Generation and Manipulation of H-Anim CAESAR Scan Bodies

Qiming Wang
National Institute of Standards and Technology
qwang@nist.gov

Sandy Ressler
National Institute of Standards and Technology
sressler@nist.gov

Abstract

In this paper we present a procedure to create animated human models, compliant with the H-Anim standard, from 3D CAESAR scan bodies, which were captured using a whole body scan device. We also present a VRML prototype of an "Animated CAESAR Viewer" to view and manipulate the generated CAESAR body animations interactively on the Web. The animated body model follows the H-Anim skinned body geometry specification. The vertex blending method has been used for smoother skin deformations. The model can be integrated with motion capture data. Although the process to generate an H-Anim body involves several different techniques, the discussion is focused on the methods of creating segments and assigning vertex weights. The Viewer provides the functions for a user to explore the components of the digital human model, to adjust the joint locations, to make body postures with a direct kinematics method, and control the animation using VCR-like controls. The aim of the Viewer is to help digital human modelers create more realistic postures and motion sequences intuitively.

Keywords: H-Anim, VRML, 3D scan, CAESAR, 3D animation

1. Introduction

The CAESAR (Civilian American and European Surface Anthropometry Resource) project has collected whole body 3D scans, seventy three 3D anthropometric landmarks, and traditional measurements for about 5,000 subjects. The 3D high-resolution data were scanned using a Cyberware WB4 whole body scanner. Each subject has three scanned postures: one standing and two seated as shown in Figure 1.

The CAESAR scan database was designed for anthropometric measurement analysis, and does not support applications related to human motions without further effort. If we can create animated human models from the scan and landmark data, it will stimulate research and designs in various industries.

Figure 1: A CAESAR body with three postures.

Another consideration is that the animated human model can be displayed and interactively manipulated in the web browser. They can also be embedded in a virtual environment. Therefore, we choose the H-Anim (Humanoid Animation) skinned body geometry standard as our target human model representation. The body surface is a continuous piece of geometry, a skin, which has an underlying skeleton. The articulated skeleton is defined as a hierarchical tree of joints.

Some previous work has been done for animating scanned human models. One of them is presented in the paper [Oliveira et al. 2003]. They implemented the methods for creating a layered human model, which includes generation of the skeleton by the use of automatically extracted key landmarks or manually selected landmarks, a surface to bone mapping technique, and a vertex blending algorithm for deformation modeling. Since the H-Anim standard [Humanoid Animation Working Group 2004] has been published, some software, tools, and papers have contributed to H-Anim avatars and articulated humans. "H-Animator" [Buttussi et al. 2006] provided a fast and easy-to-use visual tool to help in each of the modeling phases.

2. Generating H-Anim CAESAR Body

The conversion from CAESAR to H-Anim involves technical challenges in various areas, such as mesh reconstruction, 3D human models, and animation of the human body. Many of the issues are still in research stages. We are not trying to solve these critical technical problems. The purpose of this work is to bring together some existing technologies, and develop a tool kit to implement an experimental conversion from scan data to H-Anim models, which can be used for further research in digital human modeling.

Web3D 2007, Perugia, Italy, April 15–18, 2007.
© 2007 ACM 978-1-59593-652-3/07/0004 $5.00

2.1 Mesh Reconstruction

A CAESAR scan is a 3D polygonal mesh of more than 200,000 vertices and 300,000 triangles with color per vertex. These models make storage, transmission, computation, and display less efficient. Therefore, the first step of the processing is surface simplification. We only considered publicly available polygon compression software for our purpose. After the research in the Web, we decided to adopt QSlim 2.0, [Garland 1999] proposed and developed by Michael Garland [Garland and Heckbert 1998]. It takes about 15 seconds for a PC with an Intel Pentium 4 2.40 GHz processor to simplify the 300,000 triangle polygons to 50,000 polygons. QSlim 2.0 not only maintains high fidelity to the original model, but also preserves the detail color and texture information in geometrically smooth areas. The figures shown in this paper were compressed to 50,000 triangles using this software.

The second processing step is to fill the holes on the surface. 3D scanning systems often cannot observe the whole surface; the resulting 3D models may be incomplete. CAESAR scanned human bodies contain many holes, which have serious effects on the modeling. Within publicly available hole filling software, we tested two software packages: "Volfill" [Stanford 2005], based on volumetric diffusion theory; and "Tight Cocone" [Dey and Samrat 2003], based on Voronoi diagram and Delaunary triangulation. In general, both methods could fill the small and simple holes. For the complicated holes, such as around hands, under armpits, and around ears, they are not perfect. However, the "Volfill" generated some surface protrusion with extra polygons in some areas, so that the resulting model is not acceptable. At present, we use "Tight Cocone". In addition, the hole filling software does not consider color, so we developed a simple program to use original vertex color to estimate the color for each vertex in reconstructed model.

2.2 Generation of H-Anim Humanoid Node

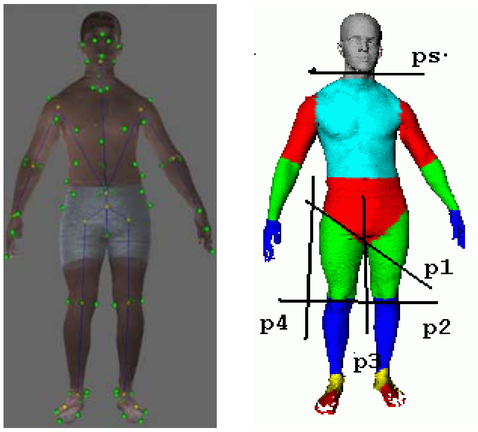
In this section, we discuss the technical issues involved in generating skin based H-Anim Humanoid: joint hierarchy, and skin deformation model.

Based on the available information in the landmark data, we adopted H-Anim LOA ONE skeletal hierarchy. The joint centers are critical to digital human animation. Determination of all of the joint centers using only surface data (including landmarks) is still a research topic. At present, we estimate the joint centers using the average of the landmarks around the areas. Figure 2(a) shows the locations of the landmarks and joints of one CAESAR body. The green balls indicate landmarks on the surface. The yellow balls present the joints inside the body. The lines between the joints represent the joint hierarchy.

The skinCoordIndex and skinCoordWeight fields defined in H-Anim Humanod Node, are two important attributes related to the skin model. The skinCoordIndex field indicates what vertices will articulate when the joint rotates. The skinCoordWeight presents the weights of vertices defined in the skinCoordIndex. In our implementation, we used the method discussed in [Oliveira et al. 2003] to calculate these attributes. Two techniques are involved: segmentation method and vertex blending deformation.

We used joint separation planes and additional clipping planes to divide the scan surface into segments. The division should insure that the segment distribution is as accurate as possible, and each vertex on the surface belongs to one and only one segment, except on the joint separation planes. A joint separation plane, or a clipping plane, is a 3D plane specified by a normal and a point on the plane. Suppose that a 3D plane is represented as A x + B y + C z + D = 0, where vector (A, B, C) is unit normal. The value of D can be determined by substituting in the known point (Px, Py, Pz) on the plane. For any vertex (Qx, Qy, Qz) on 3D space, the expression, side (Q) = A Qx + B Qy + C Qz + D, computes which side of the plane the vertex lies on. If it is positive, the point lies on the same side as the normal. If negative, it lies on the other side; if zero, it lies on the plane.

Figure 2(b) shows the separation of the body, and some clipping planes. The segments are distinguished with different colors. Four planes, **p1**, **p2**, **p3**, and **p4** were used to generate the segment associated with right hip joint. We could not create all of the clipping planes automatically due to the complexity of the scan surface. Therefore, we developed a semi-automatic method to define a set of clipping planes for an individual body interactively. This significantly improves the result.



(a) (b)

Figure 2: (a) Landmarks and Joints. (b) Segments.

We used the following vertex blending deformation model after the body segmentation. Suppose one joint links two segments, S1 and S2, as shown in Figure 3. S1 is between this joint and child joint, S2 is between it and its parent. We divide the whole area into two regions: a deformable region (the shaded area in the figure) and an un-deformable region (the un-shaded area). If the distance between a vertex and the joint separation plane is less than ¼ of the segment length, it belongs to the deformable region. Otherwise, it is in the un-deformable region. For the vertex on the separation plane, the weight is 0.5. In the deformable region of S1, the vertex has weight linearly changing from 0.5 to 1.0. In the un-deformable region of S1 side, the weight is 1. In deformable region of S2, the vertex has weight linearly changing from 0.5 to 0. The weight is 0 for un-deformable area.

Figure 4 shows a comparison of the skin deformation between a method without vertex blending and with vertex blending around the knee. The smoothness of the motion with vertex blending has been improved, especially in the areas of the knees and elbows.

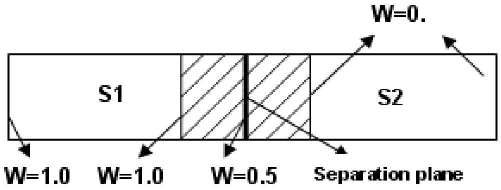


Figure 3: An illustration of weight assignment.

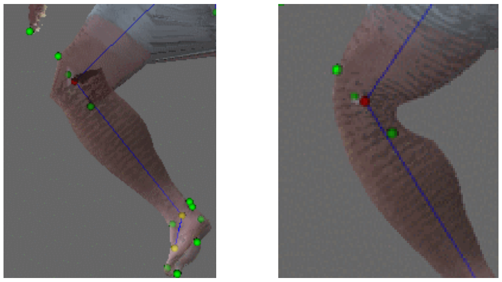


Figure 4: A comparison skin deformation without vertex blending and with vertex blending.

The generated H-Anim CAESAR bodies are able to move after the motion information is embedded. We converted some BVH motion capture data to VRML interpolator nodes and integrated them to the bodies to form an animated CAESAR body.

3. Animated CAESAR Viewer

The created H-Anim CAESAR model can be viewed in any VRML/X3D web browser with an H-Anim implementation. We used the BSContact browser from Bitmanagement to view the bodies because of its native implementation of H-Anim. In order to let users and human modelers explore and manipulate the bodies visually and interactively, we developed a VRML prototype “Animated CAESAR Viewer” called “Viewer” in this paper. The Viewer has numerous functions, which include exploring the information of body parts visually, generating postures, adjusting the location of selected joints, and controlling the animation using VCR-like controls. An example H-Anim CAESAR H-Anim body with Viewer can be found at http://ovrt.nist.gov/anthroindex.html.

There are two user interface control panels. The first one is the skeleton control panel shown in the left side of Figure 5. In the middle part of the panel, there are eighteen ball-shaped buttons with joint names, cylinder-shaped buttons with segment names, and box-shaped buttons. The ball buttons are for selecting a joint, the cylinder buttons for displaying the vertex weight distribution of the selected joint, and the box buttons are for showing the selected segment. The top three sliders are used to set rotation values in X, Y, Z directions around the selected joint. The bottom sliders are used for translating the location of the selected joint in X, Y, and Z directions. The X, Y, and Z directions are in accordance with the world coordinate system. There are three buttons on the right side of the rotation sliders and three buttons on the right side of the translation sliders. They are used for generating body posture, adjusting joint location, and printing the rotation or translation data for the posture.

When the user clicks on one ball button, it becomes red, and the joint on the body becomes red. When moving the bead on one of the top sliders, the body parts responding to the joint will rotate around the selected joint. In addition, the text above the slider displays the rotation angles between -180 degree and 180 degrees. Moving the bead on the bottom slider, the selected joint of the body will translate in the X, Y, or Z directions. The number above the slider shows the distance moved. In Figure 5 joint “r_hip” is selected, the rotation around Z direction is set to -19 degrees. The body on the right side of the figure displays the “r_hip” joint rotation.

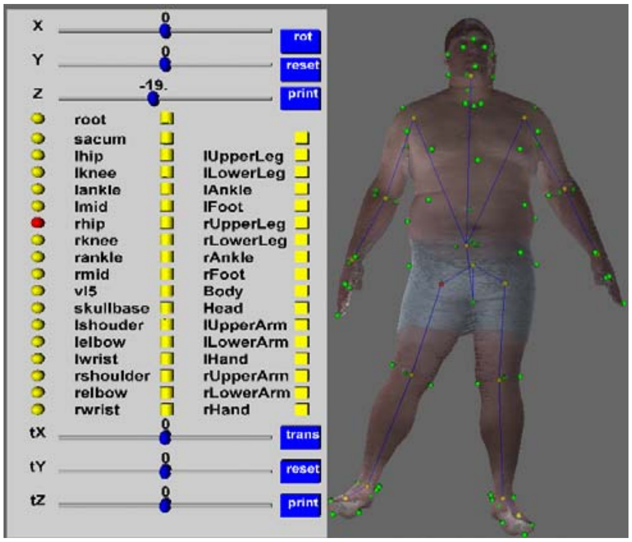


Figure 5: The skeleton control panel.

In the prototype, we allow the user to specify the joint motion limitations including the upper limit and the lower limit for any of the joints. If the rotation value on the slider bar is out of the limits specified for the selected joint, the body rotation will stop at the limit position and the color of the bead becomes red. An example is shown in Figure 6. The l_hip rotation limits are set to [-90, 90]. The “l_hip” joint moves and stops at the 90 degree location, even though the bead moves to 106 degrees.

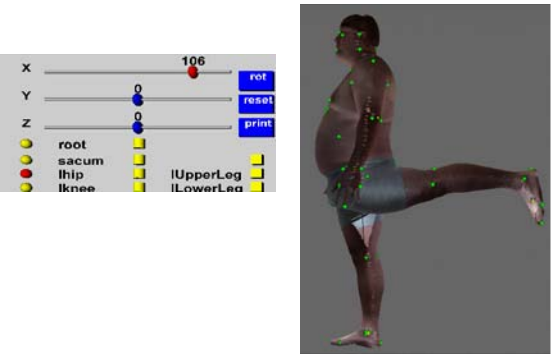
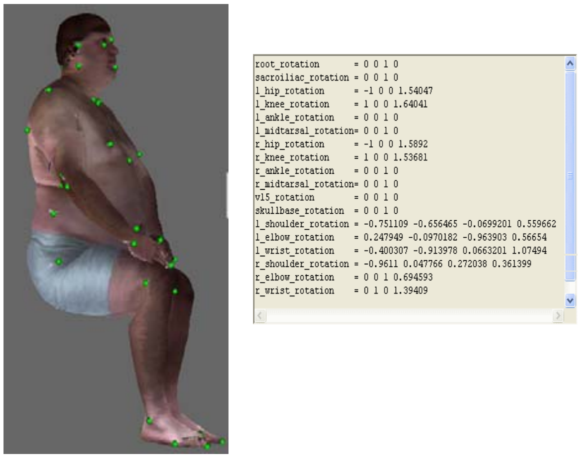


Figure 6: An example of joint rotation limits.

If we want to generate one posture with multiple rotations around more than one joint, we need a mechanism to make the rotation sequence. Each rotation in the sequence is based on the previous rotations. We specified three buttons on the top right: “rot”, “reset” and “print”. Clicking the “rot” button, the system will save the current rotation and integrate it into the previous result through performing VrmlMatrix multiplications. It is implemented in a Script node. The “reset” button is used to reset the selected joint rotation to its original location. The “print” button prints the joint rotations on the console, which only works for BSContact. The BSContact browser has a “print” function to display information on console. The user can save the result using cut and paste to capture the text on console. Figure 7 shows a sitting posture generated from the standing posture using the skeleton interface, and the joint rotation data.



```
root_rotation       = 0 0 1 0
sacroiliac_rotation = 0 0 1 0
l_hip_rotation      = -1 0 0 1.54047
l_knee_rotation     = 1 0 0 1.64041
l_ankle_rotation    = 0 0 1 0
l_midtarsal_rotation= 0 0 1 0
r_hip_rotation      = -1 0 0 1.5892
r_knee_rotation     = 1 0 0 1.53681
r_ankle_rotation    = 0 0 1 0
r_midtarsal_rotation= 0 0 1 0
vl5_rotation        = 0 0 1 0
skullbase_rotation  = 0 0 1 0
l_shoulder_rotation = -0.751109 -0.656465 -0.0699201 0.559662
l_elbow_rotation    = 0.247949 -0.0970182 -0.963903 0.56654
l_wrist_rotation    = -0.400307 -0.913978 0.0663201 1.07494
r_shoulder_rotation = -0.9611 0.047766 0.272038 0.361399
r_elbow_rotation    = 0 0 1 0.694593
r_wrist_rotation    = 0 1 0 1.39409
```

Figure 7: A sitting posture with console display.

The joint centers that we used are the estimations from landmark data. In the skeleton control panel, we provide controls for the user to adjust and print the center positions using slider bars and buttons. The process is similar to the described rotation process.

The second control panel in the “Viewer” is the Animation control panel. It has several VCR-like buttons to control the animation: play, pause, backward play, rewind and fast forward. It also includes a slider bar to play the animation manually. Figure 8 shows some snapshots from one female H-Anim CAESAR body’s motion.

4. Conclusion and Discussion

In this paper, we have presented an experimental implementation that converts a CAESAR scan and landmark data to an H-Anim body with smooth deformation. The methods are feasible for CAESAR scan data. We have generated more than 30 H-Anim bodies. It is more difficult to find good clipping planes to do segmentation for surfaces with complex and big holes. Vertex blending deformation works better for small to moderate rotations, and it works better for simple joints like the knee and elbow than for complex joints like the shoulder.

One issue presented in this paper is joint center determination. There are some possible solutions. One is to adjust the joint locations visually and interactively in order to make the motion acceptable. Another possibility is to use multiple posture data, such as the other two postures from the same CAESAR person.

The VMRL prototype “Animated CAESAR Viewer” provides an intuitive way to explore and use the digital human models. The integration of the H-Anim human body and motion capture data has great potential for the applications involving human actions. In the future, we will use the EAI (External Application Interface) to save joint rotations and translations for the posture designs; and obtain realistic joint motion limits from other sources.

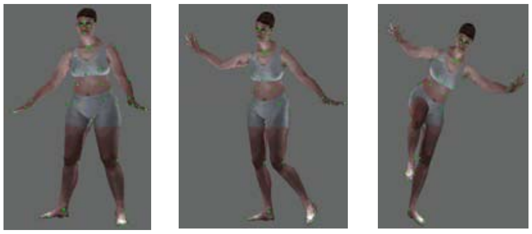


Figure 8: Snapshots of one body’s animation.

5. Acknowledgements

We would like to thank Kathleen Robinette and Dennis Burnsides of Wright Patterson Air Force Base for their help with the CAESAR data.

6. References

BUTTUSSI, F. CHITTARO, L. and NADALUTTI, D. 2006. H-Animator: A Visual Tool for Modeling, Reuse and Sharing of X3D Humanoid Animations, *Proceedings Web3D 2006*, 109-117.

DEY, T. and GOSWAMI, S. 2003. Tight Cocone: A Water-tight Surface Reconstructor, *Proc. 8th ACM Symposium. Solid Modeling Application, 2003*, 127-134.

GARLAND, M. and HECKBERT, P. 1998. Simplifying Surface with Color and Texture Using Quadric Error Metrics, *IEEE Visualization 1998.*

GARLAND, M. 1999. http://graphics.cs.uiuc.edu/~garland/software/qslim20.html

HUMANOID ANIMATION WORKING GROUP, 2004. http://www.hanim.org

STANFORD UNIVERSITY, 2005. http://graphics.stanford.edu/software/volfill/

OLIVEIRA, J., ZHANG, D., SPANALANG, B. 2003. Animating Scanned Human Models, *Journal of WSCG, Vol. 11*, No. 1.

Generation and Manipulation of H-Anim CAESAR Scan Bodies

Qiming Wang, Sandy Ressler

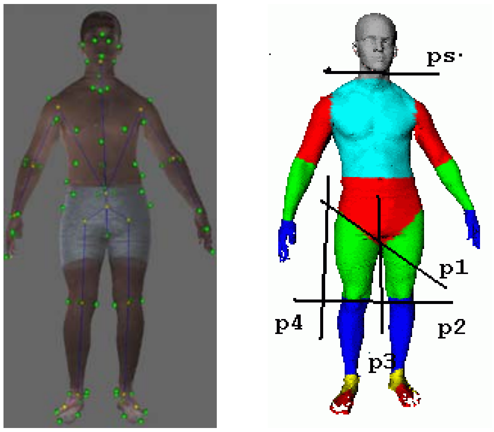


Figure 2: (a) Landmarks and Joints. (b) Segments.

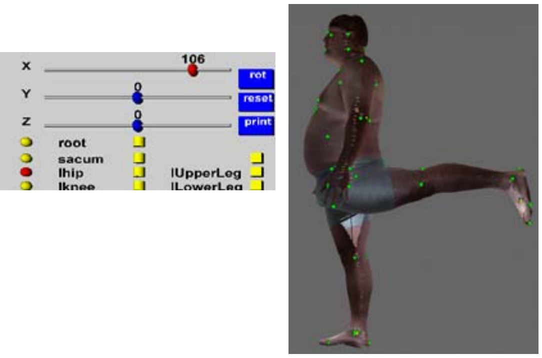


Figure 6: An example of joint rotation limits.

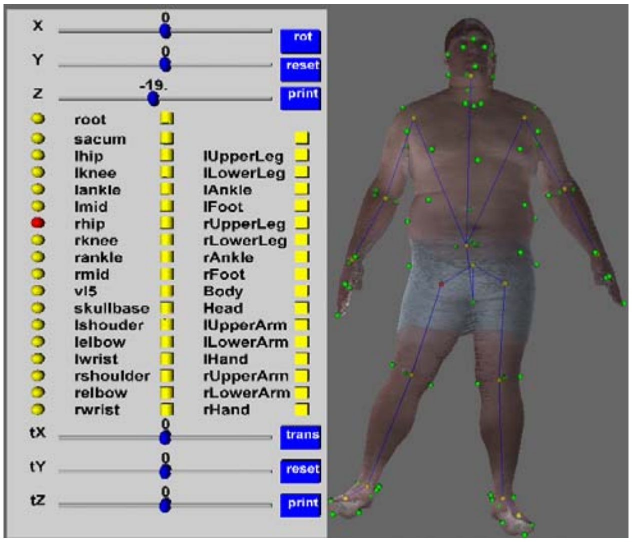
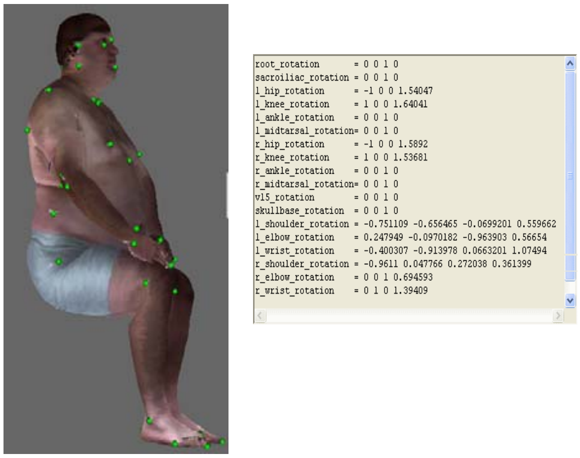


Figure 5: The skeleton control panel.



```
root_rotation       = 0 0 1 0
sacroiliac_rotation = 0 0 1 0
l_hip_rotation      = -1 0 0 1.54047
l_knee_rotation     = 1 0 0 1.64041
l_ankle_rotation    = 0 0 1 0
l_midtarsal_rotation= 0 0 1 0
r_hip_rotation      = -1 0 0 1.5892
r_knee_rotation     = 1 0 0 1.53681
r_ankle_rotation    = 0 0 1 0
r_midtarsal_rotation= 0 0 1 0
vl5_rotation        = 0 0 1 0
skullbase_rotation  = 0 0 1 0
l_shoulder_rotation = -0.751109 -0.656465 -0.0699201 0.559662
l_elbow_rotation    = 0.247949 -0.0970182 -0.963903 0.56654
l_wrist_rotation    = -0.400307 -0.913978 0.0663201 1.07494
r_shoulder_rotation = -0.9611 0.047766 0.272038 0.361399
r_elbow_rotation    = 0 0 1 0.694593
r_wrist_rotation    = 0 1 0 1.39409
```

Figure 7: A sitting posture with console display.

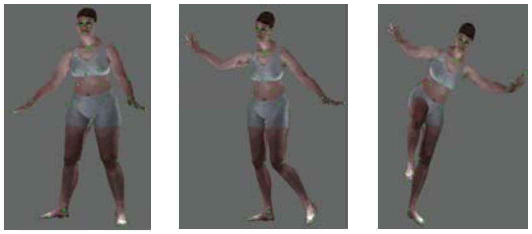


Figure 8: Snapshots of one body’s animation.