VOLUME BASED RECONSTRUCTION OF ARCHAEOLOGICAL ARTIFACTS

Martin KAMPEL, Ro*b*ert SABLATNIG, Sr*d*an TOSOVIC

Vienna University of Technology,
Institute of Computer Aided Automation, Pattern Recognition and Image Processing Group
Favoritenstr. 9, 183-2, A-1040, Vienna, Austria ,
{kampel,sab}@prip.tuwien.ac.at

KEY WORDS: 3D, scanning, close range, archaeology, surface models.

ABSTRACT:

An algorithm for the automatic construction of a 3d model of archaeological vessels using two different 3d algorithms is presented. In archeology the determination of the exact volume of arbitrary vessels is of importance since this provides information about the manufacturer and the usage of the vessel. To acquire the 3d shape of objects with handles is complicated, since occlusions of the object's surface are introduced by the handle and can only be resolved by taking multiple views. Therefore, the 3d reconstruction is based on a sequence of images of the object taken from different viewpoints with different algorithms; shape from silhouette and shape from structered light. The output of both algorithms are then used to construct a single 3d model.
Images for both algorithms are acquired by rotating the object on a turntable in front of a stationary camera. Then an octree representation of the object is built incrementally, by performing limited processing of all input images for each level of the octree. Beginning from the root node at the level 0 a rough model of the object is obtained quickly and is refined as the processed level of the octree increases. Results of the algorithm developed are presented for both synthetic and real input images.

1. INTRODUCTION

The combination of the *S*hape fro*m* *S*ilho*u*ette method with the *S*hape fro*m* *S*tr*u*ct*u*tre*d* *L*ight method presented in this paper was performed within the *C*o*m*p*u*ter Ai*d*ed *C*lassification of *C*era*m*ics [11,6] project, which aims to provide an objective and automated method for classification and reconstruction of archaeological pottery. The final goal is to provide a tool which helps archaeologists in their classification process.

Pottery was made in a very wide range of forms and shapes. The purpose of classification is to get a systematic view of the material found, to recognize types, and to add labels for additional information as a measure of quantity [15]. In this context, decoration of pottery is of great interest. Decoration is difficult to illustrate since it is a perspective projection of an originally spherical surface. This fact induces distortions that can be minimized by 'unwrapping' the surface. In order to be able to unwrap the surface it is necessary to have a 3d representation of the original surface. Furthermore, the exact volume of the vessel is a great interest to archaeologists too, since the volume estimation allows also a more precise classification [15]. Since pottery is manufactured on a turntable we use a turntable based method for the 3d-reconstruction of the original. To acquire images from multiple views we put the archaeological vessel on a turntable which rotates in front of a stationary camera.

Shape from Silhouette is a method of automatic construction of a 3D model of an object based on a sequence of images of the object taken from multiple views, in which the object's silhouette represents the only interesting feature of the image [19,16]. The object's silhouette in each input image corresponds to a conic volume in the object real-world space. A 3D model of the object can be built by intersecting the conic volumes from all views, which is also called *S*pace *C*arving [8].

Shape from Silhouette is a computationally simple algorithm (it employs only basic matrix operations for all transformations). It can be applied on objects of arbitrary shapes, including objects with certain concavities (like a handle of a cup), as long as the concavities are visible from at least one input view. This condition is very hard to hold since most of the archaeological vessels do have concavities (like a cup for instance) that have to be modeled. Therefore a second, active shape determination method has to be used to discover all concavities. The acquisition method used for estimating the 3d-shape of objects is shape from structured light, based on active triangulation [2]. Both methods requires only a camera and illumination devices as equipment, so they can be used to obtain a quick initial model of an object which can then be refined stepwise.

There have been many works on construction of 3D models of objects from multiple views. Baker [1] used silhouettes of an object rotating on a turntable to construct a wire-frame model of the object. Martin and Aggarwal [10] constructed volume segment models from orthographic projection of silhouettes. Chien and Aggarwal [4] constructed an object's octree model from its three orthographic projections. Veenstra and Ahuja [22] extended this approach to thirteen standard orthographic views. Potmesil [16] created octree models using arbitrary views and perspective projection. For each of the views he constructs an octree representing the corresponding conic volume and then intersects all octrees. In contrast to this, Szeliski [19] first creates a low resolution octree model quickly and then refines this model iteratively, by intersecting each new silhouette with the already existing model. The last two approaches project an octree node into the image plane to perform the intersection between the octree node and the object's silhouette. Srivastava and Ahuja [18] in contrast, perform the intersections in 3D-space. The work of Szeliski [19] and Niem [13] were used as a basis for the shape from silhouette approach presented in this paper. For the active triangulation method we use an approach bis Liska developped for a next view planing strategy using structured light [9].

The paper is organized as follows. Section 2 describes the equipment used. Section 3 describes the strategy for and

Section 4 presents experimental results Finally, in Section 5 conclusions are drawn and future work is outlined.

2. ACQUISITION SYSTEM

The acquisition system (Figure 1) consists of the following devices:

- a turntable (Figure 1a) with a diameter of 50 cm, whose desired position can be specified with an accuracy of 0.05°. The turntable is used to obtain multiple views of the object observed.
- two monochrome CCD-cameras (Figure 1b and 1c) with a focal length of 16 mm and a resolution of 768x576 pixels. One camera (Camera-1 in Figure 1) is used for acquiring the images of the object's silhouettes and the other (Camera-2 in Figure 1) for the acquisition of the images of the laser light projected onto the object.
- a laser (Figure 1d) used to project a light plane onto the object. The laser is equipped with a prism in order to span a plane out of the laser beam. The color of the projected light is red.
- a lamp (Figure 1e) used to illuminate the scene for the acquisition of the silhouette of the object. The object should be clearly distinguishable from the background independent from the object's shape or the type of its surface. For that reason back-lighting [5] is used. A large (approx. 50x40 cm) rectangular lamp is put behind the turntable (as seen from the camera). In addition, a white piece of paper, larger than the lamp, is put right in front of the lamp, in order to make the light more diffuse.

The whole system is protected against the ambient light by a thick black curtain.

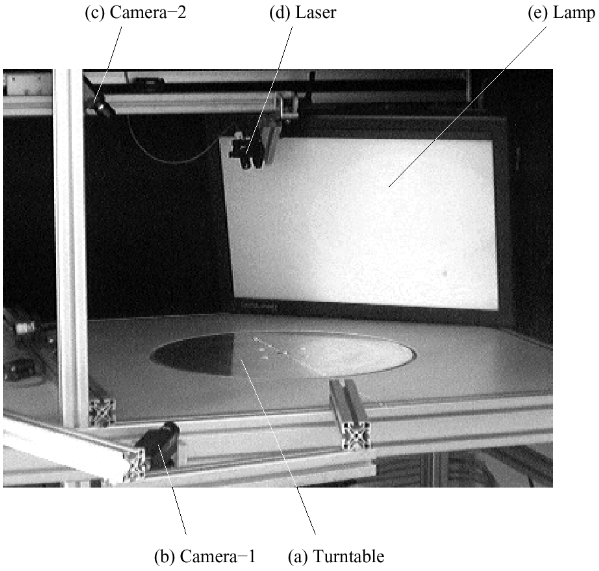


Figure 1. Acquisition System

The geometrical setup of the acquisition devices is shown in Figure 2. Both cameras are placed about 50 cm away from the rotational axis of the turntable. Ideally the optical axis of the camera for acquiring object's silhouettes (Camera-1 in Figure 2) lies nearly in the rotational plane of the turntable, orthogonal to the rotational axis. The camera for acquiring the projection of the laser plane (Camera-2 in Figure 2) onto the object views the turntable from an angle of about 45°(β in Figure 2). The laser is directed such that the light plane it projects contains the rotational axis of the turntable. Camera-2 in Figure 2 views the light plane also from an angle of about 45° (α in Figure 2). The relative position of the two cameras to one another is not important, since the acquisition of the silhouettes and the acquisition of the laser light projection are independent from one another.

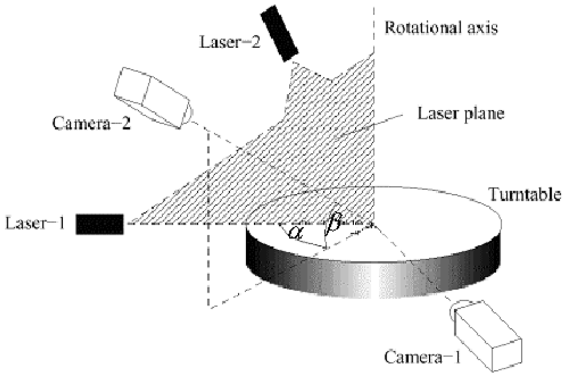


Figure 2. Geometrical setup of acquisition system

Prior to any acquisition, the system is calibrated in order to determine the inner and outer orientation of the camera and the rotational axis of the turntable. The calibration method used was exclusively developed for the Shape from Silhouette algorithm presented and it is described in detail in [20] and [7,21].

3. MODEL REPRESENTATION

There are many different model representations in computer vision and computer graphics used. Here we will mention only the most important ones. Surface-based representations describe the surface of an object as a set of simple approximating patches, like planar or quadric patches [12]. Generalized cylinder representation [17] defines a volume by a curved axis and a cross-section function at each point of the axis. Overlapping sphere representation [14] describes a volume as a set of arbitrarily located and sized spheres. Approaches such as these are efficient in representing a specific set of shapes but they are not flexible enough to describe arbitrary solid objects.

Two of the most commonly used representations for solid volumes are boundary representation (B-Rep) and constructive solid geometry (CSG) [17]. The B-Rep method describes an object as a volume enclosed by a set of surface elements, typically sections of planes and quadratic surfaces such as spheres, cylinders and cones. The CSG method uses volume elements rather than surface elements to describe an object. Typical volume elements are blocks, spheres, cylinders, cones and prisms. These elements are combined by set operations into the modeled object. The B-Rep and CSG method suffer from quadratic growth of elemental operations as the complexity of the modeled object increases.

An octree [3] is a tree-formed data structure used to represent 3-dimensional objects. Each node of an octree represents a cube subset of a 3-dimensional volume. A node of an octree which represents a 3D object is said to be:

- black, if the corresponding cube lies completely within the object
- white, if the corresponding cube lies completely within the background, i.e., has no intersection with the object

- gra*y*, if the corresponding cube is a boundary cube, i.e., belongs partly to the object and partly to the background. In this case the node is divided into 8 child nodes (octants) representing 8 equally sized sub-cubes of the original cube

All leaf nodes are either black or white and all intermediate nodes are gray. An example of a simple 3D object and the corresponding octree is shown in Figure 3.

This octree contains binary information in the leaf nodes and therefore it is called a binary octree and it is suitable for representation of 3D objects where the shape of the object is the only object property that needs to be modeled by the octree. Non-binary octrees can contain other information in the leaf nodes, e.g., the cube color in RGB-space. For the Shape from Silhouette algorithm presented, a binary octree model is sufficient to represent 3D objects, and in the remainder of this paper the term octree will always refer to a binary octree.

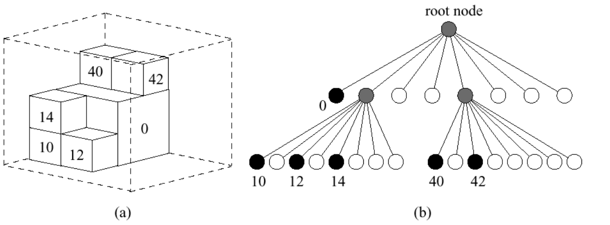
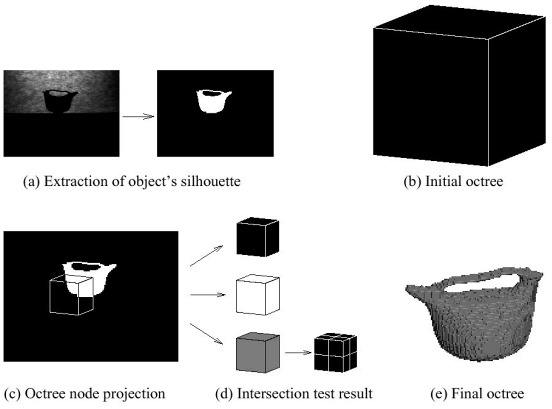


Figure 3. A simple object (a) and the corresponding octree (b)

3.1 Construction of an octree

The algorithm builds up a 3D model of an object in the following way: first, all input images are transformed into binary images where a "black" pixel belongs to the object observed and a "white" one to the background (Figure 4a). In the implementation, black means background and *w*hite means object, but it is more intuitive to describe an object pixel as "black" and a background pixel as "white". Then, the initial octree is created with a single root node (Figure 4b) representing the whole object space, which will be "carved out" corresponding to the shape of the object observed.


(a) Extraction of object's silhouette

(b) Initial octree

(c) Octree node projection

(d) Intersection test result

(e) Final octree

Figure 4. Algorithm overview

Then, the octree is processed in a level-by-level manner: starting from level 0 (with root node as the only node), all octree nodes of the current level marked as "black", i.e., belonging to the object, are projected into the first input image (Figure 4c) and tested for intersection with the object's image silhouette. Depending on the result of the intersection test, a node can remain to be "black", it can be marked as "white" (belonging to the background) or in case it belongs partly to the object and partly to the background, it is marked as "grey" and divided into 8 black child nodes of the next higher level (Figure 4d). The remainder of the black nodes of the current level is then projected into the next input image where the procedure of ntersection testing with the object's silhouette is repeated. Once all input images have been processed for the current octree level, the current level is incremented by one and the whole procedure (starting from the projection of the black nodes of the current level into the first input image) is repeated, until the maximal octree level has been reached. The remaining octree after the processing of the last level is the final 3D model of the object (Figure 4e).

We define the following coordinate systems relevant for the projection of an octree node into the image plane:

- octree coor*d*inate s*y*ste*m*: rooted at the intersection of the rotational axis of the turntable and its rotational plane. For the first input image, it is identical to the object coordinate system and it rotates with the object observed
- ob*j*ect coor*d*inate s*y*ste*m*: rooted at the same point as the octree coordinate system, but it is static, i.e., it doesn't rotate with the object. The *y* axis is the rotation axis of the turntable and the *z* axis is orthogonal to the image plane
- *im*age coor*d*inate s*y*ste*m*: lies in the image plane and it is rooted at the image center point

Figure 5 depicts these coordinate systems and their relative positions to one another.

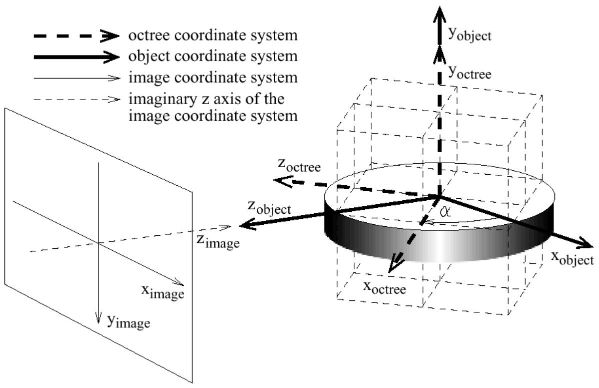


Figure 5. The octree-, object- and image coordinate system

3.2 Determination of octree no*d*es

An octree node is projected into the image plane in the following way: as a preprocessing step, the translation and rotation matrices and are multiplied for all possible view angles *α*, and the resulting matrices of these multiplications are stored in a lookup table. This is done before any processing of octree nodes starts. Once it starts, all vertices of the current node are projected into the image plane by multiplying their octree coordinates with the matrix from the lookup table corresponding to the current view angle and then multiplying the result with the appropriate scaling matrix. The result of the projection of an octree node into the image plane are image coordinates of all of the vertices of the node's corresponding cube. In the general case, the projection of a node looks like a hexagon, as depicted in Figure 6(a). To find the hexagon corresponding to the eight projected vertices is a costly task, because it requires to determine which points are inside and which outside the hexagon, and there can be hundreds of thousands of octree nodes that need to be processed. It is much simpler (and therefore faster) to compare the bounding box of the eight points. Figure6 shows a projected octree node and the corresponding bounding box.

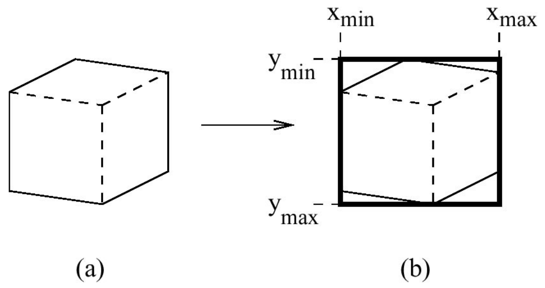


Figure 6. Projection of a node (a) and its bounding box (b)

The bounding box is tested for intersection with the object's silhouette in the current input (binary) image. All image pixels within the bounding box are checked for their color, whether they are black or white. The output of the intersection testing procedure is percentage of the black pixels of the bounding box, i.e., the percentage of pixels belonging to the object. If this percentage is equal or higher than a user definable threshold for black nodes, the node is marked as black. If the percentage is smaller than or equal with a user definable threshold for white nodes, the node is marked as white. Otherwise, the node is marked as gray and it is divided into eight child nodes representing eight sub-cubes of finer resolution.

The calculated image coordinates of the cube's vertices can lie between two image pixels, and a pixel is the smallest testable unit for intersection testing. Which pixels are considered to be "within" the bounding box? Figure 7 illustrates our answer to this question. We decided to test only pixels that lie completely within the bounding box (Figure 7a), because that way the number of pixels that need to be tested is smaller than testing all pixels that are at least partly covered by the bounding box The pixels at the border of the bounding box are excluded, because most of them do not lie within the hexagon approximated by the bounding box. In the special case if there are no pixels that lie completely within the bounding box (Figure 7b) the pixel closest to the center of the bounding box is checked for the color.

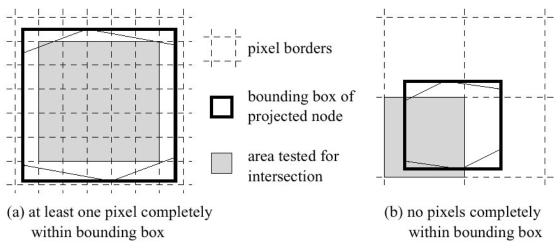

pixel borders
bounding box of projected node
area tested for intersection
(a) at least one pixel completely within bounding box
(b) no pixels completely within bounding box

Figure 7: Selection of pixels for the intersection test

The octree representation has several advantages [3]: for a typical solid object it is an efficient representation, because of a large degree of coherence between neighboring volume elements (voxels), which means that a large piece of an object can be represented by a single octree node. Another advantage is the ease of performing geometrical transformations on a node, because they only need to be performed on the node's vertices. The disadvantage of octree models is that they digitize the space by representing it through cubes whose resolution depend on the maximal octree depth and therefore cannot have smooth surfaces.

4. COMBINATION OF ALGORITHMS

An input image for *S*hape fro*m* *S*ilho*u*ette defines a conic volume in space which contains the object to be modeled (Figure 8a). Another input image taken from a different view defines another conic volume containing the object (Figure 8b). Intersection of the two conic volumes narrows down the space the object can possibly occupy (Figure 8c). With an increasing number of views the intersection of all conic volumes approximates the actual volume occupied by the object better and better, converging to the 3D visual hull of the object. Therefore by its nature *S*hape fro*m* *S*ilho*u*ette defines a vol*um*etric model of an object.

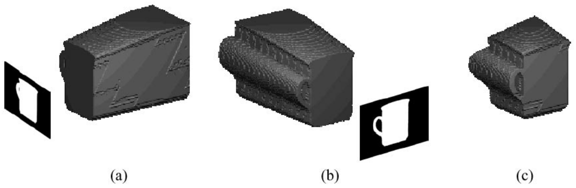


Figure 8. Two conic volumes and their intersection

An input image for *S*hape fro*m* *S*tr*u*ct*u*re*d* *L*ight using laser light defines solely the points on the surface of the object which intersect the laser plane (Figure 9a). Using multiple views provides us with a cloud of points belonging to the object surface (Figure 9b), i.e. with the s*u*rface model of the object.

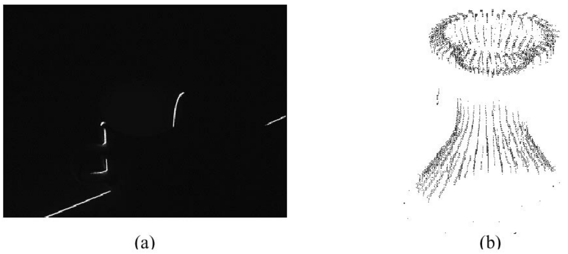


Figure 9. Laser projection and cloud of points

The main problem that needs to be addressed in an attempt to combine these two methods is how to adapt the two representations to one another, i.e. how to build a common 3D model representation. This can be done in several ways:

- Build the *S*hape fro*m* *S*ilho*u*ette's volumetric model and the *S*hape fro*m* *S*tr*u*ct*u*re*d* *L*ight's surface model independently from one another. Then, either convert the volumetric model to a surface model and use the intersection of the two surface models as the final representation or convert the surface model to a volumetric model and use the intersection of the two volumetric models as the final representation.
- Use a common 3D model representation from the ground up, avoiding any model conversions. That means either

design a volume based Shape from Structured Light algorithm or a surface based Shape from Silhouette algorithm.

With the former method both underlying algorithms would build their "native" model of the object. However, conversion and intersection of the models would not be a simple task. While conversion of the Shape from Silhouette's volumetric model to a surface model is straightforward --- one only has to find 3D points of the volume belonging to the surface --- an intersection of two surface models can be rather complex. One could start from the points obtained by Shape from Structured Light (because they really lie on the object's surface, whereas points on the surface of the volume obtained by Shape from Silhouette only lie somewhere on the object's visual hull) and fill up the missing surface points with points from the Shape from Silhouette model.

There are several problems with this approach. There could be many "jumps" on the object surface, because the points taken from the Shape from Silhouette model might be relatively far away from the actual surface. The approach would also not be very efficient, because we would need to build a complete volumetric model through Shape from Silhouette, then intersect it with every laser plane used for Shape from Structured Light in order to create a surface model, and then, if we also want to compute the volume of the object, we would have to convert the final surface model back to the volumetric model.

Another possibility would be converting the surface model obtained by Shape from Structured Light to a volumetric model and intersect it with the Shape from Silhouette's model. In this case the intersection is the easier part - for each voxel of the space observed one would only have to look up whether both models "agree" that the voxel belongs to the object - only such voxels would be kept in the final model and all others defined as background. Also the volume computation is simple in this case - it is a multiplication of the number of voxels in the final model with the volume of a single voxel. But the problem with this approach is the conversion of the Shape from Structured Light's surface model to a volumetric model - in most cases, the surface model obtained using laser plane is very incomplete (see the model of an amphora in Figure 9(b) because of the light and camera occlusions (Figure 10), so one would have to decide how to handle the missing parts of the surface.

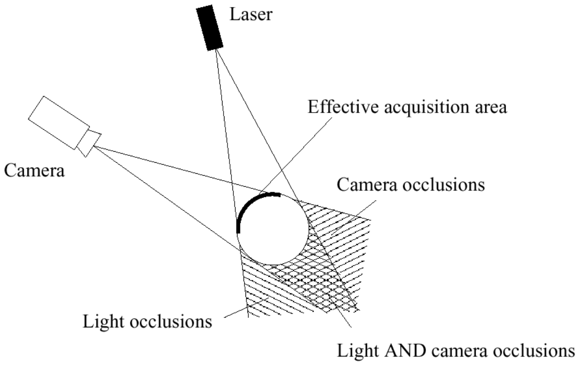


Figure 10. Light and camera occlusions

And generally, the conversion of a surface model to a volumetric model is a complex task, because if the surface is not completely closed, it is hard to say whether a certain voxel lies inside or outside the object. With closed surfaces one could follow a line in 3D space starting from the voxel observed and going in any direction and count how many times the line intersects the surface. For an odd number of intersections one can say that the voxel belongs to the object. But even in this case there would be many special cases to handle, e.g. when the chosen line is tangential to the object's surface.

This reasoning lead us to the following conclusions:

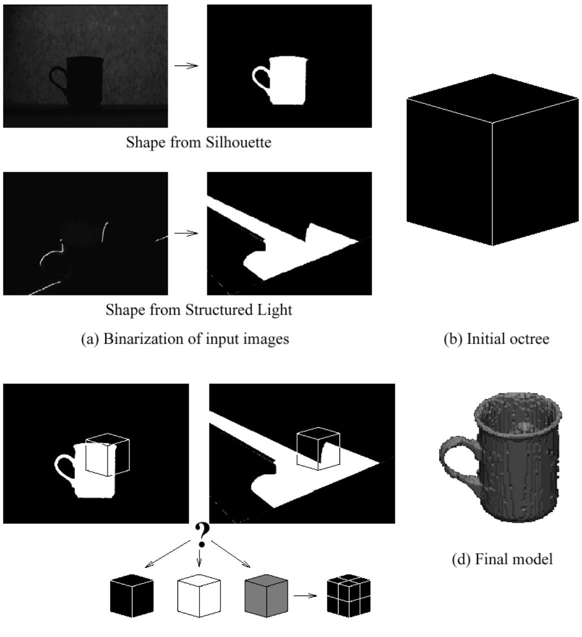
- Building a separate Shape from Structured Light surface model and a Shape from Silhouette volumetric model followed by converting one model to the other and intersecting them is mathematically complex and computationally costly.
- If we want to estimate the volume of an object using our model, any intermediate surface models should be avoided because of the problems of conversion to a volumetric model.

When building a 3D volumetric model of an object based on a number of its 2D images, there are two possibilities regarding the decision whether a certain voxel is a part of the object or belongs to the background. Therefore, our approach proposes building a single volumetric model from the ground up, using both underlying methods in each step (illustrated in Figure 11):

1. Binarize the acquired images for both Shape from Silhouette and Shape from Structured Light in such a way that the white image pixels possibl*y* belong to the object and the black pixels for s*u*re belong to the background. This is shown in Figure 11a.
2. Build the initial octree, containing one single root node marked "black". (Figure 11b). This node is said to be at the level 0. Set the `current level` to 0.
3. All black nodes of the `current level` are assumed to be in a linked list. Set the `current node` to the first node in the list. If there are no nodes in the `current level`, the final model has been build so jump to Step 8. Otherwise, continue with Step 4.
4. Project the `current node current level` into all Shape from Silhouette binarized input images and intersect it with the image silhouettes of the object (by simply counting the percentage of white pixels within the projection of the node). As the result of the intersection the node can remain "black" (if it lies within the object) or be set to "white" (it lies outside the object) or "grey" (it lies partly within and partly outside the object). This is illustrated in Figure 11c. If at least one image says "this node is white", it is set to white. Otherwise, if at least one image says "this node is grey", it is set to grey and only if all images agree that the node is black, it stays black.
5. If the `current node` after Step 4 is not white, it is projected into two binarized Shape from Structured Light images representing two nearest laser planes to the node - one plane is the nearest among the images acquired with the turntable's rotation angle between 0° and 180° and the other the nearest among images taken with the angle between 180° and 360°. The separation into 2 intervals is done because if we use a single nearest plane, it could happen that the projection of the node lies completely in the occluded part of the image. Two nearest planes defined this way are almost identical, because they both contain the rotational axis of the turntable (because of the way we set the laser plane, see Figure 2, so if the nearest plane in the range 0° - 180° was with the angle α, then the nearest plane in the range 180° - 360° will be with the angle α +

180°. This way we increase the chance that the node does not lie in the occluded area in at least one of the planes. The projected node is now intersected with both images in the same way as in Step 4 (Figure 11c). If at least one image says "this node is white", it is set to white. Otherwise, if at least one image says "this node is grey", it is set to grey and only if both images agree that the node is black, it stays black. The intersection with the object in the image is performed in the same way as the intersection of a node with object's silhouettes in Shape from Silhouette input images.

6. If the node is set to grey it is divided into 8 child nodes of the `current level + 1`, all of which are marked "black"
7. Processing of the `current node` is finished. If there are more nodes in the `current level` set the `current node` to the next node and go back to Step 4. If all nodes of the `current level` have been processed, increment the `current level` and go to Step 3.
8. The final octree model has been built (Figure 11d).


Shape from Silhouette

Shape from Structured Light

(a) Binarization of input images

(b) Initial octree

(c) Intersection testing

(d) Final model

Figure 11. Algorithm overview

5. RESULTS

For tests with synthetic objects we can build a model of a virtual camera and laser and create input images in such way that the images fit perfectly into the camera model. This way we can analyze the accuracy of the constructed models without impact of camera calibration errors. The parameters and the position of the camera and the laser are arbitrary, so we choose realistic values. We assume having a virtual camera with focal length f = 20 *mm*, placed on the *y* axis of the world coordinate system, 2000 *mm* away from its origin (Figure 12). We set the distance between two sensor elements of the camera to $d_x\ d = 0.01$ *mm*. The laser is located on the *z* axis of the world coordinate system, *850 mm* away from its origin, and the turntable 250 *mm* below the *x-y* plane of the world coordinate system, with its rotational axis identical to the *z* world axis, as shown in Figure 12.

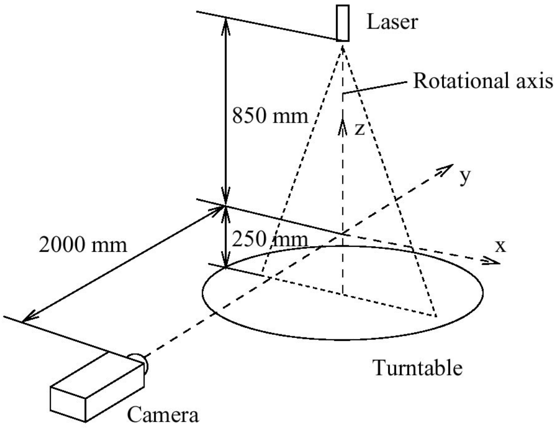


Figure 12. Virtual acquisition system

We build input images with size 640 x 480 pixels, in which 1 pixel corresponds to 1 mm in the *x-z* plane of the world coordinate system.

Having built the camera model and the input images we can test our 3D modeling algorithms with varying modeling parameters. As the measure of the accuracy of the models we compare the size (width, height and length) and the volume of the model with the size and the analytical volume of the object.

5.1 S*y*nthetic o*b*ject

As the synthetic object we create a sphere with radius r=200 *mm*, shown in Figure 13a. If we place the center of the sphere in the origin of the world coordinate system (see Figure 12), the sphere will look the same from all possible views. For our virtual acquisition system we can assume having neither camera nor light occlusions and we can construct perfect input images of the sphere (Figure 13b and c) which can be used for any view.

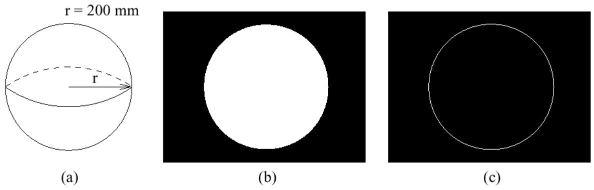
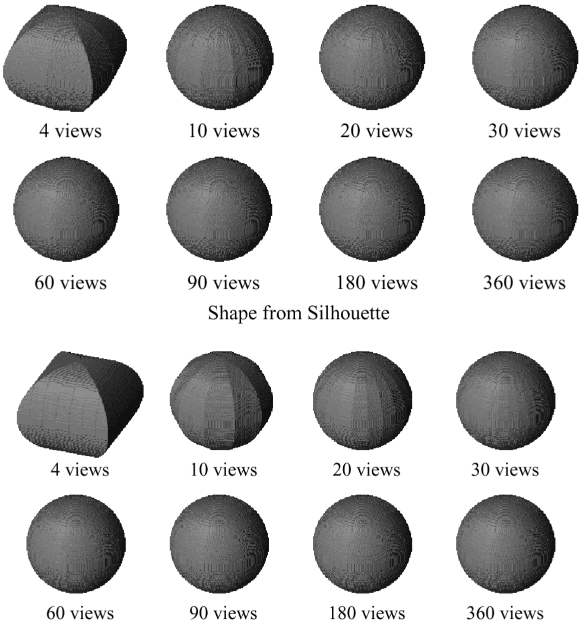


Figure 13. Synthetic sphere (a) and an input image for Shape from Silhouette (b) and Shape from Structured Light (c)

Note that the image from Figure 13c can not be obtained using the laser from Figure 12. Instead, we assume seeing the complete profile of the sphere, in order to be able to reconstruct the complete object using Shape from Structured Light only. Since the sphere does not contain any cavities, Shape from Silhouette can also reconstruct it completely. Therefore, we can measure the accuracy of each of the methods independently, as well as of the combined method.

In the first test we build models using 360 views with the constant angle of 1° between two views, while increasing octree resolution. It turned out that the shape from Silhouette method performed best with an octree resolution of 128^3, where the approximation error was +0.83% of the actual volume, the structured light method with a resolution of 256^3 and +0.29% error (the other method produced there an errror of -1.42%).

In the second test we build models with constant octree resolution of 256^3 and increasing number of views. The angle between two neighboring views is always constant. Voxel size is 2 mm for all models. The models computed are shown in Figure 14.



Shape from Silhouette

Shape from Structured Light

Figure 14. 3D models of synthetic sphere with increasing number of views

In the tests with synthetic sphere we compared Shape from Silhouette and Shape from Structured Light against one another. With respect to octree resolution there was no significant difference in the behavior of the two methods -- the accuracy of the models built was approximately the same, with exception of octrees 256^3 and 512^3, where the volume and size of the Shape from Silhouette model started being smaller than the analytical values, while the Shape from Structured Light model truly converged to the analytical one. This can be explained through the complexity of building a Shape from Silhouette based octree -- each node is projected into all 360 input images by projecting its 8 vertices, which means 2880 world-to-image projections of points per node. In the Shape from Structured Light method, a node is projected into two nearest images only, i.e. there are 16 world-to-image projections per octree node. Therefore, when dealing with octree nodes of finer resolution (when the projection of the node has approximately the size of a pixel), errors due to numerical instabilities are more likely to happen in Shape from Silhouette, especially when using a large number of views.

Regarding number of views, there was also no significant difference between the two methods. Using 20 instead of 360 input views was sufficient for both methods to create models less than 1% different from the models built using 360 views.

As second test object we used a synthetic cone where the relative error of the computed volume of models built with increasing octree resolution was much larger than the error of the models of the sphere built with same parameters. The reason for this is that the cone has a large number of octree nodes belonging to the surface, and these nodes are the ones contributing mostly to the error. The models of the cone built with different number of views showed the same behavior as the models of the sphere -- starting from 20 input views the volume error relative to the volume of the object built with 360 views falls to 1% or less .

If an object only needs to be visualized, without calculating its olume, a model built with the octree resolution 32^3 and 10 input views can give a satisfactory result. However, it should be noted that the sufficient octree resolution as well as the sufficient number of view depend on the properties of the camera, the geometry of the acquisition system and the properties of the object observed. In the tests with synthetic data we dealt with rotational symmetric objects only. For more realistic cases with more complex data sets tests with real objects are necessary.

5.2 Real objects

For tests with real objects we use 7 objects shown in Figure 15: a metal cuboid, a wooden cone, a coffee mug, two archaeological vessels and two archaeological sherds.

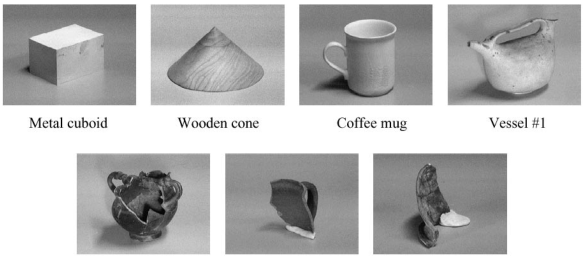


Figure 15. Real objects used for tests

The cuboid and the cone have known dimensions so we can calculate their volumes analytically and compare them with the volumes of their reconstructed models. Using these two objects we can also measure the impact of ignoring camera lens distortion on the accuracy of the models. The other objects have unknown volume, so we will just show the models constructed. All models shown in this section are built using 360 views, with constant angle of 1° between two neighboring views.

Object	Voxel size	Measured dimensions (mm)
Vessel #1	0.74 mm	141.2 x 84.8 x 93.7
Vessel #2	0.53 mm	114.2 x 114.6 x 87.4
Sherd #1	0.84 mm	51.8 x 67.0 x 82.2
Sherd #2	0.76 mm	76.0 x 107.3 x 88.5
Cup	0.66 mm	113.3 x 80.0 x 98.9
Object	Volume (mm^3)	Calculated dimensions (mm)
Vessel #1	336131	139.2 x 83.2 x 91.4
Vessel #2	263696	113.0 x 111.9 x 86.4
Sherd #1	35911	51.0 x 66.0 x 79.4
Sherd #2	38586	74.9 x 103.9 x 86.2
Cup	276440	111.6 x 79.0 x 98.3

Table 1. Reconstruction of two vessels, two sherds and a cup

The exact volumes of these objects are unknown and therefore the accuracy of the volume calculated through reconstruction can not be estimated. However, we can measure the bounding cuboid and compare it with the dimensions of the model. Table 1 summarizes the results. The resulting models, shown from three views, are depicted in Figure 16. All models are built with an octree resolution of 256^3 and using 360 views.

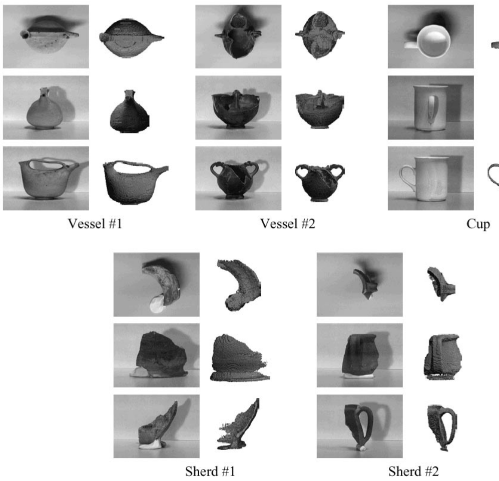


Figure 16. 3D models of two vessels, two sherds and a cup

6. CONCLUSION

Many In this paper a combination of a Shape from Silhouette method with a shape from Structured Light method was presented, which create a 3D model of on object from images of the object taken from different viewpoints. It showed to be a simple and fast algorithm, which is able to reconstruct models of arbitrarily shaped objects, as long as they do not have hidden concavities, i.e., concavities not visible in any of the input images. For concavities visible to the active system, results show that the computed volumes provide the correct model. The algorithm is simple, because it employs only simple matrix operations for all the transformations and it is fast, because even for highly detailed objects, a high resolution octree (256^3 voxels) and a high number of input views (36), the computational time hardly exceeded 1 minute on a Pentium II. Already for a smaller number of views (12) the constructed models were very similar to the ones constructed from 36 views and they took less than 25 seconds of computational time.

For archaeological applications, the object surface has to be smoothed in order to be applicable to ceramic documentation, for classification, however, the accuracy of the method presented is sufficient since the projection of the decoration can be calculated and the volume estimation is much more precise than the estimated volume performed by archaeologists.

7. REFERENCES

[1] H. Baker. Three-dimensional modelling. In *Proc. of 5th Intl. Conf. on AI,* pages 649-655, 1977.

[2] P. Besl. Active, optical range imaging sensors. MVA, 1(2):127-152,1988.

[3] H. H. Chen and T. S. Huang. A survey of construction and manipulation of octrees. *Computer Vision, Graphics, and Image Processing,* 43:409-431, 1988.

[4] C. H. Chien and J. K. Aggarwal. Volume/surface octrees for the representation of three-dimensional objects. *CVGIP,* 36:100-113,1983.

[5] R. M. Haralick and L. G. Shapiro. Glossary of computer vision terms. PR, 24(1):69-93, 1991.

[6] M. Kampel and R. Sablatnig. On 3d Modelling of Archaeological Sherds. In *Proceedings of International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging, Santorini, Greece,* pages 95-98, 1999.

[7] M. Kampel and S. Tosovic. Turntable calibration for automatic 3D-reconstruction. In R. Sablatnig, editor, *Proc. of 24th AAPR Workshop,* pages 25- 31, 2000.

[8] K. Kutulakos and S. Seitz. A theory of shape by space carving. IJICV, 38(3):197-216, 2000.

[9] C. Liska and R. Sablatnig. Estimating the next sensor position based on surface characteristics. In *ICPROO,* volume I, pages 538-541, 2000.

[10] W. N. Martin and J. K. Aggarwal. Volumetric description of objects from multiple views. *PAMI,* 5(2):150-158, 1983.

[11] C. Menard and R. Sablatnig. Computer based Acquisition of Archaeological Finds: The First Step towards Automatic Classification. In H. Kamermans and K. Fennema, editors, *Interfacing the Past, Computer Applications and Quantitative Methods in Archaeology,* number 28, pages 413-424, Leiden, March 1996. Analecta Praehistorica Leidensia.

[12] V S. Nalwa. *A Guided Tour Of Computer Vision.* AddisonWesley, 1993.

[13] W. Niem. Error analysis for silhouette-based 3D shape estimation from multiple views. In N. Sarris and M. Strintzis, editors, *Proc. of Intl. Workshop on Synthetic-Natural Hybrid Coding and 3D Imaging,* pages 143-146, 1997.

[14] J. O'Rourke and N. Badler. Decomposition of threedimensional objects into spheres. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* PAMI-1 (3):295-305, 1979.

[15] C. Orton, P. Tyers, and A. Vince. *Pottery in Archaeology.* 1993.

[16] M. Potmesil. Generating octree models of 3D objects from their silhouettes in a sequence of images. *CVGIP, 40:1-29,* 1987.

[17] Y. Shirai. *Three-Dimensional Computer Vision.* Springer-Verlag, 1987.

[18] S. K. Srivastava and N. Ahuja. Octree generation from object silhouettes in perspective views. *CVGIP,* 49:68-84, 1990.

[19] R. Szeliski. Rapid octree construction from image sequences. *CVGIP: Image Understanding,* 58(1):23-32,1993.

[20] S. Tosovic. Lineare Hough-Transformation und Drehtellerkalibrierung. Technical Report PRIP-TR-59, Institute of Computer Aided Automation, Pattern Recognition and Image Processing Group, Vienna University of Technology, Austria, 1999.

[21]S. Tosovic and R. Sablatnig. 3d modeling of archaeological vessels using shape from silhouette. In *Proc. of Conf. on 3-D Digital Imaging and Modeling, pages* 51-58, 2001.

[22] J. Veenstra and N. Ahuja. Efficient octree generation from silhouettes. In *CVPR,* pages 537-542, 1986.

8. ACKNOWLEDGEMENTS

This work was partly supported by the Austrian Science Foundation (FWF) under grant P13385-INF, the European Union under grant IST-1999-20273 and the Federal Ministry of Education, Science and Culture.