Carving 3D Models from Uncalibrated Views

M. Sainz, N.Bagherzadeh
Department of Electrical and Computer Engineering
University of California Irvine, USA
{msainz,nader}@ece.uci.edu

A. Susin
Departament de Matemàtica Aplicada 1
Universitat Politècnica de Catalunya, Spain
toni.susin@upc.es

ABSTRACT
A general automatic method for recovering a 3D volumetric representation of a scene and the camera parameters from multiple uncalibrated images is presented. Using automatically tracked 2D features a first projective approximation is calculated and upgraded to an Euclidean structure by computing the projective distortion matrix in a way that is analogous to estimate the absolute quadric. With the Singular Value Decomposition (SVD) as a main tool, and a careful study of the rank of the matrices involved in the process we are able to get excellent results. Moreover, in contrast to other approaches our process is essentially a linear one. After the camera autocalibration process, an improved voxel carving algorithm is used to recover the external surface of the objects in the scene. This carving algorithm uses optimized data structures for speed purposes and the information from the calibration process, such as the 3D coordinates of the tracked points, to select automatically the internal threshold used when carving.

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

1 Introduction

In recent years Image Based Rendering has shown the importance of using real imagery to highly improve the quality of final renders. New rendering algorithms have been presented([20],[3],[6]) that reach a photorealistic quality and interactive speeds when rendering 3D models by using images of real objects and some geometric information (i.e. a geometric proxy). However, while these methods have emphasized the rendering speed and quality, most of them need a tremendous preprocessing effort in order to obtain well calibrated images and geometric approximations of the objects. Moreover, most of these algorithms abuse of 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 an object from different views of it taken with a still camera or handheld video camera. More specifically, the goal of the problem is to recover the 3D geometry of a scene from the 2D projections obtained from multiple view images, taking into account the motion of the camera. But, neither the camera calibration (intrinsic parameters and pose) nor the geometry of the scene are known.

The first part of the problem is to calibrate the cameras in the scene. This is known as the *structure from motion problem* (SfM). Since we are working with uncalibrated cameras, we use an stratification approach to recover both camera parameters and structure of the scene ([18],[8],[9],[4],[12]). The idea is to first obtain a projective reconstruction and then upgrade it to Euclidean structure. For a good review of different methods we suggest [7].

One of the advantages of the method proposed in this paper is that it allows to recover an Euclidean reconstruction of the model without any initial guess, 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. The knowledge of the geometric and rank properties of the different transformations represented by the matrices of the process allows us to obtain a valid Euclidean reconstruction. As shown in [14] high accuracy can be obtained when synthetic is used, and when noise is added the precision is still acceptable. The accuracy of these results is maintained when real data examples are used.

Once after the autocalibration process, the second part of the problem is to extract the 3D geometry of the objects in the scene, based on the camera locations and the images. Different approaches have faced this problem using, for example, photogrammetry, stereo vision, contour and shadow analysis techniques . Recently a set of volumetric techniques based on voxel algorithms ([11], [1], [2], [5]) have been used for reconstructing complex shape objects with good results. We have improved the space carving algorithm based on voxel coloring [15] by using an automatic threshold selection and a statistical color consistency criterion.

2 Structure reconstruction and Calibration

Next we will address the SfM problem using a small set of points or features and we assume the 2D trajectories of these features along the image sequence are known. We assume no prior knowledge of their coordinates in the 3D space, the relative motion between the camera and the scene (extrinsic parameters), and the camera's internal geometry (intrinsic parameters) and we wish to recover this information only from the 2D measurements corresponding to the set of features we are considering. Following

[19], we use standard image coordinates, that is, we scale image pixels to lie in $[-1,1] \times [-1,1]$ which guarantees good numerical conditioning.

2.1 Projective factorization method

The projective factorization method ([18],[9], [4],[12]) is a generalization of the factorization method which was first developed in [17] and [13], for the orthographic projection and the paraperspective models respectively. Because of the restrictions in the application of the previous models, projective factorization gives a more general framework to recover 3D shape and motion from multiple view images. Moreover, if no information is known about the camera intrinsic parameters, the motion and the object, only a reconstruction up to an unknown projective transformation is possible to compute.

Our goal is to recover 3D structure and motion from m uncalibrated perspective images of a scene and 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 vectors, $P_i, \quad i = 1, \ldots, m$ the unknown 3×4 image projections, and $\mathbf{x}_{ij} = (u_{ij}, v_{ij}, 1)$ the measured homogeneous image point vectors.

We call *projective depths* the non-zero scale factors λ_{ij} relating the world points and its projections

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

Each object is defined only up to re-scaling. With correctly normalized points and projections the λ's become true optical depths.

We can state the problem using matrix notation as $\mathbf{W} = \mathbf{P}\mathbf{X}$, 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*. As is stated in [18], the projective depths depend on the 3D structure, which in turn derives from the depths. To recover the values of λ_{ij} an iterative projective algorithm is proposed, based on Singular Value Decomposition (SVD). Matrix $\mathbf{W}$ has to be a rank-4 matrix if it is the matrix associated to a projection of a set of real points. Consequently, for points in general positions, a rank-4 factorization of the scaled matrix produces a projective reconstruction of the points. There exist different approaches ([4],[8], [9],[12],[18]) to construct an iterative algorithm that converges to a rank-4 decomposition of the measurement matrix (see [14] for more details).

2.2 Metric reconstruction

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

$$W = \hat{P}\hat{X} = \hat{P}HH^{-1}\hat{X} = PX \tag{2}$$

with $P = \hat{P}H$ and $X = H^{-1}\hat{X}$. We need to impose metric constraints to recover the correct Euclidean motion and shape. This process is called *normalization*. As it is stated in [14] this is equivalent to recover the absolute quadric what makes this approach closer to the ones based on epipolar geometry. Now, for each frame i, the projective 3×4 matrix P_i can be decomposed into

$$P_iH = \mu_i K_i \left(R_i | \mathbf{T}_i\right) \quad i = 1, \ldots, m. \tag{3}$$

where

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

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

Where μ_i is a scale factor, the calibration matrix K_i encodes the intrinsic parameters of the camera, (u_{0i}, v_{0i}) is the principal point, α_i is the aspect ratio, β_i is the skew and f_i is the focal length. R_i and $\mathbf{T}_i$ are the ith rotation 3×3 matrix and translation vector of the camera for each frame. Although different cases can be considered according to the unknown intrinsic parameters of the camera, we will consider the focal length f as the only unknown parameter. Therefore the aspect ratio is 1, the principal point is at the origin and the camera has no skewing(see [9] for the other possibilities). These are acceptable assumptions when the ratio between the position of the scene and the focal length is not too large. The later occurs when using aerial imagery, where the objects are very far away from the camera and the motion of the camera is small.

2.2.1 Normalization algorithm

We will give now a description of the normalization algorithm as is stated in [9] and also the improvements we have added in order to obtain a more robust one. As we will show below, the method presented in [9] does not guarantees to obtain good results by just applying a rank 3 decomposition.

Combining Equation (3) for all the frames, we can get the global matrix $P = (M|T)$, where $M_i = K_iR_i$, and $T_i = K_i\mathbf{T}_i$. If we express the 4×4 matrix PDM as $H = (A|B)$ where A is 4×3 and B is 4×1. Since from (2) we have $P = \hat{P}H$, then $(M|T) = \hat{P}\,(A|B)$. At this moment, we decouple the computation of the translation from the rotation one. This way, we will be able to compute the Euclidean reconstruction using essentially linear algorithms, instead of the nonlinear ones related with Kruppa's equations [7].

Taken into account that the shape matrix $\mathbf{X}$ is related to the geometry of the object and therefore, independent of the frame, we can express each point in local object coordinates

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

where $s_j = (s_{xj}, s_{yj}, s_{zj})$ are the local coordinates. We can also consider the origin of the world coordinate system placed at the center of mass of the scaled object points.

Now, if we look at the sum of the first coordinates of the projected points, using the center of mass we can obtain

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

Analogously, a similar expression can be obtained for the other coordinates. Next, we can use the translation terms to compute B solving a linear least square problem. For that, we consider (2.2.1) to get

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

From (6) we obtain the quotient

$$\frac{T_{xi}}{T_{zi}} = \frac{\sum_{j=1}^{n} \lambda_{ij} u_{ij}}{\sum_{j=1}^{n} \lambda_{ij}}, \qquad \frac{T_{yi}}{T_{zi}} = \frac{\sum_{j=1}^{n} \lambda_{ij} v_{ij}}{\sum_{j=1}^{n} \lambda_{ij}} \tag{8}$$

Finally, from (7) and (8) we can set up an homogeneous system of $2n$ linear equations for the 4 unknowns elements of B. The kernel of the system gives us the elements of B.

On the other hand, we have to compute the matrix A to complete the desired distortion matrix. The information embedded in A is the orientation of the PDM. To express that in a compressible way, first from (3) and (2.2.1) we obtain

$$M_{xi} = \mu_i f_i \mathbf{i}_i + \mu_i u_{0i} \mathbf{k}_i,$$
$$M_{yi} = \mu_i \alpha_i f_i \mathbf{j}_i + \mu_i v_{0i} \mathbf{k}_i, \quad M_{zi} = \mu_i \mathbf{k}_i. \tag{9}$$

In the case we are considering $\alpha_i = 1$, $u_{0i} = v_{0i} = 0$, and using that the rotation axis are orthogonal we get the metric relations

$$|M_{xi}|^2 = |M_{yi}|^2, \quad |M_{zi}|^2 = \mu_i^2,$$
$$M_{xi} \cdot M_{yi} = M_{xi} \cdot M_{zi} = M_{yi} \cdot M_{zi} = 0 \tag{10}$$

From (10), the metric constraints can be written as linear constraints

$$MM^T = \hat{P} A A^T \hat{P}^T = \hat{P} Q \hat{P}^T. \tag{11}$$

obtaining a set of $4m$ linear equations for the 10 unknowns of Q. In [9] they solved the problem adding a new non-homogeneous metric equation (based on the scale factors on (3)) fixing the first factor to one, $\mu_1 = 1$, and adding the equation $|M_{z1}|^2 = 1$ to (11). After obtaining the least square solution they make a rank 3 decomposition of Q to get the matrix A and this way completing the projective distortion matrix. As we have stated in [14] this is not in general a robust method of solution because the essential condition (see [19]) rank(Ω)=3 is not imposed to the solution Q obtained from (11).

In the general case the homogeneous system (11) turns to be of rank 8. This is because, as we have shown in [14], the unknowns involved in the system (11) are essentially the components of the *absolute quadric* Ω and the *dual absolute conic* ω. From [19] we know that there is an additional constraint that forces the angles between visual planes measured using Ω to agree with those measured from the corresponding image lines using ω. When projected to epipolar planes this constraint gives the Kruppa linear constraint. Therefore, we have one extra degree of freedom for the solution which can be used to force the final matrix Q to be rank 3. Indeed, we consider a linear combination of the vectors associated to the zero singular values and we impose $\det(Q) = 0$. This gives us a four order polynomial, using the linear combination coefficients quotient as unique variable, and we choose the best root that gives us a rank 3 matrix Q. After that, we obtain matrix A as a rank 3 approximation of Q using again the SVD decomposition.

3 Shape Recovery

From the above autocalibration process we have recovered the position and orientation of the cameras from which our images had been taken. The next step is to recover the 3D shape of the objects in the scene. As mention before, we use a volumetric method known as *voxel carving* and has been analyzed in ([11], [1], [2], [5]).

We start from an initial voxelized volume that is larger than the object being reconstructed. At each iteration some voxels are carved away until the resulting shape is a good approximation of the external visible part of the scene. A voxel is carved if it is not consistent with the input images. The resulting voxelized object reprojects to the images and matches closely the original images. This object is called the *Photo Hull* as an analogy to the convex hull because it reprojects properly although it could be a superset of voxels of the real model. As shown in [11], the accuracy of the final model will depends on the number of views and their coverage of the scene. A crucial part in the method is the *Consistency Check Criterion* which is the mechanism that decides whether a voxel will be kept or carved.

Our current implementation uses a plane sweep space carving algorithm [11] with some modifications. The main steps are

1. Estimate threshold.
2. Initialize voxel volume to only those voxels visible in a minimum number of cameras (more than one) and that do not lay in background areas. Store the voxels in an Octree data structure.
3. Determine active cameras. Draw frontal faces of voxels that lay in the sweep plane.
4. Draw Shadow voxels in front of the sweep plane.
5. Scan the images and build a list of voxels with their RGB statistics per image.
6. Perform a consistency check at each voxel. If is not consistent carve the voxel.
7. go to step 3.

The following subsections will provide more detail of the key issues of the algorithm.

3.1 Initial Voxelized Volume

One of the first steps of the algorithm is to determine the initial volume to be carved. The size of this volume is cal-

culated by upscaling the bounding box that contains the recovered 3D points from the SFM reconstruction. The orientation of the initial volume will be determined by averaging the recovered camera orientations. This will allow to maximize the exposed voxel surface in all the cameras.

One issue to take into account is that some of the initial voxels will only be visible in one or few cameras and it is reasonable to eliminate them to reduce the total amount of voxels to analyze. Furthermore, when reconstructing an object, there will be areas in the images that correspond to the background and eliminating the voxels that project into those areas before the consistency analysis will speed up the carving process.

In order to achieve this initial simplification, we create an octree data structure that will be recursively divided and tested 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 subdivided. We call this test the *Visibility Test*. Moreover, if the images used for the carving process can be segmented into background and foreground, an additional test can be performed in order to stop the subdivision process if an octree cell falls into the background in any of the images. We call this the *Background Test*

The result of this initialization step will be a pre-carved voxel space adjusted to the object's convex hull. The octree structure will be also very useful to access the different planes of voxels during the plane sweep carving process.

3.2 Sweep plane projection

As shown above, the plane sweep algorithm is and iterative process that in each step projects a plane of voxels to all the cameras. Each voxel's projection into an image or camera focal plane is called *footprint*. These footprints will store the average color and statistical values of each voxel per each image. Since this step is performed several times, we have optimized our implementation by relying on OpenGL accelerated hardware. Each voxel is assigned a unique RGB color and all the voxels that intersect the sweep plane are drawn for each camera settings and then each framebuffer is grabbed from the videocard memory. The resulting set of images will contain the projections of voxels.

The voxels that belong to the active sweep plane will only be painted as a single square face, opposing the sweep direction. However, voxels that have not been carved in previous iterations will be fully painted in color black 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

We follow the ideas of the recent work by Kutulakos and Seitz [11] and by Broadhurst and Cipolla [1] but in their approach the user is required to select the threshold that will be used to determine if a voxel has to be carved. However, in our approach we have more information about the scene due to the reconstruction process we have done. In particular, we know the 3D coordinates of the points we have used along the autocalibration process. As explained above, we use these *true points* to determine the initial voxelized volume. Moreover, since we know the projections of the points in the images, we can evaluate a threshold estimation that would avoid to carve the true points from the voxel space. This solves the problem of the manual selected threshold.

Here on we will assume that the object's surface can be considered Lambertian, so regardless of the viewing angle the color does not change and a voxel consistency can be determined by measuring and comparing the color of its projection among the different camera planes.

Because the voxels project to larger areas than a single pixel, we agree with [1] that using a single pixel color per voxel, as in the original algorithm (see [11]), we lose important information and the method is very sensible to the sensor's noise. Therefore we will use a statistical consistency check instead of the traditional one. For each footprint of a voxel we will store the number of pixel samples, and per each R,G and B channel the mean, the variance, the sum and the sums of squares (SS) of its pixel values. During the consistency check we will perform 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 other words, when there is sufficient evidence to suggest that the voxel can not correspond to the same part of object in all the images, it will be carved. If not, the voxel will be kept and marked to be rendered as a shadow voxel.

4 Experiments and Results

In this section we will present several tests that have been performed with real imagery and we will introduce a detailed analysis of our model reconstruction pipeline stages and its performance for the different datasets. Figure 1 shows a mosaic of different stages of the reconstruction pipeline and the final results.

The first stage of the pipeline is to digitize the sequences and track them to obtain enough 2D information to calibrate the cameras. We use a standard handheld digital video camera that transfers digital video to the computer using a FireWire connection. We have tested three tracking strategies because of the differences of footage and quality of the images of the presented datasets.

The House dataset was tracked automatically with an in-house tracker implementation. The images do not have reflections, there are also sharp corners in the objects that facilitate their detection and tracking, and most important the change of the object's position between frames is very small. We used 30 frames of 512x512 pixels and a total of 39 points were found by the tracking system.

For the second dataset, the Dragon, we only had a few still pictures of 720x480 pixels from different angles. An initial set of 15 images was reduced to 5 useful ones

	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

because specularities in the object. We used 10 features and they were selected manually.

Finally, the Anteater dataset has 393 frames of 720x480 pixels. We put markers in the object and we used a pseudo automatic tool, were the user selects the point to track in one frame, and the software does the tracking through the rest of frames. A total of 9 features have been used.

The next stage of the pipeline is the autocalibration of the cameras, where we will obtain the cameras intrinsec and extrinsec parameters related to the 3D location of the object itself. The accuracy of the reconstruction is determined with a 2D reprojection error analysis (Table 1) in addition of a visual quality check will serve as verification of the results because we do not have ground true data.

Most of the computation time for the autocalibration is used in the iterative projective depth recovery. On the other hand the metric reconstruction takes an nonsignificant amount of time compared to the iterative part. We run our tests in a PC P4 at 1.8GHz and the real datasets take around 0.5s to 5s to complete.

The last stage of the pipeline is the voxel carving algorithm. We have used initial volumes of 256 voxels per side and a downsampled set of images: for the House dataset we have used 7 images and 6 for the anteater. The dragon dataset used all the images available (a total of 5).

	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

In order to accelerate the carving process we have masked of the background of the images to enforce the octree optimization and background testing. Table 2 shows the initialization times, initial volume, volume reduction and memory used to store the octree. This initialization step is very important because 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 thresholds have 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 camera planes and the threshold is adjusted to enforce the consistency of these voxels. The value represents the confidency of discaring the consistency of a voxel. For our samples, any value with a F-value greater than 5%, 1% or 1.9% will be carved.

The carving algorithm is very time consuming but with the octree optimization and the background test step we have been able to obtain small running times for large voxel resolutions. The carving tests have also been done in a PC P4 at 1.8Ghz and the total amount of model carving is between 9 and 18 minutes.

The final voxelized models have a very good appearance from the calibrated cameras, and for any location around to those cameras. For novel views that are too far apart from the original views the models are not that accurate. This is a well know limitation of all the model reconstruction algorithms based on images, because we can not reconstruct, without any additional information, the parts of the model one can not see. However, with a set of images that provide a good coverage of the object a fairly accurate model can be reconstructed easily, 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 need any initial solution or arbitrary additional constraints to compute the final result. Another advantage is that the *Sfm* method is linear in the unknowns therefore computationally faster than other nonlinear approaches.

We have developed an optimized version of the space carving algorithm that used statistical measurements and a automatic threshold selection to determine voxel consistency. Moreover it uses an octree based data structure to optimize the process and reduce the computational time drastically.

We have tested the pipeline with both multiple views and long real image sequences. In all the datasets excellent reconstructions have been obtained, although some of the models have non convex surfaces. Moreover, the proposed method is computationally fast in standard PC computers, making it a very attractive solution.

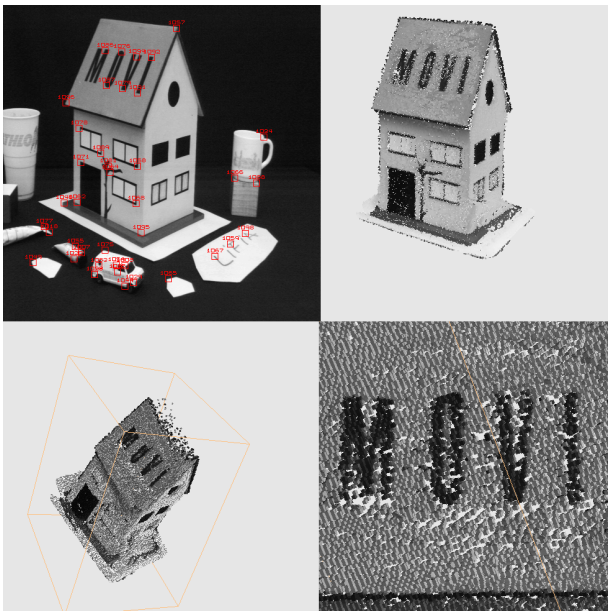
As a future work, we plan to extend our optimized space carving algorithm to perform multiple plane sweeps and to generate directly triangulated meshes for straightforward integration with an IBR algorithm such the ones mentioned.

6 Acknowledgements

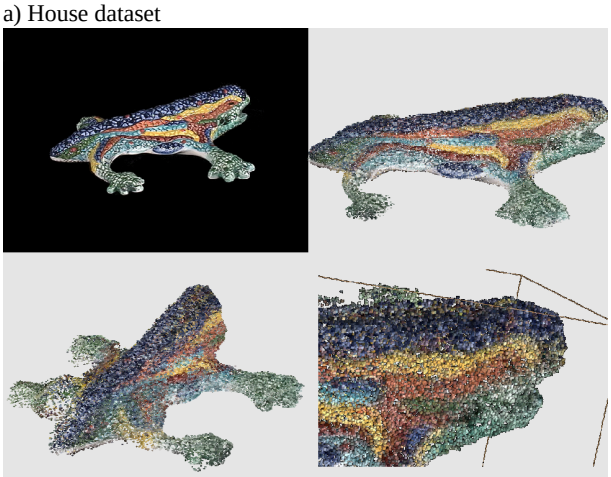
This research was partially funded by grants from NSF (CCR-0083080) and from CIRIT, Gaspar de Portolà grant C01–02.

References

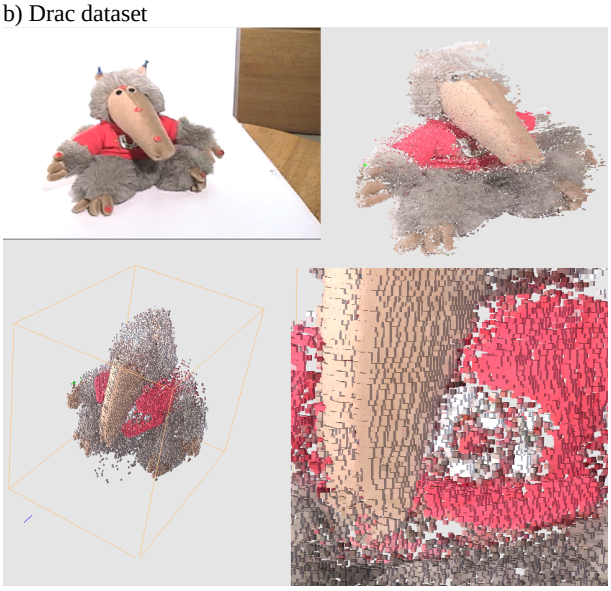
[1] Broadhurst A. and Cipolla R., A statistical consistency check for the space carving algorithm. *Proc. British Machine Vision Conference*, I, 282–291, 2000.



a) House dataset



b) Drac dataset



c) Anteater dataset

Figure 1. For each of the datasets (a,b,c), the top left image shows one of the original images. The top right image is a view of the reconstructed model from one of the calibrated cameras. The bottom images show novel views of the reconstructed model.

[2] Broadhurst A., Drummond T.W. and Cipolla R., A probabilistic framework for space carving. *International Conference on Computer Vision*, I, 282–291, 2001.

[3] Buehler C., Bosse M., McMillan L., Gortler S., Cohen M., Unstructured Lumigraph Rendering, *SIGGRAPH '01 Proceedings*, 2001, 425-432.

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

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

[6] Debevec P., Taylor C., and Malik J. Modeling and rendering architecture from photographs.*SIGGRAPH '96 Proceedings*, 1996, pages 11-20.

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

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

[9] Han M. and Kanade T., Creating 3D Models with Uncalibrated Cameras *IEEE Computer Society Workshop on the Application of Computer Vision (WACV2000)*, 9(2), 137–154, 2000.

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

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

[12] Mahamud S. and Hebert M., Iterative projective reconstruction from multiple views. motion *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00)*, 2, 430–437, 2000. 2000.

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

[14] Sainz M., Bagherzadeh N. and Susin A., Recovering 3D Metric Structure and Motion from Multiple Uncalibrated Cameras. *To appear in IEEE Proc. International Conference on Information Technology: Coding and Computing* ,Las Vegas, 2002. (preprint available at http:\\ibmrlab.ece.uci.edu\)

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

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

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

[18] Triggs B., Factorization methods for projective structure and motion *IEEE CVPR96*, 845–851, 1996.

[19] Triggs B., Autocalibration and the Absolute Quadric *IEEE CVPR97*, 609–614, 1997.

[20] Wood D., Azuma D., Aldinger K., Curless B., Duchamp T., Salesin D., Stuetzle W., Surface Light Fields for 3D Photography, *SIGGRAPH '00 Proceedings*, 2000, 287-386.