Fast Human 3D Voxelized Shape Reconstruction for Human-Computer Interaction

Yaiza Vélaz, Luis Unzueta, and Ángel Suescun

***Abstract*—**At present, there is an increasing interest in multimodal interaction, which supports multiple modes of interaction of the user with the computer. Among these modes of interfacing, those obtained from face expressions, voice, body postures and movements are the most important and researched within a human face-to-face interaction context. Some of the tools used to achieve this communication are: motion capture, speech recognition and force feedback. Thus, in order to carry out the human-computer interaction (HCI), some features and parameters are needed, which can be extracted via cameras or haptics depending on the communication mode.

The 3D shape of the target person constitutes one of the most used features in markerless motion capture for a posterior pose and motion pattern recognition, which can be obtained from a multi-camera system. The process starts by subtracting the background from the images in order to get the projected silhouettes of the subject. Then, having established the relation between the real 3D world and the camera projections, the 3D shape reconstruction can be attained with voxel carving methods. The reconstruction quality depends on the number of cameras, the size of the voxels and the used method. The more precise the 3D shape reconstruction is, the higher the computational cost will be, which may prevent its use for HCI applications.

We present a novel approach that accelerates existing voxel carving methods. It goes from coarse to fine, preserving the matching with the captured silhouettes, and thus achieving a good reconstruction quality. Results show its suitability for real-time markerless motion capture, applicable to HCI applications, such as videogames.

***Keywords*—**Voxel Carving, Human-Computer Interaction, Markerless Motion Capture, Computer Vision.

I. Introduction

In the last decades many efforts have been invested in representing a real moving human inside the 3D virtual world, i.e., human motion capture, due to the variety of applications this computer vision field presents, including virtual reality, surveillance, or motor skills capture and transfer, among others.

Manuscript received May 28, 2008: Revised version received October 14, 2008.

Y. Vélaz, L. Unzueta and A. Suescun are in the Applied Mechanics Department in the CEIT and Tecnun of the University of Navarra, Paseo de Manuel Lardizabal, 15, 20018, Donostia-San Sebastián (Spain).

E-mail: yvelaz@ceit.es, lunzueta@ceit.es, asuescun@ceit.es).

In 1994, Laurentini [1] defined the foundations of the 3D space reconstruction, although before those days some related papers had already been published to reconstruct 3D objects from silhouettes [2-6]. Laurentini defined and demonstrated the *visual hull* of an object S as the closest approximation that gives the same silhouette of S obtained from all views outside the convex hull of the object. However, as stated by Slabaugh et al. [7], volumetric representations use a finite number of viewpoints computing the inferred visual hull. Many of the methods which implement the 3D reconstructions from silhouettes are based on Shape-From-Silhouette (SFS), or *voxel carving*. A voxel representation has the advantage that it is simple to implement, and additionally it is a look-up table, i.e., the voxel structure can be accessed with a simple array indexing operation.

The resulting 3D space representation can be used as input for *voxel coloring* or *space carving* [8-14], which are methods to colorize the 3D reconstruction in order to get a more realistic representation.

The coloring process has the handicap of needing to determine which parts of the reconstruction are visible to the observers, in order to assign the proper color to each voxel. This color information is useful among other applications, for the identification and tracking of the positions and orientations of body parts.

In this paper, an algorithm for human 3D shape reconstruction that accelerates current voxel carving approaches is presented. The algorithm is additionally quickened by implementing part of the system with parallel threads (those processes that can be parallelized). Thus, the procedure achieves real-time performance for small voxel sizes, obtaining good reconstruction quality results compared to the approaches that have been accelerated. Along with this procedure, we show two fast approaches to extract the user's silhouettes from static backgrounds: (1) for general color static backgrounds and (2) for backdrops with a predefined color (*chroma-key*). The latter procedure enhances the classical chroma-key technique, as it can handle more robustly with casted shadows, including those coming from backdrop folds, and also even if those folds change during the subject's performances. Finally, we evaluate our algorithm accelerating an existing reconstruction approach, using as input data four points of view, in order to extract the four silhouettes of a subject in each frame.

This paper is organized as follows. Firstly, the general state of the art of markerless 3D shape reconstruction and its usability for HCI applications are explained. Then, the reconstruction process is described, starting from the silhouette extraction to continue with the carving procedure of the 3D space. Subsequently, two strategies, the Basic Accelerated Voxel Carving (BAVC) and the Enhanced Voxel Carving (EAVC), which accelerate the reconstruction process, are presented.

Experimental results show the suitability of the proposed algorithms for their use in HCI applications.

II. Related work

SFS or voxel carving is a challenging problem in computer vision. The first step to be attained by voxel carving approaches is the extraction of the 2D projections of the subject to be reconstructed, which can be obtained with background subtraction. It is difficult to discern which pixels correspond to the subject and which not, due to illumination changes and moving background. A review on these techniques is presented in [15], where it can be stated that there is not an optimal approach for every context and the robustness of the algorithm implies a higher computational cost.

For HCI applications a static background with a controlled illumination is usually supposed, which simplifies the silhouette extraction procedure. Furthermore, a green or blue backdrop can be used behind the subject, in order to ease the background subtraction.

This procedure is known as *chroma-key*. Green or blue colors, corresponding to the pure G and B tones in RGB color space, are the most employed hues as they are the furthest away from skin colors. Thus, the color channel corresponding to the backdrop is removed in order to subtract the background. This implies that the scene must be correctly illuminated in order to avoid casted shadows, which degrade the extracted silhouettes.

Next step consists on establishing the relations between the observed 3D world and its projections on the views. This can be done by calibrating the cameras, i.e., establishing the *intrinsic* and *extrinsic camera parameters*. Intrinsic parameters are the attributes that affect the image, whereas extrinsic parameters are the camera position and orientation in the real world.

Some methods to determine these parameters can be found in [16-18]. See [19] for detailed explanations on multi-view geometry.

Most of the voxel carving methods in literature use calibrated cameras [3, 4, 6, 20-27]. The final 3D shape reconstruction depends on the calibration quality. On the other hand, when the cameras are weakly calibrated, i.e., only the fundamental matrices that relate the views are known, but no the intrinsic and extrinsic parameters of each camera, the projective space representation can be defined. Saito and Kanade [28] adapted the voxel carving procedure to create projective shapes of subjects. Thus, they achieve the projective reconstruction by matching the correspondence points between the images.

SFS approaches usually work with a rectangular bounding volume where the subject lies on, which is discretized into small cubes of same shape and size called *voxels*. Another spatial division technique are *octrees* [3-6, 20]. A general octree approach starts from a bounding cube, which is divided into smaller cells or octants in case it corresponds to the subject being captured instead of the background. It works efficiently when there are large space regions as stated by Dyer [29]. Thus, it is often used for scene reconstruction. On the contrary, when the space has little free spaces the algorithm becomes slower as more operations have to be undertaken. On the other hand, ray space representations [27, 30] have less memory requirements, as they work directly with the resulting rays that start from the camera center and pass through every pixel of the subject's silhouettes. For a review of volumetric representations see [7].

The reconstructed subject's volume may be directly used for applications where the subject being captured interacts in a virtual world, by detecting the collisions with other virtual objects.

A higher level interaction may also be attained by estimating the positions and orientations of each body part of the subject, i.e., the kinematical structure. An example of HCI using reconstructed body shapes is that of the commercial system Softkinetic [36], which is capable of identifying and tracking the subject's body parts, from a depth sensing camera in real-time (about 15 fps). This system can also be used for animating industry standard 3D skeletons.

Using discretized representations such as voxels and octrees, each cube has to be labelled as belonging to the subject or not. This is usually a binary process and the conditions to label them can change depending on the approach. Many papers have been published during the last years [2, 22, 24-26] and two reviews can be found in [31, 32].

The algorithms presented in this paper for accelerating the voxel carving procedures are focused on voxel representations with the same size.

III. Three Dimensional Silhouette extraction with voxel carving

In the present section we describe a general voxel carving method, explaining which are the steps to achieve it.

Consider we have a subject being observed by n cameras around it. Firstly, we compute the n projected silhouettes, which consist on binary images which separate the subject from the background. These can be obtained by applying background subtraction algorithms (see [15] for a review).

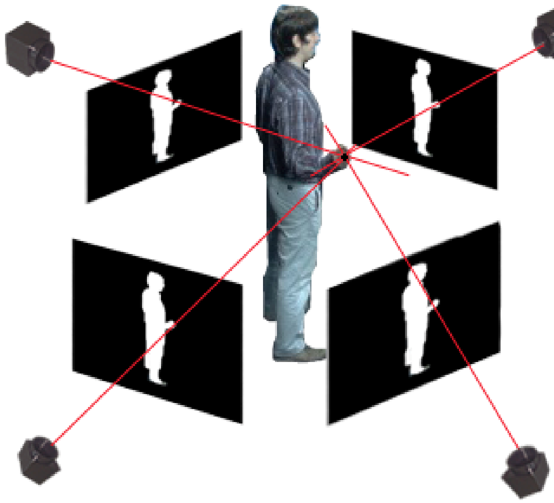

Fig. 1: An example of how the rays intersect in the hand of the subject. Each ray starts from the camera center, and crosses its corresponding image silhouette.

Then, having determined the camera parameters (which relate the projections of the views and the 3D world), we can define a ray in the scene space which starts from each camera focus, crosses a point in the projected silhouette and reaches its corresponding real point on the subject.

If this procedure is extended to all the silhouette points a cone-like volume is drawn, and the intersection of all the cones from each image forms the visual hull (Fig. 1).

There are many algorithms that reconstruct the inferred visual hull. Some methods work directly in the ray space representation, but this approach is computationally expensive. On the other hand, most of the solutions define a rectangular prismatic bounding volume of size M × N × P where the subject lies on. The volume is divided into cubes of same size, i.e., voxels. Thus, a general voxel carving algorithm works as shown in Algorithm 1.

ALGORITHM 1: General Voxel Carving Algorithm
(1)**Procedure** VoxelCarving(Volume)
(2) **For** each voxel of the Volume **Do{**
(3) Project the voxel onto the silhouette images(Projecting the 8 vertices of a cube becomes a convex hull)
(4) Set the voxel occupancy as *opaque* or *transparent* depending whether the projected voxel is contained in all the projected silhouettes or not
(5) }**End For**
(6)End procedure

There are different ways to set the occupancy of a voxel. The most robust, but also the most expensive consists on checking all the pixels that compose the projected image. Cheung et al. [25] accelerate this process using an algorithm called SPOT, which tests some uniformly distributed pixels within the convex hull. Similarly, Peroutka [26] projects the center of each cube. Some information can be lost depending on the voxel's size, but on the other hand the speed of the reconstruction can be increased significantly.

Fig 2 shows an example of this procedure where the occupancy of a voxel's projection is checked respect to the silhouette's pixels. On the other hand, Fig. 3 shows an example of how only the center of the voxel is checked for the reconstruction of a forearm.

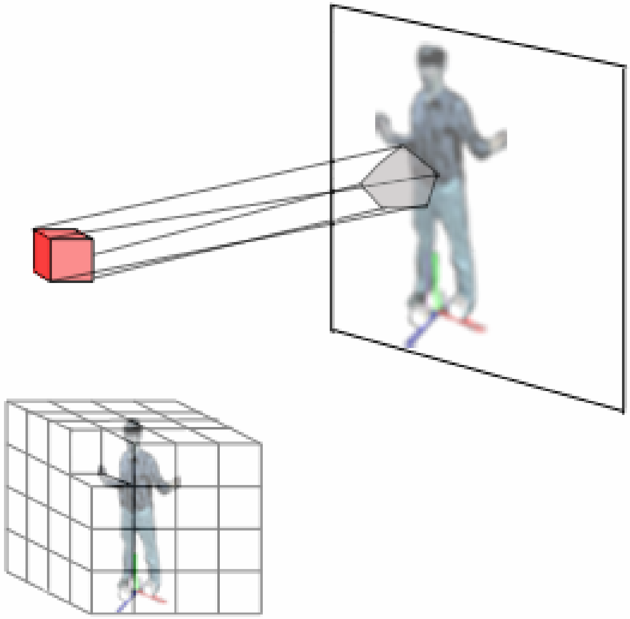

Fig. 2: An example of how the bounding volume is discretized into cubes with the same size and how on of these is projected into one of the silhouettes.

IV. Accelerated Voxel Carving with Controlled Backgrounds

The first step to achieve the 3D reconstruction is to extract the projected silhouettes for each point of view from which the subject is being observed .In the following subsection we show how this process can be achieved in a fast way, appropriate for HCI applications, (1) for general color static backgrounds and (2) for backdrops with a predefined color (chroma-key). The second subfield explains in detail the 3D reconstruction acceleration algorithms having as input data the silhouettes extracted from a general color static background.

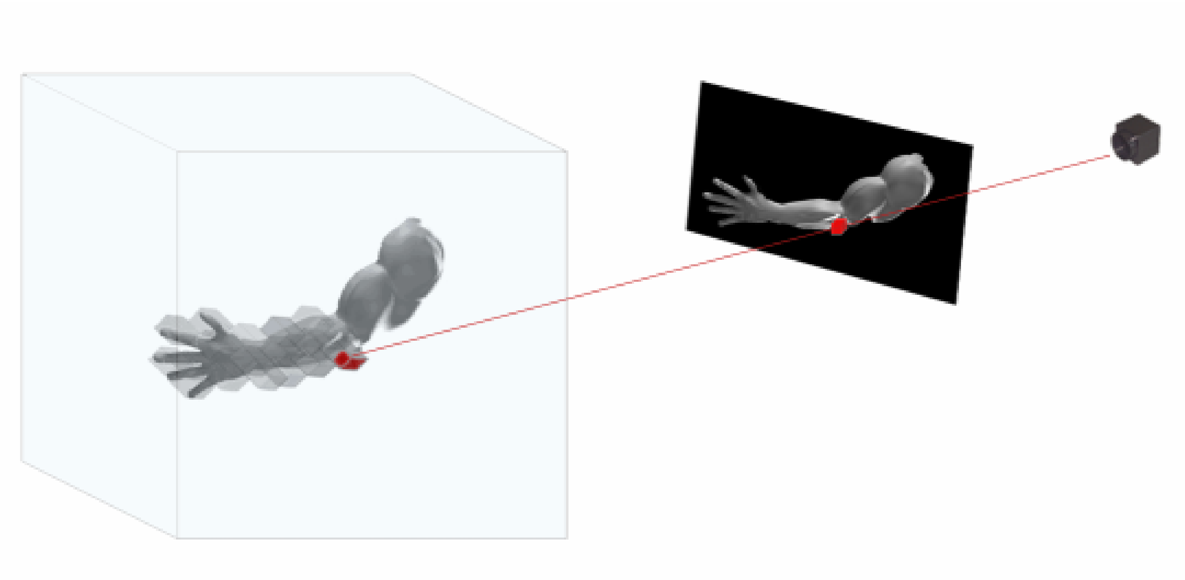

Fig. 3: A voxel carving method is being computed to the forearm checking Orly the occupancy of the voxel's center [26]. The ray crosses a silhouette edge from the camera viewpoint and goes into the bounding volume.

A. Silhouette Extraction

The input of this process are the background model and the images being grabbed during the capture. The output is in the form of a binary image, *the silhouette mask*, where white pixels correspond to the foreground (or subject's silhouette) and the black pixels to the subtracted background. This mask combined with the incoming RGB images results in an images where only the subject is visible. The textures of these images can be used for a further voxel coloring or space carving procedures. On the other hand, the use of the projected binary masks on their own are enough for the 3D shape reconstruction as only the voxel occupancies must be checked.

ALGORITHM 2: Gaussian RGB Model Background Subtraction

(1)**Procedure** GaussianRGBModelBS(I, BG_{mean}, BG_{stdDev}, colorThr)

(2) $T = (I - BG_{mean}) \cup (BG_{mean} - I) - BG_{stdDev}$

(3) Convert T channels (RGB) to binary separately (**If** pixel of T channel ≥ colorThr **Then** pixel = *white* **else** pixel = *black*) $\rightarrow TBinary_R, TBinary_G, TBinary_B$

(4) SilhouetteMask = $TBinary_R \cup TBinary_G \cup TBinary_B$

(5)End procedure

Algorithm 2 shows the procedure to subtract the background for general color static backgrounds. Initially, before the subject enters the scene, the background is recorded for a certain time. During that time pixel values vary around certain RGB values with a noise coming mainly from the fluorescent light flickering and the gain applied to the pixels to get brighter images. Using these background images a Gaussian RGB model for each pixel can be built, representing this way a background model.

The main advantage of the Gaussian RGB model is that it does not require having and installing any backdrop, but on the other hand casted shadows are present (Fig. 4).

There are many techniques to remove them (see [33] for a review). In our case, as the background colors are darker than those of the subject's clothes and skin, a threshold can be used in the grey-level image to remove them, because their intensity values are lower than those of the non-shadow regions. A diffuse illumination can also diminish the size of the casted shadows.

Nevertheless, for many HCI applications it is not strictly necessary to eliminate completely these shadows. A small shadow below the feet, as the one shown in Fig.4, does not constitute a problem for the interaction of the user with virtual objects as these are normally done with user's upper body parts.

Algorithm 3 shows an approach to extract the subject's silhouette using a green backdrop behind. It can also be applied to blue backdrops replacing green by blue and vice versa in the algorithm.

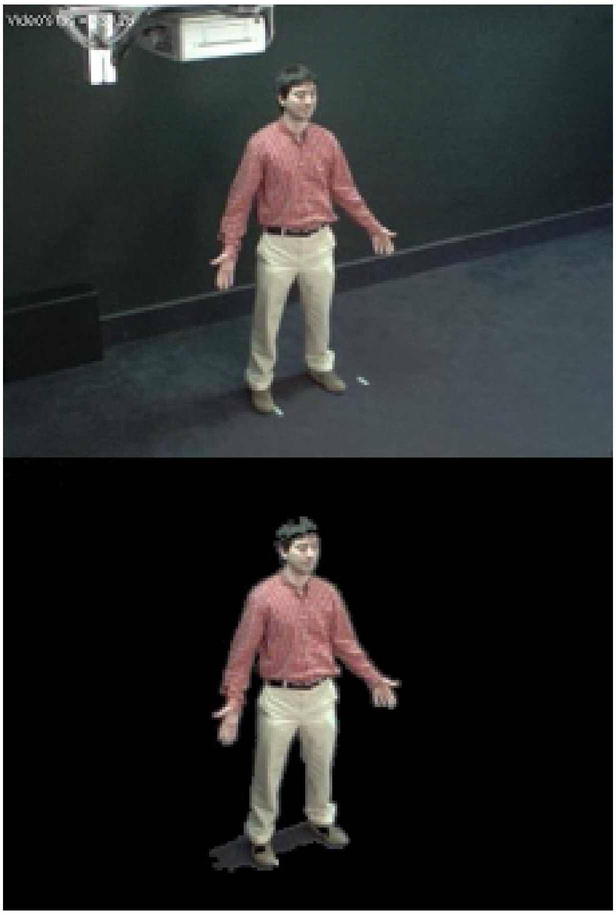

Fig. 4: Fast background subtraction by Gaussian RGB model of pixels.

Fig. 5: Fast background subtraction with a green chroma-key.

ALGORITHM 3: Background Subtraction with a Green Backdrop

(1)**Procedure** GreenChromaKeyBS(I, colorThr)
(2) Split *I* into separate color channels $\rightarrow I_R, I_G$ and I_B
(3) $T_1 = I_G - I_R$
(4) Convert T1 to binary (**If** pixel of $T_1 \geq$ colorThr **Then** pixel = *white* **else** pixel = *black*) $\rightarrow TBinary_1$
(5) $T_2 = I_G - I_B$
(6) Convert T_2 to binary (**If** pixel of $T_2 \geq$ colorThr **Then** pixel = *white* **else** pixel = *black*) $\rightarrow TBinary_2$
(7) $TBinary_3 = TBinary_1 \cap TBinary_2$
(8) SilhouetteMask = inverted binary image of $TBinary_3$
(9)**End procedure**

It can be observed in Fig. 5 that as the backdrop tone is very similar to that of the RGB green channel, it is possible to attain clean silhouettes without casted shadows, even those coming from backdrop folds, and also even if those folds change during the subject's performances. This way, the classical chroma-key background subtraction technique is enhanced as there is not need to illuminate in a strictly diffuse mode the scene.

The use of this type of backdrops allows a wider range of clothes and skin colors. Apart from these two methods, in the case that a depth-sensing camera is used, such as in the case of Softkinetic system [36], a depth threshold can be used in order to ignore elements behind the subject. This approach is also appropriate for background subtraction for indoors HCI applications, but is not included in the results of this paper.

The more cameras are used, the higher the computational cost will be, as the background subtraction procedure must be performed in each camera. As the resulting silhouettes are not related among them for their extraction, i.e., they can be obtained separately, the procedure can be easily parallelized using *multithreading programming* [37].

B. *Accelerated Voxel Carving*

For a correct reconstruction of the 3D shape of the subject it is important that all the silhouette masks correspond to the same time instant. This requires all cameras to be synchronized, especially when a high number of cameras are used, as the computer's performance may be affected, even if multithreading programming is used. This is usually achieved by hardware, in which electric pulses emitted by the network or an external trigger specifies when cameras grab images.

Depending on the size and the number of voxels that compose the reconstructed shape it can be computationally expensive to check all the voxel occupancies, even in the more simplified versions like those of Cheung et al. [25] and Peroutka [26]. Therefore, in this section we present some improvements to accelerate the voxel carving process, independently on how the voxel occupancy is checked.

We will define first what the basic idea of the method is, which we refer to as Basic Accelerated Voxel Carving (BAVC). Then, with the addition of some improvements to this basic procedure we can obtain higher framerates with a slight decrease on the reconstruction quality. We refer to the latter as Enhanced Accelerated Voxel Carving (EAVC).

Once the subject's silhouettes are extracted for all the views with a background subtraction technique, and having determined the camera parameters (which are obtained by camera calibration before motion capture starts), then the voxel carving procedure can be applied.

Bearing in mind the general voxel carving explained in section 3, in the BAVC method, instead of projecting all the voxels onto the reference images one by one, we go through the bounding volume with a step *S*, i.e., jumping S voxels each time one is checked. In the case that one of the tested voxel belongs to the subject (i.e., it is within all the projected silhouettes at the same time), the surrounding voxels (which were skipped by the step forming a cube shape subset of size $S \times S \times S$) are then projected onto the cameras and their occupancies are determined. Algorithm 4 shows the BAVC overall procedure, and Fig. 6 and 7 depict in a graphical way an overview of this approach.

The method starts checking voxel occupancies from one corner of the bounding volume where the subject's 3D shape is being calculated, following the X direction until there are no more voxels to project in that row. Later, it continues through the next rows advancing firstly through X, then through Y and finally through Z directions until all the bounding volume is covered.

Going further with the previous concept, the neighborhood of a voxel has six bounding walls. Each of them can be *interior* or *exterior*, depending whether it is next to another neighborhood area or not.

In the BAVC algorithm the field of search is extended by testing the surrounding neighboring cube subsets, which recursively calls to other neighbors until all the voxels inside a cube are transparent (i.e., none of them contain the subject). This method has proved to be very robust compared to a typical voxel carving procedure, and accelerates it considerably. Algorithm 5 shows the recursive procedure embedded in Algorithm 4, in order to go through other neighboring $S \times S \times$ S cube subsets.

ALGORITHM 4: Basic Accelerated Voxel Carving

```
Procedure BAVC(S, volume)
  For each voxel of the VOLUME (using step S) Do {
    Set the voxel occupancy as opaque or transparent
    depending whether the projected voxel is contained in
    all the projected silhouettes or not
    If voxel is opaque then {
      For each voxel of the subset (S×S×S) Do{
        Set the voxel occupancy as opaque or transparent
        depending whether the projected voxel is
        contained in all the projected silhouettes or not
      }End For
      For each of the six bounding walls of the subset Do{
        CheckWallsNeighborhood(subset, S)
      }End For;
    }End if;
  }End For;
End procedure
```

The BAVC algorithm can be improved, thus transforming it into the EAVC algorithm, by pruning some of the neighboring subsets instead of checking the six bounding walls each time. Therefore, another condition is added to the BAVC algorithm: if there is an opaque voxel in a bounding wall, then the neighboring subset to that wall is checked, and otherwise the recursive algorithm does not continue through that wall.

ALGORITHM 5: Check Walls Neighborhood

```
Procedure CheckWallsNeighborhood(Subset, S)
  For each voxel of the SUBSET (using step S) Do{
    Set the voxel occupancy as opaque or transparent
    depending whether the projected voxel is contained in
    all the projected silhouettes or not
  }End For
  If (one or more voxels from the subset are opaque)
  Then{
    For Each of the six bounding walls of the subset Do{
      CheckWallsNeighborhood(subset_Side_k, S)
    }End For;
  }End If;
End procedure
```

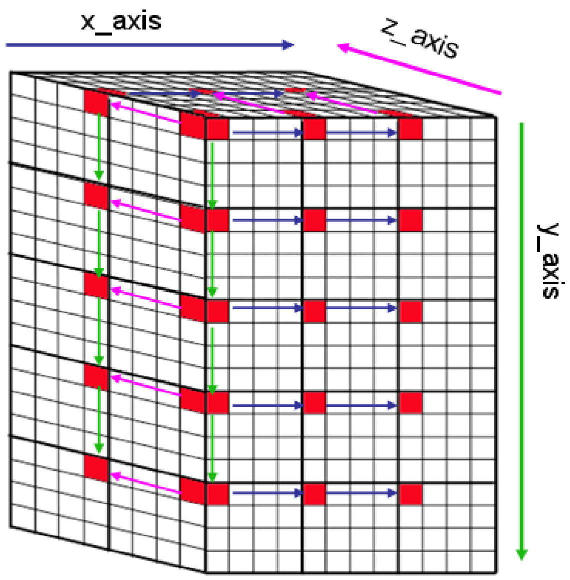

Fig. 6: The grid of voxels is tested with a step value of 4. The boundaries of the checked voxel neighborhoods are outlined with rough lines.

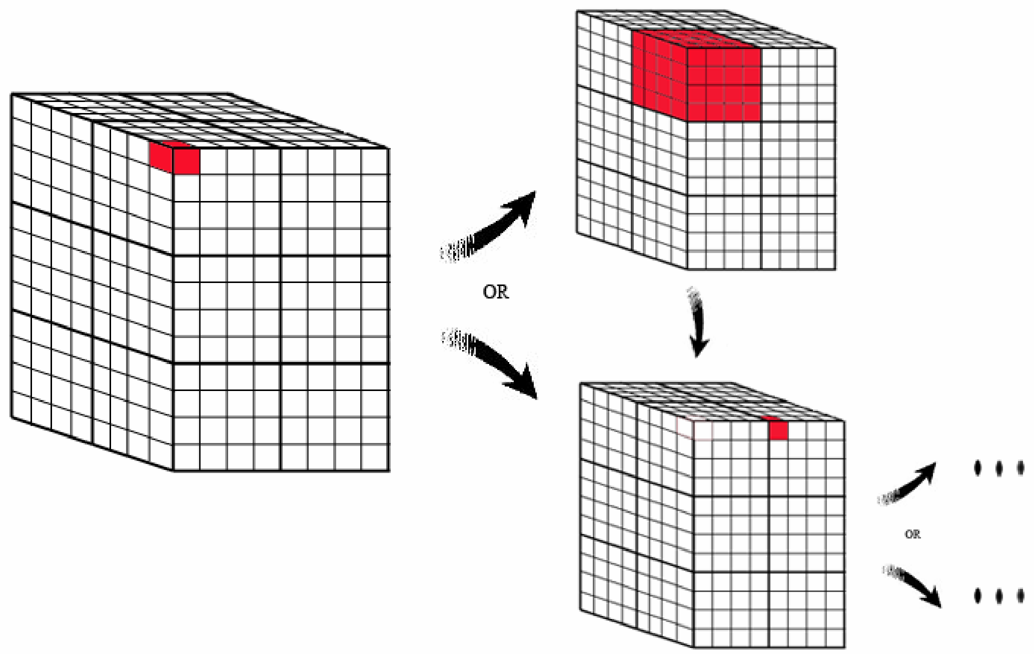

Fig.7: The BAVC algorithm step by step. Firstly, it starts checking one voxel, and depending how it is labelled the procedure follows by checking the neighbors or, by checking the next voxel with a step S.

V. EXPERIMENTAL RESULTS

We have implemented and tested both algorithms, BAVC and EAVC, taking Peroutka's [26] method as a basis for voxel carving, and capturing the subject from four different viewpoints. Images are grabbed with FireWire cameras working at 30 FPS with a resolution of 320 × 240, calibrated using Zhang's method [16] and connected to a 2.4 GHz Intel Core 2 Duo with 2 GB of RAM. The software has been programmed in C++ with OpenCV computer vision library [34]. The background subtraction algorithms have been implemented using multithread programming, which speeds up considerably the image processing, compared to that of single-thread. The rendering process is undertaken in a single thread using OpenGL graphics library [35].

The considered rectangular prismatic bounding volume size is of 150 × 200 × 150 cm^3, and the tested voxel lengths vary from 2 to 5 cm. On the other hand, voxel steps *S* of 4, 7 and 10 are used in the BAVC and EAVC algorithms. We represent the reconstruction quality with the obtained number of voxels relative to a method that checks the occupancy of every single voxel of the bounding volume (in this case Peroutka's method). This way, the reconstruction qualities and the computation times obtained by the three methods are shown in Table 1. Fig. 8 shows a sample of the obtained 3D shape reconstruction.

VI. CONCLUSION

In this paper we have presented two strategies, the Basic Accelerated Voxel Carving (BAVC) and the Enhanced Voxel Carving (EAVC), which accelerate voxel carving procedures, in order to obtain in real time the approximated 3D shape (*visual hull*) of a subject being observed from different points of view for HCI applications.

TABLE I
VOXEL CARVING ALGORITHMS COMPARISON

		Number of Voxels (%)			Median Computation Time (s)		
	Step / Voxel Size(cm)	4	7	10	4	7	10
Peroutka	2		100			4.2768	
	3					1.1735	
	4					0.4820	
	5					0.2516	
BAVC	2	99.98	99.98	99.98	0.5056	0.4472	0.4372
	3	99.68	99.97	96.36	0.1933	0.1870	0.1746
	4	100	96.88	100	0.1108	0.1012	0.1059
	5	100	88.86	100	0.0796	0.0712	0.0744
EAVC	2	99.25	99.38	99.25	0.3822	0.3253	0.3094
	3	99.37	97.58	95.55	0.1415	0.1197	0.1159
	4	98.14	95.02	95.36	0.0764	0.0730	0.0646
	5	98.17	88.61	98.86	0.0557	0.0482	0.0549

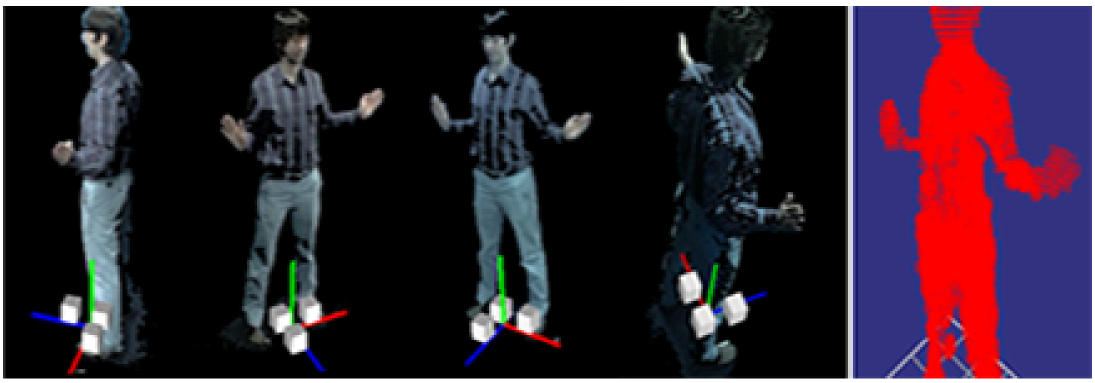

Fig. 8: On the left, the projected subject silhouettes obtained from four different points of view, and on the right its corresponding 3D shape reconstruction.

Besides, we have also presented two fast background subtraction techniques for controlled environments: (1) for general color static backgrounds and (2) for backdrops with a predefined color (*chroma-key*).

Experimental results have shown that both background subtraction techniques allow to obtain clean enough silhouettes of the subject for a further 3D shape reconstruction. It has also been shown that the latter procedure can handle more robustly with casted shadows, including those coming from backdrop folds, even if those folds change while the subject is moving. Hence, it enhances the classical chroma-key technique, where the color channel corresponding to the backdrop is removed in order to subtract the background.

Regarding experimental results on 3D shape reconstruction, they have demonstrated that both BAVC and EAVC algorithms accelerate the reference voxel carving procedures, which have a volumetric representation based on a regular discretized bounding volume, maintaining a considerable good quality respect to their basic implementation. The obtained frame rates are good enough for real-time applications, and hence HCI applications, depending on the voxel size and the method to be accelerated.

Future work will focus on adding texture information (i.e., color) to the reconstructed 3D shapes implementing all or part of the system on the GPU instead of the CPU in order to speed up the renderization process. This way, higher level interactions may be attained with the computer as it opens up the possibility of tracking body parts separately. It will also be studied how to identify motor semantic gestures from sequences containing the temporal evolution of the reconstructed 3D shapes. And finally, we will study how to accelerate other volumetric representation approaches for 3D reconstruction, like octrees.

ACKNOWLEDGMENT

This work has been subsidized by the SKILLS project Multimodal Interfaces for Capturing and Transfer of Skill funded by the European Union, FP6-IST-2005-035005, the Ministry of Education of Spain within the framework of the Torres Quevedo Program, and cofunded by the European Social Fund.

REFERENCES

[1] Laurentini, A., The Visual Hull Concept for Silhouette-Based Image Understanding, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 2, 1994, pp. 150-162.

[2] Martin, W, Aggarwal, J. Volumetric Descriptions of Objects from Multiple Views, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 5, No. 2, 1983, pp. 150-158.

[3] Potmesil, M., Generating Octree Models of 3D Objects from their Silhouettes in a Sequence of Images, *Computer Vision, Graphics and Image Processing*, Vol. 40, No. 1, 1987, pp.1-20.

[4] Szeliski, R., Rapid Octree Construction from Image Sequences, *Computer Vision, Graphics and Image Processing: Image Understanding*, Vol. 58, No. 1, 1993, pp. 23-32.

[5] Szeliski, R., Real-time Octree Generation from Rotating Objects, *Technical Reports 90/12*, Cambridge Research Lab, Cambridge, Massachusetts, USA, 1990.

[6] Ahuja, N., Veenstra, J., Generating Octrees from Object Silhouettes in Orthographic Views, *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 11, No. 2, 1989, pp. 137-149.

[7] Slabaugh, G., Culbertson, B., Malzbender, T., Schafer, R., A Survey of Methods for Volumetric Scene Reconstruction from Photographs, *Proceedings of the International Workshop on Volume Graphics*, 2001, pp. 81-100.

[8] Kutulakos, K., Seitz, S., A Theory of Shape by Space Carving, *Technical Reports TR692*, Computer Science Department, University of Rochester, New York, USA, 1998.

[9] Zach, C., Karner, K., Reitinger, B., Bischof, H., Space Carving on 3D Graphics Hardware, *VRVis Technical Report TR-VRVis-2004-013*, Institute for Computer Graphics and Vision Technical University Graz, Austria, 2005.

[10] Montenegro, A.A., Pinto, P.C., Gattas, M., Rodrigues, L.C.P., Adaptive Space Carving, *Proceedings of the 3D Data Processing, Visualization, And Transmission*, 2004, pp. 199-206.

[11] Yang, Y.K., Lee, J., Kim, S.K., Kim, C.H., Adaptive Space Carving with Texture Mapping, *Computational Science and Its Applications, LNCS 3482*, 2005, pp. 1129-1138.

[12] Seitz, S.M., Dyer, C.R., Photorealistic Scene Reconstruction by Voxel Coloring, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 1067-1073.

[13] Prock, A.C., Dyer, C.R., Towards Real-time Voxel Coloring, *Proceedings of the Image Understanding Workshop*, 1998, pp. 315-321.

[14] Culbertson, W. B., Malzbender, T., Slabaugh, G., Generalized Voxel Coloring, Vision Algorithms: Theory and Practice. *Proceedings International Workshop on Vision Algorithms*, *LNCS 1883*, 2000, pp. 100-115.

[15] Piccardi, M., Background Subtraction Techniques: A Review, Proceedings of the IEEE International Conference on Systems, Man and *Cybernetics*, Vol. 4, 2004, pp. 3099-3104.

[16] Zhang, Z., A Flexible New Technique for Camera Calibration, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 11, 2000, pp. 1330-1334.

[17] Svoboda, T., Martinec, D., Pajdla, T., A., Convenient Multi-Camera Self-Calibration for Virtual Environments, *PRESENCE: Teleoperators and Virtual Environments*, Vol. 14, No. 4, 2005, pp. 407-422.

[18] Tsai, R. Y., An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, USA, 1986, pp. 364-374.

[19] Hartley, R., Zisserman, A., *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[20] Franquesa, M., Brunet, P., Analysis of Methods for Generating Octree Models of Objects from Their Silhouettes, *Technical Reports LSI-96 46-R*, The Pennsylvania State University, Pennsylvanian, USA, 1996.

[21] Li, M., Magnor, M., Seidel, H.P., Hardware-Accelerated Visual Hull Reconstruction and Rendering, *Graphics Interface*, 2003, pp. 65-71.

[22] Cheung, K.M, Baker, S., Kanade. T., Shape-From-Silhouette of Articulated Objects and its Use for Human Body Kinematics Estimation and Motion Capture, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, 2003, pp. 77-84.

[23] Franco, J.S., Lapierre, M., Boyer, E., Visual Shapes of Silhouette Sets, *3D Processing, Visualization and Transmission,* Chapel Hill, USA, 2006.

[24] Cheung, K.-M., Baker, S., Kanade, T., Shape-From-Silhouette Across Time Part I: Theory and Algorithms, *International Journal of Computer Vision*, Vol. 62, No. 3, 2005, pp. 221-247.

[25] Cheung, K.-M., Kanade, T., Bouguet, J-Y., Holler, M., A Real Time System for Robust 3D Voxel Reconstruction of Human Motions, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, 2000, pp. 714-720.

[26] Peroutka, R., Acquisition of Voxel Data for Human Body Models, *RITE Technical Reports*, University of Maryland, Baltimore, Maryland, USA, 2005.

[27] Matusik, W., Buehler, C., Raskar, R., Gortler, S. J., McMillan, L., Image-Based Visual Hulls, *Proceedings of the International Conference on Computer Graphics and Interactive Techniques*, 2000, pp. 369-374.

[28] Saito, H., Kanade, T. Shape Reconstruction in Projective Grid Space from a Large Number of Images, *Proceedings of the IEEE Computer Vision and Pattern Recognition*, Vol. 2, 1999.

[29] Dyer, C.R., Volumetric Scene Reconstruction, *Foundations of Image Understanding*, Kluwer, 2001, pp. 469-489.

[30] Kutulakos, K. N., Shape from the Light Field Boundary, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 53-59.
[31] Dyer, C.R., Volumetric Scene Reconstruction, *Foundations of Image Understanding*, Kluwer, 2001, pp. 469-489.
[32] Szeliski, Stereo Algorithms and Representations for Image-Based Rendering, *Proceedings of the British Machine Vision Conference*, 1999, pp. 314-328.
[33] Prati, A., Mikic, I., Trivedi, M. M., Cucchiara, R., Detecting Moving Shadows: Algorithms and Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 7, 2003, pp. 918-923.
[34] Intel, OpenCV: Open Source Computer Vision Library, 2006, <http://sourceproject.net/project/opencvlibrary>.
[35] SGI, OpenGL: Open Graphics Library, 2006, <http://www.opengl.org/>.
[36] Softkinetic, Building Natural Interfaces, 2008, <http://www.softkinetic.net/>.
[37] S. Prasad, *Multithreading Programming Techniques*, J. Ranade Workstation Series, 1997.
[38] Suzuki, M., Yaginuma, Y., Kato, H., Yamada, T., *A Pattern Recognition Technique for Voxel Data Based on 3D Masks*, WSEAS Transactions on Electronics, 3, vol. 1, 562-567, 2004.
[39] Lascu, M., Lascu, D., A New Morphological Image Segmentation with Application in 3D Echographic Images, 3, vol. 5, pp. 72-82, 2008.
[40] Miidla, P., Rannat, K., Uba, P., *Simulated Studies of Water Vapour Tomography*, WSEAS Transactions on Environment and Development, 3, vol. 4, pp. 181-190, 2008.