1978), 350 – 355. https://doi.org/10.1016/0010-4485(78)90110-0

Tao Chen, Zhe Zhu, Ariel Shamir, Shi-Min Hu, and Daniel Cohen-Or. 2013. 3Sweepp: Extracting Editable Objects from a Single Photo. *ACM Trans. Graph.* 32, 6, Article 195 (Nov. 2013), 10 pages. https://doi.org/10.1145/2508363.2508378

Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. 2015. High-quality Streamable Free-viewpoint Video. *ACM Trans. Graph.* 34, 4, Article 69 (July 2015), 13 pages. https://doi.org/10.1145/2766945

Yotam Doron, Neill D. F. Campbell, Jonathan Starck, and Jan Kautz. 2015. User Directed Multi-view-stereo. In *Computer Vision - ACCV 2014 Workshops*, C.V. Jawahar and Shiguang Shan (Eds.). Springer International Publishing, Cham, 299–313.

Martin Habbecke and Leif Kobbelt. 2009. An Intuitive Interface for Interactive High Quality Image-Based Modeling. *Computer Graphics Forum* 28, 7 (2009), 1765–1772. https://doi.org/10.1111/j.1467-8659.2009.01553.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2009.01553.x

R. I. Hartley and A. Zisserman. 2004. *Multiple View Geometry in Computer Vision* (second ed.). Cambridge University Press, ISBN: 0521540518.

Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 559–568. https://doi.org/10.1145/2047196.2047270

R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. 2011. KinectFusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. 127–136. https://doi.org/10.1109/ISMAR.2011.6092378

Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization* (second ed.). Springer, New York, NY, USA.

Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. 2006. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 1. 519–528. https://doi.org/10.1109/CVPR.2006.19

Sudipta N. Sinha, Drew Steedly, Richard Szeliski, Maneesh Agrawala, and Marc Pollefeys. 2008. Interactive 3D Architectural Modeling from Unordered Photo Collections. *ACM Trans. Graph.* 27, 5, Article 159 (Dec. 2008), 10 pages. https://doi.org/10.1145/1409060.1409112

Noah Snavely, Steven M. Seitz, and Richard Szeliski. 2006. Photo Tourism: Exploring Photo Collections in 3D. *ACM Trans. Graph.* 25, 3 (July 2006), 835–846. https://doi.org/10.1145/1141911.1141964

Richard Szeliski. 2010. *Computer Vision: Algorithms and Applications* (1st ed.). Springer-Verlag, Berlin, Heidelberg.

Anton van den Hengel, Anthony Dick, Thorsten Thormählen, Ben Ward, and Philip H. S. Torr. 2007. VideoTrace: Rapid Interactive Scene Modelling from Video. *ACM Trans. Graph.* 26, 3, Article 86 (July 2007). https://doi.org/10.1145/1276377.1276485

Mingliang Xu, Mingyuan Li, Weiwei Xu, Zhigang Deng, Yin Yang, and Kun Zhou. 2016. Interactive Mechanism Modeling from Multi-view Images. *ACM Trans. Graph.* 35, 6, Article 236 (Nov. 2016), 13 pages. https://doi.org/10.1145/2980179.2982425

Z. Zhang. 2000. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 11 (11 2000), 1330–1334. https://doi.org/10.1109/34.888718

Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, and Marc Stamminger. 2014. Real-time Non-rigid Reconstruction Using an RGB-D Camera. *ACM Trans. Graph.* (2014). https://doi.org/10.1145/2601097.2601165