# CARVING 3D MODELS FROM UNCALIBRATED VIEWS

M. SAINZ, N. BAGHERZADEH
Image Based Modeling and Rendering Lab.
Department of Electrical and Computer Engineering
University of California Irvine, USA
{msainz,nader}@uci.edu

A. SUSIN
Dynamic Simulation Lab.
Departament de Matemàtica Aplicada 1
Universitat Politècnica de Catalunya, Spain
toni.susin@upc.es

**ABSTRACT**
In this paper we present an automatic method for the reconstruction of a 3D volumetric representation of real world scenes from a set of multiple uncalibrated images. The process is divided in two steps (1) an automatic calibration of the cameras and (2) a scene reconstruction consistent with the input views. The calibration of the cameras is performed using automatically tracked 2D features, and consists in calculating a projective approximation and upgrading it to an Euclidean structure by computing the projective distortion matrix in a way that is analogous to estimate the absolute quadric. Moreover, in contrast to other approaches our process is essentially a linear one. The underlying technique is based on the Singular Value Decomposition (SVD) and the process is enhanced with a careful study of the rank of the matrices involved in order to get the excellent results shown in the paper. The volumetric reconstruction of the scene is performed using an improved voxel carving algorithm. The result is a voxel-based model of the external surface of the physical objects present in the scene. Optimized data structures and graphics hardware acceleration are used to achieve a substantial reduction in computation time. Furthermore, the spatial information obtained from the the camera calibration process about the 2D tracked measurements is used to automatically set the internal thresholds of the carving algorithm, achieving a full automation of the method.

**KEY WORDS**
3D reconstruction, Structure from Motion, Singular Value Decomposition, Voxel Carving.

## 1 Introduction

In recent years Image Based Modeling and Rendering (IBMR) techniques have demonstrated the advantages of using real image data to greatly improve the rendering quality. New rendering algorithms have been presented that reach a photorealistic quality at interactive speeds when rendering 3D models by using images of real objects and some additional shape information (i.e. a geometric proxy). While these methods have emphasized the rendering speed and quality, they generally require extensive preprocessing in order to obtain accurately calibrated images and geometric proxies of the target objects. Moreover, most of these algorithms require user interaction for the camera calibration and image registration part or need the use of expensive equipment such as calibrated gantries and 3D scanners.

In this paper we present a method for extracting a 3D volumetric representation of a real object using a set of different views taken with an inexpensive sensor such as a digital camera or a camcorder. More specifically, the goal is to recover the 3D geometry of a scene from the 2D projections obtained from the digital images of multiple reference views, taking into account the motion of the camera. Neither the camera calibration (intrinsic parameters and pose) nor the geometry of the scene are known.

The first part of the process is the calibration of the different reference views of the scene. This is known as the *structure from motion problem* (SFM). Since we are working with uncalibrated cameras, we use a stratification approach to recover both camera parameters and structure of the scene ([16], [6], [7], [3], [10]). The idea is to obtain a projective reconstruction and then upgrade it to an Euclidean structure. For a good review of different methods we suggest [5].

One advantage of the calibration approach presented in this paper is that allows to recover an Euclidean reconstruction of the scene without any initial solution, which is one of the drawbacks of most of the existing methods. Another important feature is that the entire process is based on solving *linear systems* using the SVD decomposition algorithm. The knowledge of the geometric meaning and rank properties of the different transformations represented by the matrices of the process allows to enforce a valid Euclidean reconstruction. As shown in [12] high accuracy can be obtained when synthetic data is used, and when noise is added or real data is used, the precision is still acceptable.

The second part of the process is the reconstruction of the 3D geometry of the objects in the scene, based on the reference images and their registration information. Different approaches such as photogrammetry, stereo vision, contour and shadow analysis techniques have been used to solve this problem. Recently a set of volumetric techniques based on voxel coloring algorithms ([1], [2], [4], [9]) have been used for reconstructing complex object shapes with excellent results. These methods consist in carving a voxelized piece of virtual material that contains the objects in a similar way as and artist sculpting a raw block of marble. The tradeoff is the extended computational cost. The method presented in this paper improves the run-time speed of the space carving algorithm based on voxel coloring [13]

by using optimized data structures and the OpenGL API which is fully supported by hardware. Moreover, the internal threshold selection is performed automatically and the color consistency test is improved using a statistical criterion.

## 2 Image Calibration

The presented solution of the SFM problem considers a set of reference views as a sequence of images taken with a single camera over time. The input data consists of a sparse set of features, 2D points, selected in the reference views. We assume that the features and their individual correspondences along the set of images are given, characterizing perfectly the 2D trajectories of each feature in the sequence of images.

The method does not require prior knowledge of the 3D coordinates of the features, the relative position for of the camera in the scene for each reference view (extrinsic parameters of the camera), and the internal geometry (intrinsic parameters). The goal is to recover all this information by analyzing the variation of the 2D measurements of each of the features in the set of reference images.

Here on, and to guarantee good numerical conditioning, we assume that the 2D features are expressed in normalized image coordinates, that is, following [17] the image pixels coordinates are scaled to lie in a $[-1,1]\times[-1,1]$ image rectangle.

### 2.1 Projective Factorization

The projective factorization method ([16],[7], [3],[10]) is a generalization of the factorization methods developed in [15] and [11] for the orthographic and the paraperspective projection models respectively. The projective factorization gives a more general framework to recover 3D shape and motion from multiple view images, overcoming the restrictive assumptions of the other two models. However, it is only possible to compute a reconstruction up to an unknown projective transformation unless additional information about the camera intrinsic parameters, the motion and the object dimensions can be obtained.

The goal is to recover 3D structure and camera parameters from $m$ uncalibrated perspective views of a scene and the 2D projection on each image of $n$ 3D object points. Let $\mathbf{X}_j = (X_j, Y_j, Z_j, 1)^T, \quad j = 1,\ldots,n$, be the unknown homogeneous 3D point coordinates, $P_i, \quad i = 1,\ldots,m$ the unknown $3\times 4$ camera projection matrices, and $\mathbf{x}_{ij} = (u_{ij}, v_{ij}, 1)$ the measured 2D homogeneous projections of the 3D points.

We introduce the *projective depths* as the non-zero scale factors $\lambda_{ij}$ that relate the 3D points in camera coordinates and their projection in 2D image space

$$\lambda_{ij}\mathbf{x}_{ij} = P_i\mathbf{X}_j \quad i = 1,\ldots,m \quad j = 1,\ldots,n. \qquad (1)$$

This can be stated using matrix notation as $\mathbf{W} = \mathbf{PX}$, where $\mathbf{W}$ is the $3m\times n$ *scaled measurement matrix*, $\mathbf{P}$ is the $3m\times 4$ *perspective matrix* and $\mathbf{X}$ is the $4\times n$ *shape matrix*. Since a scale factor can be introduced in the previous equation without altering the result, each reconstructed scene is defined up to re-scaling. With the correct scaling, normalized points and projections the $\lambda$ factors become true optical depths.

The projective depths and 3D structure of the scene are dependant on each other and can not be calculated separately. On the other hand, since the matrix $\mathbf{W}$ is associated to a projection of 3D points in 2D image space, its rank has to be equal to four. Consequently, for points in general positions, a rank-4 factorization of the scaled measurement matrix produces a valid projective reconstruction of the points.

There exist different approaches ([3], [6], [7], [10], [16]) to construct an iterative algorithm that converges to a rank-4 decomposition of the measurement matrix. We propose an iterative projective factorization algorithm to recover the values of $\lambda_{ij}$ based on Singular Value Decomposition (SVD) (see [12] for more details).

### 2.2 Metric Reconstruction

The projective factorization of Equation (1) recovers the motion and the shape up to a linear projective transformation $H$ known as the *Projective Distortion Matrix* (PDM)

$$W = \hat{P}\hat{X} = \hat{P}HH^{-1}\hat{X} = PX, \qquad (2)$$

with $P = \hat{P}H$ and $X = H^{-1}\hat{X}$ being $P$ and $X$ unknowns.

We define as *normalization* the process of imposing a set of metric constraints to calculate the PDM and recovering a correct Euclidean motion and shape. In [12] is shown this process is equivalent to the calculation of the absolute quadric of the epipolar based methods.

For each frame $i$, the $3\times 4$ projective matrix $P_i$ can be decomposed into

$$P_iH = \mu_iK_i\,(R_i|\mathbf{T}_i) \quad i = 1,\ldots,m, \qquad (3)$$

where

$$K_i = \begin{pmatrix} f_i & \beta_i & u_{0i} \\ 0 & \alpha_i f_i & v_{0i} \\ 0 & 0 & 1 \end{pmatrix},$$

$$R_i = \begin{pmatrix} \mathbf{i}_i^T \\ \mathbf{j}_i^T \\ \mathbf{k}_i^T \end{pmatrix}, \quad \mathbf{T}_i = \begin{pmatrix} \mathbf{T}_{xi} \\ \mathbf{T}_{yi} \\ \mathbf{T}_{zi} \end{pmatrix}, \quad i = 1,\ldots,m. \qquad (4)$$

Here $\mu_i$ is a scale factor, the matrix $K_i$ encodes the intrinsic parameters of the camera where $(u_{0i}, v_{0i})$ is the principal point, $\alpha_i$ is the aspect ratio, $\beta_i$ is the skew and $f_i$ is the focal length. $R_i$ and $\mathbf{T}_i$ are the $3\times 3$ rotation matrix and translation vector of the camera in the $i$th frame.

Different cases can be considered according to the number of unknown intrinsic parameters of the camera. In our approach we assume zero skews, an aspect ratio of one and the principal point located in the center of the image, at the origin of coordinates (see [7] for the other possibilities). Our study considers the focal length $f$ as the only unknown variable, which is an acceptable assumption when the ratio between the position of the scene and the focal length is small. This is not valid in in aerial imagery, where the objects are very far away from the camera and the motion is small in compared to the distance.

### 2.2.1 Normalization

Without lack of generalization we can assume $\mu_i = 1$ in Equation (3) and define the global matrix $P = (M|T)$, where $M_i = K_i R_i$, and $T_i = K_i \mathbf{T}_i$.

If we express the $4 \times 4$ PDM matrix as $H = (A|B)$ where $A$ is $4\times3$ and $B$ is $4\times1$, from (2) we have $P = \hat{P}H$, then $(M|T) = \hat{P}\,(A|B)$. The computation of the translation component ($T = \hat{P}B$) can be decoupled from the rotation one ($M = \hat{P}A$), allowing to compute the Euclidean reconstruction using essentially linear algorithms, instead of the nonlinear ones related with Kruppa's equations [5].

Taking into account that the matrix $\mathbf{X}$ is related to the geometry of the object and therefore is independent of the camera parameters, we can define a local coordinate system with arbitrary orientation and the origin at the center of mass of the object

$$\mathbf{X}_j^T = (\tau_j s_{xj}, \tau_j s_{yj}, \tau_j s_{zj}, \tau_j)\,, \quad j = 1, \ldots, n, \qquad (5)$$

where $s_j = (s_{xj}, s_{yj}, s_{zj})$ are the local coordinates. Looking at the sum of the $X$ coordinates of the projected points and using the center of mass property, we obtain

$$\sum_{j=1}^{n} \lambda_{ij} u_{ij} = T_{xi} \sum_{j=1}^{n} \tau_j. \qquad (6)$$

Analogously, a similar expression is obtained for the other coordinates $Y$ and $Z$. A system in terms of $B$ can be defined considering the translational component $T = \hat{P}B$

$$T_{xi} = \hat{P}_{xi} B, \qquad T_{yi} = \hat{P}_{yi} B, \qquad T_{zi} = \hat{P}_{zi} B. \qquad (7)$$

Finally, a homogenous system can be constructed dividing (7) by $T_{zi}$ and combining with (6). We obtain $2n$ linear equations for the $4$ unknown elements of $B$. The base of the null space of the system gives us the elements of $B$.

The next step is to compute the matrix $A$ to complete the desired projective distortion matrix. The information embedded in $A$ can be seen as the orientation of the PDM. From (3) and the rotational component $(M) = \hat{P}\,(A)$ we obtain

$$M_{xi} = \mu_i f_i \mathbf{i}_i + \mu_i u_{0i} \mathbf{k}_i,$$
$$M_{yi} = \mu_i \alpha_i f_i \mathbf{j}_i + \mu_i v_{0i} \mathbf{k}_i, \quad M_{zi} = \mu_i \mathbf{k}_i. \qquad (8)$$

Considering the assumptions on the intrinsic camera parameters, $\alpha_i = 1$, $u_{0i} = v_{0i} = 0$, and imposing the Euclidean rotational axis to be orthogonal we get the following metric relations

$$|M_{xi}|^2 = |M_{yi}|^2, \quad |M_{zi}|^2 = \mu_i^2,$$
$$M_{xi} \cdot M_{yi} = M_{xi} \cdot M_{zi} = M_{yi} \cdot M_{zi} = 0 \qquad (9)$$

From (9), the metric constraints can be written as linear constraints

$$M M^T = \hat{P} A A^T \hat{P}^T = \hat{P} Q \hat{P}^T. \qquad (10)$$

obtaining a set of $4m$ linear equations for the $10$ unknowns of $Q$. In [7] Han and Kanade avoid the homogeneous condition by adding an extra metric equation fixing the first scale factor to one, $\mu_1 = 1$, and adding the equation $|M_{z1}|^2 = 1$ to (10). A least square solution of $Q$ is calculated and by performing a rank $3$ decomposition of $Q$ a solution for $A$ can be extracted completing the projective distortion matrix.

As shown in [12], the method proposed in [7] does not impose to (10) the essential rank condition of the absolute quadric: rank($\Omega$)=3 (see [17]). Moreover, according to [12] the homogeneous system (10) turns to be of rank 8 since the unknowns involved in the system (10) are essentially the components of the *absolute quadric* $\Omega$ and the *dual absolute conic* $\omega$. In [17] it is shown that there is an additional constraint that forces the angles between visual planes measured using $\Omega$ to agree with those measured from the corresponding image lines using $\omega$. When projected to epipolar planes this gives the Kruppa linear constraint.

Therefore, there is an extra degree of freedom on the solution of (10) which can be used to enforce the final matrix $Q$ to be rank 3. In our method we consider a linear combination of the two vectors associated to the null space of (10) in terms of an unknown constant. Imposing $\det(Q) = 0$, we obtain a fourth degree polynomial in terms of the unknown constant. We choose the root that enforces a rank 3 matrix $Q$. Then, the matrix $A$ will be calculated as a rank 3 approximation of $Q$ using the SVD decomposition.

## 3 Shape Reconstruction

The result of the image registration process is the set of intrinsic parameters, the position and orientation of the cameras and the 3D position of the 2D features. The next step is to recover the 3D shape of the objects in the scene. As mention before, the proposed solution is an improved version of the *space carving* algorithm presented in [9].

Six progressive carving steps are performed by sweeping a *carving plane* along the positive and negative direction of each axis of a bounding volume that contains the objects to reconstruct. At each iteration the voxels in the carving plane are tested for color consistency in all the images where they are visible, and removed if not consistent. The resulting voxel-based object is called the *Photo Hull* because when reprojected to each view, it matches closely the original images. As shown in [9], the accuracy of the final model is related to the number of reference views and their coverage of the scene. A crucial part in the method is the *Consistency Check Criterion*, the mechanism used to decide whether a voxel will be kept or carved.

The main steps of the proposed improved space carving algorithm are

1. Estimate threshold.
2. Determine initial voxel volume.
3. Store the voxels in an octree data structure.
4. Determine active cameras. Draw frontal faces of voxels that lie in the sweep plane.
5. Draw shadow voxels in front of the sweep plane.
6. Scan the images and build a list of voxels with their RGB statistics per image.
7. Perform a consistency check at each voxel. If is not consistent carve the voxel.
8. go to step 4.

The following subsections will provide more detail about the primary components of the algorithm.

### 3.1 Initial Voxel Volume

The first step of the algorithm is to determine the initial volume to be carved. The size of this volume is calculated by upscaling the bounding box that contains the recovered 3D points from the SFM reconstruction. The orientation of the initial volume is determined by averaging the recovered camera orientations, averaging the exposed voxel surface in all the cameras.

A large amount of the initial volume projects to the reference images in areas that can be considered as background. Moreover some of the initial voxels are visible in one or few images and will not contribute to the final volume. To improve the performance of the algorithm it is reasonable to eliminate such voxels prior to start the carving process.

We use an octree data structure to keep track of the set of consistent voxels throughout the carving process. The root node is initialized and a *visibility test* is performed to remove the set of voxels not visible. This is done by recursively subdividing the octree and testing the cells for intersection against the camera frustums. If an octree cell is visible in less that a minimum number of cameras it will be marked as initially carved and it will not be further subdivided. Moreover, if the reference images can be segmented into background and foreground, a *background test* can be performed in order to stop the subdivision process if an octree cell falls into the background in any of the images.

The result of this initialization is a pre-carved voxel space adjusted to the object's convex hull, that improves the performance of the algorithm eliminating voxels from the analysis. The octree data structure plays an important role in storing and managing the voxels during the plane sweep carving process.

### 3.2 Carving Plane Projection

The space carving algorithm is an iterative process that in each step projects a plane of voxels to all the cameras. A *footprint* is the projection of a voxel in one camera focal plane. For each footprint the average color and variance are calculated and stored together with the other footprints of the same voxel. Since this step is performed several times, we have optimized our implementation using OpenGL accelerated hardware to generate the footprints.

Each voxel is assigned with a unique RGB color encoding its location in space. Then all the voxels of the carving plane are drawn for one of the reference views. The rendered set of voxels determine a color mask that encodes for each pixel of the image to which voxel belongs. The footprint statistics are obtained by scanning the reference image using the color mask. This process is then repeated for each of the reference views.

In order to avoid evaluating voxels occluded to the reference images by other voxels previously determined as consistent, we first render each of the voxels of the active carving plane as a single colored square. Then all the non-carved voxels previously analyzed are rendered in black color enabling the zBuffer of the videocard in order to mask those parts of the images already assigned to previous voxels. We call these the *shadow voxels*.

### 3.3 Consistency Check Criterion

The consistency check tests each voxel of the carving plane to determine if it belongs to the object or not and then mark it as carved. Assuming the object surface is Lambertian, the color of a voxel does not change regardless of the viewing angle and a voxel consistency can be determined by measuring and comparing the color of its footprints. If they have a similar color the voxel will be kept and rendered as a shadow voxel in subsequent iterations. Otherwise it will be carved.

In the approach of Kutulakos and Seitz [9] they consider a unique color value for each footprint. However the voxels project to larger areas than a single pixel, and important information is lost making the method very sensitive to sensor noise. Therefore, as in [1], we use a statistical consistency check instead of the traditional one. For each footprint of a voxel we store the number of pixel samples, and for each R,G and B channel the mean, the variance, the sum of values and the sums of squares of all the contributing pixels. The consistency check is a *One Way ANOVA with Unequal Sample Sizes* test considering the different footprints of a voxel as groups and testing if all those groups can be assimilated to a single color distribution.

In the work by Kutulakos and Seitz [9] and by Broadhurst and Cipolla [1] the user is required to select a threshold for the consistency check. To avoid a manual selection our approach exploits the additional scene information obtained during the calibration. By performing an initial consistency check of the voxels that contain the 3D points of the object used during calibration, a threshold estimation that would avoid carving those voxels can be calculated.

## 4 Experiments and Results

Several validation tests have been performed with real uncalibrated images. The experiments consisted in obtaining the required 2D measurements, calibrating the different views and performing a volumetric reconstruction of the objects. All tests were performed on a PC system with a 1.8GHz P4, 512Mb of RAM and a 64Mb NVIDIA Quadro Pro videocard. Figure 1 shows a mosaic of one of the reference images and three different views of the reconstructed model for each of the datasets. A detailed analysis of the performance of the different stages of the reconstruction pipeline is presented.

The first stage of the pipeline is the data acquisition and preprocess to obtain 2D measurements for the camera calibration. Given the differences of footage and image quality of the datasets we have used three different tracking strategies.

The *house* dataset has been tracked automatically with an in-house tracker implementation. The images do

| | House | Dragon | Anteater |
|---|---|---|---|
| # Features | 39 | 10 | 9 |
| # Images | 30 | 5 | 39 |
| Projection Time(sec.) | 4.15 | 0.421 | 1.5 |
| Metric Time (sec.) | 0.031 | 0.016 | 0.016 |
| Max 2D Error (pixels) | 4 | 11 | 5 |

Table 1. Autocalibration results

not show any specular reflections on the object surface and it presents sharp corners facilitating feature detection and tracking. The pose variation of the object between frames is very small. The dataset consists of 30 frames of 512x512 pixels and a total of 39 points found by the tracking system.

The second dataset, the *dragon*, consists of a few still pictures. An initial set of 15 images of 720x480 pixels was reduced to 5 useful ones due to specularities in the object. We manually selected 10 features.

Finally, the *anteater* dataset has 393 frames of 720x480 pixels. We put markers in the object and we used a semiautomatic tool that allows the user to select a point to track in one frame, and the software does the tracking through the rest of frames. A total of 9 features were used.

The second stage of the pipeline is the camera autocalibration, where the camera intrinsic and extrinsic parameters related to the 3D location of the object itself are calculated. To evaluate the accuracy of the reconstruction a 2D reprojection error was calculated (Table 1). Additionally a visual quality check was performed since no ground truth data was available.

The last stage of the pipeline is the volumetric reconstruction by voxel carving. The initial resolution of the carving volume was set to 256 voxels per side and a reduced subset of the images was used: 7 images for the *house* dataset, 5 images (all the available) for the *dragon* dataset and 6 for the *anteater* dataset.

| | House | Dragon | Anteater |
|---|---|---|---|
| Volume | $256^3$ | $256^3$ | $256^3$ |
| # Initial Leafs | 2747120 | 2333184 | 1438448 |
| Initial Memory (Kb.) | 173429 | 147518 | 91827 |
| # Final Leafs | 116306 | 112438 | 107704 |
| Threshold | 4.6e-2 | 1.0e-2 | 1.9e-2 |
| Init. Time (sec.) | 501 | 278 | 539 |
| Carving Time (sec.) | 472 | 291 | 560 |

Table 2. Voxel carving results

To reduce the computation time of the carving process a set of background/foreground masks have been manually created for each of the datasets to enable the use of the background testing during the initialization of the volume. Table 2 shows the initialization time, initial volume, volume reduction and memory used to store the octree. This initialization plays an important role since it reduces the initial volume to a 16.3%, 13.9% and 8.6% of the original $256^3$ voxels for the house, dragon and anteater datasets respectively.

The threshold for the consistency check has been calculated using the voxels containing the recovered 3D points of the objects. Before starting the initialization of the octree, these voxels are projected into the reference views and the threshold is adjusted to guarantee their consistency. The value represents the confidence of rejecting a voxel as consistent, which in our datasets are 5%, 1% or 1.9% respectively.

The carving algorithm has a large computational cost but with the octree optimization and the background test short running times (between 9 and 18 minutes per sweep direction) have been obtained for large voxel resolutions.

The final voxelized models have an excellent visual appearance when viewed from the calibrated cameras, and for any location around to those cameras. For arbitrary views that are too far apart from the original views the models are not that accurate. This is a well know limitation of the image based model reconstruction algorithms, since information not captured in the images cannot be reconstructed unless additional information is provided. However, with a set of images that provide a good coverage of the object a fairly accurate model can be easily reconstructed, as shown in this work.

## 5 Conclusions

A completed pipeline for reconstructing 3D models from uncalibrated images of real objects has been presented.

One of the advantages of the proposed method is that it does not require any initial solution or arbitrary additional constraints to perform the reconstructions. Another advantage is that the SFM method is linear in the unknowns, therefore computationally faster than other nonlinear approaches.

The original space carving technique has been improved by adding a statistical consistency criterion and an automatic consistency threshold selection. Moreover, the use of an octree based data structure to optimize the process has shown drastic reduction of the computational time.

The entire reconstruction pipeline has been tested with both multiple views and long image sequences. Excellent reconstructions have been obtained for all the datasets, although some of the models present concavities on the surface. Moreover, the proposed method is computationally fast in standard PC computers, making it a very attractive solution.

As a future work, we plan to enhance the quality of the final voxel based reconstructions by generating a polygonal mesh representation which then can be texture mapped with the original images.

## 6 Acknowledgements

This research was supported by the National Science Foundation under contract (CCR-0083080) and by the Comissio Interdepartamental de Recerca i Innovacio Tecnologica (CIRIT), Gaspar de Portola grant C01-02.

## References


[1] A. Broadhurst and R. Cipolla, A statistical consistency check for the space carving algorithm, *Proc. British Machine Vision Conference* **I**, 2000, 282-291.

[2] A. Broadhurst, T.W. Drummond and R. Cipolla, A probabilistic framework for space carving, *Proc. Int. Conf. on Computer Vision* **I**, 2001, 282-291.

[3] Q. Chen and G. Medioni, Efficient iterative solution to M-view projective reconstruction problem, *Proc. IEEE Computer Vision and Pattern Recognition* **1**, 1999, 55-61.

[4] W.B. Culbertson, T. Malzbender and G. Slabaugh, Generalized Voxel Coloring, *Proc. of Vision Algorithms Theory and Practice Workshop*, Corfu, Greece, 1999. 100-114.

[5] A. Fusiello, Uncalibrated Euclidean reconstruction: a review. In *Image and Vision Computing*, **18**, 2000, 555-563.

[6] A. Heyden, R. Berthilsson and G. Sparr, An iterative factorization method for projective structure and motion from image sequences. *Image Vision and Computing*, **17**, 1999, 981-991.

[7] M. Han and T. Kanade, Creating 3D Models with Uncalibrated Cameras, *Proc. IEEE Computer Society Workshop on the Application of Computer Vision*, **9**(2), 2000, 137-154.

[8] D. Jacobs, Linear fitting with missing data for structure-from-motion, *Computing Vision and Image Understanding*, **82**, 2001, 57-81.

[9] K. Kutulakos and S. Seitz, A theory of shape by space carving, *International Journal of Computer Vision* **38**(3), 2000, 198-218.

[10] S. Mahamud and M. Hebert, Iterative projective reconstruction from multiple views, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*,**2**, 2000, 430-437.

[11] C.J. Poelman and T. Kanade, *A paraperspective factorization method for shape and motion recovery*(Technical Report CMU-CS 93-219, School of Computer Science, Carnegie Mellon University, December 1993).

[12] M. Sainz, N. Bagherzadeh and A. Susin, Recovering 3D Metric Structure and Motion from Multiple Uncalibrated Cameras, *Proc. IEEE Int. Conference on Information Technology: Coding and Computing* , 2002, 268-273.

[13] S. Seitz and C. Dyer, Photorealistic scene reconstruction by voxel coloring, *International Journal of Computer Vision* **35**(2), 1999, 1067-1073.

[14] P. Sturm, Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction, *Proc. IEEE Computer Vision and Pattern Recognition*, 1997, 1100-1105.

[15] C. Tomasi and T. Kanade, Shape and Motion from Image Streams Under Orthography: a factorization method, *Intl. Journal of Computer Vision*, **9**(2), 1992, 137-154.

[16] B. Triggs, Factorization methods for projective structure and motion, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1996, 845-851.

[17] B. Triggs, Autocalibration and the Absolute Quadric, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1997, 609-614.


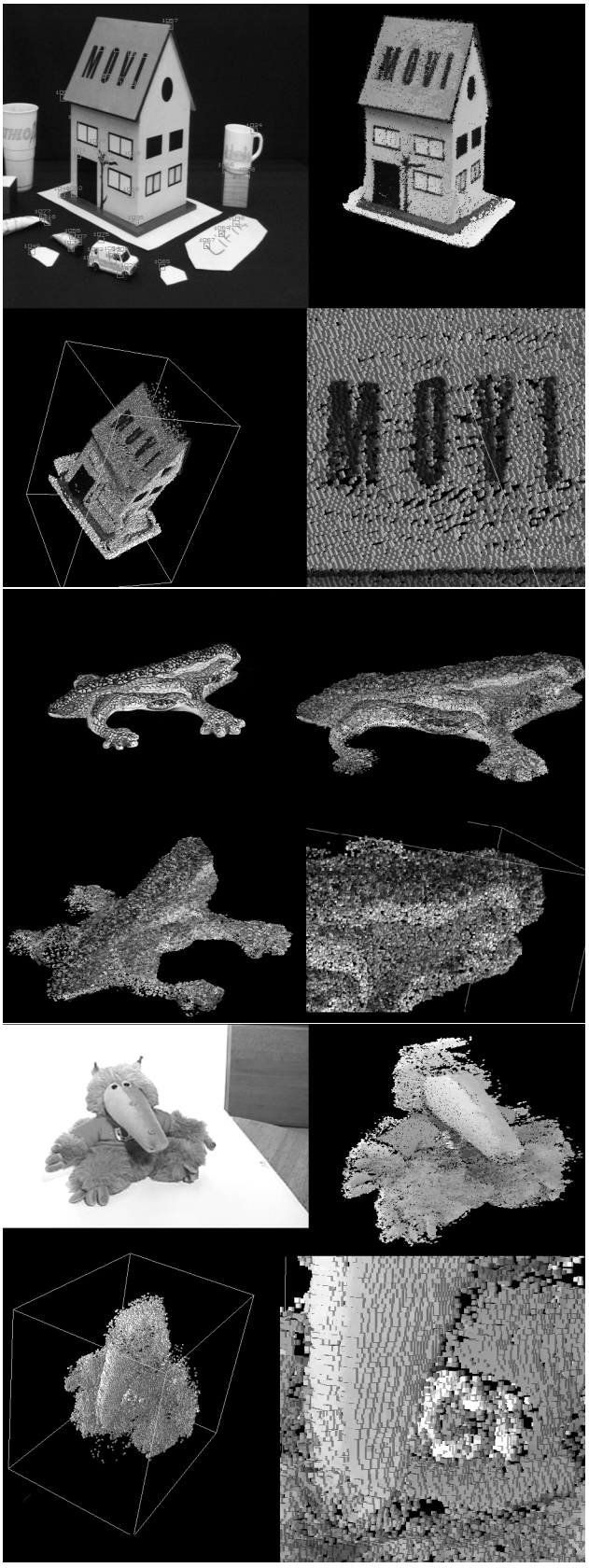

Figure 1. For each of the datasets the top left image shows one of the reference views. The other images are different views of the reconstructed model.