Interactive Hierarchical Space Carving with Projector-based Calibrations

Martin Granger-Piché, Emric Epstein, and Pierre Poulin

LIGUM
Dép. I.R.O., Université de Montréal

Abstract

We present an interactive reconstruction system based on space carving. The user controls the camera and object positions and their impact on the reconstruction is immediately displayed. The flow of images is used first for silhouette carving, and then representative images are stored in a hemispherical structure for balanced color carving. Projected color points on the measured background or on the object (acquired by structured light) allow for automatic and adapted calibration of the camera. The octree object representation is central to our space carving algorithm: voxels are efficiently carved out at their largest size, images are treated at their appropriate level, and 3D regions are subdivided only when necessary. We conclude by analyzing our results and by discussing our acquired experience.

1 Introduction

Computer graphics is ubiquitous in computer entertainment, computer games, movie special effects, e-commerce, training in virtual environments, etc. In recent years, the astonishing progress of computer graphics hardware, as well as processing power and memory size at reasonable costs, have stirred the imagination for the development of new applications. This explosion of computer graphics applications fosters a growing need for accurate and realistic 3D models of real objects. Cheaper and flexible scanning devices that are adapted to specific needs and conditions must respond to this challenging demand.

Image-based modeling and rendering is attempting to respond to this need. Important contributors to these techniques and to model representations include light fields [14] and lumigraphs [10, 4], visual and opacity hulls [16, 17], automated 3D computer vision reconstruction pipelines [19, 15], etc. Unfortunately, large memory requirements, incident real illumination captured with the model, or expensive calibrated equipment are common limitations that affect the dissemination of these techniques for general 3D acquisition.

Traditional 3D computer vision has also developed many active and passive reconstruction techniques, including stereo vision, optical flow, laser scanner, structured light, space carving, etc. Several surveys address the pros and cons of these techniques [2, 5, 30].

Typically, all reconstruction processes follow the pipeline:

1. photo/video acquisition;
2. manual/automatic image/frame cleanup and calibration;
3. system variables initialization;
4. 3D reconstruction;
5. reconstructed 3D model display.

However this linear process, with some lengthy steps that provide no intermediate feedback, is less appropriate to acquire realistic 3D models specifically adapted to one's needs.

1.1 Interactive Reconstruction

Several academic and industrial reconstruction systems have recognized this limitation and have benefited from user intervention at different stages of the reconstruction process [6, 20, 7, 21, 24, 18, 3]. Because the user understands the semantics of objects in images and his specific needs, he is better suited to indicate where to add underlying polygons, extract meaningful textures, reject defective images, etc. Even though these techniques have proved to reconstruct better 3D models, they all suffer from relatively tedious user interventions, as each new photo requires more work.

Rusinkiewicz *et al.* [27] break away from these interactive systems; they let the user slowly manipulate the real object while displaying the most robust 3D points being reconstructed from a structured-

light algorithm in real time. Only consumer-level equipment is used in this technique. When a post-processing global optimization is performed on all acquired images, they obtain 3D models of quality comparable to laser-scanned models. Unfortunately, their system suffers from the traditional limitations of structured light: the real object must be mostly diffuse with limited texture.

This strategy inspired the development of our system. Our goal is to obtain quality 3D reconstructions with affordable equipment and under a flexible setup. This quality is achieved thanks to a larger number of images acquired and treated on the fly, as well as stored in a balanced structure during the reconstruction process. Interactivity is paramount: the user should be able to move the video camera and/or the object, while observing in real time the 3D model being reconstructed. Therefore automatic and accurate calibration of the camera and pose estimation of the object are essential. They are achieved with a projector that adaptively projects robust calibration color points on the setup or on the object. Our octree representation is also crucial to the efficiency of our space carving algorithm, and its integration in all aspects of reconstruction and display will be presented.

1.2 Space Carving

We chose to integrate interactivity into space carving [29, 13], a popular technique to reconstruct 3D models. In space carving, voxels from a regular 3D grid are coherently projected in calibrated images. The colors of each voxel are gathered for each image it is visible in. Voxels with non-consistent colors or projecting in the background are carved out from the set of voxels. The remaining colored voxels form the reconstructed 3D model. The high coherency of the regular 3D grid traversal allows for very efficient visibility determination and the consistent and conservative color comparison (even for non-diffuse surfaces [31]) ensures that only impossible voxels are carved out. While space carving is most appropriate for textured color objects, it can also work with less textured surfaces as long as silhouettes can be used to chop portions exterior to the object.

2 Our Interactive Reconstruction System

Our simple setup consists only of consumer-level equipment: a computer linked to a video *camera* and a DLP *projector*, a *background* formed by a measured room corner, and a *stand* with real feature color points, over which lies the object to reconstruct. Our setup is illustrated in Figure 1 and is shown in action in the accompanying video [1]. An overview of our interactive reconstruction process is schematized in Figure 2 and is presented in the remaining part of this section, while details about the more important steps are provided in the following sections.

The reconstruction proceeds as follows. After a few initialization steps, the user moves the video camera around the object he wants to reconstruct. The hand-calibrated projector projects feature color points on the background to automatically calibrate each new camera position (Section 2.1). Any new input image is automatically cleaned up (Section 2.5), calibrated, and entered in the dynamic (FIFO) list. If it is considered the most representative in the directional bin it lies in, the new image is also inserted in the static hemispherical structure (Section 2.3). During space carving, each voxel currently on the estimated surface is efficiently treated (silhouette carving and color carving) at its appropriate octree subdivision level for each image it is visible in (Section 2.2). This integration of an octree representation allows for efficient carving and more accurate color comparison. The set of peripheral uncarved voxels are displayed with extracted colors (Section 2.6), so the user can immediately see how the 3D model improves, and concentrate new views where more effective.

The user can move the stand over which the object lies (Section 2.1), and then resume video acquisition and treatment. When finer details are needed in a region, the user first executes a pass of structured light [2, 27], and robust 3D points that are well distributed over the surface are kept. The projector uniquely illuminates an adapted subset of these 3D points, thus allowing the camera to zoom on a region while being calibrated with them instead (Section 2.4). The exploitation of the projector to create calibration points on smaller regions, integrated with the adaptive octree representation, allow us to add details where needed, without facing an explo-

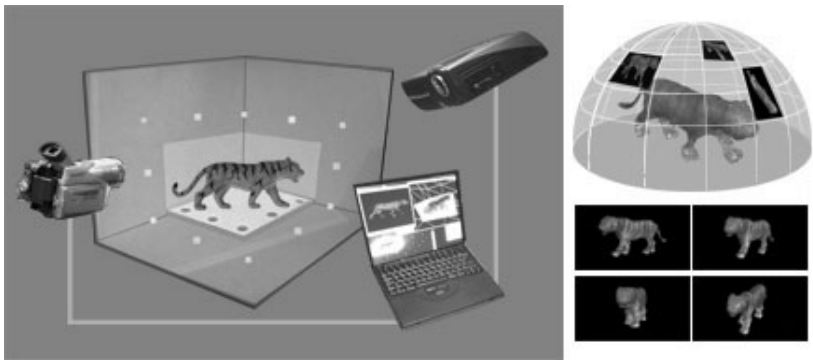

Figure 1: System overview: The stand over which is lying the 3D object to reconstruct is color marked for precise and automatic pose estimation. The calibrated projector adaptively projects color points over the measured room corner in order to automatically calibrate the video camera image. Each processed image from the live video camera is stored in a hemispherical distribution of views (top right) or in a most-recent image buffer list (bottom right) to ensure a balanced set of images for space carving. The current set of color voxels are interactively displayed for immediate reconstruction feedback from the latest set of images.

sion of memory usage.

All these user interventions, augmented by immediate display of the current results, allow better exploiting of the space carving strengths to the particularities of the object being reconstructed, as well as to the expectations of the user. The next sections describe in more detail some of these techniques.

2.1 Calibration and Pose Estimation

The object to reconstruct is positioned in a room corner (background), which is marked with a few measured real feature points. To calibrate the projector, a set of 13 markers of known 3D positions are projected and manually placed over the corresponding feature points of our background. These associated 2D-3D pairs are used in a calibration procedure [8, 23] to extract one affine projection, then transformed in two OpenGL matrices (GL_PROJECTION and GL_MODELVIEW) for the projector. The choice of resolving an affine calibration is critical for the projector, as the image center is often away from the optical axis, as opposed to normal perspective calibration.

Corners and edge midpoints of a synthetic cube are then projected over the background. The image of these projected color landmarks is captured by the camera and cleaned up (Section 2.5) to simplify the association of the 3D projected feature points, with their 2D positions in the camera image. The 2D-3D pairs are used to automatically calibrate the camera, as for the projector. This automatic camera calibration allows the user to freely move the camera and compute its parameters while the flow of images is acquired.

The object is placed over a rigid stand, marked with real feature color landmarks. Initially, the user roughly indicates in the calibrated camera image a pixel in each marker. The system automatically centers all these points so they better fit the camera image of the markers. When the object is moved, the markers identified in the camera image are used to compute their corresponding 3D positions, which enables the extraction with a least-squares approach [22] of a rigid body transformation (horizontal translation and rotation). The redundant information of our 8 or more markers is useful when some markers disappear due to occlusion by the object.

All these calibration points are illustrated in Figure 3.

2.2 Hierarchical Object Representation

Usually, space carving produces better 3D models when all visible voxels project within about one pixel in all images. It also simplifies the color comparison process because fewer colors need to be acquired and tested for a valid voxel visible in an image. However this requires that images be taken

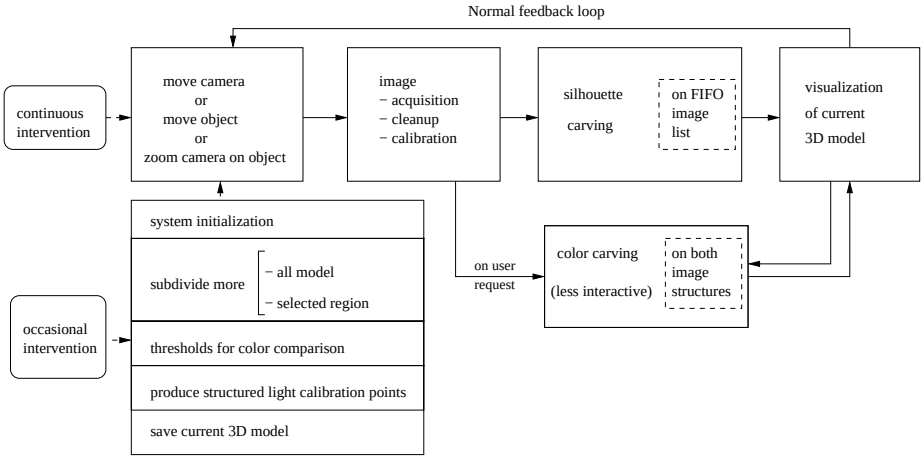

Figure 2: Pipeline of our interactive space carving system

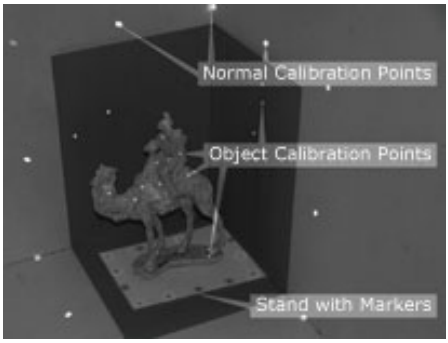

Figure 3: Two sets of projected points are used to calibrate the camera: color points projected on the measured background for normal camera calibration; white points acquired by structured light and projected on the object and the background around it. The pose of the stand is estimated with its real color markers.

from about the same distance. As object features are not always of the same size, an adaptive representation is preferable to capture these details.

We use an octree to represent our 3D model, and we integrate this representation in several aspects of the space carving algorithm (Section 2.3). All voxels of the octree are treated hierarchically, from the top to the subdivision level corresponding to the projection of about one pixel in the current image, or to the specified maximum subdivision level. This representation reduces the memory consumption of our system since only visible voxels that lie on the surface are subdivided. Consequently, the memory space required is $O(n^2)$ instead of $O(n^3)$. The representation also improves the efficiency of the space carving procedure because large chunks of voxel space can be discarded rapidly at smaller resolutions. Moreover, the octree structure allows for images taken at different distances of the object to be treated correctly for both silhouette carving and color carving. That would be impossible if we were to use a regular grid.

2.3 Space Carving

Every image from the flow of incoming images is automatically calibrated, segmented, and added to our image structures. In silhouette carving, a voxel is removed when it projects completely on background pixels in an image. The voxel's children are removed as well in that case. Otherwise, the voxel is subdivided until its projected area occupies between 1 to 4 pixels in the images it is visible in. A user threshold controls the maximum level of voxel subdivision to carve silhouettes at lower resolutions. The complete reconstruction process takes a user about 1 to 3 minutes to produce a coarse model carved out of a 32^3 voxel space. This includes image acquisition, clean up, calibration, segmentation, and silhouette carving. It is then possible to gradu-

ally subdivide the voxels in order to reach the appropriate level for a given image resolution. We found this strategy to respond well to our expectations.

Color carving removes *visible* voxels that are considered impossible (incoherent) due to color differences in images. It operates best in highly-textured regions, and it even applies to concave regions without holes where silhouette carving is impossible. Color carving is however more computationally intensive, and is applied only when requested by the user. It must test the colors of a voxel in all visible images to check if the voxel can be removed, and this test must be conservative. Because of the continuous flow of images, we cannot keep all images at all times; yet, we need a balanced set of static images (well distributed) and dynamic images (more weight on most recent ones) to ensure proper color comparisons.

We subdivide the hemisphere above the 3D model into a number of $\theta \times \phi$ directional bins (typically 3×32) [10]. We only keep the calibrated image that falls closer to its bin center and that is oriented toward the object center.

Each new image is added to a FIFO dynamic list of images, and is treated from the most to the least recent. A typical dynamic list has 8 images. All dynamic images and a subset of the static images are used for each pass of color carving during the acquisition process. This subset has typically the same number of images as the dynamic list, and they are randomly chosen at each pass among the hemisphere bins. For user-requested offline color carving, taking typically seconds to a few minutes, the images in all the bins are used.

To gather the colors of all visible voxels, we project all voxels as transformed OpenGL cubes (with backface culling) in each image of the set. Only the voxels residing at the proper level of subdivision for a given image are projected in it. A voxel is colored with its ID encoded in a 32 bit color stored in a hash table. For each such rendered image, we read back the IDs and assign the corresponding image colors to its visible voxel. Therefore a voxel that is visible in at least one current image potentially contains a list of colors (if the voxel projects in more than one pixel) for each image it is visible in. We always consider at least a 4×4 window of pixels to limit the problems due to possible calibration error, image noise, and compression artefacts; we apply the color variance test of Kutulakos [12]. This color test tries to find similar pixel colors in each window within every image the voxel appears. If this color does not exist, the voxel is removed.

Each time the stand is moved, the illumination may change the appearance of the object. Because we usually take several images with the object in the same position, we can apply the color test on each group of images with the same illumination. If one group fails the test, the voxel can be safely removed.

Because hidden interior voxels can become visible each time a voxel is carved out, color gathering must be performed for all current images and all projector configurations after each pass of space carving. We did not optimize this process; we re-render all voxel IDs in all current images and all projector/stand configurations. This ensures that voxel information is up-to-date, which is crucial in our context. Nonetheless our system remains interactive.

By moving the video camera, the user can influence the space carving voxel elimination. New views with easily identified new silhouettes quickly slice out entire sections of voxels. New views from a nearby neighbourhood increase the ratio of similar colors and thus reduce color variance for visible voxels. Direct feedback display of the resulting carving process allows the user to quickly decide what to do to improve the reconstruction of his 3D model.

2.4 Close-up Reconstruction

When the user would like to add fine details over a region of the object, moving the camera closer would likely lose all projected feature color points. Therefore new calibration points are needed, but this time directly over the object.

We execute a pass of structured light [26] to generate 3D points. The camera is first calibrated for its current un-zoomed position with the usual color landmarks on the background. Then a group of 3D points generated by structured light are tested for robustness according to the current calibrated camera. From these points we select only those that reproject as close as our calibration would expect, that is, sufficiently distant from each other to not confuse them in the images and as less coplanar as possible. If both calibrations (current and with the subset of structured light points) are equivalent, we

can then use the structured light points to zoom on the object (Figure 4). Because we use more such calibration points than necessary, we can afford to lose some of these calibration points due to occlusion by the object as the camera moves closer and around the region of interest. New points from the unused robust structured light calibration points are selected when more points from the subset are lost.

2.5 Input Image Processing

In order to efficiently calibrate the equipment, carve out the silhouettes, identify the color landmarks, perform color comparison, etc., we need to clean up the incoming images. In fact, we must carefully consider the limited quality of our video images: 720×480 pixels, some JPEG compression artefacts, and darker and somehow noisy colors (images must be treated with and without projector illumination, while avoiding over-saturated colors as much as possible). While none of the used well-known techniques [9] are crucial, they help for both robustness and efficiency while exploiting the advantages of the projector setup. An M.Sc. thesis [11] goes well within these details and is omitted here due to space constraints.

2.6 Display

As we know the current video camera calibration and its equivalent OpenGL matrices, we use these parameters to display the 3D model during its reconstruction. This lets the user quickly assess the impact of acquired images on the space carving. The voxels can be displayed at the current level of subdivision or at the level requested by the user. The assigned color of a voxel corresponds to the *most similar color* from the color carving test.

An intermediate or final 3D reconstructed model (exterior voxels) can be saved as 3D color points, with approximative surface normals, to a Q-splat [28] or Pointshop [25] display system. This allows us to manipulate the viewpoint, surface shading, and incoming illumination while maintaining real time display. All the results shown in this paper are displayed with Q-splat [28].

3 Results

All these steps (image acquisition, calibration, cleanup, silhouette carving, valid voxel projection, color acquisition and color comparison for voxel carving, color selection for display) are processed at interactive rates on a dual Pentium IV Xeon processor running at 2.4 GHz with 1 GB of memory and an nVIDIA GeForce4 Ti 4200 graphics card.

The digital video of 720×480 pixels from a Panasonic PVGS-70 camera is delivered through a firewire connection at 29.97 fps, with a slight JPEG compression. The DLP Compaq MP4800 projector emits 2100 lumens at a 1024×768 pixel resolution.

The room corner measures $81 \times 61 \times 63\ cm$ and is coated with mat white paint. It has a few dozens of measured calibration pencil marks.

Image calibration, including time to clean up the input data, identify the 2D color points, and segment the background takes about 0.17 seconds per new image. On average, we achieve subpixel error for the reprojected 3D feature color points.

In the next table, we show an estimate in seconds of some carving experiments we ran for different voxel resolutions. The first column shows the average time to identify voxel IDs and to carve out those that are projecting in the background for one image. The second column shows the average time to apply the color test on voxels once all the colors are gathered during the visibility step.

Voxel Space Resolution	Silhouette Carving	Color Carving
32^3	0.06 sec.	1.3 sec.
64^3	0.15 sec.	1.9 sec.
128^3	0.25 sec.	3.2 sec.
256^3	0.50 sec.	4.6 sec.

A conservative estimate of the distribution of time for a typical reconstruction session with our system consists of 3 mins for projector calibration and pre-processing steps, 10 to 30 mins for video acquisition with 3D reconstruction feedback and 20 mins or more for closer image acquisition and offline color carving.

In the current setup, given the 3D objects we scanned, an 128^3 to 256^3 voxel octree provided satisfying results while a 512^3 voxel octree provided more surface details. Usually, the final 3D models consist of a few hundred of thousands of final (subdivided) voxels, obtained after treating about 400 video images.

Some of our results are illustrated in the color

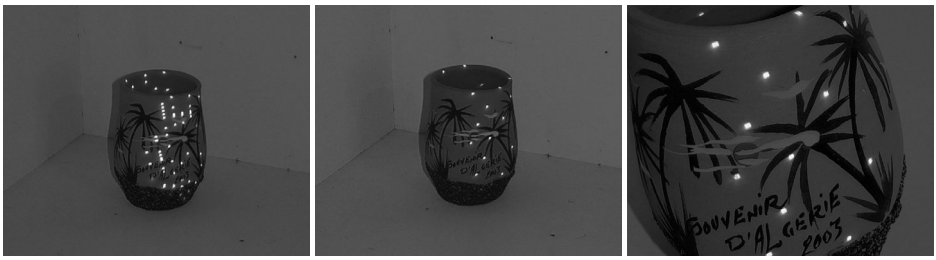

Figure 4: (Left) A set of points acquired from structured light are stored as candidates for further calibration. (Center) A few points from the entire set are selected given their robustness with respect to the current camera position. (Right) These calibration points and other from the set are used for close-up reconstruction.

plate. More results are presented on our website [1] and in the accompanying video.

4 Conclusion and Future Work

We have presented a functional interactive 3D reconstruction system based on space carving involving affordable consumer-level equipment. The user can manipulate live input (video camera) and change the object's position to interactively control where details are needed, using the simultaneous feedback display of the currently reconstructed 3D model. By organizing the treatment of the flux of incoming images into two structures, we ensure good distribution of images for color comparisons, while providing some emphasis on the most recent images. The exploitation of a projector for adaptive calibration has proven flexible and useful. The refinement of voxels in an octree allows for efficient memory management and for improved 3D models. The experiments and direct feedback are also very useful as an image selection, automatic calibration, and system variable setting to be used as pre-processing for an offline space carving. Validated positions of robust 3D reconstructed points have been used as new calibration points to zoom even further over the 3D object. The octree is therefore more appropriate to provide detailed 3D models and could be extended to a capture process for level-of-detail 3D filtered geometry.

Our framework has only touched the wide potential of using a projector in the context of reconstruction. A tighter integration of the structured light reconstruction algorithm appears quite promising. As well, BRDF extraction and shadow silhouettes should benefit from our controlled illumination. Other highly interesting avenues involve capturing reflective and refractive light patterns in order to reconstruct surface properties.

5 Acknowledgment

We would like to thank Mathieu Ouimet, Denis Vontrat, and Neil Stewart for their help, and Caroline Lacasse for her culinary support. We acknowledge financial support from FCAR and MITACS.

References

[1] www.iro.umontreal.ca/labs/infographie/papers.

[2] F. Bernardini and H. Rushmeier. The 3D model acquisition pipeline. In *Eurographics 2000: STAR*, September 2000.

[3] Boujou. www.2d3.com.

[4] C. Buehler, M. Bosse, L. McMillan, S.J. Gortler, and M.F. Cohen. Unstructured lumigraph rendering. In *Proc. SIGGRAPH 2001*, pages 425–432, August 2001.

[5] Q. Chen and G. Medioni. A volumetric stereo matching method: Application to image-based modeling. In *Proc. Computer Vision and Pattern Recognition*, pages 29–34, June 1999.

[6] P.E. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proc. SIGGRAPH 96*, pages 11–20, August 1996.

[7] S. Dedieu. *Adaptation of a 3D Model Reconstruction System from Photos to User Knowl-*

edge. PhD thesis, Université Bordeaux I, December 2001.

[8] O. Faugeras. *Three-Dimensional Computer Vision — A Geometric Viewpoint*. MIT Press, 1993.

[9] R.C. Gonzalez and R. Woods. *Digital Image Processing*. Prentice Hall, 2nd edition, 2002.

[10] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen. The lumigraph. In *Proc. SIGGRAPH 96*, pages 43–54, August 1996.

[11] M. Granger-Piché. Interactive hierarchical space carving with projector-based calibrations (in french). M.Sc. thesis, DIRO, Université de Montréal, December 2004.

[12] K.N. Kutulakos. Approximate n-view stereo. In *European Conference on Computer Vision*, pages 67–83, 2000.

[13] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

[14] M. Levoy and P.M. Hanrahan. Light field rendering. In *Proc. SIGGRAPH 96*, pages 31–42, August 1996.

[15] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, to appear, 2004.

[16] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan. Image-based visual hulls. In *Proc. SIGGRAPH 2000*, pages 369–374, July 2000.

[17] W. Matusik, H. Pfister, A. Ngan, P. Beardsley, R. Ziegler, and L. McMillan. Image-based 3D photography using opacity hulls. *ACM Trans. on Graphics*, 21(3):427–437, July 2002.

[18] Photomodeler. www.photomodeler.com.

[19] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.

[20] P. Poulin, M. Ouimet, and M.-C. Frasson. Interactively modeling with photogrammetry. In *Eurographics Workshop on Rendering 1998*, pages 93–104, June 1998.

[21] P. Poulin, M. Stamminger, F. Duranleau, M.-C. Frasson, and G. Drettakis. Interactive point-based modeling of complex objects from images. In *Graphics Interface 2003*, pages 11–20, June 2003.

[22] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1992.

[23] R. Raskar. Camera calibration. *SIGGRAPH Course Notes 21*, July 2003.

[24] RealViz. www.realviz.com.

[25] L. Ren, H. Pfister, and M. Zwicker. Object space ewa surface splatting: A hardware accelerated approach to high quality point rendering. In *Proc. Eurographics 2002*, pages 461–470, 2002.

[26] C. Rocchini, P. Cignoni, C. Montani, P. Pingi, and R. Scopigno. A low cost 3d scanner based on structured light. *Computer Graphics Forum*, 20(3):299–308, 2001.

[27] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3D model acquisition. *ACM Trans. on Graphics*, 21(3):438–446, July 2002.

[28] S. Rusinkiewicz and M. Levoy. Qsplat: A multiresolution point rendering system for large meshes. In *Proc. SIGGRAPH 2000*, pages 343–352, July 2000.

[29] S.M. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999.

[30] R. Szeliski and P. Golland. Stereo matching with transparency and matting. *International Journal of Computer Vision*, 32(1):45–61, August 1999.

[31] R. Yang, M. Pollefeys, and G. Welch. Dealing with textureless regions and specular highlights – a progressive space carving scheme using a novel photo-consistency measure. In *Proc. Int. Conf. on Computer Vision*, pages 576–584, July 2003.