

# High Resolution 3D Content Creation using Unconstrained and Uncalibrated Cameras

Hoang Minh Nguyen, Burkhard C. Wünsche, Patrice Delmas,  
Christof Lutteroth, Wannes van der Mark

Department of Computer Science, University of Auckland, New Zealand  
hngu039@aucklanduni.ac.nz, burkhard@cs.auckland.ac.nz, p.delmas@cs.auckland.ac.nz,  
lutteroth@cs.auckland.ac.nz, w.vandermark@auckland.ac.nz

**Abstract**—An increasing number of applications require 3D content. However, its creation from real-world data either necessitates expensive equipment, artistic skills, or is constrained, for example, by the range of the utilized sensors. Image-based modeling is rapidly increasing in popularity since cameras are very affordable, widely available, and have a wide image acquisition range suitable for objects of vastly different size. The technique is especially suitable for mobile robotics involving low cost equipment and robots with a light payload, for example, small UAVs. In this paper we describe a novel image-based modeling system, which produces high-quality 3D content automatically from a collection of unconstrained and uncalibrated 2D images. The system estimates camera parameters and a 3D scene geometry using *Structure-from-Motion* (SfM) and *Bundle Adjustment* techniques. The point cloud density of 3D scene components is enhanced by exploiting silhouette information of the scene. This hybrid approach dramatically improves the reconstruction of objects with few visual features, for example, unicolored objects, and improves surface smoothness. A high quality texture is created by parameterizing the reconstructed objects using a segmentation and charting approach which also works for objects which are not homeomorphic to a sphere. The resulting parameter space contains one chart for each surface segment. A texture map is 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.

## I. INTRODUCTION

A key task in mobile robotics is the exploration and mapping of an unknown environment using the robot's sensors. SLAM algorithms can create a map in real time using different sensors. While the resulting map is suitable for navigation, it usually does not contain a high quality reconstruction of the surrounding 3D scene, e.g., for use in virtual environments, simulations, and urban design.

High quality reconstructions can be achieved using image input and conventional modeling systems such as Maya, Lightwave, 3D Max or Blender. However, the process is time-consuming, requires artistic skills, and involves considerable training and experience in order to master the modeling software.

The introduction of specialized hardware has simplified the creation of models from real physical objects. Laser scanners can create highly accurate 3D models, but are expensive and have a limited range and resolution. RGB-D sensors, such as the Kinect, have been successfully used for creating large scale reconstructions. In 2011 the *Kinect-*

*Fusion* algorithm was presented, which uses the Kinect depth data to reconstruct a 3D scene using the Kinect sensor like a handheld laser scanner [28]. Since then a wide variety of new applications have been proposed such as complete 3D mappings of environments [16]. The Kinect is very affordable, but has a very limited operating range (0.8 - 3.5 m), a limited resolution and field-of-view, and it is sensitive to environmental conditions [30].

Reconstruction 3D scenes from optical sensor data has considerable advantages such as the low price of sensors, the ability to capture objects of vastly different size, and the ability to capture highly detailed color and texture information. Furthermore optical sensors are very light weight and have a low energy consumption, which makes them ideal for mobile robots, such as small Unmanned Aerial Vehicles (UAVs).

This paper proposes a novel system that employs a hybrid multi-view image-based modeling approach coupled with a surface parameterization technique as well as surface and texture reconstruction for automatically creating a high quality reconstruction of 3D objects using uncalibrated and unconstrained images acquired using consumer-level cameras. This makes the technique particularly suitable for recognizance and surveillance, e.g., producing 3D reconstructions with high resolution texture maps from video data of miniature UAVs.

One key challenge is that reconstructing 3D scenes from a sequence of images requires knowing where each photo was taken and into what direction the camera was pointing (extrinsic parameters), as well as the internal camera settings, such as zoom and focus (intrinsic parameters), which influence how incoming light is projected onto the retinal plane. Our algorithm automatically estimates the intrinsic and extrinsic parameters of the camera being used and computes the 3D coordinates of a sparse set of points in the scene. This is accomplished using *Structure-from-Motion* and *Bundle Adjustment* techniques.

In order to deal with feature-poor objects, additional 3D points are extracted and added by exploiting the silhouette information of the object. The benefit of integrating shape-from-silhouette and shape-from-correspondence approaches is that the new hybrid system is capable of handling both featureless objects and objects with concave regions. These classes of objects often pose great difficulty for algorithms

using only a single approach. As the result, our solution is able to produce satisfactory results for a much larger class of objects.

The system performs 3D reconstruction using the following steps:

- 1) Camera parameter estimation and scene geometry generation
- 2) Increase the density of the obtained point cloud by exploiting objects' silhouette information
- 3) 3D surface reconstruction
- 4) Surface parameterization and texture reconstruction

The remainder of this paper is structured as follows. In section II, we review related work in the field of image-based modeling. Section III presents the design of our solution. Results are discussed in section IV. In section V we conclude the paper and discuss directions for future research.

## II. RELATED WORK

3D image-based reconstruction algorithms can be classified and categorized based on the visual cues used to perform reconstruction, e.g., silhouettes, texture, shading or correspondence. Amongst them, shape-from-silhouette and shape-from-correspondence have proven to be the most well-known and successful visual cues. Classes of reconstruction methods exploiting these visual cues can offer a high degree of robustness due to their invariance to illumination changes [17].

Shape-from-silhouette algorithms obtain the 3D structure of an object by establishing an approximate maximal surface, known as the visual hull, that progressively encloses the actual object. Shape from silhouette-based methods can produce surprisingly good results with a relatively small number of views and have been proven to be stable with regard to object surface properties (color, texture and material). Silhouette-based methods, however, are very limited in the object geometries they can handle, such as concave regions [13], [26], [29].

The first attempt of obtaining a 3D representation of a scene by exploiting the silhouette information was made by Baumgart in 1974. In his pioneering work Baumgart [2] employed the silhouette contours extracted from four input images to derive the 3D structure of a baby doll and a toy horse. Following Baumgart's footsteps, many different variations of silhouette-based paradigm have been studied and proposed.

Aggarwal et al. [25] presented a method that used an intensity threshold-based segmentation method to separate the object foreground and background in each input image. A connected component analysis of the segmented image produces the silhouette. In order to compute the intersection of different silhouette cones, the authors use a run-length encoded uniformly discretized volume.

Chien et al. [8] improve the efficiency of Aggarwal's method by exploiting alternative data structures such as an octree representation for storing the in/out status of voxels. The method uses three images acquired from orthogonal viewing directions, and converts them to three quadtree

representations, which are fused into an octree representation of the visual hull. The main drawback of this method is the requirement of strict orthogonality of camera viewing directions, which limits practical applications.

Grauman et al. [15] use a Bayesian approach to compensate for errors introduced as the result of false segmentation. This method has been shown to produce excellent error-compensated models from erroneous silhouette information. Its main disadvantage is that it requires prior knowledge about the objects to be reconstructed and large ground-truth training data. This makes the method impractical for real-world applications. Cheung et al. [8], [6], [7] improve the reconstruction quality of the visual hull by aligning multiple silhouette images of a moving object over time.

In recent years, various multi-view reconstruction techniques have been explored. However, most of these methods were designed to tackle a particular class of objects. As a result, their applicability in real-world applications is often limited.

Früh et al. [14] use a combination of aerial imagery, ground color, and LIDAR scans to create textured 3D models of an entire city. While the proposed method produces visually acceptable results, it suffers from a number of drawbacks that render it impractical for general use. In particular, the method requires a vehicle equipped with fast 2D laser scanners and a digital camera to acquire texture data for an entire city at the ground level and a LIDAR optical remote sensor. Additionally, the required manual selection of features and the correspondence in different views is very tedious, error-prone, and cannot be scaled up well.

Xiao et al. [38] presented a semi-automatic image-based approach to reconstruct 3D façade models from a sequence of photographs. The method uses a recursive subdivision scheme to partition façades into small segments, while still preserving their architectural structure. Users are required to provide feedback on the façade partition.

Another semi-automatic image-based technique was presented by Quan et al. for modeling plants [32]. The algorithm performs segmentation both in image space (by manually selecting areas in input images) and in 3D space. Using the segmented images and 3D data, the geometry of leaves is recovered by fitting a deformable model. Users are required to provide hints on segmentation. Branches are modeled through a simple user interface. The main disadvantage of this method is that it requires full coverage of the observed model (360° capture), which may not always be possible in practice.

## III. DESIGN

### A. Algorithm Overview

In order to recover the scene geometry, the system automatically detects and extracts points of interest such as corners (edges with gradients in multiple directions) in the input images. The points are matched across views and changes of their relative position across multiple images are used to estimate camera parameters and 3D coordinates of

the matched points using a *Structure from Motion* technique. The method requires that input images partially overlap.

Feature matching is achieved using an incremental approach starting with a pair of images having a large number of matches, but also a large baseline. This is to ensure that the 3D coordinates of observed points are well-conditioned. The remaining images are added one at a time ordered by the number of matches [5], [36]. The Bundle Adjustment technique is subsequently applied to refine and improve the solution.

The density of the obtained scene geometry is enhanced by exploiting the silhouette information in the input images. The end result of this stage is a dense point cloud of the scene to be reconstructed. A 3D surface mesh is obtained by interpolating the 3D point cloud. The surface is then parameterized and a texture map is obtained by back projecting the input images and fusing them together using graph-cut techniques.

Figure 1 summarizes the stages of our image-based modeling system.

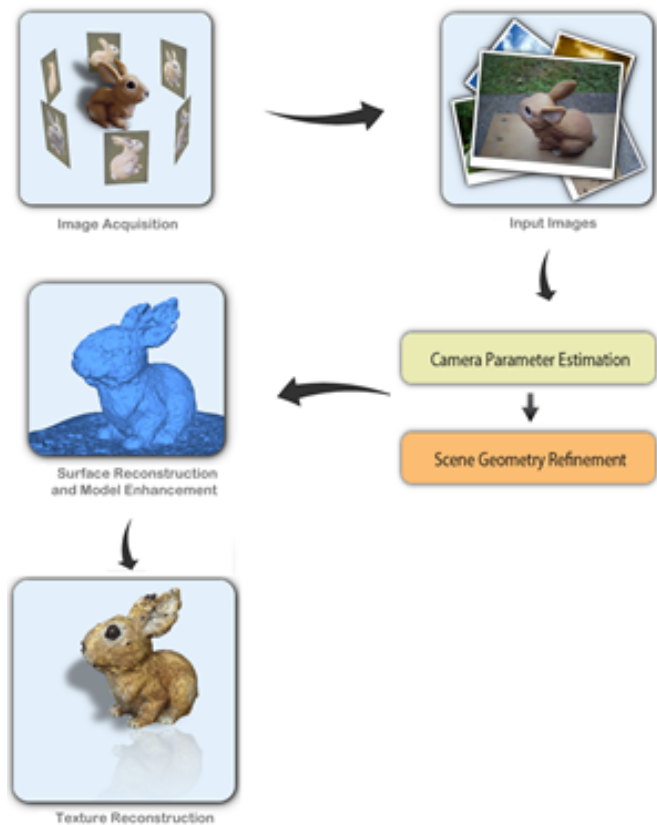


Fig. 1. Stages of the reconstruction process.

### B. Camera Parameter Estimation

The objective of this stage is to recover the intrinsic and extrinsic parameters of each view. This is accomplished in two steps: First, salient features are extracted and matched across views. Second, the camera parameters are estimated

using *Structure-from-Motion* and *Bundle Adjustment* techniques.

In our system we use the SIFT feature detector [23], which transforms an image into a large set of local feature vectors, called SIFT features or SIFT keys. These features are high-dimensional vectors that represent local image measurements. Features extracted by SIFT are also invariant to image transformations, and partially invariant to illumination changes, noise, and camera positions. The features are information rich, which makes it possible to correctly match any single feature against a large database of features with high probability [23], [22].

Once features have been detected and extracted from the input images, they are matched in order to find pairwise correspondences between them. This is achieved by using a distance metric to compute the similarity of each feature of a candidate image with features of another image. A small distance signifies that the two key points are close and thus similar. However, a small distance does not necessarily mean that the points represent the same feature. For instance, the corners of windows of a building look similar regardless of whether two photos show the same or different parts of the building. In order to accurately match a key point in the candidate image, we identify the closest and second closest key point in the reference image using a nearest neighbor search strategy. If their ratio is below a given threshold, the key point and the closest matched key point are accepted as correspondences, otherwise the match is rejected [23], [22].

At this stage, we have a set of potentially matching image pairs, and for each pair, a set of individual feature correspondences. For each pair of matching images, we compute Fundamental matrices using RANSAC. Erroneous matches are eliminated by enforcing a geometric consistency, which is known as *the epipolar constraint*. The epipolar constraint requires that a pair of corresponding features in two images,  $(x_1, y_1)$  and  $(x_2, y_2)$ , satisfies the equation

$$\begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix} F \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = 0 \quad (1)$$

where  $F$  denotes the *Fundamental matrix*, which defines a bilinear constraint between the coordinates of corresponding image points. Matched features that do not agree with the epipolar constraint are discarded.

Given a set of matching images, the system estimates the scene geometry and the motion information of the camera simultaneously using the *Structure from Motion* technique [35], [37], [5]. Once the structure of the scene and the motion information have been estimated, they are further refined using *Bundle Adjustment*.

The process begins with a pair of images being selected. To improve the robustness of the process, this initial pair must have a large number of matches, yet also possess a relatively large baseline (the distance between camera optical centers). Once the initial pair is selected, its *Essential matrix* is approximated using the *five-point algorithm*. The projection matrix can be then recovered by decomposing the

obtained *Essential matrix*. Feature tracks visible in the two images are then triangulated, producing an initial set of 3D points.

We then use an iterative process where we add in each step the  $n$  images with the largest number of shared feature tracks, whose 3D locations have already been estimated. In our system the default is  $n = 3$ . Each new added image is initialized with the same orientation, and focal length as the image that it matches best. This has proved to work very well even though images have different rotation and scale. *Bundle Adjustment* is then used to refine the solution. This procedure is repeated until no more images can be added.

Figure 2 demonstrates several stages of the SfM algorithm.

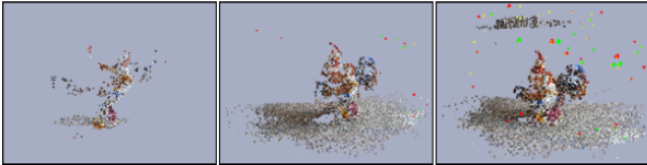


Fig. 2. Several stages of the *Structure from Motion* process for the Rooster dataset. Left: the initial two-frame reconstruction. Middle: an intermediate stage after 20 images have been added. Right: the final result using 42 images.

### C. Scene Geometry Enhancement

At this stage, we have successfully acquired both the camera parameters and the scene geometry. Due to the sparseness of the scene geometry, the surface and texture reconstruction frequently produce artifacts. Most previous works approached this problem by constraining the permissible object types or requiring manual hints for the reconstruction process. However, these requirements breach our goal of creating an easy-to-use system capable of reconstructing any type of object where shape and texture properties are correctly captured by the input photos.

We improve the reconstruction results by exploiting the silhouette information to further enrich the density of the point cloud: First, the silhouette information in each image is extracted using the *Marching Squares algorithm* [21], which produces a sequence of all contour pixels. To construct a visual hull representation of the scene using an immense silhouette contour point set will inevitably increase computational costs. In order to avoid this, the silhouette contour data is converted into a 2D mesh using a *Delaunay triangulation*, and the mesh is simplified using a mesh decimation algorithm [27]. This effectively reduces the number of silhouette contour points.

Each set of silhouette contour points together with the camera parameters of that view form a viewing cone. The visual hull of the scene can be created by a series of 3D constructive solid geometry intersections of these viewing cones. However, these 3D geometry intersections are non-trivial and often computational expensive. Matusik et al. have shown that equivalent results can be obtained using a ray-casting technique [26]. The intersection of two viewing cones

$A$  and  $B$  can be then calculated as follows: each viewing ray of  $A$  (a ray that passes through the camera center and a silhouette point of  $A$ ) is projected onto silhouette  $B$  forming a 2D line. This 2D line will intersect the silhouette contours of  $B$  at certain points. These intersected points are then lifted to 3D using a ray-tracing process yielding a set of 3D points, which defines a face of the polyhedral visual hull.

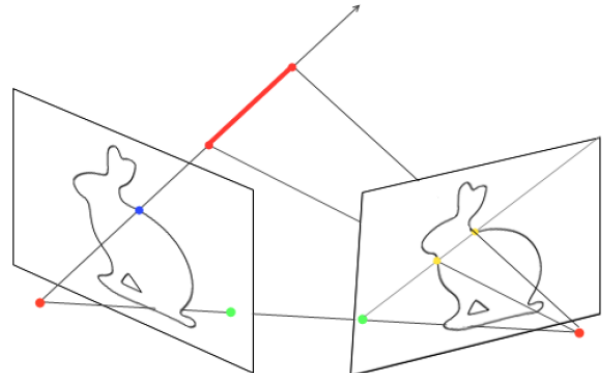


Fig. 3. Lifting 2D silhouette points to 3D points.

This lifting process is achieved by shooting a ray from the camera center of  $B$  through each of the intersection point and computing the intersection point of this newly created ray and the original ray. Figure 3 illustrates the process. In practice two rays rarely meet at a single 3D point due to noise and numerical errors. We hence use a least-squares approximation instead. The 3D points generated during this stage are merged together with the point cloud from the previous stage forming a more comprehensive geometric representation of the observed scene. Figure 4 shows the point cloud of our *General* dataset before and after enhancement.



Fig. 4. Left: The point cloud obtained using only the shape-from-correspondence approach. Right: the results obtained by integrating the shape-from-silhouette approach.

#### D. Surface Reconstruction

At this stage we have successfully obtained a quasi-dense 3D point cloud. The next step is to construct surfaces for the point cloud. Our objective is to find a smooth closed surface (without holes) that accurately approximates the underlying 3D models from which the point clouds were sampled. We tested several surface reconstruction techniques including the power crust algorithm [1],  $\alpha$ -shapes [11], and the ball-pivoting algorithm [3]. We decided to employ the *Poisson Surface Reconstruction* algorithm [19], since it produces a closed surface and works well for noisy data. In contrast to many other implicit surface fitting methods, which often segment the data into regions for local fitting and then combine these local approximations using blending functions, Poisson surface reconstruction processes all the sample points at once, without resorting to spatial segmentation or blending [19]. Figure 5 demonstrates an example.

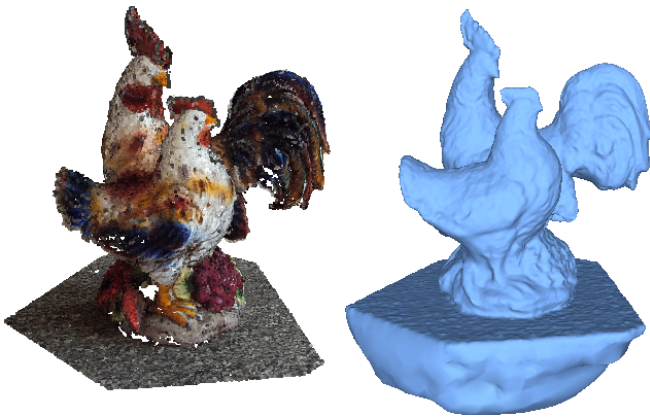


Fig. 5. The input dense 3D point cloud (left) and the resulting smooth surface reconstructed with the Poisson surface reconstruction (right).

#### E. Texture Reconstruction

A high-resolution texture for the reconstructed 3D object is obtained by parameterizing the 2D mesh and computing a texture map.

*a) Surface Parameterization:* We tested surface parameterization algorithms provided by existing libraries and tools, such as Blender. We found that they either required manual hints, only worked for objects homeomorphic to a sphere, or created a surface parameterization using many disconnected patches. The latter result is undesirable since it creates visible seams in the reconstructed texture, and since it makes postprocessing steps, such as mesh reduction, more difficult.

In order to use the resulting 3D models in a large variety of applications and professional production pipelines, we need a texture map which consists of a small number of patches, which ideally correspond to geometric features (which can be maintained in a postprocessing step such as mesh reduction). The *Feature-based Surface Parameterization* technique by Zhang et al. fulfills these criteria [39]. The algorithm consists of three stages:

**1. Genus reduction:** In order to identify non-zero genus surfaces, a surface-based *Reeb* graph [33] induced by the *average geodesic distance* [18] is constructed. Cycles in the graph signify the existence of handles/holes in the surface, i.e., the surface is not homomorphic to a sphere. Examples are donut and teacup shaped objects. The genus of the surface is reduced by cutting the surface along the cycles of the graph. The process is repeated until there are no more cycles.

**2. Feature identification:** Tips of surface protrusions are identified as leaves of the *Reeb* graph. The features are separated from the rest of the surface by constructing a closed curve.

**3. Patch creation:** The previous two steps segment the surface into patches which are homomorphic to a disk. Patches are “unwrapped” using *discrete conformal mappings* [10]. The algorithm first positions the texture coordinates of the boundary vertices, and then finds the texture coordinates of the interior vertices by solving a closed form system. Distortions are reduced by using a post-processing step, which optimizes the position of interior vertices’ texture coordinates by first computing an initial harmonic parameterization [12] and then applying a patch optimization technique [34].

Figure 6 illustrates the resulting parameterization of our *Rooster* model. Each disk in the 2D texture map corresponds to a surface segment of the 3d model.

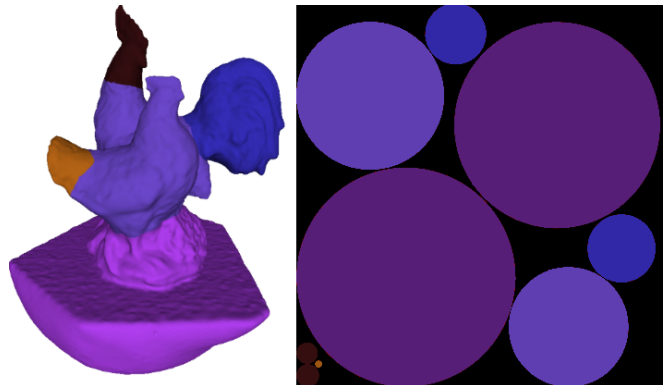


Fig. 6. The *Rooster* model segmented into patches (left) and the corresponding regions in the texture map (right).

*b) Texture Generation:* The texture map for the parameterized surface is computed in three steps:

**1. Identify regions of input images:** The objective of this step is to compute for each patch of the texture map (the disks in the right-hand side image of figure 6) pixel colors, which accurately represent the surface colors of the 3D object at the corresponding points. This is achieved by projecting the corresponding surface patch, one triangle at a time, onto all input images where it is visible. We call the resulting section of the input image the *backprojection map* and we call the resulting mapping between surface triangles and input image regions the *backprojection mapping*. The projection is only performed if the angle between a triangle’s normal and the ray shooting from the triangle’s centroid to the estimated camera position of the input image is larger

than  $90^\circ$ .

**2. Texture map computation:** The image regions defined by the backprojection map define the color information for the corresponding patch of the texture map. Using *backprojection mapping* and the surface parameterization we can compute for each triangle of the surface mesh a mapping from the input image to the texture’s parameter space. The algorithm is repeated for all patches of the reconstructed surface texture region and yields a set of overlapping textures covering the object.

**3. Minimize seams between overlapping textures:** Seams between overlapping textures are minimized by using a *graphcut algorithm* [20]. We investigated different parameters settings for image fusion applications and found that Kwatra et al.’s cost function (gradient weighted color distance) in combination with the RGB color space and the  $L_2$  norm works well for most applications [9].

Figure 7 shows the texture map obtained by backprojection 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 forgot to make photos of the underside of objects. In this case the 3D point cloud contains large gaps. The Poisson surface reconstruction will still create a smooth surface interpolating the gaps, but the corresponding regions of the texture map have no color information. In figure 7 these regions are indicated in red.

Figure 8 illustrates the level of detail obtainable with our texture reconstruction process.



Fig. 7. The texture map obtained by backprojection surface patches onto the input images (right) and the resulting textured 3D model (left). Regions that were not visible in any of the input images are colored red.

#### IV. RESULTS

We tested our image-based modeling system using more than 40 data sets of both indoor and outdoor scenes, and of objects of different scale. Our system produces qualitatively good results for both uniformly colored and feature-poor objects, and for objects with concave 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 a simple consumer-level hand-held camera, including a Smartphone camera. Our systems

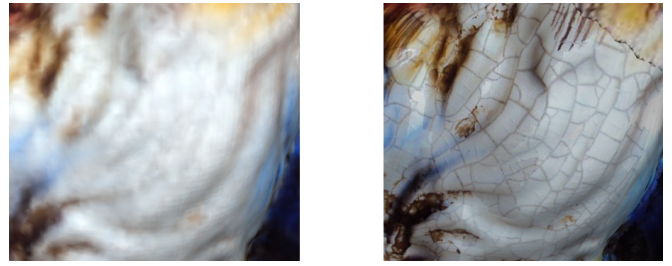


Fig. 8. Texture reconstruction by computing vertex colors and interpolating them (left) and the texture obtained using our approach (right). Note that both images show the neck section of the rooster in figure 7. The cracks in the image on the right reflect accurately the appearance of the object’s material.

fails for objects which have viewpoint dependent surface appearance, e.g., refractive and reflective materials within complex environments. The following paragraphs present three examples of our results.



Fig. 9. Three out of 37 input images of the *horse model* data set.



Fig. 10. 3D reconstruction obtained with the *horse model* data set illustrated in figure 9.

### A. Horse Model

The dataset consists of 37 images of a wooden horse model. The images were acquired outdoors on a sunny day and have a resolution of  $2592 \times 1944$  pixels. Three of the images are shown in figure 9. The original object has a very smooth, reflective and shiny surface with few distinctive visual features.

The resulting reconstructed model, shown in figure 10, is of excellent quality and bears a high resemblance to the original object. The resulting model consist of 329,275 polygons.



Fig. 11. Two out of 27 input images of the *miniature house model* data set.



Fig. 12. 3D reconstruction obtained with the *miniature house model* data set illustrated in figure 11.

### B. Miniature House Model

This dataset, depicted in Figure 11, consists of 27 images of a replica of the famous house in Alfred Hitchcock’s movie “Psycho”. The images have a resolution of  $2592 \times 1944$

pixels and were acquired with a consumer-level SONY DSC-W180 camera under complex lighting condition (multiple spotlights and diffuse lights). The model’s surface has a complex shape with many small features and holes.

The resulting reconstructed object, shown in figure 12, consists of 208,186 polygons and has an acceptable visual quality. The detailed fence-like structure on top of the roof and the tree leaves could not be accurately reconstructed since they were too blurry in the input images. Hence neither the shape-from-correspondence approach, nor the shape-from-silhouette approach could create a sufficiently high number of points for capturing the 3D geometry.

### C. Elephant Model

The *elephant model* dataset in figure 13 (top) consists of 21 images. The images have a resolution of  $2592 \times 1944$  pixels and were acquired with a consumer-level SONY DSC-W180 camera in an indoor environment with relatively low light. The object has a complex surface geometry with many bumps and wrinkles, but few distinctive textural features. The resulting 3D reconstruction, shown in figure 13 (bottom), has 198,857 faces and is of very good quality. The texture and surface geometry of the object contain surprisingly accurate surface details. This example illustrates that our system performs well for objects with dark, rough surfaces and under illumination conditions causing self-shadowing.



Fig. 13. Two out of 21 input images of the *elephant model* data set (top) and the resulting 3D reconstruction (bottom).

## V. CONCLUSION AND FUTURE WORK

We have described a novel image-based modelling system creating high quality 3D models fully automatically from a moderate number (20-40) of camera images. Input images are unconstrained and uncalibrated, which makes the system especially useful for low-cost and miniature mobile robots.

In contrast to laser scanners our system also works for shiny and dark objects.

The system still has some drawbacks which need to be addressed in future research. Missing regions in the texture map occur if the input images do not cover the entire object. We are currently working on texture inpainting techniques to fill these regions [4], [31]. The techniques are likely to fail for large missing patches, such as an inaccessible side of a building. In such instances we propose to use exemplar-based texture synthesis techniques we developed in previous research [24].

## REFERENCES

- [1] N. Amenta, S. Choi, and R. K. Kolluri. The power crust. In *Proceedings of the sixth ACM symposium on Solid modeling and applications (SMA '01)*, pages 249–266. ACM Press, 2001.
- [2] B. G. Baumgart. *Geometric modeling for computer vision*. Doctoral Dissertation, Stanford University, 1974.
- [3] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, Oct. 1999.
- [4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques (SIGGRAPH '00)*, pages 417–424. ACM Press, 2000.
- [5] W. Cheng, W. T. Ooi, S. Mondet, R. Grigoras, and G. Morin. Modeling progressive mesh streaming: Does data dependency matter. *ACM Transaction on Multimedia Computing*, pages 1–24, 2011.
- [6] K.-M. G. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time part i: Theory and algorithms. *International Journal of Computer Vision*, 62(3):221–247, May 2005.
- [7] K.-M. G. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time part ii: Applications to human modeling and markerless motion tracking. *International Journal of Computer Vision*, 63(3):225–245, July 2005.
- [8] C. H. Chien and J. K. Aggarwal. A volume surface octree representation. In *Seventh International Conference on Pattern Recognition, Montreal, Canada*, pages 817–820, 1984.
- [9] X. B. Clark, J. G. Finlay, A. J. Wilson, K. L. J. Milburn, M. H. Nguyen, C. Lutteroth, and B. C. Wünsche. An investigation into graph cut parameter optimisation for image-fusion applications. *Proceedings of the 27th Conference on Image and Vision Computing New Zealand (IVCNZ '12)*, pages 480–485, 2012.
- [10] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (SIGGRAPH '95)*, pages 173–182. ACM, 1995.
- [11] H. Edelsbrunner. Smooth surfaces for multi-scale shape representation. In *Proceedings of the 15th Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 391–412. Springer-Verlag, 1995.
- [12] M. S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14(3):231–250, Apr. 1997.
- [13] J.-S. Franco, M. Lapiere, and E. Boyer. Visual shapes of silhouette sets. In *3D Data Processing, Visualization and Transmission*, pages 397–404, 2006.
- [14] C. Früh and A. Zakhor. Constructing 3d city models by merging ground-based and airborne views. *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2:562–569, 2003.
- [15] K. Grauman, G. Shakhnarovich, and T. Darrell. A bayesian approach to image-based visual hull reconstruction. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 1:187–194, 2003.
- [16] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *Int. J. Rob. Res.*, 31(5):647–663, Apr. 2012.
- [17] C. Hernandez, G. Vogiatzis, and R. Cipolla. Multi-view photometric stereo. In *IEEE Transaction on Pattern Recognition and Machine Intelligence*, 30:548–554, 2008.
- [18] M. Hilaga, Y. Shinagawa, T. Komura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. *Computer Graphics Proceedings, Annual Conference Series (SIGGRAPH 2001)*, pages 203–212, 2001.
- [19] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing (Aire-la-Ville, Switzerland, Switzerland, 2006)*, pages 61–70, 2006.
- [20] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.*, 22(3):277–286, July 2003.
- [21] W. E. Lorenson. Marching through the visible man. In *Proceedings of the 6th conference on Visualization '95*, pages 368–373. IEEE Computer Society, 1995.
- [22] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, 2:1150–1157, 1999.
- [23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, 60:91–110, 2004.
- [24] F. Manke and B. Wünsche. Fast spatially controllable multi-dimensional exemplar-based texture synthesis and morphing. In *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, volume 68, pages 21–34. Springer, 2010.
- [25] V. Martin and J. K. Aggarwal. Volumetric descriptions of objects from multiple views. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):150–158, 1983.
- [26] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques (SIGGRAPH '00)*, pages 369–374. ACM Press, 2000.
- [27] S. Melax. Simple, fast, and effective polygon reduction algorithm. *Game Developer Magazine*, pages 44–49, 1998.
- [28] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Proceedings of the 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR '11)*, pages 127–136, Washington, DC, USA, 2011. IEEE Computer Society.
- [29] H. M. Nguyen, B. Wünsche, P. Delmas, and C. Lutteroth. Realistic 3d scene reconstruction from unconstrained and uncalibrated images. In *Proceedings of GRAPP 2011, Algarve, Portugal*, 31:67–75, 2011.
- [30] A. Oliver, S. Kang, B. C. Wünsche, and B. MacDonald. Using the kinect as a navigation sensor for mobile robotics. In *Proceedings of IVCNZ 2012*, pages 509 – 514, Nov. 2012.
- [31] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, July 2003.
- [32] L. Quan, P. Tan, G. Zeng, L. Yuan, J. Wang, and S. B. Kang. Image-based plant modeling. In *ACM Transactions on Graphics*, 25(3):599–604, 2006.
- [33] G. Reeb. Sur les points singuliers d'une forme de pfaff complètement intégrable ou d'une fonction numérique [on the singular points of a completely integrable pfaff form or of a numerical function]. *Comptes Rendus Acad. Sciences Paris* 222, pages 847–849, 1946.
- [34] P. V. Sander, S. J. Gortler, J. Snyder, and H. Hoppe. Signal-specialized parameterization. *Proceedings of the 13th Eurographics Workshop on Rendering*, pages 87–100, 2002.
- [35] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. In *ACM Transactions on Graphics*, 25(3):835–846, 2006.
- [36] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *ACM Transactions on Graphics*, 25(3):835–846, 2006.
- [37] R. Szeliski. Image alignment and stitching. A tutorial in *Computer Graphics and Vision*, 2006. Available at <http://research.microsoft.com/apps/pubs/default.aspx?id=70092>. Last accessed on May 21<sup>st</sup> 2011.
- [38] J. Xiao, T. Fang, P. Tan, P. Zhao, E. Ofek, and L. Quan. Image-based façade modeling. In *ACM Transactions on Graphics*, 27(5):26–34, 2008.
- [39] E. Zhang, K. Mischaikow, and G. Turk. Feature-based surface parameterization and texture mapping. *ACM Transactions on Graphics (TOG)*, 24(1):1–27, 2005.