Silhouette and Stereo Fusion for 3D Object Modeling

Carlos Hernández Esteban and Francis Schmitt
Signal and Image Processing Department, CNRS URA 820
Ecole Nationale Supérieure des Télécommunications, France
{carlos.hernandez,francis.schmitt}@enst.fr

Abstract

In this paper we present a new approach to high quality 3D object reconstruction. Starting from a calibrated sequence of color images, the algorithm is able to reconstruct both the 3D geometry and the texture. The core of the method is based on a deformable model, which defines the framework where texture and silhouette information can be fused. This is achieved by defining two external forces based on the images: a texture driven force and a silhouette driven force. The texture force is computed in two steps: a multi-stereo correlation voting approach and a gradient vector flow diffusion. Due to the high resolution of the voting approach, a multi-grid version of the gradient vector flow has been developed. Concerning the silhouette force, a new formulation of the silhouette constraint is derived. It provides a robust way to integrate the silhouettes in the evolution algorithm. As a consequence, we are able to recover the apparent contours of the model at the end of the iteration process. Finally, a texture map is computed from the original images for the reconstructed 3D model.

1. Introduction

As computer graphics and technology become more powerful, attention is being focused on the creation or acquisition of high quality 3D models. As a result, a great effort is being made to exploit the biggest source of 3D models: the real world. Among all the possible techniques of 3D acquisition, there is one which is specially attractive: the image-based modeling. In this kind of approach, the only input data to the algorithm are a set of images, possibly calibrated. Its main advantages are the low cost of the system and the possibility of immediate color. The main disadvantage is the quality of the reconstructions compared to the quality of more active techniques (range scanning or encoded-light techniques). We present in this paper an image-based modeling approach that offers the possibility of high quality reconstructions by mixing two orthogonal image data into a same framework: silhouette information and texture information. Our two main contributions are a new approach to the silhouette constraint definition and the high quality of the overall system (see Fig.1).

2. Related Work

Acquiring 3D models is not an easy task and abundant literature exists on this subject. There are three main approaches to the problem of 3D acquisition: pure image-based rendering techniques, hybrid image-based techniques, and 3D scanning techniques. Pure image-based rendering techniques as [2, 20] try to generate synthetic views from a given set of original images. They do not estimate the real 3D structure behind the images, they only *interpolate* the given set of images to generate a synthetic view. Hybrid methods as [5, 19] make a rough estimation of the 3D geometry and mix it with a traditional image-based rendering algorithm in order to obtain more accurate results. In both types of methods, the goal is to generate coherent views of the real scene, not to obtain metric measures of it. In opposition to these techniques, the third class of algorithms try to recover the full 3D structure. Among the 3D scanning techniques, we can distinguish two main groups: active methods and passive methods. Active methods use a controlled source of light such as a laser or a coded light in order to recover the 3D information [25, 4, 14]. Passive methods use only the information contained in the images of the scene and are commonly known as *shape from X methods*. They can be classified according to the type of information they use. A first class consists of the shape from silhouette methods [1, 23, 28, 22, 18]. They obtain an initial estimation of the 3D model known as visual hull. They are robust and fast, but because of the type of information used, they are limited to simple shaped objects. We can find commercial products based on this technique. A second class corresponds to the shape from shading methods. They are based on the diffusing properties of Lambertian surfaces. They mainly work for 2.5D surfaces and are very dependent on the light conditions. A third class of methods uses color consistency to carve a voxel volume [26, 17]. But they only provide an output model composed of a set of voxels which makes difficult to obtain a good 3D mesh representation. A fourth class of methods are the shape from stereo approaches. They recover the shape by maximizing a stereo-based energy [10, 24]. As a specialization of this group, there are particular methods that try to use at the same time another type of information such as silhouettes or albedo. Although very good results are obtained, the quality is still limited, and the main problem is the way the fusion of different data is done. Some, such as [17, 3] use a volume grid for the fusion. Others like [15] do the fusion in the image domain. Finally, a deformation model framework can be used as in [8]. The algorithm we present in this paper can be classified in this last group. We perform the fusion of both silhouettes and texture information by a deformation model evolution. The main difference with the methods mentioned above is the way the fusion is accomplished, which allows us to obtain very high quality reconstructions.

3. Algorithm Overview

The goal of the system is to be able to reconstruct a 3D object only from a sequence of geometrically calibrated images. To do so, we dispose of several types of information contained in the images. Among all the information available, shading, silhouettes and features of the object are the most useful for shape retrieval. Shading information needs a calibration of the light sources, which implies a strongly controlled environment for the acquisition. The use of the silhouettes requires simply a good extraction of the object from the background, which, however, is not always very easy to accomplish. Finally, of all the features available from an object, such as texture, points, contours, or more complicated forms, we are mainly interested in texture. Therefore, the information we try to use consists of silhouettes and texture. The next step is to decide how to mix these two types of information to work together. As we will see, this is not an easy task, because those types of information are very different, almost orthogonal.

The framework that we have chosen for the fusion of both data is a classical snake [9]. This deformable model allows us to define an optimal surface which minimizes a global energy $\mathscr{E}$. In our case, the minimization problem is posed as follows: find the surface S of R^3 that minimizes the energy $\mathscr{E}(S)$ defined as follows:

$$\mathscr{E}(S) = \mathscr{E}_{tex}(S) + \mathscr{E}_{sil}(S) + \mathscr{E}_{int}(S), \tag{1}$$

where $\mathscr{E}_{tex}$ is the energy term related to the texture of the object, $\mathscr{E}_{sil}$ the term related to the silhouettes and $\mathscr{E}_{int}$ is a regularization term of the surface model. Minimizing Eq.1 means finding S_{opt} such that:

$$\begin{aligned}\nabla\mathscr{E}(S_{opt}) &= \nabla\mathscr{E}_{tex}(S_{opt}) + \nabla\mathscr{E}_{sil}(S_{opt}) + \nabla\mathscr{E}_{int}(S_{opt}) = 0,\\ &= \mathscr{F}_{tex}(S_{opt}) + \mathscr{F}_{sil}(S_{opt}) + \mathscr{F}_{int}(S_{opt}) = 0,\end{aligned} \tag{2}$$

where the gradient vectors $\mathscr{F}_{tex}$, $\mathscr{F}_{sil}$ and $\mathscr{F}_{int}$ represent the forces which drive the snake.

Equation 2 establishes the equilibrium condition for the optimal solution, where the three forces cancel each other out. A solution to Eq.2 can be found by introducing a time variable t for the surface S and solving the following differential equation:

$$S_t = \mathscr{F}_{tex}(S) + \mathscr{F}_{sil}(S) + \mathscr{F}_{int}(S). \tag{3}$$

The discrete version becomes:

$$S^{k+1} = S^k + \Delta t(\mathscr{F}_{tex}(S^k) + \mathscr{F}_{sil}(S^k) + \mathscr{F}_{int}(S^k)). \tag{4}$$

Once we have defined the energies that will drive the process, we need to make a choice for the representation of the surface S. This representation defines the way the deformation of the snake is done at each iteration. We have chosen the triangular mesh representation, because of its simplicity and well known properties.

To completely define the deformation framework, we need an initial value of S, i.e., an initial surface S_0 that will evolve under the different energies until convergence.

In this paper, we describe the snake initialization in Section 4, the force driven by the texture of the object in Section 5, the force driven by the silhouettes in Section 6, how we control the mesh evolution in Section 7. We finally discuss our results in Section 8.

4. Snake Initialization

The first step in our minimization problem is to find an initial surface *close enough* to the optimal surface in order to guarantee a good convergence of the algorithm. *Close* has to be considered in a geometrical and topological sense. The geometric distance between the initial and optimal surfaces has to be reduced in order to limit the number of iterations in the surface mesh evolution process and thereby the computation time. The topology of the initial surface is also very important since classical deformable models maintain the topology of the mesh during its evolution. On the one hand, this imposes a strong constraint that makes the initialization a very important step since the initial surface must capture the topology of the object surface. Level-set based algorithms [27] have the intrinsic capability to overcome this problem but in practice the topology remains very difficult to control. On the other hand, the topology-constant property of a classical snake provides more robustness to the evolution process.

If we make a list of possible initializations, we can establish an ordered list, where the first and simplest initialization is the bounding box of the object. The next simplest surface is the convex hull of the object. Both the bounding box and the convex hull are unable to represent surfaces with a genus greater than 0. A more refined initialization, which lies between the convex hull and the real object surface is the visual hull [12]. The visual hull can be defined as the intersection of all the possible cones containing the object. In practice, a discrete version is usually obtained by intersecting the cones generated by back projecting the object silhouettes of a given set of views. As a difference with the convex hull, it can represent surfaces with an arbitrary number of holes. However, the topology of the visual hull depends on the discretization of the views, and some real objects may have holes that cannot be seen as a silhouette hole from any point of view. In such case, the visual hull will fail to represent the correct topology of the object.

Computing the visual hull from a sequence of images is a very well known problem of computer vision and computer graphics [23, 22, 19]. We are interested in producing good quality meshes (eulerian, smooth, high aspect ratio) with the right topology. Volume carving methods are a good choice because of the high quality output meshes that we can obtain through a marching cube [16] or marching tetrahedron algorithm. The degree of precision is fixed by the resolution of the volume grid, which can be adapted according to the required output resolution. However this adaptability can also generate additional problems of topology: if the resolution of the grid is low compared to the size of the visual hull structures, the aliasing produced by the sub-sampling may produce topological artifacts that the theoretic visual hull does not have. But they can be avoided by increasing the precision of the algorithm or by filtering the silhouettes.

5. Texture Driven Force

In this section we define the texture force $\mathscr{F}_{tex}$ appearing in Eq.2 which contributes to recover the 3D object geometry during the snake evolution process. We want this force to maximize the texture coherence of all the cameras that see the same part of the object. It is based on the following projective geometry property: if two cameras see the same textured surface, then the two images are related by a geometric relation that depends only on the 3D geometry of the object. This property is only valid under the common hypothesis of perfect projective cameras, perfect Lambertian surface and the same lighting conditions. This property can be used in two different types of applications: i) for image based rendering by simulating a virtual view from a set of known views and an estimation of the 3D geometry of the object, ii) for 3D recovery from a set of known views, where we search the surface that maximizes the texture coherence of the images. Our problem corresponds to the second type of applications. We define first in section 5.1 a texture coherence measure. In section 5.2 we describe the voting approach which we propose to recover 3D geometry. In section 5.3 we explain how the texture driven force is derived from the data collected by the voting approach using a multi-grid gradient vector flow.

5.1. Texture Coherence Criterion

If we know the 3D geometry, there exist various methods to measure the texture coherence. The simplest measure would be the color difference between the two pixels. But it is too local a measure since it only uses one sample per texture. It is then very sensitive to noise. More robust measures use several samples per texture by considering a neighborhood around the pixels. A usual criterion is the normalized cross-correlation between the two neighborhoods. This measure compares the intensity distributions inside the two neighborhoods. It is invariant to changes of the mean intensity value and of the dynamic range inside the neighborhoods. This criterion is more robust to failures in the Lambertian hypothesis for the surfaces.

If we do not know the 3D geometry, we can try to recover it by maximizing the texture criterion for a given set of views. Two different types of approaches for this optimization have been proposed in the literature. In the first type the texture similarity is used to evaluate a current model. If the measure is improved by deforming the model locally, then the model is updated and the process iterated as in [8, 7] or as in level-set based methods [10, 24] where a volumic band is explored around the current model. In this first type of approaches the exploration remains locally dependent on the current model. Since the exploration does not test all the possible configurations, the algorithm can fail because of local maxima of the texture coherence criterion. The second type of approaches consists of testing all the possible configurations. This allows making a more robust decision. In order to improve even more the robustness, we can cumulate the criterion values into a 3D grid by using a voting approach as in [17, 21]. We use this kind of approach since it is very robust in the presence of highlights (e.g. see details in Fig.1 left) and it allows us to pass from the image information to a more usable information of the sort “probability of finding a surface”.

5.2. Proposed Voting Approach

The proposed voting approach is based on a multi-stereo correlation criterion such as the one described in [6]. The multi-stereo computation is done for all the pixels of all the images and the resulting 3D estimations are cumulated into a voxel grid. The problem with this sort of algorithm is the computation time. For large images as those we are using

(2000x3000), it can reach 16 hours on a fast machine. However this computation time can be strongly reduced with almost no loss by using the computation redundancy. We distinguish the redundancy between different images and the redundancy inside an image.

The redundancy between images happens when several images see a same part of the object surface. If we have already computed a surface estimation using one image, we can back project the 3D points into the other images, giving an initial estimation of the distance to the surface. The problem is that if the previous image did not correlate well, errors may propagate and prevent the following images from attenuating it.

The redundancy inside an image is due to the content of the image. In our case, it is a picture of an object, and we can expect it to be locally continuous. This implies that, if the surface is correctly seen and if there is no occlusion, the depth values for neighboring pixels should not be very different. This can be exploited in order to reduce the depth interval for the correlation criterion. In the greedy algorithm, for each pixel, we test the entire depth interval defined by the visual hull without taking into account if its neighbors have already found a coherent surface. To be able to benefit from already computed correlations, the image can be partitioned into different resolution layers. Then the greedy algorithm is run on the lowest resolution pixels, with the depth interval defined by the visual hull. For consecutive layers, the depth interval is computed using the results of the precedent layer. To estimate the depth interval of a pixel based on the results of the previous layer, a record of the correlation values is maintained in order to control the reliability of the estimation. Besides the improvement in computation time, an improvement in storage space is obtained by substituting the 3D volume grid by a more compact octree structure. The proposed algorithm can be coded as:

```
For each image in imageList
  For each layer in image
    For each pixel in layer
      If layer = first layer
        Compute the depth interval from the visual hull
      Else
        Compute the depth interval from the previous layer
      Compute the correlation curves
      Transform all the curves into the same coordinate system
      Find the best candidate depth
      If correlation value < threshold, continue with next pixel
      Compute the 3D position P of the candidate depth
      Add the correlation value to the octree
```

The theoretical maximum improvement that we can reach with this method in the case of 3 layers is 16 times faster than the greedy method. The worst case would correspond to a non textured image where correlations become unreliable and the depth interval estimation fails, which leads to the same computation time than the greedy method. In practice, the improvement is about 5 or 6 times faster for well-textured images.

The result of the correlation step is a 3D octree containing the cumulated hits of all the pixel estimations. This volume by itself cannot be used as a force to drive the snake. A possible force could be the gradient of the correlation volume. The problem is that this is a very local force defined only in the vicinity of the object surface. The proposed solution to this problem is to use a gradient vector flow (GVF) field to drive the snake.

5.3. Octree-based Gradient Vector Flow

The GVF field was introduced by [29] as a way to overcome a difficult problem encountered with traditional external forces: their capture range. This problem is caused by the local definition of these forces, and the absence of an information propagation mechanism. To eliminate this drawback, and for all the forces derived from the gradient of a scalar field, they proposed to generate a vector field force that propagates the gradient information. The GVF of a scalar field f is defined as the vector field $\mathbf{v}$ that minimizes the following energy functional $\mathscr{E}$:

$$\mathscr{E} = \int \mu ||\nabla \mathbf{v}||^2 + ||\mathbf{v} - \nabla f||^2 ||\nabla f||^2,$$

where μ is the weight of the regularization term.

The GVF can be seen as the original gradient smoothed by the action of a Laplacian operator. This smoothing action allows us at the same time to eliminate strong variations of the gradient vector field and to produce a propagation of the gradient. The degree of smoothing/propagation is controlled by μ. If μ is zero, the GVF is the original gradient, if μ is very large, the GVF is a constant field whose components are the mean of the gradient components.

Since our data have been stored in an octree structure, a multi-resolution version of the GVF algorithm has been developed.

6. Silhouette Driven Force

The silhouette force is defined as a force that makes the snake match the original silhouettes of the sequence. If it is the only force of the snake, the model should converge towards the visual hull. Since we are only interested in respecting silhouettes, the force will depend on the self occlusion of the snake. If there is a part of the snake that already matches a particular silhouette, the rest of the snake is no more concerned by that silhouette, since the silhouette is already matched. If we compare a visual hull and the real object, we see that the entire real object matches the silhouettes, but not all the points of the object. The object concavities do not obey any silhouette because they are

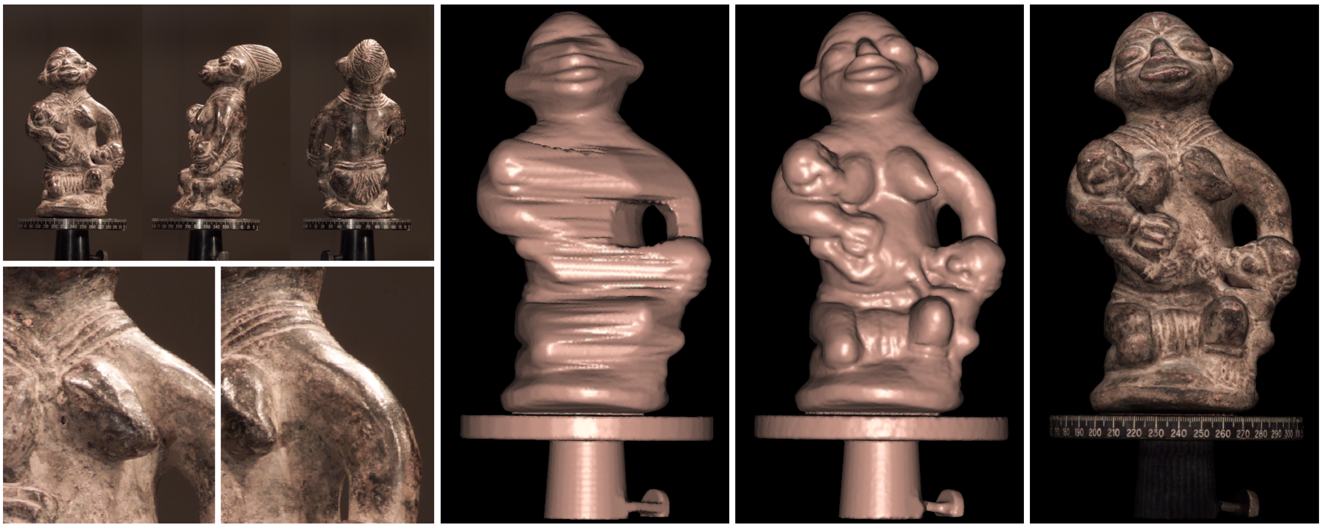

Figure 1. Different steps in the reconstruction process of the Twins model. From left to right: image acquisition (top: some sequence images, bottom: details), visual hull initialization, shaded final model and textured final model. The reconstructed model has 83241 vertices.

occluded by a part of the object that already matches the silhouettes. The main problem is how to distinguish between points that have to obey the silhouettes and those that have not. This is equivalent to finding the apparent contours of the object. The silhouette force can be in fact decomposed into two different components: a component that measures the silhouette fitting, and a component that measures how strongly the silhouette force should be applied. The first component is defined as a distance to the visual hull. For a 3D vertex $\mathbf{P}_M$ on the snake mesh this component can be implemented by computing the smallest signed distance d_{VH} between the contours of a silhouette S_i and the projection $\mathscr{P}_i\mathbf{P}_M$ of the point into the image corresponding to S_i:

$$d_{VH}(\mathbf{P}_M) = \min_i d(S_i, \mathscr{P}_i\mathbf{P}_M).$$

A positive distance means that the projection is inside the silhouette, and a negative distance that the projection is outside the silhouette. Using only this force would make the snake converge towards the visual hull. The second component measures the occlusion degree of a point of the snake for a given view point. The view point is chosen as the camera that defines the distance to the visual hull:

$$\begin{aligned} \alpha(\mathbf{P}_M) &= \begin{cases} 1 & \text{for } d_{VH}(\mathbf{P}_M) \leq 0 \\ \frac{1}{(1+d(S_c snake, \mathscr{P}_c\mathbf{P}_M))^n} & \text{for } d_{VH}(\mathbf{P}_M) > 0 \end{cases}, \\ c(\mathbf{P}_M) &= arg\min_i d(S_i, \mathscr{P}_i\mathbf{P}_M). \end{aligned}$$

In the definition of $\alpha(\mathbf{P}_M)$, there are two cases. If d_{VH} is negative, it means that the point is outside the visual hull. In that case, the force is always the maximum force. For a point inside the visual hull, c is the camera that actually defines its distance to the visual hull. $S_c snake$ is the silhouette created by the projection of the snake into the camera c. The power n controls the decreasing ratio of α. This function gives the maximum silhouette force to the points that compose the apparent contours. The rest of the points, which are considered as concavities, are weighted inversely to the distance to the silhouette. This allows the points of the snake to *detach* themselves from the visual hull. A big value of n allows an easier detachment, but the force becomes too local and does not allow smooth transitions between concavities and contours. The value used in practice is $n = 2$, which is a compromise between smoothness and concavity recovery.

The final silhouette force for a given point of the snake is a vector directed along the normal to the snake surface $\mathbf{N}_M$ and its magnitude is the product of both components:

$$\mathscr{F}_{sil}(\mathbf{P}_M) = \alpha(\mathbf{P}_M) d_{VH}(\mathbf{P}_M) \mathbf{N}_M(\mathbf{P}_M)$$

7. Mesh Control

Having defined the texture and silhouette forces $\mathscr{F}_{tex}$ and $\mathscr{F}_{sil}$, i.e. the external forces, the last force to detail is the internal force $\mathscr{F}_{int}$. The goal of the internal force is to regularize the effect of the external forces. We define the internal regularization as a force that tries to move a given point $\mathbf{v}$ of the mesh to the center of gravity of its 1-ring neighborhood:

$$\mathscr{F}_{int}(\mathbf{v}) = \left(\frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i - \mathbf{v} \right),$$

where $\mathbf{v}_i$ is the i^{th} neighbor of $\mathbf{v}$. If the only force in the snake is the internal force, the mesh will collapse under the action of the barycentric filtering.

Since the texture forces $\mathscr{F}_{tex}$ can sometimes be parallel to the surface of the snake, in the snake evolution we use as texture force its projection $\mathscr{F}_{tex}^N$ over the normal of the surface:

$$\mathscr{F}_{tex}^N(\mathbf{v}) = (\mathscr{F}_{tex}(\mathbf{v}) \cdot \mathbf{N}(\mathbf{v}))\mathbf{N}(\mathbf{v}).$$

This avoids problems of coherence in the texture force of neighbor points and helps the internal force to keep a well-shaped surface. The snake evolution process (Eq.4) at the k^{th} iteration can then be written as the evolution of all the points of the mesh $\mathbf{v}_i$:

$$\mathbf{v}_i^{k+1} = \mathbf{v}_i^k + \Delta t(\mathscr{F}_{tex}^N(\mathbf{v}_i^k) + \beta \mathscr{F}_{sil}(\mathbf{v}_i^k) + \gamma \mathscr{F}_{int}(\mathbf{v}_i^k)), \quad (5)$$

where Δt is the time step and β and γ are the weights of the silhouette force and the regularization term, relative to the texture force. Equation 5 is iterated until steady-state of all the points of the mesh is achieved. The time step Δt has to be chosen as a compromise between the stability of the process and the convergence time. An additional step of remeshing is done at the end of each iteration in order to maintain a minimum and a maximum distance between neighbor points of the mesh. This is achieved by a controlled decimation of the mesh based on the edge collapse operator and a controlled refinement based on the $\sqrt{3}$-subdivision algorithm [11].

8. Results

In this section we present several results obtained with the proposed approach. We use also a texture mapping method similar to the one in [13]. But we have further improved the quality of the texture by filtering the highlights. This is possible thanks to the availability of several images seeing a given triangle.

All the reconstructions presented in this paper where obtained from a single axis rotation sequence of 36 images, each image having 2008x3040 pixels. The values of β and γ are the same for all the reconstructions: $\beta = 0.2$, $\gamma = 0.15$. Because the snake iteration is always done in the voxel coordinate system of the GVF octree, the value of β only depends on the ratio between the images size and the octree size. Typical values of γ are between 0.1 and 0.25, depending on the required smoothness.

Computation times are dominated by the correlation voting step: a typical computation time for 36 images of 6 Mpixels is about 3 hours on a P4 1.4GHz machine.

Figure 2 illustrates the influence of the silhouette force. The support of the object does not provide any texture information and cannot be reconstructed using the only texture force (Fig.2 left). Adding silhouette constraints solves this problem and guarantees that the deformable model respects the visual hull in this region (Fig.2 right).

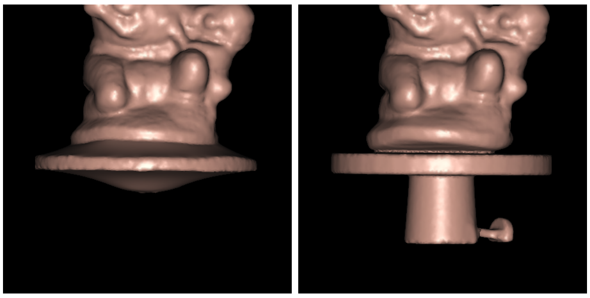

Figure 2. Twins model detail after convergence. Left: evolution under the texture force only. Right: evolution under the texture force and the silhouette force.

In Fig.3 we illustrate the different forces used in the deformable model. Ten octree levels are used in the voting approach (top and middle left), which provides a high precision in the gradient vector flow computation (top and middle right). At the end of the iterative process, a steady-state for the entire mesh is achieved, and concavities are automatically detected (bottom).

In Fig.4 a complete reconstruction is presented using both silhouette and stereo. We are able to recover many details with high accuracy as can be seen in Fig.4 (bottom): observe for instance the quality of reconstruction of the bracelet and of the rope.

Another example is shown in Fig.5. We can appreciate the quality of the tunic folds reconstruction. It shows that the method is powerful even for the reconstruction of small concavities. They are correctly detected and detached from the visual hull.

In Fig.6 we present how the lack of good image correlations is handled by the proposed approach. In Fig.6 top left we can clearly see parts with a dark brilliant non-textured material which produces bad correlation results (Fig.6 top right). The final model will converge to the textured surface whenever it exists and to the visual hull everywhere else (Fig.6 bottom), producing some surface reconstruction errors in this area.

In Fig.7 we present a comparison between a laser acquisition method (left) and the proposed method (right).We have less resolution in our reconstructed model due to the stereo limitations and the regularization term. However, the mesh quality is quite good and the main object concavities are well recovered too.

9. Conclusion and future work

We have presented a new approach to 3D object reconstruction based on the fusion of texture and silhouette information. Our two main contributions are the definition and the fusion of the silhouette force into the snake framework,

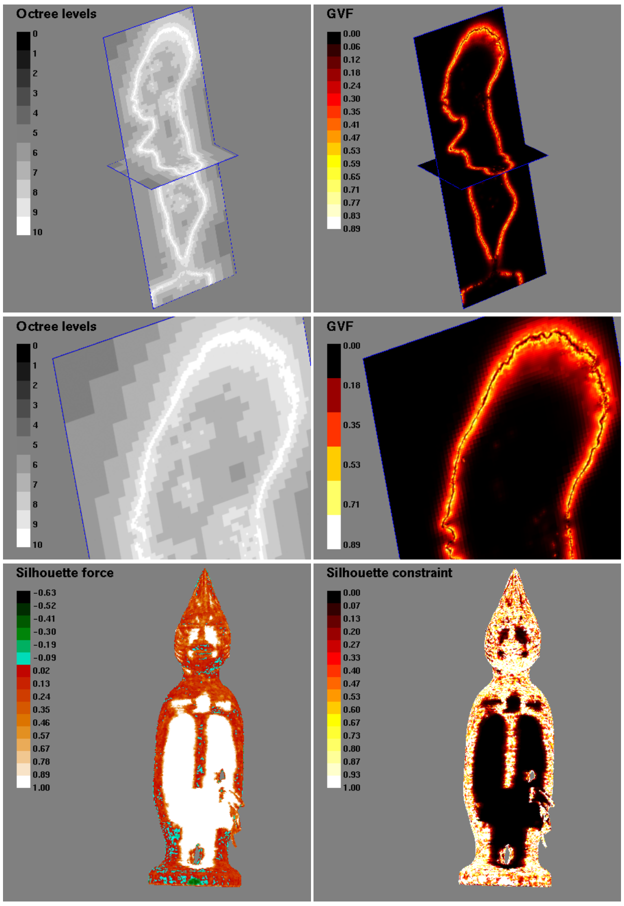

Figure 3. External forces used in the reconstruction of the African model. Top left: the octree partition used in the computation of the gradient vector flow field. Top right: norm of the gradient vector flow field. Middle: details. Bottom left: d_{VH} silhouette component after convergence. Bottom right: α component of the silhouette force after convergence.

and the full system approach where different known techniques are used and improved in order to obtain high quality results. The two main limitations of the algorithm are also its two main sources of robustness: the volume voting approach and the topology constant snake approach. The voting approach allows good reconstructions in the presence of highlights, but the 3D grid used also limits the maximum resolution of the 3D model. A way to overcome this limitation could be to introduce the final model into another snake evolution where the texture energy computation would take into account the current surface (tangent plane or quadric based cross-correlation). Since the initial model is already very close to the real surface, only some iterations would suffice to converge. The second drawback is the topology constant evolution. It allows a guaranted topology of the final model but it is also a limitation for some kind of objects where the topology cannot be captured by the visual hull concept. A possible solution would be to detect self collisions of the snake, and to launch a local level-set based method in order to recover the correct topology. Further work includes: i) the self calibration of the image sequence using both the silhouettes and traditional methods, ii) an improved strategy for the converge of the snake in order to accelerate the evolution in the empty concavity regions, iii) the possible use of the surface curvatures to allow a multi-resolution evolution of the mesh, iv) a more advanced work in the generation and visualization of the texture mapping.

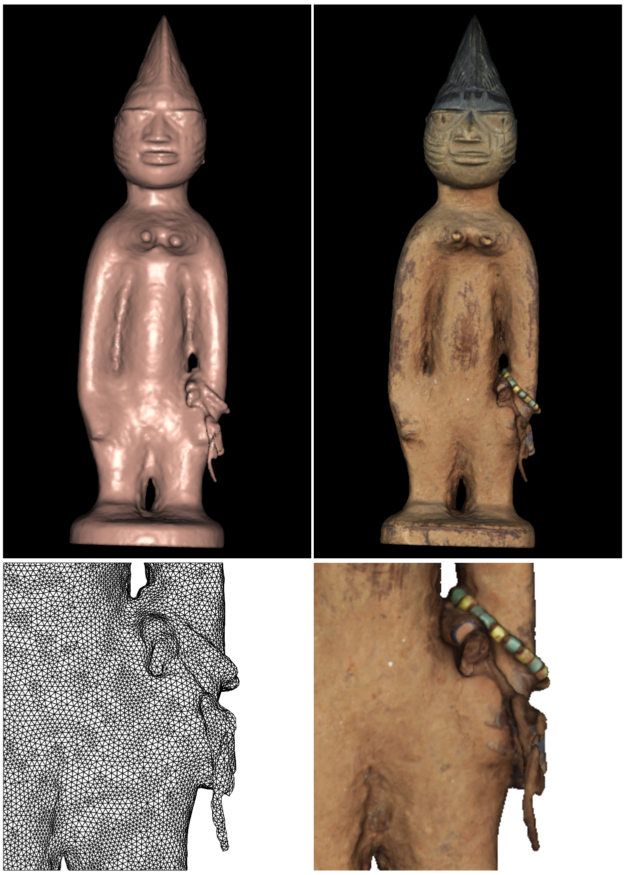

Figure 4. African model after convergence (57639 vertices). Top left: Gouraud shading. Top right: same view with texture mapping. Bottom: Detail of the model after convergence.

Acknowledgements: This work has been partially supported by the SCULPTEUR European project IST-2001-35372. We thank the Thomas Henry museum at Cherbourg for the image sequences corresponding to Fig.6 and 7.

URL: www.tsi.enst.fr/3dmodels/

References

[1] B. G. Baumgart. *Geometric Modelling for Computer Vision*. PhD thesis, Standford University, 1974.

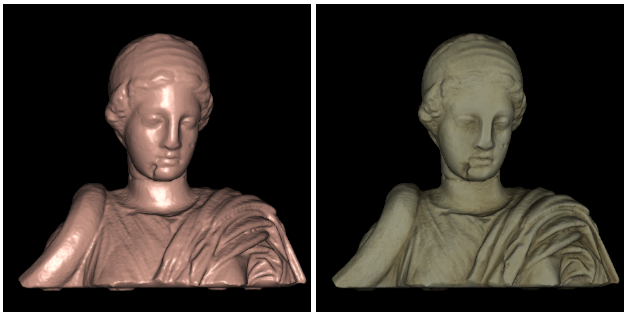

Figure 5. Hygia model after convergence (159534 vertices). Left: Gouraud shading. Right: texture mapping.

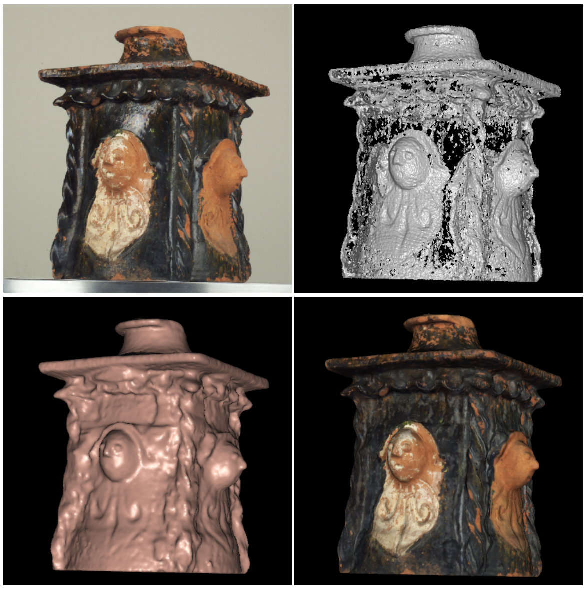

Figure 6. Example of a reconstruction with bad image correlations. Top left: one of the original images. Top right: rendering of the correlation voting volume. Bottom left: snake mesh after convergence. Bottom right: textured mesh.

[2] S.E. Chen and L. Williams. View interpolation for image synthesis. In *SIGGRAPH'93*, pages 279–288, 1993.

[3] G. Cross and A. Zisserman. Surface reconstruction from multiple views using apparent contours and surface texture. In A. Leonardis, F. Solina, and R. Bajcsy, editors, *NATO Advanced Research Workshop on Confluence of Computer Vision and Computer Graphics, Ljubljana, Slovenia*, pages 25–47, 2000.

[4] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH'96*, pages 303–312, 1996.

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

[6] C. Hernández Esteban and F. Schmitt. Multi-stereo 3d object reconstruction. In *3DPVT'02*, pages 159–166, 2002.

[7] P. Fua. From multiple stereo views to multiple 3d surfaces. *Int. J. of Computer Vision*, 24:19–35, 1997.

[8] P. Fua and Y.G. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *Int. J. of Computer Vision*, 16:35–56, 1995.

[9] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. of Computer Vision*, 1:321–332, 1988.

[10] R. Keriven and O. Faugeras. Variational principles, surface evolution, pdes, level set methods, and the stereo problem. *IEEE Transactions on Image Processing*, 7(3):336–344, 1998.

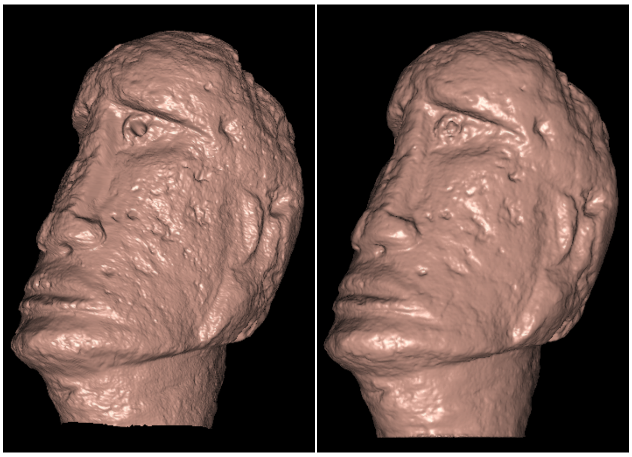

Figure 7. Comparison between the proposed passive method and a laser active method. Left: laser model of 385355 vertices obtained with a Minolta VIVID 910 3D scanner. Right: proposed method after snake convergence (233262 vertices).

[11] L. Kobbelt. $\sqrt{3}$-subdivision. In *SIGGRAPH 2000*, pages 103–112, 2000.

[12] A. Laurentini. The visual hull concept for silhouette based image understanding. *IEEE Trans. on PAMI*, 16(2):150–162, 1994.

[13] H. Lensch, W. Heidrich, and H. P. Seidel. A silhouette-based algorithm for texture registration and stitching. *Journal of Graphical Models*, pages 245–262, 2001.

[14] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project: 3d scanning of large statues. In *SIGGRAPH 2000*, pages 131–144, 2000.

[15] M. Li, H. Schirmacher, M. Magnor, and H.P. Seidel. Combining stereo and visual hull information for on-line reconstruction and rendering of dynamic scenes. In *IEEE Workshop on MMSP*, pages 9–12, 2002.

[16] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH'87*, volume 21, pages 163–169, 1987.

[17] Y. Matsumoto, K. Fujimura, and T. Kitamura. Shape-from-silhouette/stereo and its application to 3-d digitizer. In *Proceedings of Discrete Geometry for Computing Imagery*, pages 177–190, 1999.

[18] Y. Matsumoto, H. Terasaki, K. Sugimoto, and T. Arakawa. A portable three-dimensional digitizer. In *Int. Conf. on Recent Advances in 3D Imaging and Modeling*, pages 197–205, 1997.

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

[20] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. In *SIGGRAPH'95*, pages 39–46, 1995.

[21] G. Medioni, M.S. Lee, and C.-K. Tang. *A Computational Framework for Segmentation and Grouping*. Elsevier, 2000.

[22] W. Niem and J. Wingbermuhle. Automatic reconstruction of 3d objects using a mobile monoscopic camera. In *Int. Conf. on Recent Advances in 3D Imaging and Modeling*, pages 173–181, 1997.

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

[24] A. Sarti and S. Tubaro. Image based multiresolution implicit object modeling. *EURASIP Journal on Applied Signal Processing*, 2002(10):1053–1066, 2002.

[25] F. Schmitt, B. Barsky, and W. Du. An adaptative subdivision method for surface-fitting from sampled data. In *SIGGRAPH'86*, pages 179–188, 1986.

[26] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. *Int. J. of Computer Vision*, 38(3):197–216, 2000.

[27] J.A. Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences*. Cambridge University Press, 1996.

[28] R. Vaillant and O. Faugeras. Using extremal boundaries for 3d object modelling. *IEEE Trans. on PAMI*, 14(2):157–173, 1992.

[29] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, pages 359–369, 1998.