8

Reconstruction of Volumetric 3D Models

Peter Eisert

Fraunhofer Institute for Telecommunications, Berlin, Germany

8.1 Introduction

Photo-realistic 3D computer models of real objects and scenes are key components in many 3D multimedia systems. The quality of these models often has a large impact on the acceptability of applications like virtual walk-throughs (e.g., city guides or virtual museums), caves, computer games, product presentations in e-commerce, or other virtual reality systems. Although the rendering of 3D computer scenes can often be performed in real-time even on hand-held devices, the creation and design of 3D models with high quality is still time consuming and thus expensive. This has motivated the investigation of a large number of methods for the automatic acquisition of textured 3D geometry models from multiple views of an object.

The large body of work devoted to this problem can basically be divided into two different classes of algorithms. The first class of 3D model acquisition techniques computes depth maps from two or more views of the object. Here, depth is estimated from changes in the views caused by altering properties like position of cameras (shape-from-stereo, shape-from-motion), focus (shape-from-focus, shape-from-defocus), or illumination (shape-from-shading, photometric stereo). For each view, a depth map can be computed that specifies for each pixel the distance of the object to the camera. Since the object is only partially represented by a single depth map due to occlusions, multiple depth maps must be registered into a single 3D surface model.

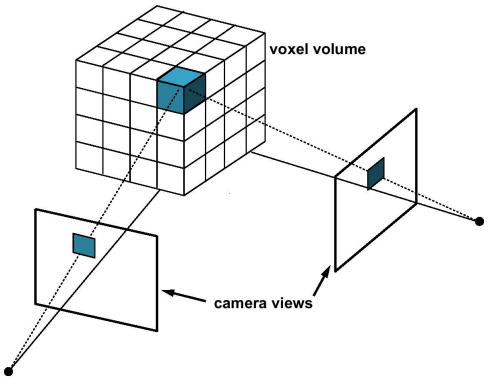

Figure 8.1 Representation of a scene by a 3D array of volume elements (voxels).

This registration process can often be avoided in a second class of reconstruction methods which relies on a volumetric description of the scene. In the simplest case, the space containing the object or scene to be reconstructed is equidistantly divided into small cubical volume elements called *voxels* (see Fig. 8.1). If a particular voxel does not belong to the object it is set transparent, whereas voxels within the object remain opaque and can additionally be colored. Thus, the entire scene is composed of small cubes approximating the volume of the objects. The finer the discretization of the 3D space, the more accurate the shape.

One advantage of volumetric representations for reconstruction purposes is the simple joint consideration of all available views. Once the cameras are calibrated, each voxel can be projected into all views as shown in Fig. 8.1 and the information of the corresponding pixels can determine whether the voxel belongs to the object or not. Instead of fusing multiple depth maps computed, e.g., from pairs of images, the joint computation leads to 3D computer models that are consistent with all views. Since less smoothness constraints are incorporated, volumetric methods can often be well exploited to reconstruct fine structures with multiple occlusions.

This chapter focuses on methods for the reconstruction of volumetric computer models from multiple camera views. The emphasis is on vision-based applications where the 3D representation usually describes a colored sampled surface. The interior of the object is here of less importance. Other applications for volumetric data that exploit the entire volume for visualization (e.g., computer graphics or medical applications like computer tomography) are not considered in the following. The chapter is organized as follows. First, the important class of *shape-from-silhouette* algorithms is reviewed. Due to its simplicity, robustness, and efficiency, this method is very popular in many multimedia applications [Goldlücke and Magnor (2003); Grau (2003); Gross (2003); Wu and Matsuyama (2003)]. One drawback of shape-from-silhouette methods is their inability to recover concavities in surfaces. If color information is additionally considered it is possible to recover such surfaces as well. Volumetric algorithms that exploit color information are, e.g., space carving algorithms described in Section 8.3 and the image cube trajectory analysis presented in Section 8.4.2.

8.2 Shape-from-Silhouette

Shape-from-silhouette or *shape-from-contour* is a class of algorithms for 3D scene reconstruction that uses the outline of the objects to recover their shape. Especially under controlled situations, e.g. in studio scenarios, the silhouette of an object can be determined very reliably. Therefore, these methods are very robust against lighting changes or photometric variations between cameras which are critical for other algorithms based on the brightness constancy assumptions [Horn (1986)].

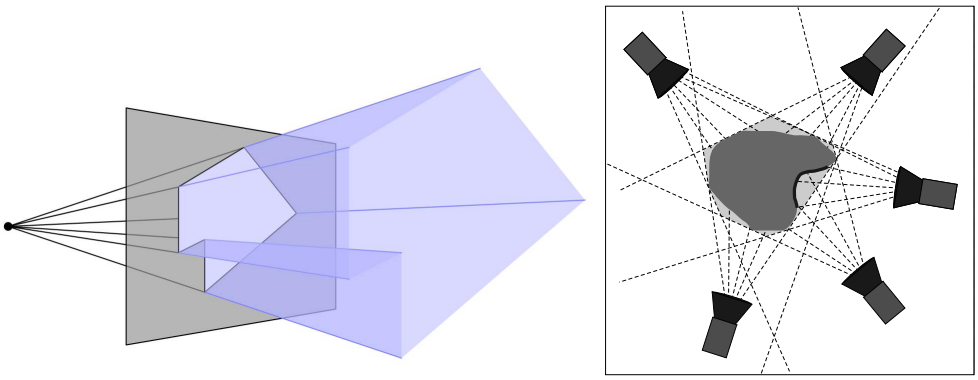

Figure 8.2: **Left:** Viewing cone through the image silhouette containing the object. **Right:** Intersection of multiple viewing cones.

The concept of shape-from-silhouette dates back to the early seventies [Baker (1977); Greenleaf et al. (1970); Martin and Aggarwal (1983)] and was initially used for medical applications. The basic idea of these approaches is that any object must be entirely located somewhere within its contour. If an object is viewed from a particular known position under perspective projection the rays from the focal point through the silhouette contour form the hull of a viewing cone. This viewing cone, illustrated on the left hand side of Fig. 8.2, defines an upper bound for the object shape – the correct object volume is definitely less or equal than this rough approximation. In this consideration, no assumption about the viewing position is made except for the knowledge of this calibration data. For any viewing position, the silhouette defines a viewing cone which entirely contains the object. Since the object volume is bounded by all particular cones, it must also reside within the intersection of these viewing cones. Only points in the 3D space that are inside all viewing cones may belong to the object to be reconstructed. This is shown on the right hand side of Fig. 8.2.

The algorithm starts without any restrictions on the object shape. A large bounding volume enclosing the entire scene can be used as initialization. Multiple views from different viewing positions are captured. Each view provides a viewing cone which is intersected with the current object shape approximation leading to a volume with increasing accuracy. However, not all possible shapes can be reconstructed with shape-from-silhouette methods. The concavity, e.g., on the right hand side of Fig. 8.2 is never visible in the silhouette (independent from the viewing direction) and cannot

be recovered. Instead of the true shape, only the *visual hull* [Laurentini (1994)] can be estimated. However, for many practical applications this leads to sufficiently accurate representations. In order to model, e.g., small bumps in the surface, additional information like color has to be evaluated as it is done in *space carving* or *voxel coloring* approaches described in Section 8.3. The entire procedure of shape-from-silhouette consists of the following steps:

- Calibration of the cameras to determine position, orientation, and intrinsic camera parameters
- Segmentation of the object from the background in the captured images to derive the object contour
- Intersection of all viewing cones

The intersection of the viewing cones can be computed exactly with polyhedral representations [Matusik et al. (2001)]. But most often, a volumetric discretization of the space is used as shown in Fig. 8.1. For a particular voxel with given 3D position, its projection into the camera frames is computed. If the voxel falls outside the silhouette in at least one view, it is discarded from the volume. After all voxels are processed, the remaining volume elements approximate the visual hull. Although some aliasing effects may occur, this discrete volume intersection is very popular due to its low complexity. With current technology, even real-time reconstruction is possible [Goldlücke and Magnor (2003); Wu and Matsuyama (2003)] enabling new interactive 3D applications in computer vision and graphics.

8.2.1 Rendering of Volumetric Models

Once a 3D volume is reconstructed, it can be viewed from different directions. For rendering, a color can also be assigned to each voxel. This color information is extracted from those camera views, where the corresponding voxel is visible. In order to consider occlusions of multiple voxels, a z-buffer may be added for rendering. Each voxel is then projected into the image plane and modifies the pixel color if it is closer to the virtual camera than the current value stored in the z-buffer. Different approaches can be classified by the direction of volume traversal (front-to-back, back-to-front, or arbitrary traversal).

The simplest approach in voxel rendering is to assume a voxel being infinitesimally small and to use it to exactly color one pixel. However, dependent on the viewing distance and the discretization of the image and volume space, this method can lead to aliasing or holes in the surface. Better results can be obtained if the finite size of the projected voxel is considered. Fig. 8.3 shows the projection of a cube into the image plane that results in a 2D polygon of up to 6 corners that may cover multiple pixels. Due to the discrete nature of pixels, special attention must be paid to the boundary of the polygon in order to avoid aliasing. Since the use of 6-sided polygons for each voxel is computationally demanding, the footprint of a voxel can also be approximated by simpler shapes. For example, the bounding box, spheres, or Gaussian functions can be used for projection leading to techniques like splatting, surfels, and point-based rendering [Pauly et al. (2003); Rusinkiewicz and Levoy (2000); Westover (1990)].

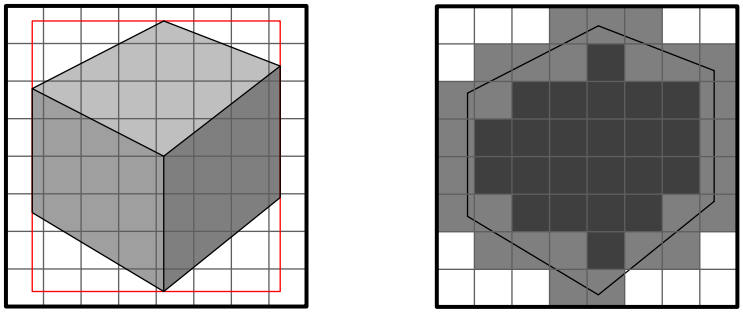

Figure 8.3: **Left:** Voxel projection onto image grid. **Right:** Voxel footprint and pixel contribution.

Figure 8.4: **Left:** Voxel volume of a Peruvian vase reconstructed from 72 images. **Middle:** Triangle mesh derived from the voxel volume. **Right:** Textured 3D model.

Often, the voxel volume is not rendered itself but converted into a triangle mesh that can efficiently be handled by a graphics card [Niem and Wingbermühle (1997)]. Modeling the outer surface of the cubical structures with polygons may lead to rough surfaces with discrete steps. Better results can be obtained using *marching cubes* [Lorensen and Cline (1987)] or *marching intersections* [Tarini et al. (2002)] methods that use a local neighborhood of voxels to derive oriented surface patches. After smoothing and mesh reduction, efficient representations of the 3D objects can be created. Fig. 8.4 shows an example of a vase reconstructed with shape-from-silhouette techniques and rendered as a voxel volume (left), a reduced triangle mesh (middle), and as a textured 3D model.

8.2.2 Octree Representation of Voxel Volumes

Volumetric descriptions tend to impose high demands on memory due to their three dimensions. Therefore, efficient representations of large voxel volumes are essential. Mostly, *octrees* are used in order to represent the 2D surface with a 3D voxel grid structure [Chien and Aggarwal (1986); Potmesil (1987); Srivastava and Ahuja (1990); Szeliski (1993); Veenstra and Ahuja (1986)]. In this case, an octree is a tree that hierarchically defines the object shape starting with a root node that covers the entire bounding volume. Each cube represented by a node in the tree is then successively subdivided into 8 smaller cubes (dependent on the contour information) with the cube's edge lengths halved. Further subdivision of a cube is stopped if it either lies completely inside or outside the object.

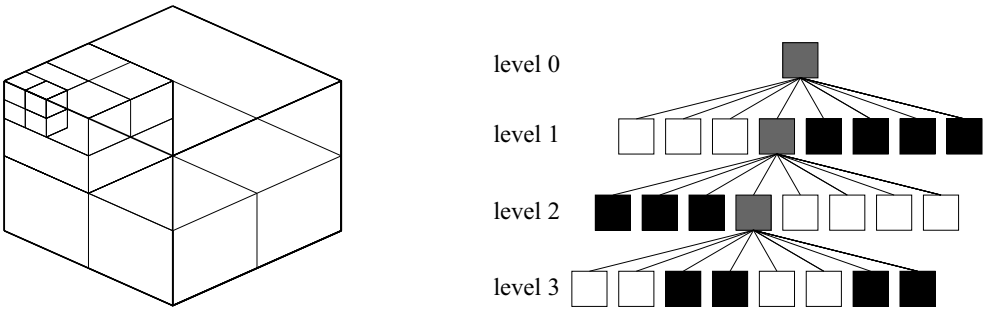

Figure 8.5: **Left:** Octree representation of an object. **Right:** Tree structure defining subdivision.

Fig. 8.5 shows an example for such an octree subdivision. The volumetric object on the left is defined by the tree on the right. Each node in this tree specifies if one of its sub-cubes lies completely inside (black), completely outside (white), or on the object surface (gray). In the latter case, the corresponding voxel is further subdivided until a maximum layer defining the highest resolution is reached. This way, it is assured that the object surface is accurately modeled while regions outside or inside the object can be efficiently described by large cubes. Fig. 8.6 gives an example of a 3D object represented with different maximum octree levels. For practical scenarios, drastic memory savings can be achieved compared to the full 3D voxel volume with the same resolution.

For hierarchical shape-from-silhouette, an octree can be constructed bottom-up or top-down. If reconstruction is started from level 0 with recursively refining the object, the spatial extension of the cube must be considered for projection and the footprint of the voxel as illustrated in Fig. 8.3 need to be computed. The intersection of all viewing cones in the shape-from-silhouette procedure can be performed as follows [Smolic et al. (2004)]. A voxel of a particular resolution level that is completely inside the silhouettes of all views belongs to the object and is not further subdivided. Similarly, a cube whose footprint is completely outside the silhouette of at least one view is marked as outside and is not further refined. All other voxels are subdivided into 8 sub-cubes and the process is continued on all sub-cubes until a predefined resolution

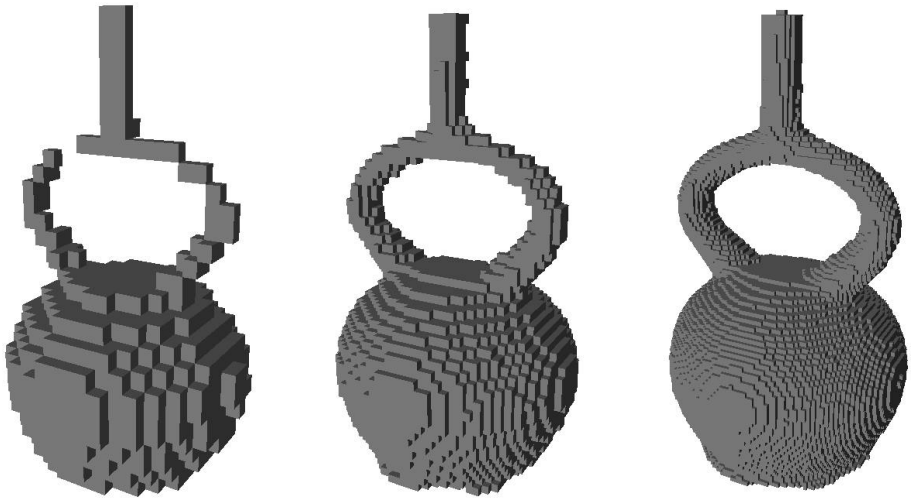

Figure 8.6: Voxel volume of the vase in Fig. 8.4 rendered up to octree level 5, 6, and 7, respectively.

is reached.

8.2.3 Camera Calibration from Silhouettes

For shape-from-silhouettes algorithms, the accuracy of the reconstructed geometry is considerably affected by the knowledge of the true camera parameters which requires an accurate camera calibration. Deviations from the correct values can lead to incorrect 3D models, since valid object parts might be cut off during viewing cone intersection. As a result, the silhouette of the reconstructed object is always smaller or equal to the true contour. However, this unwanted effect can be used to refine the camera parameters again. In Grattarola (1992), pairs of images are used and the viewing cone of one image contour is projected into the other image plane, computing silhouette mismatches. Minimization of these mismatches over all camera parameters optimizes the calibration. Similarly, Niem [Niem (1998)] minimizes the deviation of the back-projected silhouette of the reconstructed object to the true silhouette in the camera images. Both approaches have in common, that a non-linear optimization in a high dimensional space has to be performed.

For the particular case of turn-table scenarios where shape-from-silhouette is often used, additional constraints apply that simplify the optimization of camera parameters. Instead of placing multiple cameras around the object, a single camera is used capturing images while the object slowly rotates. The rotation angle between two shots can usually be accurately controlled, whereas the position of the camera relative to the turn-table is in general unknown and requires camera calibration. The knowledge of circular motion adds severe constraints which increase robustness and result in a very low dimensional parameter space that has to be searched. If the size

of the reconstructed object need not be recovered three parameters are sufficient to determine the entire extrinsic camera geometry [Eisert (2004)]. Similar constraints are also utilized in [Fitzgibbon et al. (1998)], where extrinsic and intrinsic camera parameters are derived from feature point correspondences instead of silhouette.

Figure 8.7: Geometry deviations for a turn-table sequence. **Left:** Original camera frame. **Middle:** Silhouette mismatch between original and synthetic view. **Right:** Silhouette error after refinement.

Fig. 8.7 shows the result from a camera refinement exploiting silhouette mismatches. The left hand side of this figure is an original camera frame from a turn-table sequence of a small tree. Mismatches (white pixels) between the silhouette of the camera frame and the silhouette of the reconstructed object with only inaccurate initial calibration are depicted in the middle. These deviations can be exploited in order to optimize the camera parameters leading to the smaller deviations shown on the right. Due to the improved camera parameters, the accuracy of the reconstructed 3D model is highly increased by this preprocessing step making initial camera calibration less important or even unnecessary.

8.3 Space Carving

In shape-from-silhouette methods, binary masks specifying the 2D object contours in the images are used as input in order to estimate the 3D object shape. Color information is not exploited. This leads to very fast and robust algorithms but not all object shapes can be reconstructed accurately. The recovered object is bounded by the visual hull – small dents and concavities cannot be determined. In the extreme case, for example the reconstruction of rooms in indoor environments with centered outwards facing cameras, no silhouette information is available and the reconstruction fails. This limitation can be overcome if color information from the images is used in addition to the binary contour masks. *Voxel coloring* [Seitz and Dyer (1997)], *multi-hypothesis volumetric reconstruction* [Eisert et al. (1999)], and *space carving*

[Kutulakos and Seitz (2000)] belong to this group of algorithms that exploit color in order to improve the reconstruction accuracy.

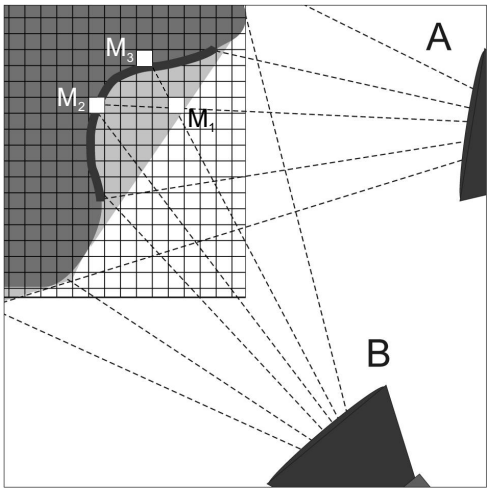

Figure 8.8 Refinement of object shape using surface color.

The benefit of exploiting surface color information is exemplarily illustrated in Fig. 8.8. The image shows an object with a small concavity which cannot be detected using contour information only. Textured surfaces, however, allow to distinguish between the correct shape and the visual hull. Imagine a point M_1 lying on the visual hull and being projected on particular pixels of cameras A and B, respectively. Since the true shape contains a concavity at that position, the pixels in both cameras are colored from two different points on the surface M_2 and M_3 instead of M_1. If the surface is textured these points and therefore the pixels may have different colors. Thus, point M_1 cannot belong to the true object volume and can be removed. Space carving methods use that phenomenon in order to carve out concavities from the visual hull like a sculptor from a block of stone until the object surface looks correct in all views.

Shape-from-silhouette approaches reconstruct a binary voxel volume that has the same silhouettes as in the camera images. Similarly, space carving or voxel coloring methods create a colored voxel representation, where the volume reprojections into the image planes show the same pixel colors as the original views. Again, this need not be the correct geometry of the object. It is only assured that the rendered views are *photo-consistent* [Seitz and Dyer (1997)] with the camera views. There are multiple possible geometries that all lead to photo-consistent views. Since generally no smoothness assumptions are made, there are different voxel configurations (with mostly local variations at the object surface) that result in the same projections into the finite number of available views. The union of all photo-consistent configurations is called *photo-hull* following the concept of the visual hull.

The basic principle of all volumetric color reconstruction methods is quite similar.

All voxels in the volume are usually processed one after the other. For each voxel, its projection into all available views is calculated and, depending on the pixel colors at the corresponding 2D image positions and considering visibility, it is determined whether the voxel belongs to the object surface or not. If it lies on the surface, the voxel color is computed from the pixel colors in those views, where the voxel is visible. In the other case, the voxel is set transparent and the algorithm proceeds with the remaining volume elements until the reconstructed object looks correct from all available viewing directions. Thus, all available views are considered simultaneously without the need of determining point correspondences or fusing multiple depth maps into a common 3D model.

Although the principle of projecting voxels into all available views is common to a wide class of algorithms, there are many differences in the way how the voxel volume is traversed, how visibility is considered, and how photo-consistency is checked. For the consistency test, the voxel is projected into all visible views and the pixel colors are compared. In the ideal case, all these colors should be the same, since they represent the same surface point. In practice, however, camera noise, mismatch in calibration data, or non-Lambertian surface reflections lead to variations between the views. This can be considered by allowing some color deviations for the test. If this range is too small, holes can occur in the model, if the threshold is too large, the resulting voxel object might be too large.

Dependent on the camera configurations, different voxel access schemes can be applied. If the convex hulls of all cameras' focal points lie completely outside the objects, the voxel volume can be traversed in a single sweep as it is done in voxel coloring [Seitz and Dyer (1997)]. In this case, a voxel access order can be defined that assures that visibility of a voxel can be determined by considering only already processed ones. For general camera configurations, this is no longer possible. Here, camera views are grouped according to their position and an iterative scheme is applied on the voxel volume, that carves away voxels at the surface until photo-consistency is reached [Culbertson et al. (1999); Eisert et al. (1999); Kutulakos and Seitz (2000)]. Similar to the shape-from-silhouette methods, many extension can be applied, for example the use of octrees to deal with large environments [Prock and Dyer (1998)], or to consider extended voxels for reducing aliasing effects [Steinbach et al. (2000)].

Fig. 8.9 shows examples of objects reconstructed with the multi-hypothesis volumetric reconstruction method described in [Eisert et al. (1999)]. The top left image shows one frame of the *Penguin* sequence recorded in CIF resolution. The algorithm was applied on 65 unsegmented frames leading to the 3D colored voxel representation shown on the upper right of Fig. 8.9. Since the bounding box of the volume does not include the background objects, an inherent segmentation is achieved. In the second example shown on the lower row of Fig. 8.9, an indoor scene of an office is reconstructed from 27 views with no available silhouette information. Although shape-from-silhouette methods would fail, the use of colors allow a photo-consistent reconstruction of the room as shown on the right hand side of Fig. 8.9.

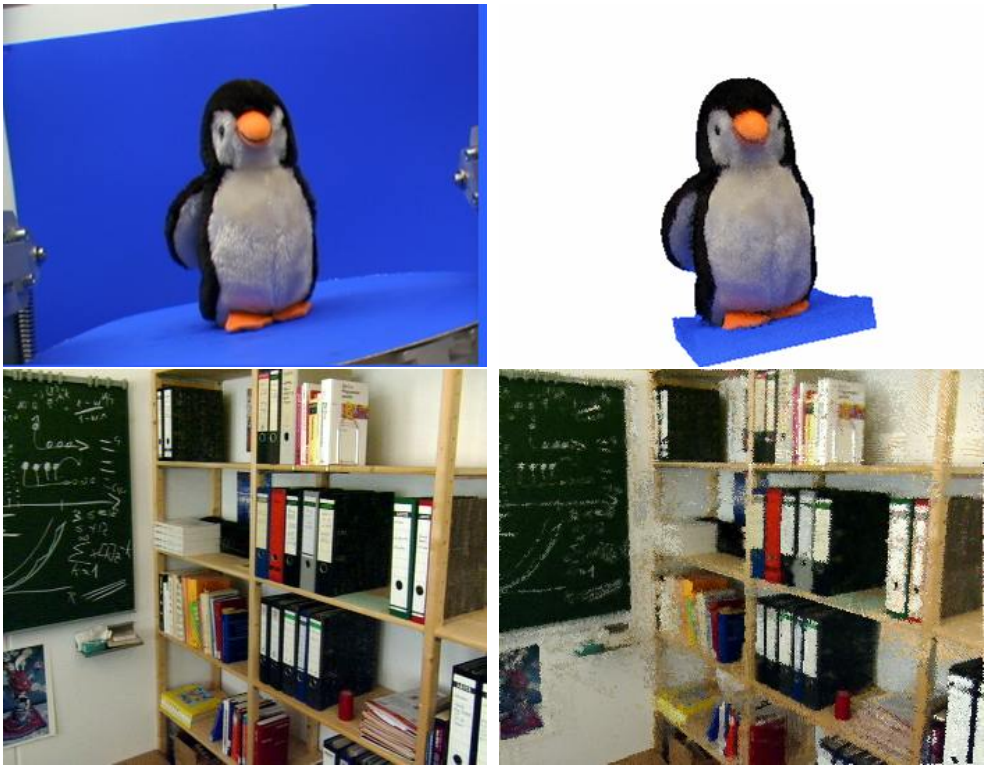

Figure 8.9: **Upper left:** Original view of the *Penguin* sequence. **Upper right:** Reconstructed view. **Lower left:** View of an indoor sequence. **Lower right:** Reconstructed view.

8.4 Epipolar Image Analysis

Similar to space carving, *epipolar image analysis* is a technique for the reconstruction of object shape using color information from multiple calibrated views of the scene. Instead of starting from the 3D volume by projecting single voxels into the available views, epipolar image analysis makes use of particular structures in 2D images to derive 3D information. For that purpose, the sequence of images is usually collated to form a 3D volume, the so called *image cube* [Criminisi et al. (2002)]. In this volume, shown on the left hand side of Fig. 8.10, vertical slices represent the original camera views.

Since the camera moves during acquisition of the slices, object points also change their position in the image cube. For the particular case of a linear and purely horizontal camera movement described in the next section, object point trajectories form lines in the image cube which can be detected easily with image processing techniques. 3D information is then derived from the line parameters. Usually, a large number of views is jointly exploited which allows the inherent consideration of mul-

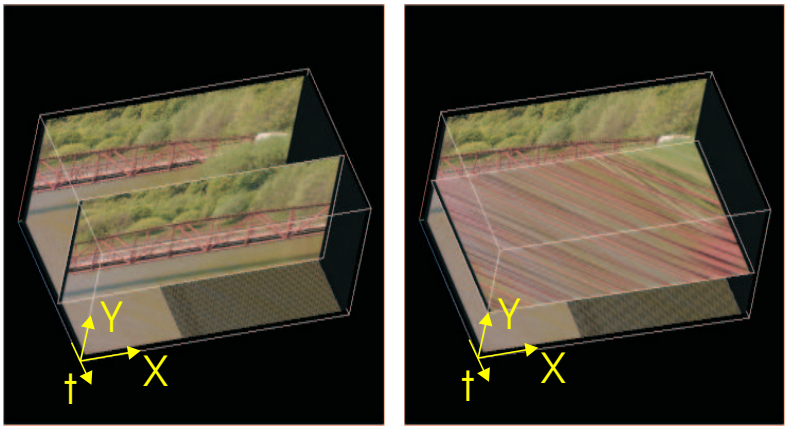

Figure 8.10: Image cube representation of an image sequence. **Left:** Time slices represent camera images. **Right:** Horizontal slices are epipolar images.

tiple occlusions as well as homogeneous regions. The initial method is restricted to linear equidistant camera movements but can be extended to other camera configurations as well. Section 8.4.2 describe a generalization to circular setups occurring, e.g., in turn-table acquisition or concentric mosaics [Shum and He (1999)].

8.4.1 Horizontal Camera Motion

The most important case of epipolar image analysis is the case of an equidistant linear camera movement parallel to the horizontal axis of the image plane [Bolles et al. (1987)]. This constraint forces all *epipolar lines* to be purely horizontal. Moreover, all epipolar lines of a particular object point coincide and remain at the same vertical position. As a result, 3D reconstruction can be performed on horizontal slices through the image cube, since all projections of 3D object points remain in the same slice throughout the entire sequence. These horizontal slices shown on the right hand side of Fig. 8.10 are called *epipolar images*.

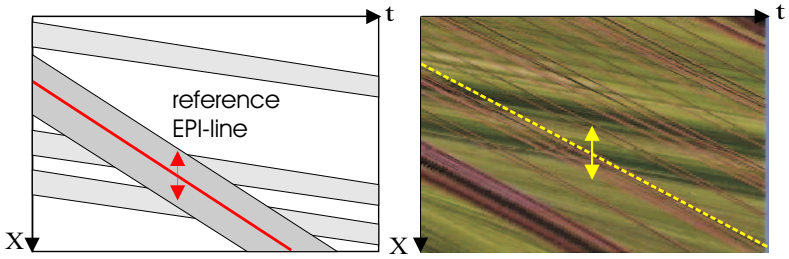

Figure 8.11: Linear camera movement. **Left:** Occlusion of EPI lines **Right:** Line structure of an epipolar image.

These epipolar images represent the trajectories of object points. If the camera is moved equidistantly, the path of an arbitrary 3D point becomes a straight line, called *EPI line* (see Fig. 8.11, right). For each object point in the image sequence, exactly one corresponding line can be detected in the epipolar image. In order to determine depth of the points, the slope of these lines is analyzed. Object points that are closer to the camera move faster through the image (large disparities), whereas projections of points further away change their position much slower. Therefore, the slopes of nearby points are higher than those of distant points and, as a result, they can be directly used to derive scene depth. Since the shape of a trajectory is explicitly known to be a straight line, conventional edge detection methods can be applied to find them.

Some points, however, may not be visible throughout the entire sequence but are occluded by other object parts. The corresponding EPI lines are therefore also partly occluded as shown on the left hand side of Fig. 8.11. Fortunately, there is a particular structure in these occlusions which can be considered for efficient occlusion handling. Object points can only be occluded by others which are closer to the camera and thus have a larger slope in the epipolar image. As a result, lines with larger slope always occlude those with a smaller one. By searching the epipolar image for lines with large slopes first, removing those lines from the image, and continuing the search with less tilted lines, occlusion handling is inherently performed. The entire process of epipolar image analysis is thus as follows:

- Calibrate the camera
- Process all epipolar images
- In each epipolar image, search for lines, starting with lines having large slopes
- Remove those lines and continue search for lines with smaller slope
- Compute depth of object point from slope of line

Since the entire length of the line is exploited to determine depth of the corresponding object point, accurate values can be determined even in the presence of multiple occlusions. Fig. 8.12 shows some results obtained with epipolar image analysis. 200 frames of a synthetic image sequence of CIF resolution are collated to an image cube. Lines are searched in the corresponding epipolar images and depth maps computed from the slopes. The right hand side of Fig. 8.12 shows one of these depth maps. No filtering, interpolation, and smoothness constraints are applied. Still, scene geometry is recovered with reasonable quality.

8.4.2 Image Cube Trajectory Analysis

Epipolar image analysis is a robust method for 3D reconstruction that jointly detects point correspondences for all available views of an image sequence. Occlusions as well as homogeneous regions can be handled efficiently. The big disadvantage of the algorithm is its restriction to linear equidistant camera movements, since the scene is

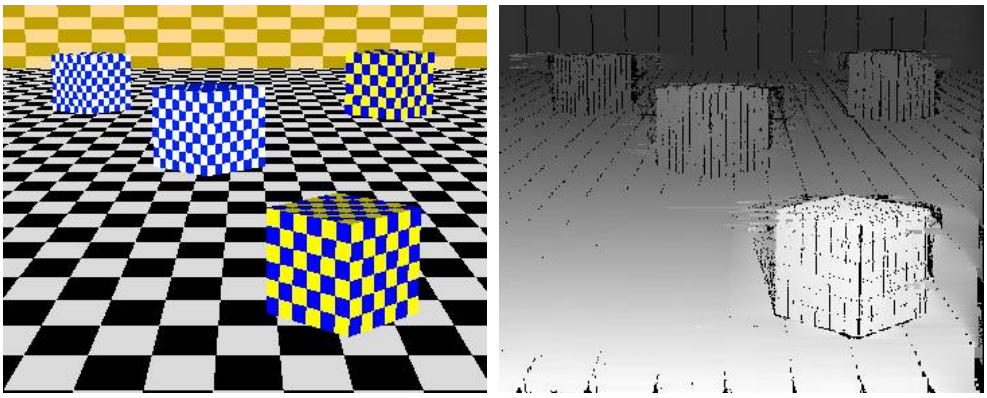

Figure 8.12 **Left:** Synthetic sequence *Boxes*, **Right:** Estimated scene depth.

viewed from one principle direction only. Other camera movements, for example circular camera configurations important for turn-table or concentric mosaic acquisitions, cannot be handled.

One idea to overcome this problem is a piecewise linear EPI analysis where small segments of the object point trajectory are approximated by straight lines. This approach can also be applied to other camera movements [Li et al. (2001)] but significantly reduces the amount of reference images and thus robustness of the 3D reconstruction.

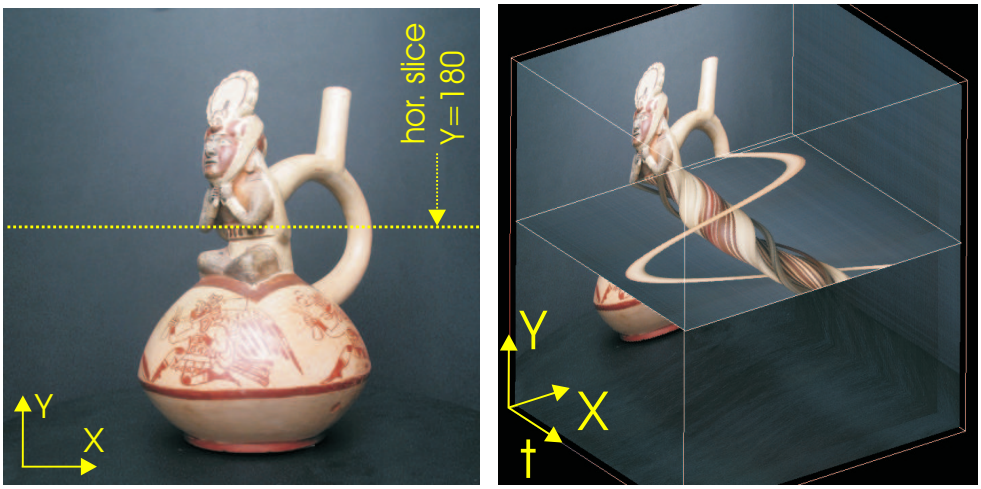

Figure 8.13: *Moche* sequence, circularly moving camera. **Left:** First camera frame. **Right:** Image cube representation with sinusoidal shaped trajectories.

This restriction to horizontal camera movements is overcome in *image cube tra-*

jectory (ICT) analysis [Feldmann et al. (2003a)] that can deal with circular camera configurations as well. For an inwards pointing circularly moving camera, e.g., the trajectories in the image planes are no longer lines but sinusoidal shaped curve as illustrated in Fig. 8.13. Instead of searching for straight lines in the epipolar images, ICT analysis first discretizes the object space similar to space carving. Either regular voxel volumes can be built or more efficient irregular structures [Feldmann et al. (2004)] can be setup. Given the camera motion, the trajectory of a particular 3D point through the image cube is computed. In a second step, color constancy along the entire trajectory is evaluated. Trajectory parameters are varied and for the best matching ICT's the corresponding 3D positions are determined. Occlusion handling is hereby performed similar to the horizontal case by using a special search order.

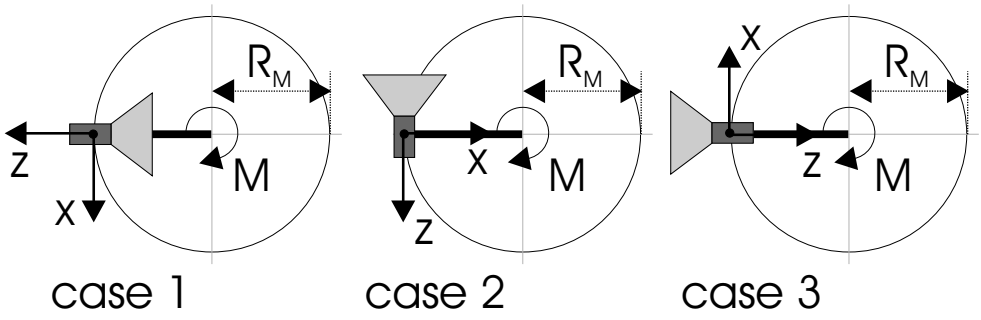

Figure 8.14: Camera configurations for three different cases. **Left:** Turn-table setup (inwards pointing camera). **Middle:** Concentric mosaic (tangential direction). **Right:** Concentric mosaic (normal direction).

Although the approach can be applied to arbitrary known camera configurations, we will restrict ourselves to circular camera motion in the following. Three cases are distinguished as illustrated in Fig. 8.14: an inwards facing camera for turn-table measurements, and an eccentrically rotating camera facing in tangential or normal direction respectively. The latter two cases are important for concentric mosaic analysis [Feldmann et al. (2003b)].

For such circular motion, an arbitrary 3D point may be described in terms of its radius R to the center of rotation, its rotation angle ϕ, and its height y. Assuming perspective projection, object point trajectories through the image cube can analytically be computed (see [Feldmann et al. (2003b)] for more details). In contrast to the horizontal case described in Section 8.4.1, the trajectories are no longer restricted to lie in a 2D plane but change in all three dimensions. The projection of the trajectory into the $X - \phi$-plane is illustrated for all three cases in Fig. 8.15. Please note that points outside and inside the camera radius R_m have to be treated differently.

Given the analytic trajectory description with the free parameters R, ϕ, and y, ICT's are computed for each 3D object point and a global search within the image cube is performed by evaluating color constancy. The parameters of the best matching ICT's determine 3D positions of valid object points. Similar to the horizontal case, occlusions may occur which hide parts of the trajectories. For the decision of the

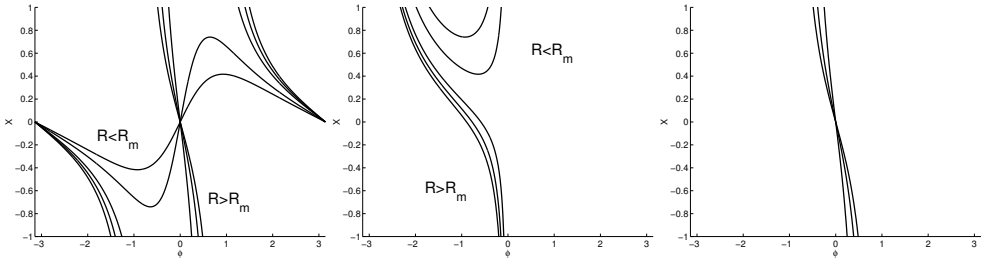

Figure 8.15: X coordinate for points with varying radius. **Left:** Trajectory for concentric mosaics with inwards facing camera. **Middle:** Tangentially facing camera, and **Right:** Outwards facing camera.

correct processing order, two cases have to be distinguished: First, trajectory parts corresponding to points lying between the center of rotation and the camera must be evaluated. For these parts, the slope of the curve is always positive and trajectories must be processed from large to small radii R. Trajectory parts being further away than the center of rotation (with negative slope) are processed next with increasing radius. Each time a matching trajectory is found, it is removed from the image cube. This way, it is assured that even partly occluded points are robustly found. For a detailed discussion of correct occlusion handling please refer to [Feldmann et al. (2003b)].

Figure 8.16: ICT reconstruction of the *Tree* sequence. **Left:** Original frame of a tree on a turn-table. **Middle:** Reconstructed geometry displayed as depth map. **Right:** Synthesized image from the voxel volume.

Fig. 8.16 shows a result obtained for a turn-table sequence with an inwards facing camera. 360 frames in CIF resolution are recorded that form the image cube. A 3D voxel model is created by searching through the image data for best matching

trajectories. The right hand side of Fig. 8.16 gives an example of a reconstructed view rendered from the voxel dataset. The geometry of the object is illustrated in the middle image by means of a depth map. Although a very simple matching strategy is used in this particular experiment, also finer details can be recovered.

8.5 Conclusions

Volumetric reconstruction methods have recently gained increasing interest since current computers allow the storage of large volumes and reconstruction of objects or scenes even in real-time. Compared to depth-based methods, a voxel representation often has the advantage that no point correspondences and registration of erroneous data are required. Due to the lack of smoothness constraints, very detailed structures can be recovered provided that accurate calibration information is available. One of the simplest methods is shape-from-silhouette that is computationally efficient and robust against lighting variations. Viewing cones computed from the contours are intersected to approximate the visual hull of the object that does not include any concavities. Higher shape accuracy can be obtained by using surface color information instead of the binary silhouette mask. Space carving or voxel coloring methods try to construct a colored voxel volume whose back-projections into the image planes result in photo-consistent synthetic views. The methods mainly differ in the way the volume is traversed, how color similarity is computed, and how visibility is determined. Occlusion handling is also elegantly solved by an occlusion compatible ordering scheme in epipolar image analysis where trajectories of object points in an image cube are analyzed. Again, all views of the scene are jointly exploited without establishing point correspondences. The classic linear camera motion assumption can also be generalized to other camera movements enabling the usage of these methods to a wide range of applications.

Bibliography

Baker H 1977 Three-dimensional modelling *Proc. of the 5th International Joint Conference on Artificial Intelligence*, pp. 649–655, Cambridge, MA, USA.

Bolles RC, Baker HH and Marimont DH 1987 Epipolar image analysis: An approach to determine structure from motion. *International Journal of Computer Vision* **1**(1), 7–55.

Chien CH and Aggarwal JK 1986 Volume/surface octrees for the representation of three-dimensional objects. *Computer Vision, Graphics and Image Processing* **36**(1), 100–113.

Criminisi A, Kang SB, Swaminathan R, Szeliski R and Anandan P 2002 Extracting layers and analyzing their specular properties using epipolar-plane-image analysis. Technical report, Tech. Rep. MSR-TR-2002-19, Microsoft Research.

Culbertson WB, Malzbender T and Slabaugh G 1999 Generalized voxel coloring *Proc. International Conference on Computer Vision (ICCV)*, Corfu, Greece.

Eisert P 2004 3-D geometry enhancement by contour optimization in turntable sequences *Proc. International Conference on Image Processing (ICIP)*, pp. 3021–3024, Singapore.

Eisert P, Steinbach E and Girod B 1999 Multi-hypothesis, volumetric reconstruction of 3-D objects from multiple calibrated camera views *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 3509–3512, Phoenix, USA.

Feldmann I, Eisert P and Kauff P 2003a Extension of epipolar image analysis to circular camera movements *Proc. International Conference on Image Processing (ICIP)*, pp. 697–700, Barcelona, Spain.

Feldmann I, Kauff P and Eisert P 2003b Image cube trajectory analysis for 3D reconstruction of concentric mosaics *Proc. International Workshop on Vision, Modeling, and Visualization*, pp. 569–576, Munich, Germany.

Feldmann I, Kauff P and Eisert P 2004 Optimized space sampling for circular image cube trajectory analysis *Proc. International Conference on Image Processing (ICIP)*, pp. 1947–1950, Singapore.

Fitzgibbon AW, Cross G and Zisserman A 1998 Automatic 3D model construction for turntable sequences *Proc. ECCV 98 Workshop on 3D Structure from Multiple Images in Large-Scale Environments*, pp. 154–170, Freiburg, Germany.

Goldlücke B and Magnor M 2003 Real-time, free-viewpoint video rendering from volumetric geometry *Proc. Visual Computation and Image Processing (VCIP)*, Lugano, Switzerland.

Grattarola AA 1992 Volumetric reconstruction from object silhouettes: A regularization procedure. *Signal Processing* **27**, 27–35.

Grau O 2003 Studio production system for dynamic 3D content *Proc. Visual Computation and Image Processing (VCIP)*, Lugano, Switzerland.

Greenleaf JF, Tu TS and Wood EH 1970 Computer generated 3-D oscilloscopic images and associated techniques for display and study of the spatial distribution of pulmonary blood flow. *IEEE Transactions on Nuclear Science* **17**(3), 353–359.

Gross M 2003 blue-c: A spatially immersive display and 3D video portal for telepresence *Proc. Computer Graphics (SIGGRAPH)*, pp. 819–827, San Diego, USA.

Horn BKP 1986 *Robot Vision*. MIT Press, Cambridge.

Kutulakos KN and Seitz SM 2000 A theory of shape by space carving. *International Journal of Computer Vision*.

Laurentini A 1994 The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**(2), 150–162.

Li Y, Tang CK and Shum HY 2001 Efficient dense depth estimation from dense multiperspective panoramas *Proc. International Conference on Computer Vision (ICCV)*, pp. 119–126, Vancouver, Canada.

Lorensen WE and Cline HE 1987 Marching cubes: A high resolution 3D surface construction algorithm *Proc. Computer Graphics (SIGGRAPH)*, vol. 21, pp. 163–169.

Martin WN and Aggarwal JK 1983 Volumetric description of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Matusik W, Buehler C and McMillan L 2001 Polyhedral visual hulls for real-time rendering *Proc. 12th Eurographics Workshop on Rendering*.

Niem W 1998 *Automatische Rekonstruktion starrer dreidimensionaler Objekte aus Kamerabildern* PhD thesis University of Hanover Germany.

Niem W and Wingbermühle J 1997 Automatic reconstruction of 3D objects using a mobile monoscopic camera *Proc. of the International Conference on Recent Advances in 3D Imaging and Modelling*, pp. 173–180, Ottawa, Canada.

Pauly M, Keiser R, Kobbelt L and Gross M 2003 Shape modeling with point-sampled geometry *Proc. Computer Graphics (SIGGRAPH)*, vol. 22, pp. 641–650, San Diego, USA.

Potmesil M 1987 Generating octree models of 3D objects from their silhouettes in a sequence of images. *Computer Vision, Graphics and Image Processing* **40**(1), 1–29.

Prock AC and Dyer CR 1998 Towards real-time voxel coloring *Proc. Image Understanding Workshop*.

Rusinkiewicz S and Levoy M 2000 QSplat: A multiresolution point rendering system for large meshes *Proc. Computer Graphics (SIGGRAPH)*, pp. 343–352, Los Angeles, USA.

Seitz SM and Dyer CR 1997 Photorealistic scene reconstruction by voxel coloring *Proc. Computer Vision and Pattern Recognition*, pp. 1067–1073, Puerto Rico.

Shum HY and He LW 1999 Rendering with concentric mosaics *Proc. Computer Graphics (SIGGRAPH)*, pp. 299–306, Los Angeles, USA.

Smolic A, Müller K, Merkle P, Rein T, Eisert P and Wiegand T 2004 Free viewpoint video extraction, representation, coding, and rendering *Proc. International Conference on Image Processing (ICIP)*, pp. 3287–3290, Singapore.

Srivastava SK and Ahuja N 1990 Octree generation from object silhouettes in perspective views. *Computer Vision, Graphics and Image Processing*.

Steinbach E, Eisert P, Betz A and Girod B 2000 3-D reconstruction of real world objects using extended voxels *Proc. International Conference on Image Processing (ICIP)*, vol. I, pp. 569–572, Vancouver, Canada.

Szeliski R 1993 Rapid octree construction from image sequences *Computer Vision, Graphics and Image Processing*, pp. 23–32.

Tarini M, Callieri M, Montani C, Rocchini C, Olsson K and Persson T 2002 Marching intersections: An efficient approach to shape-from-silhouette *Proc. Vision, Modeling, and Visualization VMV'02*, pp. 283–290, Erlangen, Germany.

Veenstra J and Ahuja N 1986 Efficient octree generation from silhouettes *Proc. Computer Vision and Pattern Recognition*, pp. 537–542, Miami Beach, USA.

Westover L 1990 Footprint evaluation for volume rendering *Proc. Computer Graphics (SIGGRAPH).*

Wu X and Matsuyama T 2003 Real-time active 3D shape reconstruction for 3D video *Proc. of 3rd International Symposium on Image and Signal Processing and Analysis*, pp. 186–191, Rome, Italy.