Available online at www.sciencedirect.com
SCIENCE DIRECT
Image and Vision Computing 22 (2004) 1137–1153
image AND vision COMPUTING
www.elsevier.com/locate/imavis

3D reconstruction of real objects with high resolution shape and texture

Y. Yemez[a,*], F. Schmitt[b]

[a]Computer Engineering Department, Koç University, 34450 Sarıyer, Istanbul, Turkey
[b]Signal and Image Processing Department, ENST-CNRS URA820, 46 rue Barrault, 75013 Paris, France

Received 23 October 2002; received in revised form 9 June 2004; accepted 23 June 2004

Abstract

We present a robust and accurate system for 3D reconstruction of real objects with high resolution shape and texture. Our reconstruction method is passive, the only information needed being 2D images obtained with a calibrated camera from different view angles as the object rotates on a turntable. The triangle surface model is obtained by a scheme combining octree construction and marching cubes algorithm, which is adapted to the shape from silhouette problem. We develop a texture mapping strategy based on surface particles to adequately address photography related problems such as inhomogeneous lighting, highlights and occlusion. Reconstruction results are included to demonstrate the attained quality.

Keywords: 3D object reconstruction; Shape from silhouette; Marching cubes; Octree; Surface particles; Texture mapping

1. Introduction

Real objects of any type via digital capture and reproduction may become virtual objects of multimedia applications over the Internet or on CD-ROM. Typical applications are, for example, interactive CD catalog creation and electronic publishing of museum art objects or commercial products. The most common existing platforms for such applications are Quick-Time VR like techniques that store and sequentially display various 2D views of an object, giving an illusion of 3D. The information stored in this way is however highly redundant and does not allow the user to select an arbitrary point of view which is not included in the view sequence.

Alternatively, it is possible to estimate the 3D object shape from its already available various 2D views so that storing 2D image information from many different views becomes unnecessary. Rather, a 3D structural model of the whole object is kept together with the texture (or surface color) information attached to it. Moreover, the object can be viewed from any angle regardless of the acquisition process, and displayed through standard rendering techniques.

Another source of information to capture 3D object shape is the direct 3D information acquired actively by laser range scanners or other coded light projecting systems. Such 3D data acquisition systems can be very precise, but have several drawbacks. Most of them are very expensive, and they require special skill and know-how for the acquisition process itself. They often necessitate specific environmental conditions, and do not perform well with objects made of materials absorbing light such as fur or velvet, or when the object surface is very shiny. In addition, these scanners concern more with shape rather than with texture. Even regarding the shape only, they require very sophisticated approaches for 3D matching and data fusion of different surface patches that are scanned separately, to reconstruct the object as a whole. Only few scanners are capable of recording 3D shape information concurrently with color texture. For those with this capability, color information is incorporated either by an RGB camera (or a linear array) [8] or by using three different laser wavelengths during the 3D acquisition process [30]. In the first case, the color acquisition device is usually different from the 3D one. The color texture and the 3D data, being acquired with two different geometrical configurations, still have to be

* Corresponding author.
E-mail addresses: yyemez@ku.edu.tr (Y. Yemez), francis.schmitt@enst.fr (F. Schmitt).

0262-8856/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2004.06.001

registered. In the second case, the color and the 3D information are perfectly registered, but the obtained colors are not colorimetrically faithful since the spectral reflectance of the object surface is sampled only at three wavelengths.

In this work, we focus on passive methods which can work in an ordinary environment with simple devices and provide possible means to extract 3D information from a set of pictures of an object. The two main passive methods are: shape from silhouette and shape from stereo. In the first one [5], 3D object shape is computed as the intersection of silhouette cones obtained by back-projecting object silhouettes from different camera views. This computation can be realized in a very efficient manner by volume carving (or silhouette extrusion) and usually gives a good and robust estimate of the visual hull of the object shape [17]. The drawback of the method is that there may exist hidden concavities on the object surface that cannot be recovered by silhouette extrusion. Shape from stereo techniques [1,12, 15,24,28] on the other hand, rely on texture information and estimate the depth from multiple image correspondences. The basic assumption here is that the pixel intensity of a 3D point does not differ significantly when projected onto different camera views. This assumption however does not hold in most practical cases due to photography related problems such as shading, inhomogeneous lighting, highlights, occlusion and radiosity. Common opinion is that it is very difficult to obtain robust and reliable shape by using only stereo information. An attempt to overcome these difficulties for object reconstruction problem is to combine shape from silhouette and shape from stereo methods [9–11, 23]. The common strategy in such a combined scheme is first to compute the visual hull with a shape from silhouette technique and then to recover the hidden concavities with stereo information. A few articles following this approach demonstrate some improvements over traditional shape from stereo techniques and benefit from starting with a good initial estimate of the object shape. However, these techniques remain complex to implement and still suffer from the above-mentioned photography related problems. Another approach is to incorporate the shape from silhouette method in active reconstruction techniques such as shape from structured light [14]. The fusion of active and passive reconstruction techniques seems very promising for obtaining more reliable and complete surface representations of objects containing strong surface concavities or non-textured surfaces. For all these combined approaches, an accurate and robust shape from silhouette object reconstruction scheme remains essential. In this paper we present such a scheme for high resolution reconstruction of object shape and texture.

There are now many works that consider the object reconstruction problem by making use of the shape from silhouette technique [2,3,6,7,13,20,22,27,32,42]. However, the published articles in this domain do not exhibit the details of the various tasks of the whole reconstruction process, and the reconstruction results that they present are generally obtained at resolutions which are lower than that of the quality that we aim at, especially for object texture. There are five basic steps to accomplish the complete reconstruction task: camera calibration, 2D silhouette extraction, 3D shape estimation, surface modeling and texture mapping.

Camera and turntable calibration is perhaps the most crucial step in obtaining an accurate 3D reconstruction, but at the same time it is the most restricting factor for the flexibility of a reconstruction system. On one side, there are fully-calibrated systems where the scene-camera geometry and internal camera parameters are either known a priori or estimated via the use of dedicated calibration patterns. On the other side, there are self-calibrated systems [9,12,24] that reduce the necessary prior knowledge about the scene-camera geometry only to a few internal and external constraints. The technique proposed in Ref. [12], for instance, can achieve a full 3D Euclidean reconstruction without need for a special calibration pattern by only imposing some constraints such as rotation around a single axis and fixed internal camera parameters. Point correspondences have to be first established with image analysis techniques for which the success mainly depends on the texture characteristics of the object to be reconstructed. Self-calibrated systems offer a desirable flexibility for the 3D reconstruction problem, eliminating most of the specific arrangements for the acquisition environment. However, their accuracy is not comparable to that of the fully-calibrated systems since they require complex optimization techniques which are slow and difficult to converge. In this respect, one can argue that there is a compromise between the reconstruction accuracy and the flexibility. To what extent the accuracy is needed in fact depends on the application and the specific needs of the user. In some applications such as 3D digitization of cultural heritage, maximum accuracy may be a very crucial requirement. Furthermore, if one attempts to incorporate stereo information or active techniques in order to recover hidden concavities, accuracy becomes almost indispensable for consistent fusion of multiple information coming from different sources and geometries. As we aim for an accurate and high resolution reconstruction, we favor here calibration accuracy over flexibility.

Most shape from silhouette techniques make use of a voxel grid structure as an intermediate representation from which the surface is then extracted using a triangulation technique such as marching cubes [21]. The vertex positions of the resulting mesh are thus limited to the voxel grid if no additional caution is taken. The general strategy is to apply post-processing on the resulting mesh such as pulling vertices towards silhouette boundaries via projection, or using smoothing and decimation to avoid visually disturbing aliasing effects [12] at the cost of a reduced approximation of the silhouette information. Vertex precision is an important issue in achieving faithful 3D shape representations. In this respect, another possibility is to compute the intersection of each voxel edge with silhouette

cones so as to localize vertex positions precisely on the visual hull of the object surface; in this manner the aliasing effects can be avoided during the triangulation process without any additional effort to guarantee manifold topology, and the object surface is more accurately reconstructed [32].

The most detailed article published in the shape from silhouette domain, and which addresses a whole reconstruction scheme, is Niem's work [27]. The particularity of this work is simultaneous acquisition of the object and the calibration pattern. This gives some flexibility in the acquisition set-up, but in turn corrupts the accuracy of silhouette extraction and calibration process and limits the useful area covered by the object in the 2D images. As for 3D shape computation, an intermediate voxel representation is first constructed by decomposing the bounding box into pillar-like volumes and then a mesh growing algorithm is applied to get the final surface model. Computationally, Niem's technique is applicable to obtain relatively low resolution models and the precision of the resulting surface, i.e. mesh vertex positions, is limited to the finest resolution of the voxel representation.

High resolution reconstruction of surfaces becomes practically feasible only with the use of space and time efficient algorithms. Adaptive carving via octree representation is one of the most appropriate solutions to the efficiency problem, providing a means to avoid unnecessary processing of all voxel grids [12,32,35]. The algorithmic complexity of adaptive carving techniques is in general in $O(R^2)$, where R stands for the resolution of the uniform voxel grid. Another possibility is to use the marching intersections (MI) data structure as proposed in Ref. [38]. In this technique, the MI structure is iteratively updated to store the precise locations of the intersections of each silhouette boundary with the voxel grid; the algorithmic complexity can still be thought of as in $O(R^2)$, however the scan-conversion algorithm employed to compute the intersections of each projected voxel grid line with the silhouettes on the image plane yields an additional significant computational load.

Texture mapping phase is crucial in improving the visual quality of 3D object models. It can even partly compensate geometric imprecisions in the reconstructed 3D models such as hidden concavities that cannot be recovered with silhouette-based techniques. At this stage, the texture of the surface model, i.e. the texture of each mesh triangle, has to be computed from available images. The usual way of doing this, is to select the most suitable image for each triangle [22,27]. Although some post-filtering can be applied on the mapped texture [27], this strategy, that treats a given triangle as a whole in source image selection, fails to address adequately the photography related problems such as highlights.

In view of the above discussion, this paper has the following contributions to the problem of 3D object reconstruction from multiple views:

1. It describes the complete reconstruction system that we have built by combining different techniques and addresses the details of the system components which determine the resulting reconstruction quality. The accuracy of the reconstructed object shape and the quality of the texture attached on it have been the main concerns during the development of the system presented here, for which the most limiting factor is the original image quality.
2. A scheme combining octree construction and isosurface extraction via marching cubes algorithm is presented for the shape from silhouette problem. The use of octree representation makes it possible to achieve very high resolutions as we aim, whereas the fast marching cubes method is adapted via an appropriately defined isolevel function so as to work with binary silhouettes, resulting in a triangle mesh with vertices located accurately on the object visual hull. The precision of vertex positions is not limited to the resolution of the constructed surface model, rather it is determined by the resolution of available images. The octree representation and the marching cubes algorithm have been used in conjunction before for the general object reconstruction problem [40], however the use of these techniques in the case of the shape from silhouette problem necessitates specific considerations.
3. A robust background removal (silhouette extraction) technique which is quite general and independent of the object type or the lighting conditions is proposed.
4. A particle-based texture mapping method is developed, that adequately addresses the photography related problems. In particular, the problem of highlights, which has been ignored in other works, is taken into account and handled during the texture mapping process.

Our basic setup for image acquisition consists of a computer, a digital CCD camera and a high precision turntable on which the target object is placed and which is driven by the computer. Computer software is also available to drive the digital camera and to download the acquired images. Once the object is posed on the turntable, a sequence of images is captured, each corresponding to a different view, by rotating the turntable stepwise (Fig. 1). The rotation step size depends on the application but a value of 10° is usually sufficient.

The whole 3D shape reconstruction process contains several tasks, each described throughout the paper (Fig. 2). An accurate camera calibration, which is the first step of the scheme, is essential in achieving high quality object reconstruction. We address this problem in Section 2 where we first describe briefly the camera calibration technique [19] and the turn-table calibration method [26] that we use. In Section 3 some photography related issues of the original image acquisition process, that in fact constrain the achievable final quality, are discussed. In Section 4 we address the problem of extracting smooth object silhouettes

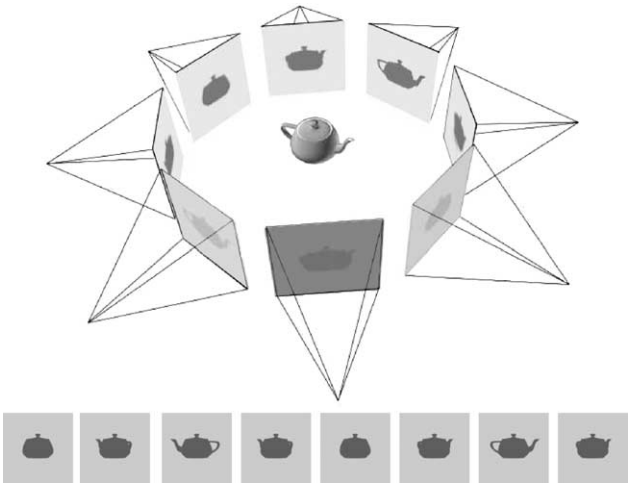

Fig. 1. Image acquisition.

that are later used to obtain an octree representation of the object via a shape from silhouette carving process described in Section 5. In Section 6, we consider the surface triangulation problem. The octree volume is triangulated by a fast marching cubes algorithm that we have adapted to the shape from silhouette problem so as to work with binary silhouettes. An important part of the work is to assign the original texture to the recovered object; this is addressed in detail in Section 7. Finally in Section 8, we give some concluding remarks. The experimental results concerning each different task of the whole reconstruction system are mostly presented throughout the text, whereas the final reconstruction quality in color is demonstrated later at the end of the paper.

2. Calibration

The calibration process consists of two phases. We first determine the intrinsic and extrinsic parameters of the camera and then proceed to the turntable calibration.

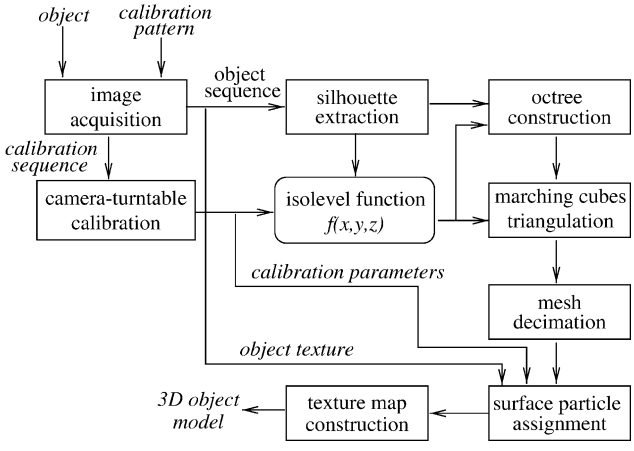

Fig. 2. Block diagram of the object reconstruction scheme.

2.1. Camera calibration

Calibration techniques usually make use of a 3D known object which is called calibration pattern. Classical methods for calibration such as Tsai's technique [39] rely mostly on the geometric precision of the fabricated pattern. Therefore, they lose their accuracy due to small possible errors that may occur in constructing the pattern or due to small geometrical deformations after temperature or humidity level changes when the pattern has been printed on paper. Since in practice it is expensive and time consuming to fabricate precise calibration patterns, Lavest et al. [19] have developed a calibration technique which does not rely on the fabrication accuracy of the calibration pattern or on the precise measurement of its geometric characteristics. This technique, that we used for camera calibration, rather relies on the precise detection of the characteristics of the projected pattern on the image plane. In addition to the intrinsic and extrinsic camera parameters, the geometric characteristics of the calibration pattern are also re-estimated and tuned concurrently via an iterative non-linear optimization technique. It has been shown that by employing multiple images and by initializing the geometric characteristics of the pattern with only their roughly estimated values, very accurate calibration results can be obtained [19].

The camera calibration process requires the acquisition of various views of the calibration pattern while maintaining the same camera adjustments chosen for the object acquisition. A set of about 10 images is acquired so that the various positions of the calibration pattern cover the whole 3D range of the object to be constructed. The positioning of the pattern is simply done manually with the help of a mechanical support. A view of a 10 cm$\times$10 cm calibration pattern and its support that we use are shown in Fig. 3. The dot center positions in the pattern coordinate system initially need to be known only with a precision of 1 or 2 mm. The calibration process finally provides us with the internal parameters of the camera (focal length, principal point position, skew factor) and the external parameters which define the rigid transformation between the fixed camera coordinate system and the coordinate system associated to the calibration pattern. This transformation varies from one view to another and each transformation is given by a different set of rotation and translation parameters, depending on the relative position of the pattern with respect to the camera.

2.2. Turntable calibration

At the second phase of the calibration process, the 3D location of the rotation axis with respect to the fixed camera coordinate system is to be determined. Since our turntable is driven by a precise step motor, we know accurately the rotation angle θ corresponding to any position of the turntable with respect to a reference position.

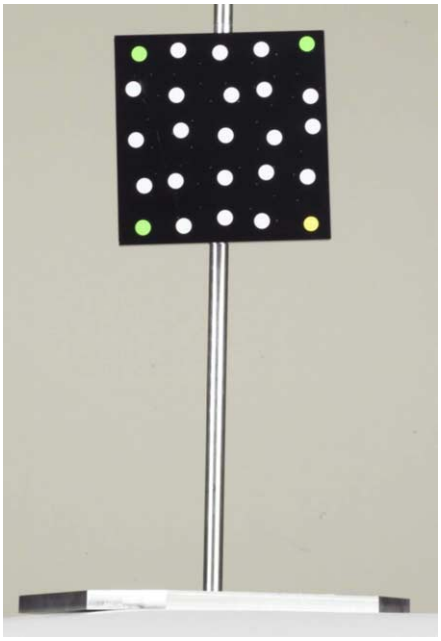

Fig. 3. A view of the calibration pattern: the pattern is composed of white dots of sticky paper attached on a black solid base. The dots in color are used to automatically recognize the orientation of the pattern.

The knowledge of the rotation axis location will then allow us to deduce in the camera coordinate system the rigid transformation of the object that corresponds to the rotation of the turntable with any given angle. This also allows us to define an object coordinate system, which is attached to the turntable, one of its axis being aligned with the rotation axis. With this information we will be able to construct the visual hull of the object, silhouette per silhouette.

The turntable calibration requires the acquisition of a second image sequence of the pattern placed on the turntable as it moves in pure rotation around the turntable axis with the same camera adjustments. We then proceed to the determination of the extrinsic calibration parameters related to this rotation sequence as described in Section 2.1. The intrinsic parameters need not be computed again since the camera adjustments remain unchanged. This provides us with a rigid transformation which relates the calibration pattern coordinate system to the fixed camera coordinate system for each view of the pattern in the rotation sequence. The rotation axis is then computed from the set of rigid transformations through the numerical procedure that has been detailed by the authors in Ref. [26].

The camera calibration process via the method in Ref. [19], followed by the rotation calibration as described in Ref. [26], results in very accurate parameter values. The estimated rotation angles come out to be almost identical to the rotation angles predefined by the turntable. The maximal variation obtained in our experiments is about 0.05°, which is very close to the 0.01° precision claimed by the turntable manufacturer. Also, different rotation axis estimates using different view combinations yields very coherent values, the rotation axis unit vector coordinates having a maximal variation of 0.2%. The precision achieved is quite satisfactory and allows us to attain a very good quality object reconstruction.

3. Photographic considerations

In our reconstruction experiments we have worked on four different objects shown in Fig. 4. Each object has been pictured horizontally from a number of different views with a fixed angle step size, covering a complete turn of 360°. The sequence of the Anyi statuette consists of 72 (2008×3040) color images, whereas for the other objects each sequence contains 36 images. The original image size is 356×512 for the Coignard statue, 1280×1024 for the Horse object and 700×1524 for the Cycladic statuette.

There are important details about how the original pictures are taken, which eventually determine the final quality. These are mostly photography related issues regarding illumination and selection of the background. Radiosity effects, highlights, and non-uniform lighting are such problems which deteriorate the quality of model texture. These problems should be handled either by avoiding them with proper adjustment of the acquisition environment or by taking them into account during the texture mapping process.

One of the most obvious problems is that of highlights as observed, for example, on the ceramic Horse object. Highlights can even be desirable as they can be used to give a hint to the shape of an object or to emphasize its metallic or ceramic composition. Highlights depend on the relative position of object, lights and camera. This means that they change position along the object surface from one image to the other. This can be problematic in recovering the diffuse texture of the original object. Another problem, that can be visually very disturbing, is texture discontinuity that can appear along triangle edges of the 3D surface model; this artifact occurs mainly due to inhomogeneous lighting that produces shaded regions and inconsistencies in the texture images of the object when pictured from different angles. Such shaded regions and highlights should be avoided in the original images by using a diffuse lighting as constant and homogeneous as possible. The spectacular effect of shading or highlights can then be simulated by adding lighting effects to the viewer once the original diffuse color of the object is mapped properly onto the model.

Another lighting problem is that of radiosity. If, for example, we have a blue background as in the case of the Coignard sequence, all the light reflected by the background will tend to tint the object border with a bluish color. This phenomenon is more severe in dark areas of the object surface. Clever use of lighting to avoid this is a common practice for photographers. The original pictures should in fact reflect as accurately as possible the diffuse nature of the object with as few lighting artifacts as possible. Yet another

Fig. 4. Example of original images from (left to right, top to bottom) the Coignard, Cycladic, Anyi and Horse sequences.

problem is whether the object or the camera is fixed or turning with respect to each other. The easiest way to have a uniform lighting for all viewing angles is to turn the object and keep the camera in fixed position, since in this case it suffices to illuminate the pictured object uniformly with respect to the field of view of the fixed camera. Finally, the question of how many pictures are needed to get a good reconstruction should be answered at this stage. In fact the more different silhouettes are seen, the more information we have for reconstruction. So if possible taking pictures from above and below is a good idea especially if some hidden concavities are revealed in this way.

4. Object extraction

Once the original photos are available, the next step is to extract the object from the background as cleanly as possible. There exist two standard techniques for object extraction. The first one relies on the uniformity of the background. A selection criterion, which is based mainly on the hue, saturation or brightness of pixel color, is defined so as to differentiate the background from the object. The second technique is the background subtraction method which necessitates two consecutive acquisitions for the same scene, one with the object and the other without the object, keeping the camera and the background unchanged. The difference of the two images when thresholded is supposed to give us directly the object silhouette. These two methods, though they might work well depending on experimental conditions (uniform lighting, camera quality, object type, etc.), may in some cases confuse the object with the background. First, there is color interference problem between the background and the object. A blue background, for example, tints the object with a bluish color at the borders, especially at dark and not sufficiently illuminated parts, or vice versa as we observe severely in the Coignard sequence (Fig. 4). Second, the background color may happen to be the same with the object color, due to lighting deficiencies or color similarity. An object may naturally have similarly colored parts with the chosen background, or shaded dark regions (as in the Cycladic sequence, for example) can be confused with a black background, or in the case of white background, highlights may cause erroneous holes inside the silhouettes, which could later appear spanning the whole reconstructed volume.

In view of the problems discussed above, we use an object extraction strategy which is quite general and independent of the object type and the lighting conditions.

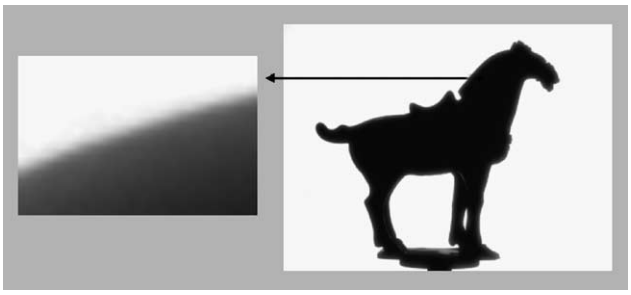

Fig. 5. The background saturated horse image and a zoom on the boundary.

This strategy requires two consecutive acquisitions for each object view. One of the acquisitions is for object texture, whereas the other is dedicated to extraction of the silhouettes to be used for recovering the object shape. The strategy is as follows. The object is first pictured with proper lighting and a background chosen so as to lessen the color interference. Then a white background is placed sufficiently distant to the object. The object is kept untouched, still in the scene, but not illuminated at all. On the other hand, the white background is exposed to light so as to be saturated totally. The total saturation can be assured by adjusting the energy of lighting or the camera exposure time. Once the scene is pictured with the saturated background, the unsaturated region of the acquired image provides us directly with the object silhouette. As observed in Fig. 5, a small interference on the object boundary may still be observable, but the transition between the background and the object comes out to be so sharp that a simple thresholding suffices to obtain the correct silhouette. In Fig. 6, we show the silhouettes of the Horse and the Anyi objects, each extracted with the technique described above.

5. Volumetric shape construction

Object silhouettes, when projected back to 3D world by means of camera calibration parameters, define some silhouette cones. The basic idea for shape recovery in shape from silhouette approach relies on the observation that the object volume must lie inside the intersection of these silhouette cones [17], as shown in Fig. 7. In this section we will describe how to compute this intersection efficiently by making use of the classical octree structure as an intermediate volumetric representation that will later in Section 6 be triangulated in order to obtain the final surface model.

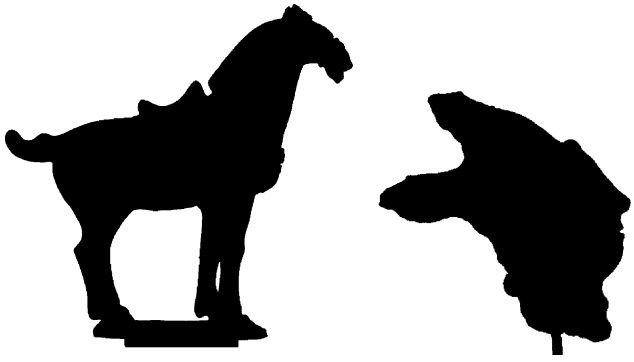

Fig. 6. Two examples from the extracted silhouettes of (left) the Horse and (right) the Anyi sequences.

5.1. Octree representation

An octree is a hierarchical tree structure that can be used to represent volumetric data in terms of cubes of varying sizes [4,5,35]. Each node of the octree corresponds to a cube in the octree space which is either totally inside the object volume, totally outside the volume, or on the object boundary surface. The object volume to be represented can be assumed to be bounded by an implicit surface $f(x,y,z)=0$ which provides the necessary information to decide whether a given cube is inside, outside or on the object surface and we mark the corresponding node with a label F equal to IN, OUT or ON, respectively. The octree space can be considered as a 3D grid of $2^R \times 2^R \times 2^R$ unit cubes, where R denotes the highest resolution level of the octree. A coordinate system is defined such that one corner of the octree space is located at the origin and its corresponding edges are aligned with the positive coordinate axes. Then each cube s_i^r in the octree space is specified by one of its corners (e.g. the one closest to the origin) located at the grid points (x_i,y_i,z_i) and its depth r in the octree hierarchy. The octree representation is obtained by recursively subdividing each parent cube s_i^r at level r into eight child cubes, starting from the root node, i.e. the bounding cube. The OUT and IN cubes need not be further subdivided. The recursive subdivision process continues only for ON cubes until the unit cubes corresponding to the leaf nodes of the highest octree level are reached. As a result, the octree representation contains larger cubes at the very inside of the object, whereas the cubes get smaller as they approach to the boundary surface, depending on the geometrical complexity and the chosen resolution.

5.2. Octree construction

The octree construction is based on a criterion to decide whether a given cube is IN, OUT or ON. Recall that the object volume lies inside the intersection of the silhouette cones. Thus an octree voxel (cube) must ideally be included by all silhouette cones if it is really inside the object volume. Following basically [35], the methodology to find this intersection corresponds to carving the bounding box in terms of voxels by iteratively excluding the parts falling outside the silhouettes, as shown in Fig. 7. Recall that the object silhouettes from different views have already been extracted and the camera model is based on the pin hole model of the perspective projection. Since a cube which is inside the object volume must lie at the inside of all

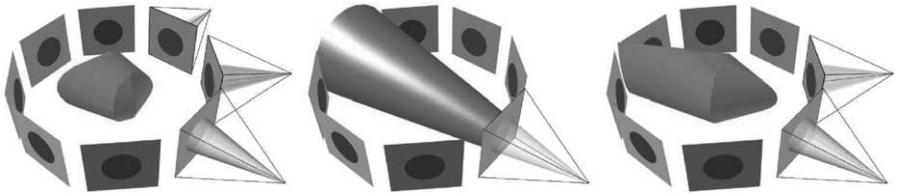

Fig. 7. Carving process by silhouette cones to locate object shape.

silhouette cones, then its projection onto each image plane of the acquisition views must also lie inside the corresponding silhouette. Thanks to the calibration process, the camera parameters can be used to project each cube of the octree to the image planes. If a cube happens to fall outside of even one single silhouette, it is labeled to be OUT. If not, but if it falls onto the boundary, i.e. if it is partially inside, of any single silhouette, it is labeled to be ON. Otherwise, the cube is assumed to be IN.

The crucial point in the octree construction is how to define the criterion that decides whether a given octree cube is inside, outside or on the object surface. This criterion, which will be defined in Section 6.4, should be consistent with the marching cubes algorithm that will be used to obtain a triangular surface representation from the ON cubes of the octree representation.

A drawback of the silhouette extrusion method described above is that there may exist concavities in the object which cannot be captured by this method [18] although it is often possible to reveal some hidden concavities by increasing the number of silhouettes, i.e. the number of viewing directions.

5.3. Voting approach

The silhouette extrusion method is highly sensitive to holes which might appear inside the silhouettes due to the problems mentioned in Section 4, whereas erroneous additional regions are mostly eliminated during the carving process. A method proposed by another group [22] to solve this problem is the voting approach. Instead of declaring a point to be OUT as soon as an image is found for which the point lies outside the silhouette, this method sets a threshold of minimum number of silhouettes for which the point must be outside in order to be considered as OUT. A drawback of the voting approach is that it tends to perturbate even pure convex objects with regular non-convex protrusions on the surface especially as the threshold for the number of silhouettes gets larger. This effect of voting approach, for the case of four different views, is shown in Fig. 8.

This method should thus be used to extract objects only from very noisy backgrounds for which the segmentation step might not have worked correctly. In our experiments we did not need to employ the voting technique since the silhouettes that we extracted with the technique described in Section 4 are quite reliable.

6. Surface triangulation

6.1. Surface representation

Different possibilities are open regarding surface representation. These are voxels, particles, triangles and more complicated parametric primitives such as splines or NURBS Voxels, the spatial equivalent of pixels, are used to represent volumes, but can also be used to represent surfaces just as we use pixels to represent silhouettes. A related primitive is the particle which is defined by its color, orientation and position. Voxels fill up space and are arranged in a regular grid, while particles are located on the surface and may be arranged in an irregular manner. A problem with particles and voxels is that the representation breaks up when zooming on the object: voxels become visible as cubes and particles become sparse and holes appear on the surface. Antialiasing is another difficult point: mainstream devices can antialias triangle representation with efficiency, but additional processing is required for particles and voxels. On the other hand, triangles or higher degree parametric surfaces are true surface representations. Their disadvantages are that they are more difficult to extract from original data and that they need to be simplified for efficiency. For representation of our 3D models, we will

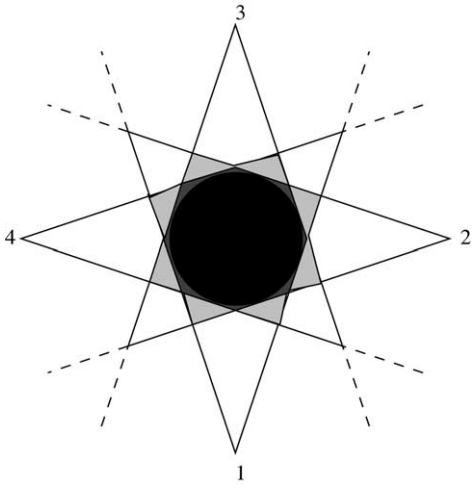

Fig. 8. Effect of the voting approach on reconstructed surface. The black region is the real object to be reconstructed by silhouette extrusion method. The intermediate dark-grey region together with the black region represents a cross-section of the reconstruction obtained without using voting approach, whereas the outer light-grey regions are additional non-convex perturbations introduced by voting with a threshold equal to 2.

use the triangle mesh which represents the surface as a set of connected triangles. Zooming without loss of quality relative to the original images and correct antialising at reconstructed object silhouettes will thereby become possible. The polygonal approximation may still be visible especially when zooming on object borders but arguably this is a less distracting problem than visible cubes or holes on the object surface.

6.2. Marching cubes

The volumetric octree representation uses multiscale voxels and has to be transformed into a triangular surface model so that it can easily be handled with standard synthesis algorithms in computer graphics. Marching cubes algorithm [21,25] is a simple and practical technique to extract surfaces from volumetric data on which an implicit continuous scalar function is available to define the isolevel information. The algorithm uses the isolevel function value at the eight corners of a cube to interpolate an isosurface passing through that cube. Since a smooth isosurface can intersect a cube only in a limited number of ways, a triangulation of all the connected cube intersections is created using a look-up table. The number of all possible configurations is 256, however because of symmetry and rotational equivalence, the number of configurations can be reduced to 15. The marching cubes algorithm uses a look-up table that locally generates the correct patch of triangles corresponding to one of these 15 cases.

6.3. Definition of the isolevel function

Marching cubes algorithm assumes that the object surface to be triangulated can be represented locally by an implicit surface $f(x,y,z)=0$. The isolevel function $f(x,y,z)$, which provides the level information necessary at the cube corners, can also be used in the octree construction to define the criterion that decides whether a given octree cube is IN, OUT or ON. The isolevel function in our case is deduced from the sequence of binary silhouettes. When directly used, the binary silhouettes give us a binary isolevel function (0 or 1) yielding jagged structured surface constructions which are visually disturbing. In order to have more realistic and visually more pleasant constructions, we define the function $f(x,y,z)$ as follows. By using the camera calibration parameters, a given space point (x,y,z) can be projected onto the image planes. Let (u_n,v_n) denote the projection of (x,y,z) onto the silhouette image I_n of the sequence, $n=0,1,\ldots,N$:

$$(u_n, v_n) = \mathrm{Proj}_{I_n}(x, y, z). \tag{1}$$

The projection (u_n,v_n) is a real valued point in the 2D continuous image plane whereas the silhouettes $I(m,n)$ are binary images defined on a discrete grid. The bilinear interpolation function $G(u,v)$ provides us with a measure of the distance of the projection to the silhouette boundary with values between 0 and 1

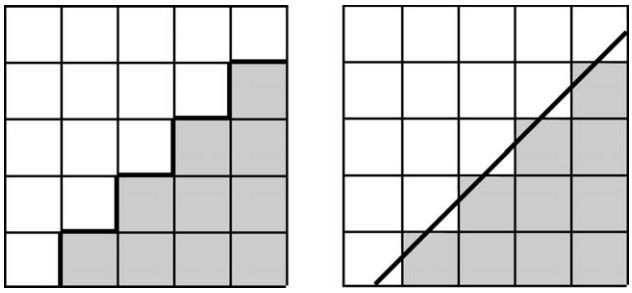

Fig. 9. Silhouette boundaries (left) without interpolation and (right) with linear interpolation ($\xi=0.5$).

$$\begin{aligned} G(u,v) = (1-\beta)((1-\alpha)I(\lfloor u \rfloor, \lfloor v \rfloor) + \alpha I(\lfloor u \rfloor \\ + 1, \lfloor v \rfloor)) + \beta((1-\alpha)I(\lfloor u \rfloor, \lfloor v \rfloor \\ + 1) + \alpha I(\lfloor u \rfloor + 1, \lfloor v \rfloor + 1)), \end{aligned} \tag{2}$$

where $(\lfloor u \rfloor, \lfloor v \rfloor)$ denotes the integer part and (α,β) is the fractional part of the coordinate (u,v). This interpolation, when thresholded with a value $\xi \in (0,1)$, smooths the staircase form of the silhouette boundaries. The effect of this first-order filtering is visualized in Fig. 9.

The function $G(u_n,v_n)-\xi$ provides isolevel values in 2D which could be used to extract isocurves from the silhouette image I_n. If we consider the 3D isosurface to be extracted as the interpolated union of the back-projected isocurves, the 3D isolevel function $f(x,y,z)$, which we need to extract isourfaces, can be defined in terms of the 2D isolevel values as follows

$$f(x, y, z) = \min_n \{G[\mathrm{Proj}_{I_n}(x, y, z)] - \xi\}, \tag{3}$$

where $0<\xi<1$. The object surface is assumed to satisfy the implicit equation $f(x,y,z)=0$. When defined as in Eq. (3), the isolevel value f of a 3D point (x,y,z) is provided by the silhouette I_k where the projection (u_k,v_k) is exteriorized the most with respect to the silhouette boundary, or in other words, where the 2D isolevel value $G-\xi$ assumes its minimum value. This definition of $f(x,y,z)$ is also consistent with the fact that a point or an octree cube whose even one single projection is outside the corresponding silhouette region is considered to be OUT.

The choice of ξ in Eq. (3) effects the position of the isosurface with respect to the object surface. The greater the value of ξ is, the closer is the isosurface to the interior of the object. Conversely, the values of ξ which are smaller than 0.5 tend to thicken the reconstructed object slightly by smoothing the small hollows along the surface.

Instead of the bilinear interpolation given in Eq. (2), higher order filters or parametric representations could also be used to better approximate the function needed to reconstruct the silhouette image sampled with the CCD.

However, this would come at the price of a higher blurring of the signal and a longer computation time.

6.4. Octree construction criterion

The isolevel function $f(x,y,z)$ provides the information to test the position of a given space point (x,y,z) with respect to the object surface $f(x,y,z)=0$. The regions where the function f takes negative, zero or positive values determine the outside, the boundary, or the inside of the surface, respectively. An octree cube is a volumetric unit, therefore another criterion, which is based on this isolevel function, should be defined to test whether its projection on a certain silhouette image, which corresponds to a two-dimensional region, is inside, outside or on the silhouette boundary. Assuming that at least two perpendicular views of the object are available, it suffices to test only the statuses of the discrete edge points of the cube s_i^r sampled in the octree space grid at the highest resolution R. Let E_i^r denote the set of grid points (x_j,y_j,z_j) which are situated along the edges of the cube s_i^r at any level r. Then the criterion that decides whether a given octree cube is IN, OUT or ON is defined as follows:

$$F(s_i^r)=\begin{cases}\text{OUT} & \text{if } f(x_j,y_j,z_j)<0,\ \forall\,(x_j,y_j,z_j)\in E_i^r,\\ \text{IN} & \text{if } f(x_j,y_j,z_j)>0,\ \forall\,(x_j,y_j,z_j)\in E_i^r,\\ \text{ON} & \text{else.}\end{cases} \tag{4}$$

The criterion defined above implies that for each octree cube s_i^r, eight corners and $(2^{R-r}-1)\times 6$ edge points should be tested before the decision. The test of eight corner points assures the compatibility of the resulting octree representation with the marching cubes algorithm, whereas the test of edge points is to preserve object details at the highest level R during the subdivision process. Since the silhouette of a cube is determined by at most six of its edges which are visible from the corresponding view direction, the test process does not need to take into account all the 12 edges of a cube. The six edges to be tested for a given view direction can easily be identified by a look-up table.

6.5. Marching octree cubes

Once constructed, an octree can be used in computer graphics to represent a 3D object as either a surface or a volume [5]. Due to our octree definition, there exist no two neighboring cubes s_i^R and s_j^R such that $F(s_i^R)=\text{OUT}$ and $F(s_j^R)=\text{IN}$. Thus there exists between the interior and the exterior of the object a layer of ON cubes s_k^R covering the full object boundary defined by $f(x,y,z)=0$. These ON cubes of the highest octree level can be fed into marching cubes algorithm, disregarding the remaining OUT and IN cubes (Fig. 10).

The level information at cube corners needed by the marching cubes algorithm is provided by the isolevel function in Eq. (3). This isolevel value can be thought of as a measure of how much a projected point is inside or outside the boundary, taking values between $-\xi$ and $1-\xi$, respectively. The real surface is assumed to cut a cube edge when one of the corners is inside and the other is outside. The triangle vertex on that cube edge is then placed at the zero level function point. This vertex position could be computed by linear interpolation of the isolevel values at the corners, as in the usual marching cubes technique. However, the isolevel function in Eq. (3) is continuous only within a band along binary silhouette boundaries and takes values in the range $(-\xi,1-\xi)$. At the outside of this band, the function becomes constant taking the value either $-\xi$ or $1-\xi$. Therefore, in order to locate precisely the intersection of the object surface with the cube edge, we employ a dichotomic subdivision process as shown in Fig. 11. This procedure ideally corresponds to a binary search of the point where the isolevel value is exactly equal to zero. In practice, the cube edge is subdivided up to a certain level according to the desired precision.

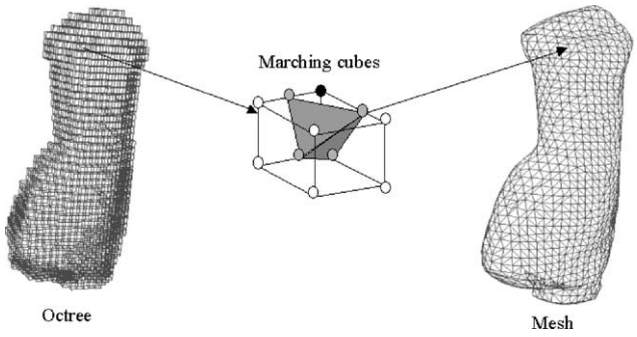

Fig. 10. Marching cube triangulation of the octree. The black and white points denote the cube corners which are inside and outside, respectively, whereas the gray ones are the triangle vertex points on the surface.

To demonstrate how the reconstructed object is effected by the choice of ξ in Eq. (3), we zoom on the horse object reconstructed at different values of ξ in Fig. 12. We observe that when ξ is chosen so as to be a small value such as 0.1, the reconstructed object tends to be thicker with smoothed small concavities like the mouth of the horse, whereas the ear, as a convexity, is sharpened compared to the case with $\xi=0.9$. The choice of ξ depends on the object shape, whether there are mostly concavities or convexities on it or which type of shape details one would like to emphasize. As an example, in our experiments we have chosen ξ as 0.4 and 0.6, respectively, for the Coignard and the Anyi statuettes.

Fig. 11. A marching octree cube and dichotomic subdivision process to find the exact vertex locations of the triangulation.

Fig. 12. Reconstructed horse at resolution $R=6$ with three different values of ξ, from left to right, 0.1, 0.6 and 0.9.

The use of the isolevel function computed via the dichotomic subdivision procedure described above makes it possible, even at low resolutions, to construct a faithful wireframe model of the object with triangle vertices located accurately on the object surface. This allows an efficient compromise between geometric accuracy and resolution especially when the object geometry is not very complex. Moreover, high resolution reconstructions do not suffer from jagged structured artifacts and are visually much more pleasant when the object model is zoomed on the boundaries (Fig. 13).

The advantage of using an octree instead of subdividing the object space into fixed size cubes is that its hierarchical structure reduces the number of voxels to be tested by the marching cubes algorithm thereby leading to very high resolution object models. Without using an octree the number of voxels in the search space becomes so large that very high resolution (e.g. one pixel precision) representations become unfeasible in terms of both memory requirements and computational aspects.

6.6. Mesh decimation

The octree construction followed by marching cubes algorithm as described above generates a triangular mesh consisting of an excessive number of triangles, which needs to be simplified due to rendering efficiency and fast transmission in network-based applications. If we consider how a cubic grid may be cut by a surface, it occurs often that a cube vertex falls inside or outside the object by a small distance, leading to the production of a small or thin triangle. Such small triangles are problematic since they introduce numerical imprecision (for normal vector computation, triangle texture definition and storage, etc.) and should preferably be merged into neighboring triangles. We have implemented a fast decimation algorithm that aims to solve these particular problems of the marching cubes algorithm, in a similar manner as in Ref. [33]. The procedure is as follows. All the triangles having an edge with a length smaller than half a cube edge are eliminated. These triangles are collapsed into points or segments depending on how many edges do not satisfy the criterion. The position of the new points or edges is determined by computing the mean of the involved points (Fig. 14). The decimation process described above eliminates almost half of the triangles produced by the marching cubes algorithm and yields triangles with very similar sizes. The triangular models of the objects reconstructed relatively at low resolutions are displayed in Fig. 15 for demonstration.

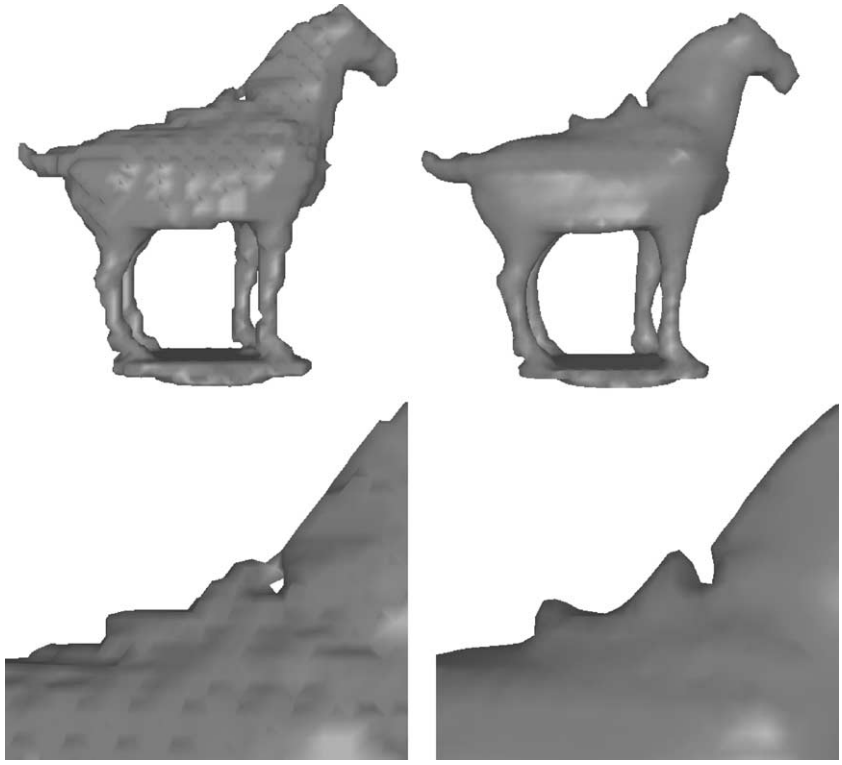

Fig. 13. The horse object and its zoomed versions, reconstructed at resolution $R=6$ (left) by the binary valued isolevel function and (right) by the interpolated isolevel function and dichotomic subdivision.

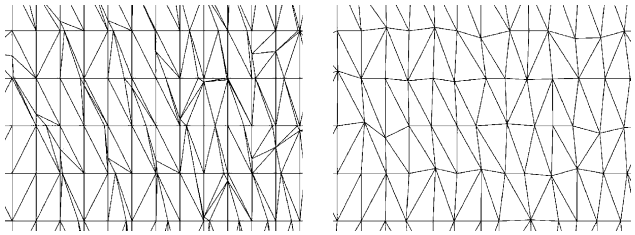

Fig. 14. A triangular mesh before and after the decimation algorithm.

7. Texture mapping

In this section, we propose a strategy to assign the optimal texture to each surface triangle of the constructed model by making use of the available color photographs of the object. We proceed on a triangle by triangle basis subdividing each triangle into particles. For each particle we determine the best color from the full set of images from which this particle is seen, taking into account self-occlusions and photography related problems. In conjunction with the texture assignment strategy, we address also the problem of constructing and storing texture maps which can then be used by standard graphic viewers to display the textured object model.

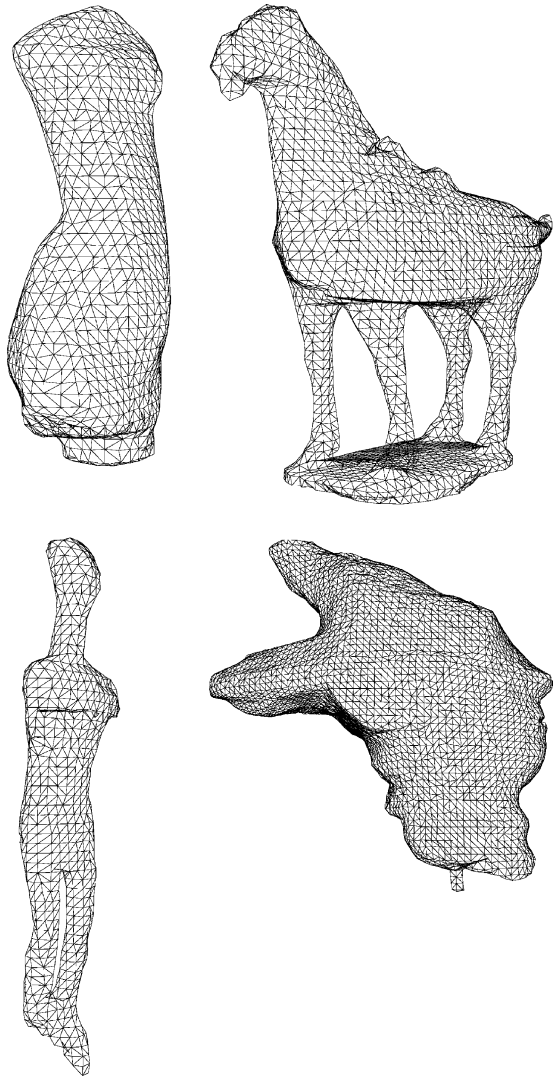

Fig. 15. Triangulated surfaces of the Coignard, Horse, Cycladic and Anyi objects at resolutions $R=6$, 6, 6 and 7, respectively. The number of triangles for each surface is 2932, 6978, 1986 and 10,792.

7.1. Particle definition

Surface particle representation is a well-known concept used in various problems of computer graphics [29,34,36,41]. A particle is usually represented by a position, a normal, and a color as shown in Fig. 16. We use surface particles in order to abstract the texture recovery algorithm from the underlying surface representation. This also allows us to express algorithms in a self-contained parallel manner and to optimize the texturing process within triangles. Particles are considered to be distributed along the whole object surface. The particle density on the object surface is chosen such that their distribution matches the resolution of the original images when projected onto the image planes pointed by their normals. Once each particle has been assigned a normal and thereby a color, we can generate the final texture in a suitable format for the surface and the underlying display system.

7.2. Particle normal computation

The accuracy of the normal vector to be computed for each particle is crucial for the resulting texture quality. The normal vectors should be as smooth as possible so as to reduce texture reconstruction artifacts like sharp texture image frontiers along the surface because of the differences between selected source images. The strategy that we use is as follows: first the geometric normal vector of each triangle is calculated using the 3D position of its vertices, and then a smoothed normal vector is generated for each triangle by averaging the geometric normal vectors of the adjacent patches. Finally, a normal vector is assigned to each individual triangle vertex by taking an area weighted average of the smoothed normal vectors of the triangles neighboring that vertex (Fig. 17).

To assign a normal vector to a particle inside a triangle, we use Phong normal interpolation which is precise enough

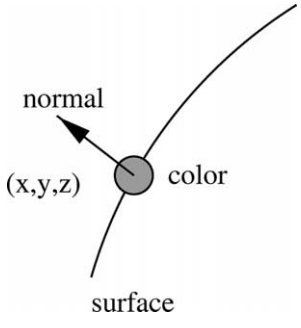

Fig. 16. Particle definition.

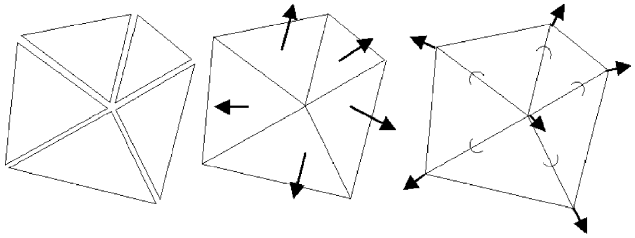

Fig. 17. Computation of normal vectors.

for our purposes since the angles between vertex normal vectors are small enough. This means that a particle normal is determined by linearly interpolating the normals of the triangle vertices.

7.3. Optimal image selection

The color of a surface point can be determined by more than one source image as shown in Fig. 18. In fact this is one of the reasons why we use surface particles. In this way, the texture within a given triangle can be determined from different source images. This flexibility avails us with the possibility of spreading and thereby smoothing texture discontinuities that occur due to the color of a surface point that inevitably differs depending on the projection view, i.e. the relative position of the lighting, the camera and the object. Texture discontinuities would be concentrated at triangle edges if each triangle were textured by a single source image, depending on the normal variation between neighboring triangles.

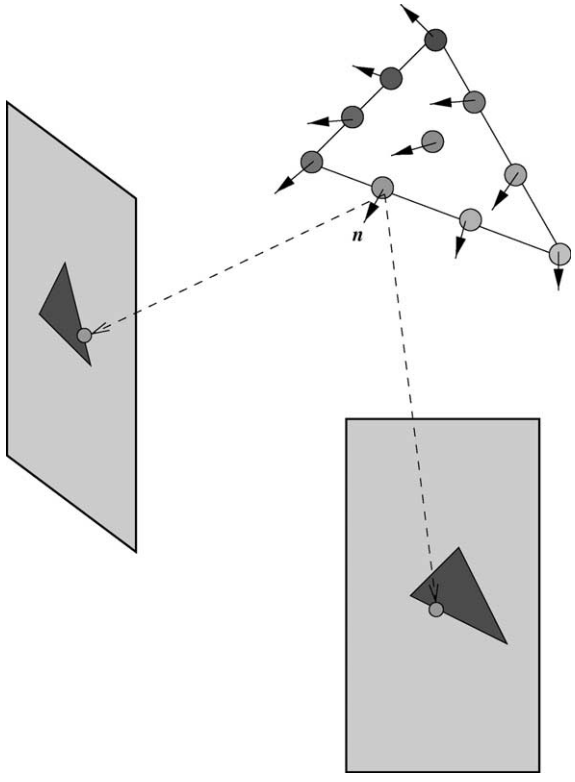

Fig. 18. Particle color assignment using two image sources selected according to the normal direction.

To determine the color of a given particle associated with a single source image, we project its center onto the image and interpolate linearly between the four nearest samples. The primary criterion to select the optimal source image is the available resolution according to the particle normal; in other words, the optimal image has its optical axis as parallel as possible to the particle normal. However, this kind of strict selection tends to enforce sharply defined source image frontiers along the surface of the object. To avoid this we determine for each particle two or more suboptimal images that are nearest to the direction pointed by the particle normal. The color from each of these source images is then weighted with the scalar product between the image normal vector and the particle normal vector indicating the direction of the theoretical optimal image.

In selecting the source images as described above, there are two important issues that should be taken into account. First, the source images selected for a given surface particle may correspond to views from which the particle is not visible due to *occlusion* problems. Such a source image, even if its optical axis is exactly parallel to the normal direction, should be ignored since the color provided by that image would obviously be wrong being the color of another surface point that occludes the underlying particle. To detect whether a particle is occluded or not with respect to a view direction, we use z-buffers constructed for all available views. A z-buffer image keeps at each pixel location the depth value (i.e. the distance from the image plane) of the surface point which is nearest to the corresponding image plane and which falls on that pixel when projected. Comparing the stored depth value with the actual distance of the particle to the image plane we can determine if there is another surface part between the particle and the image plane.

The second problem that can make an image invalid as a texture source is *highlights*. Disregarding the highlight problem for optimal image selection may lead to visually disturbing unrealistic model textures as observed in Fig. 19. Therefore for a given particle, a candidate source image for which the associated color is saturated should also be discarded. To detect the saturated source images for a given particle, we use a local threshold for each color component. The strategy is as follows: first the colors provided by the source images from which the underlying particle is visible are averaged. Then a local saturation threshold which should be a value greater than the resulting average is chosen. If any of the color components provided by a source image happens to be larger than the local threshold corresponding to that color component, the source image is assumed to be saturated and identified as an invalid image source for that particle. With this strategy highlights are properly eliminated as seen in Fig. 19, where we zoom on the reconstructed object models in order to observe better the performance of our highlight elimination strategy.

Fig. 19. 3D color horse model at resolution $R=7$ as a whole (left) and zoomed (right). The surface has been textured by two optimal images, (top) without and (bottom) with highlight elimination. Observe that highlights are spread throughout the surface resulting in a disturbing unrealistic diffuse texture if they are not taken into account during the texturing process.

We should point out that the technique described above is a simple and effective method to avoid visually disturbing highlights with almost no a priori assumptions on the object surface properties. It is not designed to recover the true reflectance properties of the object surface. In the literature, techniques exist to decompose the observed color into its specular and diffuse components by estimating surface reflection properties. These techniques assume a theoretical reflection model such as the dichromatic reflection model. To estimate the diffuse color and separate the specular reflection component, some of these techniques necessitate a large set of views presenting, for each surface point of the object, different geometrical configurations of the illuminant direction, the surface normal, and the viewing direction [31]. In the case of silhouette-based reconstruction from rotating object sequences, the geometrical configurations available at each point surface are rather limited; this reduces the appropriateness of these estimation approaches for such sequences. Some other techniques propose a single-image approach for highlight removal, but require either the object surface to be composed of a single material [16], or a user interaction to manually label each individual highlight region [37]. They thus appear to be inappropriate for our purposes since we aim at an automated 3D construction of objects with a large variety of surface properties, as it is usually the case with art objects. The highlight elimination approach we propose has the advantage of being simple and robust, especially when only a limited number of views are available.

7.4. Textures and triangles

In this section, we have so far explained how a color is associated to a given surface particle. Use of particles simplifies and optimizes the texturing process of the surface model. The last problem to address is how to sample the triangle representation that we have in hand with particles. The triangles having very similar sizes after the decimation process, we systematically sample them by placing a fixed number n of particles on each edge. This leads to $n(n+1)/2$ particles per triangle. To determine n, we project each 3D triangle onto the image where it has the largest area.

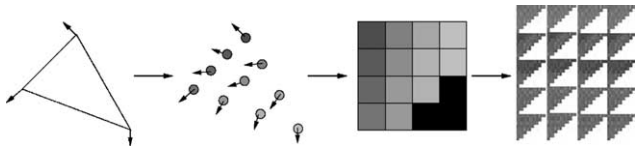

Fig. 20. Triangle, associated particles, texture bitmap, and a part of the complete texture.

We measure the length of its edges in pixels and finally average the edge length for all triangles.

The colors of all particles have to be stored in a texture map compatible with standard computer graphics systems.

Fig. 21. Various views of the Coignard, the Anyi, the Horse and the Cycladic statuettes reconstructed at resolutions $R=7$, 9, 8 and 7, respectively.

We associate to each triangle a matrix of $n \times n$ pixels and we place the colors of the $n(n+1)/2$ particles associated to the triangle into the upper left triangular part of the matrix. This triangular part can be seen as an affine transformation of the 3D triangle texture, by which the longest edge of the arbitrarily shaped surface triangle is mapped to the diagonal of the upper right triangle. The whole texture map is then obtained by just appending the matrices in a single file in the same order as the triangles appear in the data structure.

The texture mapping is performed by the viewer in hardware or in software by sampling the texture. The color of each sampled point is obtained by weighting the color of its four nearest pixels. This necessitates a particular care for the diagonal of the matrices [34]. We just add an external one-pixel-wide border to each diagonal, its colors being determined with the corresponding particles of the adjacent triangle. Triangle texturing process and its storage in a patchwork-like structure are shown in Fig. 20. An alternative is to place two adjacent triangles in a single matrix, their common edge being associated to the diagonal. This has the advantage of halving the size of each texture map, but requires the storage of the correspondences between textures and triangles.

8. Results and conclusions

We have presented a complete 3D reproduction system that reconstructs real objects in an automatic manner. The various tasks, such as camera calibration, object extraction and texture mapping, which eventually effect the final reconstruction quality have been considered one by one in detail throughout the paper. The experimental results concerning the different tasks of the whole system have

Fig. 22. Textured Coignard model at resolution $R=7$ (left) and the corresponding image of the sequence masked with the silhouette. Note the fidelity of the reconstruction both in color and shape. The smooth diffuse texture mapped on the surface is free of lighting artifacts as compared to the original image.

Fig. 23. Textured Anyi model at resolution $R=9$ (top) and the corresponding image of the sequence masked with the silhouette (bottom); (right) we zoom on the object to observe the high resolution reconstruction performance as compared to the original image.

already been presented throughout the text for demonstration purposes. Some views of the reconstructed 3D models of four objects are displayed in Fig. 21: the Coignard, the Anyi, the Horse and the Cycladic statuettes. The achieved quality both in texture and shape is quite satisfactory although the original sequences have photography related problems which are mainly due to the lighting conditions, such as highlights, lighting differences, radiosity and dark shaded regions. As observed in Fig. 22, for example, realistic textures have been associated to the object surface and rendered without loss of quality with respect to the original images.

The combined scheme of octree construction and marching cubes algorithm that we have adapted to the shape from silhouette problem makes it possible to recover the object shape and texture satisfactorily even at very high resolutions. In the case of the Anyi statuette for instance, the original image size is 2008×3040 which allows us, via the combined scheme, a very high resolution reconstruction both in shape (e.g. $R=9$) and in color. In Fig. 23, by zooming on the reconstructed Anyi model, we demonstrate the high resolution performance of our scheme with respect to the original images.

The drawback of the presented scheme which is based on a shape from silhouette technique, is that there may exist concavities in an object that cannot be captured by this method. A possible way to overcome this drawback is to incorporate stereo information that can be deduced from photographs of the object, thanks to accurate camera calibration parameters. The object shape recovered by the presented robust scheme provides us already with a very good initial estimate to search further for the hidden concavities. Such hybrid shape from stereo and silhouette techniques are very promising but much more computation intensive. We are currently continuing our research in this field [11].

Acknowledgements

This work has been supported by the European Esprit project ACOHIR (no. 23276) and by TUBITAK

(Technology and Research Council of Turkey). We would also like to thank J.M. Lavest for his invaluable help as an expert in camera calibration process, and R. Montoya Vozmediano for an early implementation of the software.

References

[1] P. Beardsley, P. Torr, A. Zisserman, 3D model acquisition from extended image sequences, Proceedings of the ECCV'96, Cambridge, UK, 1996, pp. 683–695.
[2] M. Bro-Nielsen, 3D models from occluding contours using geometric primitives, Technical Report of IMSOR, The Technical University of Denmark, 1994.
[3] H. Busch, Automatic modeling of rigid 3D objects using an analysis by synthesis system, SPIE Proceeings of the Visual Communications and Image Process. IV, Philadelphia, 1989, pp. 356–364.
[4] H.H. Chen, T.S. Huang, A survey of construction and manipulation of octrees, Computer Vis., Graph. Image Process. 43 (3) (1988) 409–431.
[5] C.H. Chien, J.K. Aggarwal, Volume/surface octrees for the representation of three-dimensional objects, Computer Vis. Graph. Image Process. 36 (1) (1986) 100–113.
[6] C.H. Chien, J.K. Aggarwal, Identification of 3D objects from multiple silhouettes using quadtrees/octrees, Computer Vis. Graph. Image Process. 36 (2/3) (1986) 256–273.
[7] G. Cross, A. Zisserman, Surface reconstruction from multiple views using apparent contours and surface texture, NATO Advanced Research Workshop on Confluence of Computer Vision and Computer Graphics, Ljubljana, Slovenia, 2000, pp. 25–47.
[8] Cyberware, Monterey, CA 93940. http://www.cyberware.com
[9] P. Eisert, E. Steinbach, B. Girod, Automatic reconstruction of stationary 3-D objects from multiple uncalibrated camera views, IEEE Trans. Circ. Syst. Video Tech. 10 (2) (2000) 261–277.
[10] C. Hernandez Esteban, F. Schmitt, Multi-stereo 3D object reconstruction, International Symposium on 3D Data Processing Visualization and Transmission, Padova, Italy, 2002, pp. 159–166.
[11] C. Hernandez Esteban, F. Schmitt, Silhouette and stereo fusion for 3D object modeling, Fourth International Conference on 3D Digital Imaging and Modeling, Banff, Alberta, Canada, 2003, pp. 46–53.
[12] A. Fitzgibbon, G. Cross, A. Zisserman, Automatic 3D model construction for turn-table sequences, European Workshop SMILE'98, LNCS 1506, 1998, pp. 155–170.
[13] T. Fromherz, M. Bichsel, Multiple depth and normal maps for shape from multiple views and visual cues, ISPRS Intercommission Workshop From Pixels to Sequences, 1995, pp. 186–194.
[14] M. Kampel, S. Tosovic, R. Sablatnig, Octree-based fusion of shape from silhouette and shape from structured light, 3DPVT02: First IEEE International Symposium on 3D Data Processing Visualization and Transmission, Padova, Italy, 2002, pp. 754–757.
[15] R. Koch, M. Pollefeys, L. VanGool, Multi viewpoint stereo from uncalibrated sequences, Proceedings of ECCV-98, Freiburg, Germany, 1998, pp. 55–71.
[16] C.J. Klinker, S.A. Shafer, T. Kanade, The measurement of highlights in color images, Int. J. Computer Vis. 2 (1990) 7–32.
[17] A. Laurentini, The visual hull concept for silhouette based image understanding, IEEE Trans. PAMI 16 (2) (1994) 150–162.
[18] A. Laurentini, How far 3D shapes can be understood from 2D silhouettes, IEEE Trans. PAMI 17 (2) (1995) 188–195.
[19] J.M. Lavest, M. Viala, M. Dhome, Do we really need an accurate calibration pattern to achieve a reliable camera calibration, Proceedings of the Fifth European Conference on Computer Vision, vol. 1, Germany, 1998, pp. 158–174.
[20] C.E. Liedtke, H. Busch, R. Koch, Shape adaptation for modeling of 3D objects in natural scenes, IEEE Conference on Computer Vision and Pattern Recognition, 1991, pp. 704–705.
[21] W.E. Lorensen, H.E. Cline, Marching cubes: a high resolution 3D surface construction algorithm, Computer Graph. 21 (4) (1987) 163–169.
[22] Y. Matsumoto, H. Terasaki, K. Sugimoto, T. Arakawa, A portable three-dimensional digitizer, International Conference on Recent Advances in 3D Imaging and Modeling, Ottowa, 1997, pp. 197–205.
[23] Y. Matsumoto, K. Fujimura, T. Kitamura, Shape-from-silhouette/stereo and its application to 3-D digitizer, DGCI 1999, 1999, pp. 177–190.
[24] P.R.S. Mendonça, K.K. Wong, R. Cipolla, Recovery of circular motion from profiles of surfaces, European Workshop on Vision Algorithms, LNCS 1883, 1999, pp. 149–165.
[25] C. Montani, R. Scateni, R. Scopigno, A. modified, look-up table for implicit disambiguation of marching cubes, The Vis. Computer 10 (1994) 353–355.
[26] A.Y. Mülayim, Y. Yemez, F. Schmitt, V. Atalay, Rotation axis extraction of a turn table viewed by a fixed camera, Vision Modelling and Visualisation VMV'99, Erlangen, Germany, 1999.
[27] W. Niem, J. Wingbermuhle, Automatic reconstruction of 3D objects using a mobile camera, Image Vis. Comput. 17 (2) (1999) 125–134.
[28] M. Pollefeys, R. Koch, M. Vergauwen, L. VanGool, Flexible acquisition of 3D structure from motion, Proceedings of the 10th IMDSP Workshop, Austria, 1998, pp. 195–198.
[29] W.T. Reeves, Particle systems: a technique for modeling a class of fuzzy objects, ACM Trans. Graph. 2 (2) (1983) 91–108.
[30] M. Rioux, F. Blais, A.B. Beraldin, G. Codin, P. Boulanger, M. Greenspan, Beyond range sensing: XYZ-RGB digitizing and modeling, IEEE Int. Conf. Robot. Automat. 1 (2000) 111–115.
[31] Y. Sato, K. Ikeuchi, Reflectance analysis for 3D computer graphics model generation, Graph. Models Image Process.: GMIP 58 (5) (1996) 437–451.
[32] F. Schmitt, Y. Yemez, 3D color object reconstruction from 2D image sequences, IEEE International Conference on Image Process. ICIP'99, Kobe, Japan, October 1999, pp. 102–106.
[33] W.J. Schroeder, J.A. Zarge, W.E. Lorensen, Decimation of triangle meshes, ACM Computer Graph. (SIGGRAPH'92, Proc.) 26 (1992) 65–70.
[34] M. Soucy, G. Godin, M. Rioux, A texture-mapping approach for the compression of colored 3D triangulations, The Vis. Computer 12 (1996) 503–514.
[35] R. Szeliski, Rapid octree construction from image sequences, CVGIP: Image Understand. 58 (1) (1993) 23–32.
[36] R. Szeliski, D. Tonnesen, Surface modeling with oriented particle systems, ACM Comp. Graph. (SIGGRAPH'92 Proc.) 26 (1992) 185–194.
[37] P. Tan, S. Lin, L. Quan, H.-Y. Shum, Highlight removal by illumination-constrained inpainting, Proceedings of the Ninth International Conference on Computer Vision, vol. 1, Nice, France, 2003, pp. 164–170.
[38] M. Tarini, M. Callieri, C. Montani, C. Rocchini, Marching intersections: an efficient approach to shape-from-silhouette, Vision Modelling and Visualisation VMV'2002, Erlangen, Germany, 2002, pp. 283–290.
[39] R.Y. Tsai, A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lens, IEEE J. Robot. Automat. 24 (6) (1981) 381–395.
[40] J. Wilhelms, A.V. Gelder, Octrees for faster isosurface generation, ACM Computer Graph. (SIGGRAPH'92 Proc.) 11 (3) (1992) 201–227.
[41] Y. Yemez, F. Schmitt, Multilevel representation and transmission of real objects with progressive octree particles, IEEE Trans. Visual. Computer Graph. 2003; 551–569.
[42] J.Y. Zheng, Acquiring 3-D models from sequences of contours, IEEE Trans. Pattern Anal. Mach. Intell. 16 (2) (1994) 163–178.