# A Hybrid Image-Based Modelling Algorithm

**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

**Abstract**

This paper explores the practical aspects associated with visual-geometric reconstruction of a complex 3D scene from a sequence of unconstrained and uncalibrated 2D images. These image sequences can be acquired by a video camera or a handheld digital camera without the need for camera calibration. Once supplied with the input images, our system automatically processes and produces a 3D model. We propose a novel approach, which integrates uncalibrated Structure from Motion (SfM), shape-from-silhouette and shape-from-correspondence, to create a quasi-dense scene geometry of the observed scene. In the second stage, surface and texture are applied onto the generated scene geometry to produce the final 3D model. The advantage of combining silhouette-based and correspondence-based reconstruction approaches is that the new hybrid system is able to deal with both featureless objects and objects with concaved regions. These classes of objects usually pose great difficulty for shape-from-correspondence and shape-from-silhouette approach. As the result, our approach is capable of producing satisfactory results for a large class of objects. Our approach does not require any a priori information about camera and image acquisition parameters. We tested our algorithm using a variety of datasets of objects with different scales, textures and shapes acquired under different lighting conditions. The results indicate that our algorithm is stable and enables inexperienced users to easily create complex 3D content using a standard consumer level camera.

*Keywords:* image-based modelling, correspondence-based reconstruction, silhouette-based reconstruction

## 1 Introduction

There is an increasing amount of applications that require high quality 3D representations, e.g. for arts, commerce, virtual heritage, training, education, computer games, virtual environments, documentation, exchanging information, and social networking applications. Conventionally, 3D digital models are constructed using modelling tools such as Maya, 3D Max or Blender. Although these applications enable graphic designers to construct highly realistic and complex 3D models, they have a steep learning curve and often require a considerable amount of training and artistic skills to use. These restrictions render them unsuitable for non-professionals. The introduction of specialised hardware, such as laser scanners, has simplified the creation of models from real physical objects. However, while many of these systems deliver highly accurate results, they are usually extremely costly and often have restrictions on the size and surface properties of objects in the scene. Consequently, there is a critical need to improve on the status quo by making 3D content creation available to a wider group of users.

In recent years image-based modelling has emerged as a new approach to simplify 3D content creation. In contrast to traditional geometry-based modelling and hardware-heavy approaches, the image-based modelling techniques confront the formidable, and still unanswered, challenge of creating a comprehensive representation of 3D structure and appearance of a scene from visual information encoded in 2D still images. Image-based modelling techniques are usually less accurate, but offer very intuitive and low-cost methods for recreating 3D scenes and models.

The ultimate goal of our work is to create a low-cost system that allows users to obtain 3D reconstruction of the observed scene using a consumer-grade camera. The idea behind the system is very simple: Once supplied with the input images, our system will automatically process and produce a 3D model without any a priori information about the scene to be reconstructed.

Reconstructing 3D scenes from a collection of 2D photographic images requires knowing where each photo was taken. A common approach to obtain this information is to perform camera calibration manually. However, this method requires a setup and preparations that are usually too sophisticated for inexperienced users. Furthermore, this method places a restriction on the types of scenes that can be reconstructed since it is not always feasible to perform camera calibration for a large and complex scene. For this reason, for each input image, our algorithm needs to automatically estimate the intrinsic and extrinsic parameters of the camera being used and compute the 3D coordinates of a sparse set of points in the scene (the scene geometry or point cloud). Surfaces and texture are then applied to this point cloud to produce the final model.

Due to the sparseness of the scene geometry, problems such as surface artifacts, noise, and blurry textures might arise during the surface and texture reconstruction processes. Most previous works approached these problems by constraining the types of objects and requiring manual user inputs to aid their systems in deducing the structure of the object to be reconstructed. However, these requirements breach our goal of creating an easy-to-use system and providing the capability of reconstructing any type of object. We overcome these problems by using a hybrid approach that integrates shape-from-correspondence and shape-from-silhouette methods. The system will perform 3D reconstruction using the following steps:

1. Camera parameter estimation
2. Initial point cloud generation from extracted key features
3. Increase the density of the point cloud by exploiting silhouette information
4. Reconstruct the object's surface and texture to produce the final model

The remainder of this paper is organised as follows. After a description of the related work on image-based modelling, a discussion about the algorithms used in our system is presented in section 3. Results are discussed in section 4. Section 5 concludes and summarises the paper and gives a brief outlook on directions for future research.

## 2 Related Work

Although there has been much interest and study of 3D modelling techniques over the last few decades, robustly and automatically obtaining 3D models is still a difficult task. For the past 30 years, creation of artiticial 3D models using conventional graphics and animation software such as Maya and 3D Max has continued to be the most popular approach. To this end, various tools have been proposed to assist the human designer by using images of the object to be modelled [THP08]. The reason why manual creation of 3D models remains the prevalent approach despite intensive study and research is that, in fact, there is no computer vision or graphics technique that works for every object. The difficulty in selecting the most suitable 3D reconstruction technique that works for a large class of objects justifies the abundant literature that exists on this subject [Est04, REH06].

Various multiple view reconstruction techniques have been explored in recent years. Amongst them, the most well-known and successful class of techniques have been *shape from silhouette* and *shape from correspondence*.

The shape from silhouette class of algorithms exploits silhouette information to create intersected visual cones, which are subsequently used to derive the 3D structure of an object. Shape from silhouette-based methods are popular for shape estimation due to their good approximation qualities for a large number of objects, their stability with respect to the object's material properties and textures, as well as their ease and speed of implementation. Exploiting silhouette information for 3D reconstruction was first considered by Baumgart in 1974. In his pioneering work [Bau74], the author computed polyhedral shape approximations of a baby doll and a toy horse by intersecting silhouette cones. Following Baumgart's work, many different variations of the shape from silhouette paradigm have been studied and proposed.

Aggarwal *et al.* [MA83] proposed 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 used a run-length encoded, uniformly discretised volume. The idea behind this is to consider each image point as a statistical occupancy sensor. The point observations are then analysed to deduce where matter is located in the observed scene. This is achieved by discretising the scene into three dimensional voxels, which are then projected into silhouette images. The task is to mark the individual voxels as either "in" or "out". If the projection of a voxel belongs to the foreground in all silhouette images, then the voxel is labeled as "in" otherwise it is labeled as "out". If a voxel is labeled as "out", it is excluded from further computations of the object structure.

Matusik *et al.* [MBR$^+$00] improve the efficency of shape from silhouette techniques by taking advantages of epipolar geometry. In their method, 3D silhouette-intersection computation is reduced to 2D by projecting one silhoutte onto another. The intersection is then carried out in image space. Franco *et al.* [FB10] attempted to improve the effciency of Matusik's method and to produce a water-tight model. Their method involves two main steps. First, point clouds of the observed scene are generated by back-projecting viewing edges of the visual hull. Next, missing surface points are recovered by exploiting local orientation and connectivity rules. A final connection walkthrough is then carried out to construct the planar contours for each face of the polyhedron.

In recent years, various correspondence-based techniques have been explored. However, most of these methods were designed to tackle reconstruction problems related to a particular class of objects. As a result, their use is often limited.

Fruh *et al.* [FZ03] used a combination of aerial imagery, ground colour, and LIDAR scan data 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 consumer-level applications. In particular, the method requires intensive use of special hardware during the data acquisition step. This includes 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.* [XFT$^+$08] presented a semi-automatic image-based approach to recover 3D structure of façade models from a sequence of street view images. The method combines a systematic and automatic decomposition scheme of façades for analysis and reconstruction. The decomposition is accomplished by recursively splitting the complete façades into small segments, while still preserving the overall architectural structure. Users are required to provide feedback on façade partitioning. This method demonstrated excellent results.

Quan *et al.* [QTZ$^+$06] presented a method for modelling plants. In their method, segmentation is performed in both image space (by manually selecting areas in input images) and in 3D space. Using the segmented images and 3D data, the geometry of each leaf is recovered by setting a deformable leaf model. Users are also required to provide hints on segmentation. The main disadvantage of this method is that it requires full coverage of the observed model (360 degree capture), which may not always be possible in practice due to obstructions and space limitations. Branches are modelled through a simple user interface.

Tan *et al.* [TZW$^+$06, LQ02] introduced a method for creating 3D models of natural-looking trees from a collection of images. Due to the large leaf count, small image footprint and widespread occlusions, it is not possible to recover accurate geometric representation for each leaf. In order to overcome this problem, the authors populate the manufactured tree with leaf replicas from segmented input images to reconstruct the overall tree shape.

## 3 Algorithms

Our proposed approach uses a coarse-to-fine strategy where a rough model is first reconstructed and then sequentially refined through a series of steps. The approach consists of two main stages: scene geometry extraction and visualisation. Both of these stages are achieved by dividing the main problem into a number of more manageable subproblems, which can then be solved by separate modules. The entire reconstruction process is illustrated in Figure 1.

The objective of the first stage is to recover the scene geometry from the input images. This stage begins with the camera parameters for each view being estimated. This is accomplished by the automatic extraction of distinctive features and establishment of point correspondences in stereo image pairs. We then isolate all matching images, selecting those that view a common subset of 3D points. Given a set of matching images, a scene geometry (point cloud) and camera pose can be estimated simultaneously by Structure from Motion and subsequently refined by Bundle Adjustment. Next, additional points are added to the computed scene geometry by exploiting the silhouette information to produce a more complete geometry. In the final step, a resampling technique is applied onto the point clouds to produce the final scene geometry. In contrast to an initial version of this algorithm [NWDL11] we integrate silhouette information, which is used in the point cloud generation and the surface reconstruction.

In the second stage, we tackle the problem of how to transform the scene geometry recovered in the preceding stage into a realistic representation of the scene. This is accomplished by applying surfaces and texture to the resulting scene geometry. The outcome of this stage is a complete 3D representation of the observed scene.

### 3.1 Camera Parameter Estimation

One key challenge in extracting 3D representation of a scene from a sequence of 2D images is that the process requires knowing where each photo was taken and in what direction the camera was

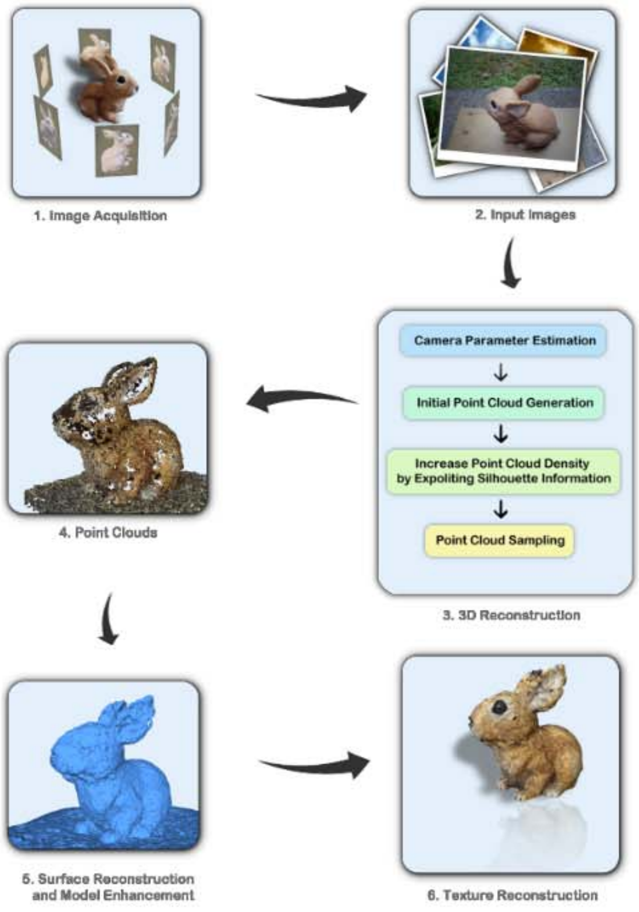
1. Image Acquisition
2. Input Images
Camera Parameter Estimation
Initial Point Cloud Generation
Increase Point Cloud Density by Expoliting Silhouette Information
Point Cloud Sampling
3. 3D Reconstruction
4. Point Clouds
5. Surface Reconstruction and Model Enhancement
6. Texture Reconstruction

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

pointed (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.

In order to recover such information, the system will first detect and extract points of interest such as corners (edges with gradients in multiple directions) in the input images. This is accomplished using the SIFT feature detector [Low06]. Feature points extracted by SIFT are highly distinctive, and invariant to different transformations and changes in illumination, and additionally have a high information content [HL07, BL05]. Once features have been identified and extracted from all the images, they are matched. This is known as the correspondence problem. Given a feature in an image $I_1$, what is the corresponding feature (the projection of the same 3D feature) in the other image $I_2$. This can be solved by using a Euclidean distance function to compare the two feature descriptors. All the detected features in $I_2$ will be tested and the one with minimum distance is selected [HQZH08].

Once all interest points have been found, we match them across views and estimate the camera parameters and 3D coordinates of the matched points simultaneously using the Structure from Motion (SfM) technique. Structure from Motion designates the computation of the camera poses and scene geometry (3D points) simultaneously from a set of images and their feature correspondences. More precisely, SfM can be formulated as an optimisation problem, where the goal is to determine the configuration of cameras and 3D points that, when related through the equations of perspective projection, best agree with the detected feature correspondences. This computation is carried out by exploiting a constraint between correspondences and the physical configuration of the two cameras. This is a powerful constraint, as two 3D rays chosen at random are very unlikely to pass close to one another. Given enough point correspondences between two images, the geometry of the system is sufficiently constrained to determine the camera poses (up to scale). The key to the success of this procedure is that successive images must not vary significantly, i.e. must contain overlapping visual features.

Our solution takes an incremental approach, in which a pair of images is selected to initialise the sequence. This initial pair should have a large number of matches, but must also have a large baseline. This is to ensure that the 3D coordinates of observed points are well-conditioned. The remaining images are added to the optimisation one at a time ordered by the number of matches [REH06, SSS06]. The Bundle Adjustment technique is followed to refine and improve the obtained solution. The accuracy of the reconstruction depends critically on the final step. Figure 2 demonstrates a scene geometry created using Structure from Motion and Bundle Adjustment.

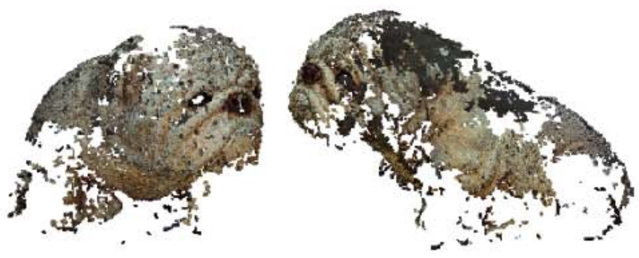
Figure 2: Two views of the scene geometry generated from 37 input images.

## 3.2 Scene Geometry Enhancement

At this stage, we have successfully acquired both camera parameters and scene geometry. As the scene geometry is derived from distintive features of the input images, they are often sparse. In order to produce a more complete and comprehensive model, the obtained scene geometry needs to be enhanced. This can be accomplished by exploiting the silhouette information of the observed scene to generate additional 3D points.

The process is as follows: First, silhouette information of the observed scene in each view is extracted using the Marching Squares algorithm [Lor95] producing sets of silhouette points. Each of these sets represents an exhaustive point-by-point isoline list of every pixel which constitutes a silhouette contour. To define a silhouette using an enormous contour point set will inevitably upsurge the computational expense. To avoid this, silhouette data must be preprocessed to reduce the number of silhouette contour points. We achieve this by first performing a *Delaunay triangulation* of the contour points. The output triangular mesh framework is then fed to a mesh simpification algorithm [Mel98] to decrease the number of triangles, which in turn effectively reduces the number of contour points.

Each set of contour points together with the camera centre of that view defines a *viewing cone*, whose apex is located at the camera's optical centre. Each viewing cone consists of a number of *cone lines*. A cone line represents a 3D line formed by a silhouette contour point and the camera's optical centre. The polyhedral visual hull information can be obtained by calculating the intersection of these viewing cones. However, 3D polygon intersection is non-trivial and often computationally expensive. Matusik *et al.* [MBR$^+$00] proved that equivalent results can be obtained by the following steps:

Assume that we want to compute the intersection of silhoutte $A$ and $B$.

1. Projecting each cone line of the viewing cone $A$ onto the silhoutte $B$.
2. Calculate the intersection of the projected line and the silhouette $B$ in 2D.
3. Lifting the computed intersection points from 2D to 3D yields a set of 3D points, which defines a face of the polyhedral visual hull.

**Projecting a cone line onto another silhouette** As each of these rays (cone lines) is defined by the camera's optical centre and the projecting ray $\overrightarrow{\mathbf{O}_A S_i}$, the projection of a ray onto the image $B$ is computed as follows:

Projecting the camera's optical centre of $A$ onto $B$ (*epipole of B*)

$$\widetilde{e_B} = \mathtt{P}_B \begin{bmatrix} \mathbf{O}_A \\ 1 \end{bmatrix} \tag{1}$$

where $\mathtt{P}_B = [Q_B; \ q_B]$ is the $3 \times 4$ projection matrix of $B$.
Projecting the point $D_i$ at the infinity of the projecting ray $\overrightarrow{\mathbf{O}_A S_i}$ onto $B$

$$\widetilde{d_B} = \mathtt{P}_B \begin{bmatrix} \mathbf{D}_A \\ 0 \end{bmatrix} = \mathtt{P}_B \begin{bmatrix} \mathtt{Q}_A^{-1} S_i \\ 0 \end{bmatrix} \tag{2}$$

These two points define a 2D line on image $B$, which will subsequently be used to compute intersections.

**Line-Silhouette Intersection** The naive approach of computing intersections of a line and a silhouette requires the traversal of each pair of silhouette contour points and performing a line-line intersection with the given line. This method, however, proves to be very inefficient. We use an alternative method [15], which is based on the observation that all projected rays intersect at one common point, the epipole. This makes it possible to subdivide the reference image into partitions so that each projected ray will intersect all the edges and only the edges in that partition. This way we only have to traverse edges in a particular partition when computing intersections. The algorithm to partition the reference image into regions is as follows:

First, the epipole of the reference image $B$ is computed by projecting the camera's optical centre of $A$ onto $B$. Each silhouette contour point along with the epipole forms a line segment. The slopes of all the line segments are stored in a sorted list in ascending order. Two consecutive line segments in the list create a partition or a bin. Each bin is defined by a minimum and maximum slope value ($\alpha_{min}$ and $\alpha_{max}$) and a set of edges associated with it.

To determine the set of edges allocated to each bin, we traverse through the silhoutte contour points while maintaining a list of edges in the current bin. When a contour point is examined, we eliminate from the current bin all edges ending at that contour point and append all edges starting at that contour point. A start of an edge is characterised as the edge end point that has a smaller slope value.

The intersections of a given line and a silhouette can be determined by first examining the slope of the input line segment to establish the bin to which the line segment belongs. Once the corresponding bin is found, we iterate through the edges assigned to that bin and perform line-to-line intersection. The result of this stage is a collection of 2D intersection points.

**2D points to 3D points** Given a 2D intersection point in the reference image, we want to compute the corresponding 3D point of this point. This is accomplished by shooting a ray from the camera's optical centre of the reference image through the intersection point and compute the intersection point of the newly created 3D rays with the original 3D ray. The intersection point of the two 3D ray is the 3D point we want to find. Figure 3 shows an example of the lifting process.

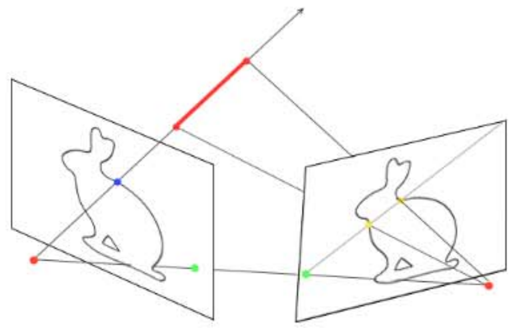

Figure 3: Lifting 2D points to 3D points.

A problem arises when the two rays do not intersect as a result of noise. We can handle this by finding the point with the smallest distance to the two rays using a *Least Square* method. Figure 4 illustrates the newly generated 3D points of the dog dataset. These point clouds together with those of the previous stage form a much more comprehensive geometric representation of the observed scene.

### 3.3 Surface and Texture Reconstruction

The final step is to reconstruct surfaces from the obtained point clouds. Surface reconstruction is the process of automated generation of a smooth surface that closely approximates the underlying 3D models from which the point clouds were sampled. The reconstructed surface is usually represented by a polygon mesh. Many sophisticated surface reconstructions have been proposed and extensively studied. In our system, we employ the Poisson surface reconstruction algorithm [KBH06] for remeshing

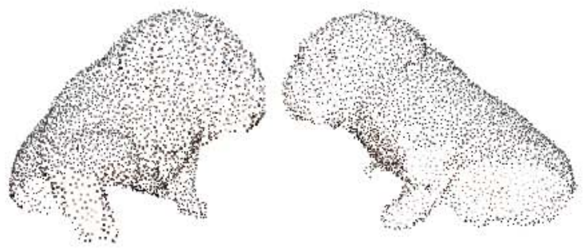

Figure 4: Additional 3D points generated by exploiting the silhouette information.

the surface.

The Poisson reconstruction method [Bol10] extracts surfaces by taking advantage of the integral relationship between oriented points sampled from the surface of an unknown model and the indicator function $\chi$ of the model. The indicator function is defined as 1 at points that lie inside the model and as 0 for points that lie outside the model.

Formally, the problem can be formulated as: Given an oriented point set $P = \{s_1, s_2, ..., s_N\}$, where a point sample $s_i$ consists of a position $p$ and an inward facing normal $n$ and is assumed to lie on or near the surface $\partial M$ of an unknown model $M$. The aim is to create a smooth and watertight approximation to the original surface by computing the indicator function $\chi$ of the model and then extract an appropriate isosurface [Bol10].

This is accomplished by first deriving a relationship between an integral of the normal field over the surface and the gradient of the model's indicator function. The gradient of the indicator function is defined as a vector field that is zero almost everywhere as the indicator function is constant practically everywhere except at points near the surface, where it points in the direction of the inward surface normal. Hence, the input oriented point samples can be thought of as samples of the gradient of the indicator function [Bol10, Kaz05]. An example of Poisson Surface reconstruction is shown in Figure 5.

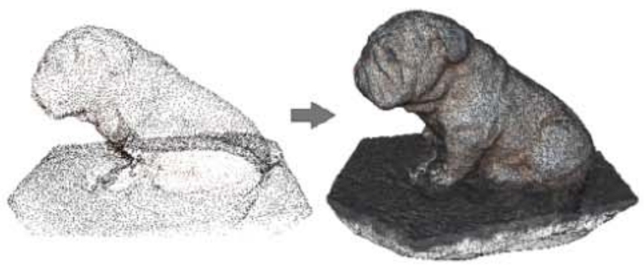

Figure 5: Poisson reconstruction. The input point samples are on the left, while the reconstructed mesh is shown on the right.

After the surface reconstruction phase, the resulting model is a shaded and textureless polygonal mesh. Such a representation is usually not an accurate reflection of the object in the input images. A more realistic representation is obtained by creating a surface texture from the colour information in the input images. In texture reconstruction, each vertex of the polygonal mesh has the RGB colour of the corresponding vertex in the attached point clouds. The resulting triangle mesh with vertex colours is rendered using Gouraud shading. Since the triangle mesh is very dense, the colour interpolation over triangles gives acceptable results.

## 4 Results

We evaluated our system using a wide range of data sets at different scales of both indoor and outdoor scenes. In most test cases, the system produces qualitatively good results. The geometries recovered using our system appear to retain high resemblance to that of the original models even for objects with smooth, uniform and limited-feature surfaces or concave regions. Datasets with both smooth, uniform surfaces and concave regions often resulted in an unsatisfied geometry since the silhouettes did not contain sufficient geometry information, and there were too few unambiguous features to allow full surface reconstruction. The size of our test datasets varies from as few as 6 images to hundreds of images, which were all taken with a simple handheld camera.

**Owl Dataset** The first dataset consists of 32 images (3648 × 2736 pixels) taken from arbitrary view directions using a normal consumer-level SONY DSC-W180 camera of a paper model of an Owl. Three of the 32 input images are shown in Figure 6. This Owl object is highly decorated with texture. On average, each image contains over 53,000 features, which would aid the reconstruction greatly.

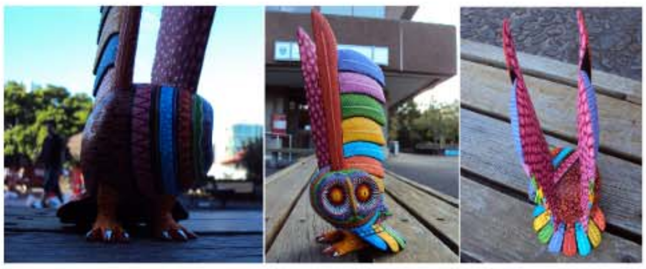
Figure 6: Owl dataset input images.

The resulting reconstructed model, illustrated in figure 7, is of excellent quality. The overall shape, along with details such as feathers, and the texture of the original model are reconstructed with no visual difference to the original model. This is due to the high number of distinct features of the original object. The resulting model has 678,210 faces in total. The process took approximately 3 hours and 30 minutes to complete on an Intel Quad Core i7 with 6GB RAM.

**Lady Statue Dataset** This dataset consists of 17 images taken from the frontside of a black copper statue. The backside was not accessible. The images were taken with the same camera as in the previous case and under very complex lighting conditions.

Some of the images which were used for the reconstruction are shown in Figure 8. The resolution of each image in this dataset is 3648 × 2736 pixels. Notice that the surface of the model contains few texture details.

The reconstructed model has 268,892 faces and is of moderate quality (Figure 9). This result is

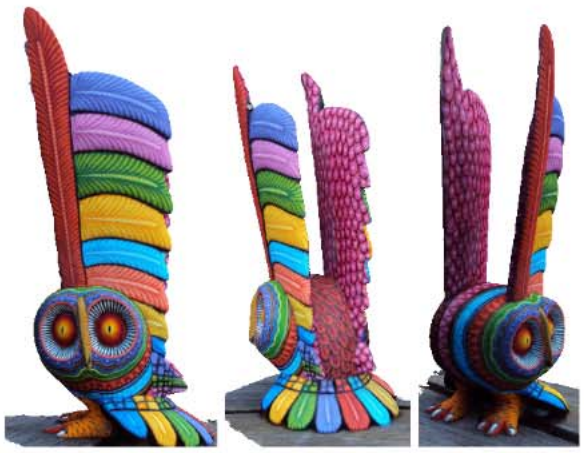
Figure 7: 3D reconstruction of the Owl model.

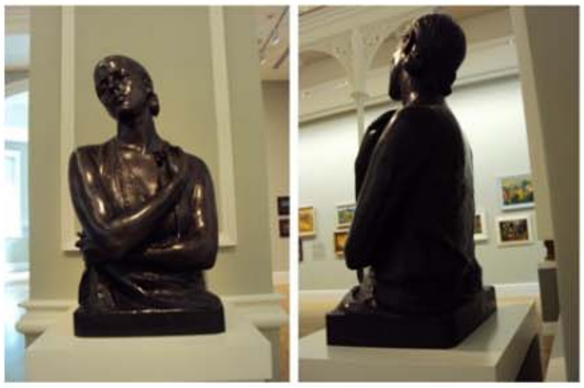
Figure 8: Two of seventeen input images of the Lady model taken at the Auckland Art Gallery.

surprisingly good considering the relatively low number of input images and the lack of distinct visual features for correspondence matching. The geometry was predominantly obtained from the silhouette information. This shows that our system is capable of producing good results for feature-poor models and few input images.

The reconstructed texture shows several significant differences to the original model. This is caused by changes in lighting conditions within the gallary. Some white patches also appear around the head region of the reconstructed model due to the lack of images from that particular direction. The current systems only computes surface colours, rather than material properties, and hence works best for diffuse surfaces and lighting conditions resulting in few isolated highlights. The reconstruction process required approximately 1 hour and 47 minutes on an Intel Quad Core i7 with 6GB RAM.

**Bunny Dataset** This data set comprises 49 images (2592 × 1944 pixels) taken from many different views of a bunny model using a normal consumer-level SONY DSC-W180 camera. The original model has a very bumpy surface, which is extremely difficult to reconstruct (Figure 10). The objective of this test is to determine if the system can effectively deal with rough surfaces and high illumination variations due to surface roughness and self shadowing.

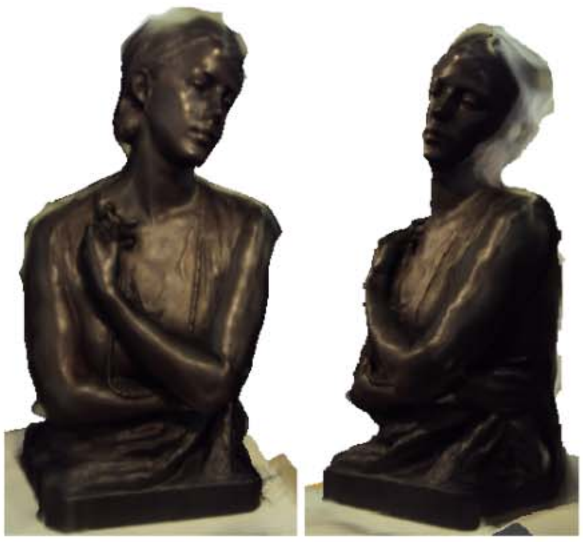
Figure 9: 3D reconstruction of the Lady Statue.

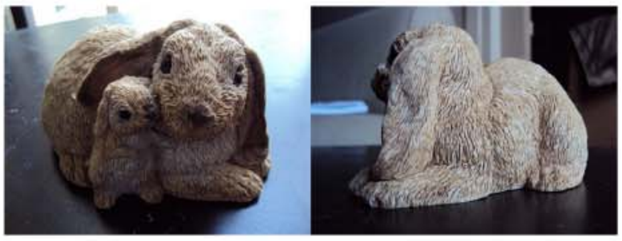
Figure 10: Two of 31 input images of a bunny model.

The reconstruction result (shown in Figure 11) is of very good quality. The final model, which is composed of 528,637 faces, bears a high resemblance to the original object.

This result demonstrates that our approach is able to recover realistic 3D models of shapes with complex surface geometries. The overall computation time is approximately 4 hours and 40 minutes on an Intel Quad Core i7 with 6GB RAM.

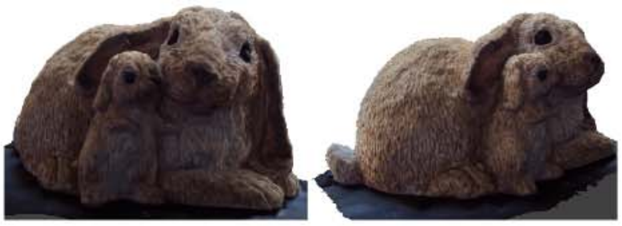
Figure 11: 3D Reconstructed model of the Bunny dataset.

**Cat Dataset** This data set was obtained from a ceramic statue of a cat. There are 44 images with a resolution of 2592 × 1944 pixels. This object has a very shiny, reflective, smooth and uniform surface. These surface characteristices pose a major problem for correspondence-based methods as very few features are available. Added to that, the observed object's surface also contains a number of concave regions, which also cause problems for silhouette-based methods. Three of the 44 input images are presented in Figure 12.

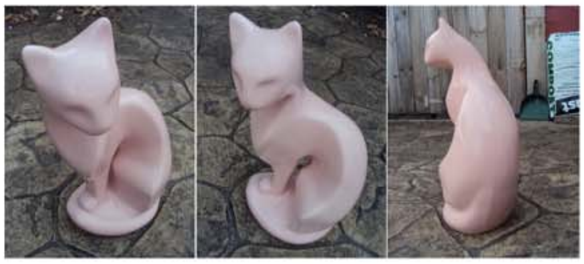
Figure 12: Input images of the Cat dataset.

The visual quality of the reconstruction is not satisfactory. While the object is recognisable, important features such as the eyes and ears are not reconstructed well, and the colour distribution varies from the original object.

This reconstruction results (Figure 13) demonstrate that our approach is able to recover 3D geometry even for models with shiny, reflective and uniform surfaces, but problems exist with concave regions. The overall computation time was approximately 2 hours and 55 minutes on an Intel Quad Core i7 with 6GB RAM.

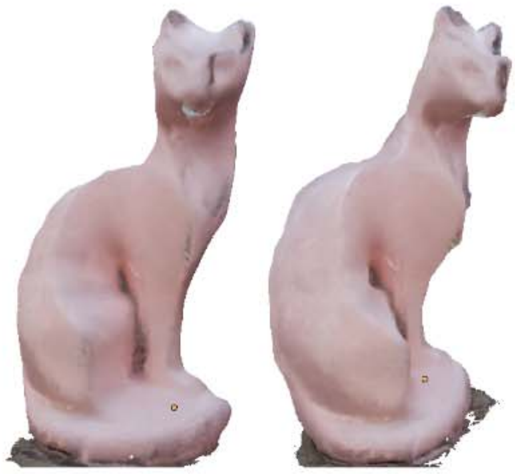
Figure 13: 3D reconstruction of the Cat model.

**Comparison with Commercial Systems**
Over the past 2-3 years a couple of prototypes of commercial systems have emerged, which are currently at various stages of user testing. While none of the commercial systems has published any information about the utilised approach, we performed an analysis, which suggest that most of them use predominantly a silhouette-based approach [NWDL12]. In this section we provide a short comparison of our novel hybrid approach with some of the most promising alternative image-based modelling systems (*123D Catch* and *Agisoft*) currently in development.

In order to evaluate these systems, we used a repository of over 40 objects. After initial tests using different objects we selected one object which reflected the main shortcomings of all tested algorithms. For this test, 44 input images of the above cat model were supplied to these systems. The reconstruction results from these two systems are shown in Figure 14 and 15.

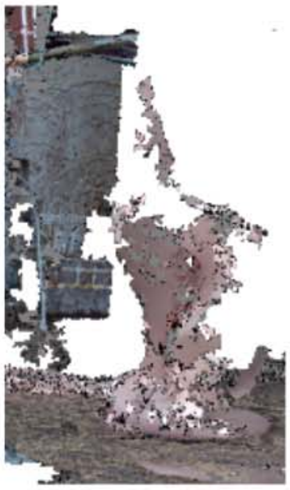

Figure 14: 3D reconstruction of the Cat model using *Agisoft* System.

*Agisoft*'s model is almost unrecognisable. The reconstruction is very incomplete and retains no visual resemblance to the original object. Large portions of the object geometry are missing. This is mostly due to the fact that the system is not able to register views when there is only a limited number of features in the input images. Additionally, the reconstructed textures are also inadequate with many black patches covering the reconstructed surfaces.

*123D Catch* produces more a complete geometry of the reconstructed model. The resulting model still somewhat reflects the structure of the original object. Compared to our reconstruction result, the reconstructed geometry appears much rougher. Regions around the back, the head and the tail of the reconstructed model are mostly distorted.

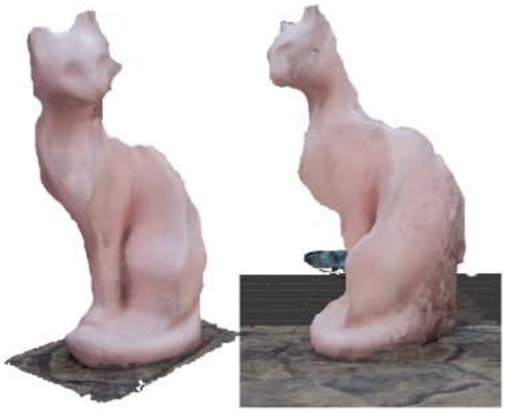

Figure 15: 3D reconstruction of the Cat model using the *123D Catch* System.

## 5 Conclusion and Future Work

Our research was motivated by the observation that there is an increasing demand for virtual 3D models. Existing modelling packages, such as Blender and Maya, enable the construction of realistic and complex 3D models, but have a steep learning curve and require a high level of artistic skills. Additionally, their use is very time-consuming and often involves tedious manual manipulation of meshes. The use of specialised hardware for 3D model reconstruction (laser scanners) makes it possible for inexperienced users to acquire 3D digital models. However, such hardware is very expensive and limited in its applications. Based on this, we concluded that the most promising approach for a general, affordable content creation process is to use an image-based modelling approach using images obtained with a consumer-level uncalibrated camera.

We demonstrated our system's ability to produce qualitatively good results for a wide range of objects including those with smooth, uniform, and textureless surfaces or containing concave regions. The system has also demonstrated its robustness in the case that there are illumination variations and shadows in the input images.

Problems, such as missing textures in some regions, still exist with the resulting 3D models. This is caused by insufficient input images of those regions. We aim to overcome this problem by employing image in-painting and exemplar-based texture synthesis techniques.

Although we have demonstrated that our system can create 3D content inexpensively and conveniently, the high computation cost partially offsets its advantages. For example, our system takes approximately 2 hours to process 17 images, and roughly 5 hours for 31 images on a Intel Quad Core i7 with 6GB RAM. The computational cost rises quadratically with the number of input images. For the current state-of-the-art of hardware technology, real time processing is impossible on consumer-level machines. However, we can reduce the processing time considerably by parallelising the computations as much as possible and executing them on the GPU.

## References

[Bau74] Bruce Guenther Baumgart. *Geometric modeling for computer vision.* Doctoral Dissertation, Stanford University, 1974.

[BL05] Matthew Brown and David Lowe. Unsupervised 3D object recognition and reconstruction in unordered datasets. *In International Conference on 3D Digital Imaging and Modelling*, pages 56–63, 2005.

[Bol10] Matthew Grant Bolitho. *The Reconstruction of Large Three-Dimensional Meshes.* Doctoral Thesis. The Johns Hopkins University, 2010.

[Est04] Carlos Hernandez Esteban. *Geometric modeling for computer vision.* Doctoral Thesis, Ecole Nationale Superieure des Telecommunications, 2004.

[FB10] Jean Sebastien Franco and Edmond Boyer. Efficient polyhedralmodeling from silhouettes. *IEEE Transaction on Pattern Analysis and Machine Intelligences*, 31:853–861, 2010.

[FZ03] Cristian Fruh and Avideh Zakhor. Constructing 3d city models by merging ground-based and airborne views. *In IEEE International Conference on Computer Vision and Pattern Recognition*, 2:562–569, 2003.

[HL07] Shungang Hua and Ting Liu. Realistic 3D reconstruction from two uncalibrated views. *In International Journal of Computer Science and Network Security*, 7:178–183, 2007.

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

[Kaz05] Michael Kazhdan. Reconstruction of solid models from oriented point sets. *In SIGGRAPH*, pages 73–82, 2005.

[KBH06] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. *In Proceedings of the fourth Eurographics symposium on Geometry processing (Aire-la-Ville, Switzerland, Switzerland, 2006)*, pages 61–70, 2006.

[Lor95] William Lorensen. Marching through the visible man. *IEEE Visualization*, pages 368–373, 1995.

[Low06] David G Lowe. Distinctive image features from scale-invariant keypoints. *In International Journal of Computer Vision*, 60(91):110–2004, 2006.

[LQ02] Maxime Lhuillier and Long Quan. Quasi-dense reconstruction from image sequence. *European Conference on Computer Vision*, pages 125–139, 2002.

[MA83] Worthy 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.

[MBR+00] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven Gortler, and Leonard McMillan. Image-based visual hulls. *In Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 369–374, 2000.

[Mel98] Stan Melax. Simple, fast, and effective polygon reduction algorithm. *Game Developer Magazine*, pages 44–49, 1998.

[NWDL11] Minh Hoang Nguyen, Burkhard C Wunsche, Patrice Delmas, and Christof Lutteroth. Modelling of 3D objects using unconstrained and uncalibrated images taken with a handheld camera. *Computer Vision, Imaging and Computer Graphics - Theory and Applications*, 274:1–5, 2011.

[NWDL12] Minh Hoang Nguyen, Burkhard C Wunsche, Patrice Delmas, and Christof Lutteroth. 3D models from the black box: Investigating the current state of image-based modeling. *Communication Proceedings, Pilsen, Czech Republic, Union Agency*, pages 25–28, 2012.

[QTZ+06] Long Quan, Ping Tan, Gang Zeng, Lu Yuan, JingdongWang, and Sing Bing Kang. Image-based plant modeling. *In ACM Transactions on Graphics*, 25(3):599–604, 2006.

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

[SSS06] Noah Snavely, Steven Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3D. *In ACM Transactions on Graphics*, 25(3):835–846, 2006.

[THP08] Thormahlen Thorsten and Seidel Hans-Peter. 3D-modeling by ortho-image generation from image sequences. *ACM Transaction Graph*, 27(3):1–5, 2008.

[TZW+06] Ping Tan, Gang Zeng, Jingdong Wang, Sing Bing Kang, and Long Quan. Image-based tree modeling. *In ACM Transactions on Graphics*, 27(3):418–433, 2006.

[XFT+08] Jianxiong Xiao, Tian Fang, Ping Tan, Peng Zhao, Eyal Ofek, and Long Quan. Image-based façade modeling. *In ACM Transactions on Graphics*, 27(5):26–34, 2008.