Modeling and Animating Virtual Humans for Real-Time Applications

Frank Hülsken, Christian Eckes, Roland Kuck, Jörg Unterberg and Sophie Jörg

***Abstract*—We report on the workflow for the creation of realistic virtual anthropomorphic characters. 3D-models of human heads have been reconstructed from real people by following a structured light approach to 3D-reconstruction. We describe how these high-resolution models have been simplified and articulated with blend shape and mesh skinning techniques to ensure real-time animation. The full-body models have been created manually based on photographs. We present a system for capturing whole body motions, including the fingers, based on an optical motion capture system with 6 DOF rigid bodies and cybergloves. The motion capture data was processed in one system, mapped to a virtual character and visualized in real-time. We developed tools and methods for quick post processing. To demonstrate the viability of our system, we captured a library consisting of more than 90 gestures.**

***Index Terms*—Real-time animation, 3D-reconstruction, optical motion capture, facial animation.**

I. INTRODUCTION

Despite the great quality of cartoon style characters and fantasy creatures, bringing a persuasive realistic virtual human to life is still a long and tedious task. However, on the other hand realistic appearance is essential for natural conversation. The use of such virtual humans for real-time applications including games and virtual reality involves additional difficulties, as there may be restrictions on the resources available at run-time.

Much progress has been made in the various fields that contribute to the realization of virtual humans. In this paper we present the full process for the creation of a life-like virtual human, from modeling and texturing to animation.

To produce a highly realistic result we base our 3D-characters on real people. A 3D scanner is used to create a detailed head scan and high resolution textures are also created. Using filtering techniques in the image acquisition phase we are able to estimate the properties of the human skin needed to render convincing human models. The remaining modeling of the body is straightforward. Here we use the standard 3D application Maya in which we also create the skeleton and the skinning of the model.

To capture the motion of the person we built a motion capture system consisting of infrared cameras, grouped retroreflective markers, which result in a high real-time robustness of the system, and two cyber gloves for the hands.

The paper is organized as follows: in section 2 we give a general overview of the current state of the art in the creation and animation of virtual humans. The next sections describe the different steps in details: the scanning of the head, the creation of blend shapes for facial animation, the modeling of the character and finally the capturing of motions to bring our character to life. Finally, our paper concludes with a summary overview of our system.

II. BACKGROUND

A good overview of virtual humans can be found in [1] which covers all aspects of creating and animating 3D characters with particular emphasis on facial expressions and natural behavior.

A guide for facial animation written for animators can be found in [2]. Osipa has created a methodology for facial animation which can be used as a step-by-step guide for modeling talking heads.

The creation of realistic speech is also a challenge, and a breakthrough in this area was made by Bregler [4]. Although they were working with video sequences of real people, this work demonstrated that it is possible to generate a new speech sequence by pasting phoneme sequences together, which nobody could recognize as fake.

The main pioneer in the field of facial animation is Parke [5]. In one of his methods the mesh is directly drawn on the face and captured with a camera to get the 3D information. In another method he captures the positions of markers fixed at specific points. This technique is also used in professional facial motion capture systems. In [6] Rahman presents a technique that makes use of dynamic regions of interest, i.e., those which provide more information. Instead of using markers it is possible to localize these features automatically to create 3D-meshes and track them over time.

The next step in facial animation is to register these captured positions on a generic animatable model. Ekman's Facial Action Coding System [7] provides a standard for mapping the motion captured from the markers to a set of predefined motions. Facial expressions represent a visible consequence of

Manuscript Received on January 18, 2007.

Frank Hülsken is a mathematician with background in pattern recognition. His research interests include speech recognition, image processing, 3D reconstruction and animation. E-Mail: frank.huelsken@iais.fraunhofer.de.

Christian Eckes received the diploma degree in Physics from the University of Dortmund in 1995. His research interests include human-computer interaction, multimedia analysis, pattern recognition and biological inspired computer vision. E-Mail: Christian.Eckes@iais.fraunhofer.de.

Roland Kuck is a computer scientist with a background in computer graphics. His research interests include visualization of large meshes, global illumination and shading. E-Mail: roland.kuck@iais.fraunhofer.de.

Jörg Unterberg studied media technology at the Technical University of Ilmenau, where he focussed on media production and interactive computer graphics. E-Mail: junterberg@gmx.de.

Sophie Jörg joined the Graphics, Vision and Visualisation Group at Trinity College Dublin in October 2006. Her current research interests include virtual human animation, perceptually adaptive graphics and crowd simulation. E-Mail: Sophie.Joerg@cs.tcd.ie.

the facial muscles' contractions. By using the 44 Action Units from Ekman' s coding system, one can assign the movement of the units to a general model, which has the same Action Unit representation, even if the faces are quite different.

Many researchers have also used 3D-scanners to create detailed head and face models and a summary is given in [3].

Another method fits 3D morphable models to an image or a 3D shape [8]. The morphable model is learned from a set of textured 3D scans of heads. The fitting algorithm uses shape and texture information to optimize the coefficients which deform the model to the reference form. Because the morphable model has a generic structure to animate, the animation for this model could be directly transformed to the 3D-shape.

To generate natural and expressive motion, which also includes the subtleties of movement that define a real person, different methods to capture the motions of real humans have been proposed. The principle ones are magnetic, optical without markers and optical with different marker sets [9]. Of these, optical marker-based systems allow the best accuracy to be achieved.

The common practice with optical motion capture systems is to use a set of 40 to 50 separate markers to capture the whole body (not including the fingers). Current commercial tracking systems also rely on this approach [10, 11]. The approximation of the joint angles of a skeleton is then computed from the positional coordinates of these markers. Occlusions and crossovers of the markers are not avoidable for every kind of move and they still remain an issue which results in time consuming manual work. Several techniques have been proposed to overcome these problems, such as using a sophisticated anatomic human model [12], Kalman filters [13] or search space reduction [14]. There have also been approaches where different kinds of markers have been used to avoid confusion [15, 16].

In this project, we adopted a different approach to optical motion capture, by forming marker targets or bodies. One target consists of a minimum of three markers on a rigid body, so that their positions are invariant with respect to each other. This allows us to compute the translation and also the orientation of each target and to differentiate between the markers. This 6-DOF data is then used to calculate the motion of the actor's body. This approach has been used in [17] for a virtual reality interface.

III. PIPELINE

We present the pipeline of work needed to achieve a realistic animatable virtual human based on a real person. There are four main steps in this process:

3.1 Head scanning

The scan system consists of a digital SLR camera and a grid projector. To obtain the information needed to calculate the 3D structure, we take one photo with the grid projected on the person's head. Additional photos without the grid are taken to create the textures for the 3D mesh. This procedure is repeated and photos are taken from different views around the person's head. After a semiautomatic offline process we achieve a high polygon model of the head plus the high resolution textures.

3.2 Creating facial animation

To use this high polygon model for animation it is necessary to simplify the mesh. For this purpose we use a generic model which already has a set of blend shapes defined and adapt the high resolution model to the generic model. With the adapted set of generic blend shapes we then construct a wide variety of facial expressions.

3.3 Modeling the character

Using photos of the real person, taken from different viewpoints, we create a low polygon model of the body. Additional texture photos are used to create the illusion of a complex model. Then a skeleton with a fixed hierarchy is adjusted to fit the model. The head is combined with the skeleton to complete the virtual character.

3.4 Animating with a motion capture system

Using an optical motion capture system with passive retroreflective markers and two CyberGloves, we record sequences of body motions. To accomplish lifelike movements, the real actor must have a figure appropriate to the 3D character. The motions from the real actor are plotted onto the virtual character and the result is visualized in real-time to give feedback to the actor.

IV. HEAD SCANNING

What distinguishes our scan system from others is the fact that it can not only obtain the information necessary to reconstruct 3D structure, but also create high resolution textures directly from photos. The textures are used to store the detailed structural and color information of the surface, which is lost during the simplification process of the model.

Our head reconstruction and texturing pipeline is shown in Fig. 1. It allows us to produce head models with very high resolution skin textures. The textures can be used for improving the appearance of the character, or more precisely as an input to a shader.

To achieve this we have developed a range of tools that help us to calibrate the system, measure the deformation of the grid lines and thus calculate the 3D structure of the geometry, generate high quality texture information from the photos, and finally merge multiple 3D geometries into one solid model with a single texture. The majority of our tools work on 3D range images, but we can also convert between range image and geometry files. Grid and color images can be aligned automatically using SIFT keypoint detection [18], in case they don't match. Some additional tasks can be accomplished with 3D animation software (i.e., Maya [19]), such as the assignment of proper uv-coordinates or modeling by hand.

In our setup, the camera takes three photos: the grid photo and two additional images while the grid is not projected.

4.1 Scanning

Scanning is performed using the ShapeCam from Eyetronics. This apparatus consists of a standard digital SLR camera, a flash with a grid pattern and a lens mounted in front of a frame, to which both are attached. The ShapeCam uses a structured light approach. The grid and the lens mounted in front of the

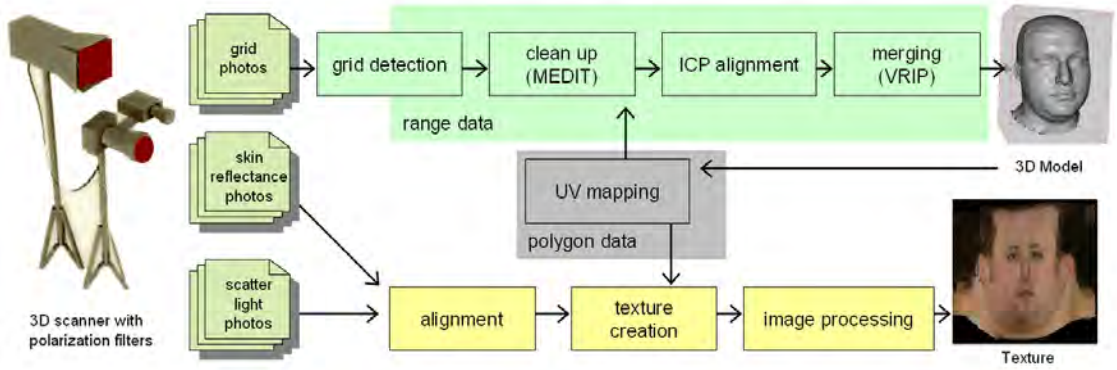

Fig. 1. Reconstruction and texture pipeline.

(a) (b) (c) (d)

Fig. 2. (a) Photo with grid; (b) extracted grid; (c) skin reflectance; (d) scattered light. See Color Plate 1.

flash are used as a projection system for the grid pattern. When the grid is projected onto the model and the camera takes the picture from a slightly different point of view, the projected grid will look deformed. The advantage of the scan system is that it works under both studio and normal lighting conditions. If only depth information is necessary then just one image is required, so the subject only has to remain motionless for the exposure time which is set to 1/60s.

The system works with normal flashes, which have a high light intensity and work well with the grid photo plus texture photo approach. The grid projector does not influence the texture image taken with a second flash, which can be positioned independently to ensure an optimal texture.

The 3D structure is reconstructed using the deformed grid in the grid images. The grid pattern consists of vertical and horizontal lines (see Fig. 2). Using the connectivity of the grid points, a coordinate system can be established that is used to create a correspondence between the light ray leaving the grid flash and the one reaching the camera lens. Thus a 3D reconstruction can be calculated.

This also means that the result of the (automatic) reconstruction largely depends on the quality of the image and the image processing step to extract the grid information. Regions where the surface shadows itself can be particularly problematic. Human faces have few of those regions and thus can be easily and effectively reconstructed.

Some regions still need to be "cleaned". To streamline the process we use our own tool MEDIT (see [20]) that is optimized for the efficient manual correction of grid data. In the worst case scenario it also allows the grid to be drawn directly on the image and is thus able to work even with poor image and grid data.

An image taken from only one position is in general not sufficient to reconstruct a human face. We therefore took multiple images from different viewpoints to cover the whole head. These 3D meshes then had to be aligned and merged to receive one final mesh. The meshes only have to be coarsely positioned. An ICP algorithm [21] is then used to align the meshes. After a first alignment to see if the patches fit, each individual mesh can be further optimized.

The individual meshes are then merged into one mesh using the VRIP algorithm [22], which renders each mesh into a volume, including an error margin set to the error of the reconstruction. Thus a smooth interpolated surface is defined in the volume and extracted using a marching cube algorithm. The resulting mesh contains a large number of (unnecessary) triangles that can be removed using standard decimation tools [23]. A special algorithm (volfill [24]) is used for filling in holes in the 3D structure based on volumetric diffusion. The final 3D geometry output is not a range image anymore, but a regular polygonal mesh.

4.2 Getting texture

After the 3D model is created, we can assign uv-coordinates and create the textures that are needed to recreate the details and color information on the surface. The textures are actually calculated directly from the color photos (the ones without the projected grid). The most challenging regions of the human

face, in terms of shading and lighting, are the eyes, hair and skin. In contrast to hair and eyes, which had to be fully computer-generated, the skin shader would benefit significantly from real skin color textures. The textures are used as color maps or as normal maps. Normal maps are used to perturb the interpolated normal on the low resolution mesh and create the effect of small details during render time. This approach is very common in real-time rendering.

Using filtering techniques in the image acquisition phase, we are able to estimate the necessary properties of the human skin. Skin is made up of multiple layers with very different properties. The outermost section of human skin is the stratum corneum, which is very thin and hence light absorption is very low. The layers below are more complex but the scattering coefficient σs/EPI (epidermis) and σs/DERM (dermis) and the phase function of both layers are approximately the same. Thus, the skin layers can be summarized as being two layers with very strong optical properties: the outer layer with strong surface reflection and minimal absorption and the inner layer with strong haemoglobin and melanin absorption and a high scattering coefficient.

Reflected light from skin contains two different components [25, 26, 27] (Fig. 3). A small amount of the incidental light at all wave length is reflected at the air-skin boundary because of the large change in the refractive index between the air and the stratum corneum. The second component is the backscattered light, which after multiple scattering and absorption, re-emerges from the skin and becomes part of the reflected light.

As in [28, 29] we are using polarized light to separate both reflected components. Light which reflects specularly off the skin will maintain the polarization of the incident light; however, light that emerges from below the surface will have been depolarized by scattering interactions. For the details and the shading we take two texture photos. With the first one we capture the whole skin reflectance. With the second one we capture the scattered light only. The light from the flashlight is polarized. A second filter in front of the camera is used to either let the polarized light pass or to completely block the polarized light from the flash.

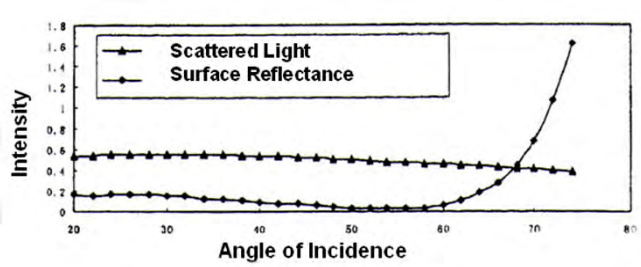

Fig. 3. Skin reflectance.

To consider the specific needs of capturing skin, we modified the ShapeCam and equipped it with a very strong flash unit for the texture photos. The ShapeCam and the flash were mounted on two height adjustable tripods.

Similar to the meshes, the texture images of the individual meshes have to be merged into one texture. Therefore, uv-coordinates are first created for the final mesh. The uv-projection can be done either automatically or by hand. The uv-coordinates will be used to create the resulting texture. The position of the camera relative to each reconstructed mesh as well as the transformation of the reconstructed mesh to the final mesh is known. We can use this projection information to draw each texture image onto the final mesh. Each pixel is weighted using several factors, including a shadowing term and a term fading out towards the edge. Thus a smooth final texture is created. From our input photos we create a scattered light map, a specular reflection map and a normal map.

V. CREATING FACIAL ANIMATION

After we have acquired the high polygon mesh of the scanned person, we have to create an animatable low resolution mesh from it. Through the use of a generic model, we can derive the low polygon model and the blend shapes simultaneously.

5.1 Blend shapes

With our scanning system we also generate high resolution models from real facial expressions. We first acquire a basic model from the 3D head. In the second session we capture the facial expression, six emotions and nine visemes defined in the FACS. For the basic model we need photos from approximately six to eight points of view, and for the morph target from 3-4 points of view (see Fig. 4). Only the components of the face that are affected by the expression must be modeled, everything else will be covered by a mask.

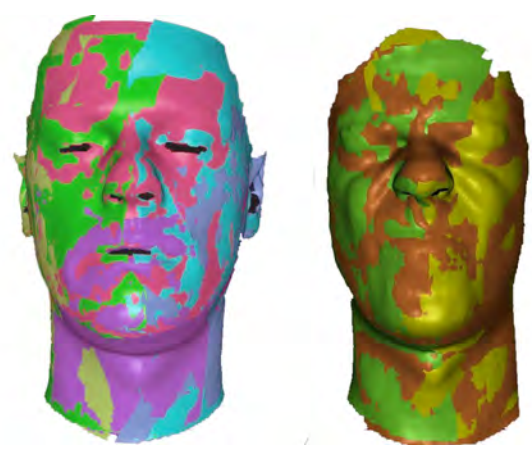

Fig. 4. Two scanned facial expressions. See Color Plate 2.

We use our tools to align the different 3D meshes in global space and apply the same uv-projection onto the model. This creates the reference space we need to apply our optical flow. By selecting the basic model and one of the expressions, we can find corresponding vertices in the two meshes and create a morph target.

A disadvantage of this technique is that the actor has to freeze while holding that expression until all photos can be taken. Therefore, real facial expressions could only be captured if it was possible to catch these faces in real time. Some promising results can be found in [30]. We build up this scanner using the color structured light approach. A pattern of

stripes with a special color sequence is projected onto the person. The edges are detected and aligned with the pattern by dynamic programming and a 3D model is generated. If high frame rates are possible, texture photos can be taken in between.

5.2 Simplify the geometry

To get a low resolution mesh which is optimized for animation, we were inspired by the work of Lee et al. [31], who use the 2D-texture map of the model to fit the generic mesh. This texture map was obtained through a cylindrical projection of the 3D face mesh. The adaption process necessitates the manual positioning of some feature points.

Even though the localization of feature points could be done automatically, there is a need for further manual processing. To facilitate the fitting process of the generic mesh, a tool was written to adjust the generic mesh directly on the texture map. By moving the mesh points over the texture map of the 3D-scan, the mesh of a generic model is matched simultaneously with the high-resolution mesh. This is achieved by copying the vertex-coordinates of the scan to the generic mesh coordinates at the positions of the uv-coordinate given by the texture map (see Fig. 5).

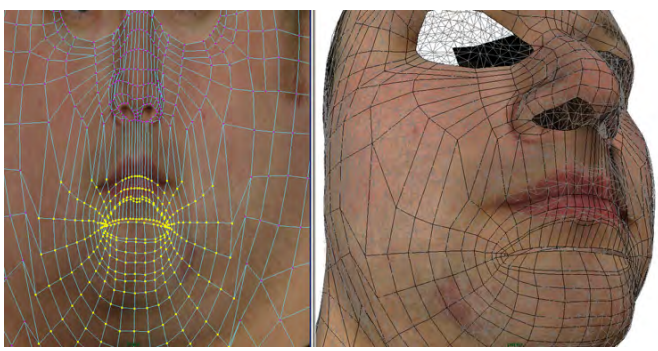

Fig. 5. (a) texture map (b) generic (dark lines) and high poly mesh (grey). To point out the functionality of the tool, the mouth uv-coordinates (light points) are moved downwards, so do the vertices of the generic model, still following the mesh structure of the high poly mesh. See Color Plate 3.

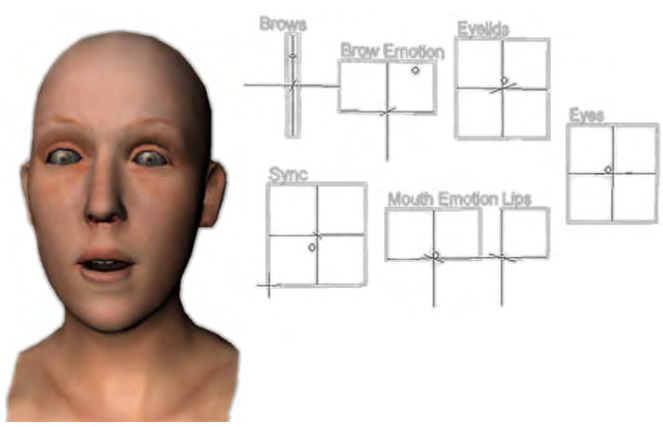

Fig. 6. Face Editor: A blend shape created with the generic defined set. The generic blend shapes are combined with some intuitive controls, like Brows and Mouth Emotion.

Because of the direct manipulation of the output mesh, problematic surface regions could be effectively handled. We have also made attempts to automate this. Here we treat the mesh as a skin with special constraints. Using the same algorithms that are used to simulate dynamic cloth, we fix some points and start the simulation. Afterwards, manual work still has to be done.

Features that could not be captured from the scanner, such as hair or inner regions, have to be handled differently. We take the inner regions from our reference model and fit them to our geometry, thus making it possible to use the blend shapes we have from our generic model. Missing parts that are not morphed are remodeled by hand.

To transfer the generic blend shapes, we take the differences for each vertex between the normal mesh and the morph and perform a weighted mapping to our low poly model. Although the blend shapes are constructed for the generic model, the results look good (Fig. 6). Therefore, we decided not to use the scanned facial expressions to get real blend shapes for the character.

VI. MODELING THE CHARACTER

Because of the requirement to animate in real time and not offline, we needed only a low resolution mesh. With commercial modelling software we built the rest of the character using reference photos of the real person. With additional photos of items of clothing we created the texture maps, thus achieving a model that appears to be quite detailed.

Although we use the same skeleton hierarchy for every model (see Fig. 7), we use two different skeletons for each character. One is optimized for animation, where the joint orientation is towards the next joint. The other is optimized for exporting the animation, so the joints here all have the same global orientation. We use a smooth bind, so that each vertex is influenced by a maximum of three joints. Overall our characters have about 9000 triangles, of which 3500 triangles are for the head.

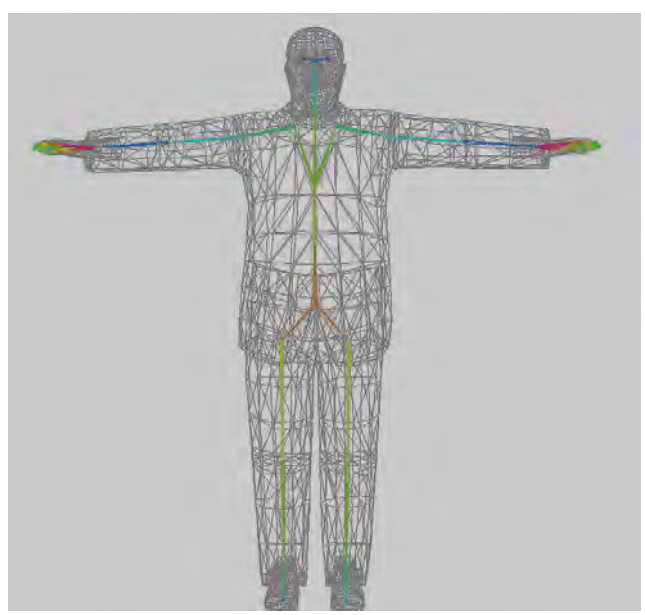

Fig. 7. Skeleton of one character.

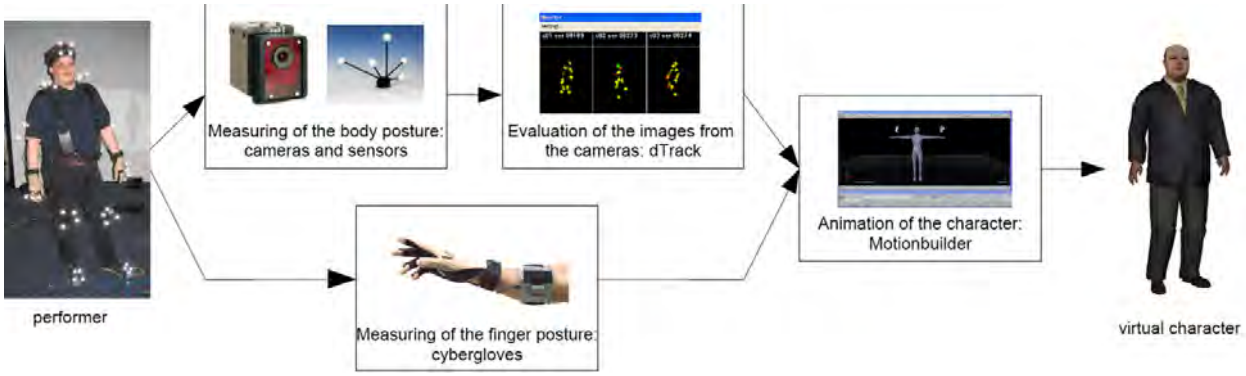

Fig. 8. Overview of the motion capture process.

VII. ANIMATION WITH A MOTION CAPTURE SYSTEM

In the following sections we will describe the capturing of the movements of a real person and the procedure to animate the virtual character using them. This includes constructing the targets, setting up the cameras, capturing the hands, integrating it all into one system with real-time visualization, capturing the motions and finally post processing the animations. Fig. 8 shows an overview of the whole process.

Fig. 9. Camera positions in our Optical Motion Capture System.

7.1 The optical motion capture system

The optical motion capture system consists of 9 infrared A.R.T cameras (880nm): five ARTTrack1 and four ARTTrack2 [32] with a frame rate of 60Hz. They are arranged to cover the 28m^2 capture area (see Fig. 9) and split into in three different synchronization groups to avoid interference between the cameras. We build a motion capture area by using a commercial system from Advanced Realtime Tracking GmbH (A.R.T.), which is common for Virtual and Augmented Reality applications where a very small set of cameras and markers is required, but has rarely been used for full body motion capture.

The A.R.T. Dtrack software recognizes the targets, which have been initialized before, and computes their 3D position and orientation.

7.2 Body optimization

At least three markers are needed to calculate the 3D translation and orientation of an object. We use four retroreflective 12 and 6 mm spherical markers for each target to add accuracy and robustness. The constellation of the markers for each individual target has to be unique and the differences of the distances between markers have to be as high as possible, so that an unambiguous recognition is assured. An overview of how we achieve this can be seen in Fig. 10.

The tracking system uses so called signatures to identify the different bodies from the set of detected markers. A signature S_i of a body M_i is defined as an ordered list of distances between all possible 3D positions of the markers. This list is sorted from large to small distance values and can be used to identify the body. There are $\binom{m}{2}$ nontrivial distances if m is the number of markers in the target. In our case, for targets with 4 markers, 6 different distances can be computed. We have further introduced a metric between the signatures $g(S_i, S_j)$ which compares the different signatures with each other. A simple euclidean metric seems sufficient. The result is a pairwise distance matrix D_{ij} which is symmetric and equal to zero on the diagonal. We used a standard technique for data visualization, namely classical multidimensional scaling (see [33] or [34]) to visualize the pairwise distances in an embedded 2-dimensional space. The marker positions within a target are iteratively modified and visualized until the targets appear most dissimilar. We have experimented with various MDS algorithms but classical, metrical MDS matched best with our observations of target misidentification.

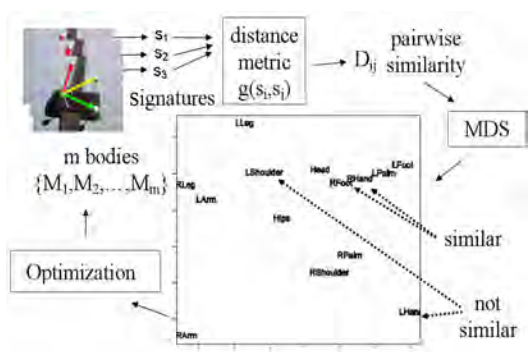

Fig. 10. Illustration of the target optimization using MDS.

Our construction of the targets ensures that there is space between the body of the actor and the retroreflective markers, thus reducing occlusion of the markers. We used a marker configuration with 12 targets common in the field of magnetic

motion capture, which also captures 6-DOF data: head, hips and left and right shoulder, arm, forearm, hand, thigh and foot.

7.3 The hands

Approaches to capturing the motion of hands are also multifaceted. Commercial systems include the DataGlove [35], with up to 16 sensors, and the low-cost P5 Glove available from Alliance Distributors [36], with five sensors. For recording the finger motions we used two CyberGloves from Immersion with 18 bending sensors [37] and the commercial software Motionbuilder from Autodesk. As the fit of the gloves varies over time and precise calibration is very time-consuming, we developed methods to compensate for this effect in post processing. To avoid having to synchronize finger and body motion later on, we integrated the recording of both in a single system that records all data simultaneously.

7.4 Integration and visualization

To integrate all of the components in one system and to generate the real-time visualization, we wrote a plug-in for Motionbuilder which loads Dtrack-data and visualizes it. This enables the capture of body and finger motion at the same time without the need for any later synchronization step. After calibration of the system, the data is used as an input source for the virtual character (consisting of 23 animated joints). The animated character is projected onto the walls, so that the actor receives feedback on his movements in real-time.

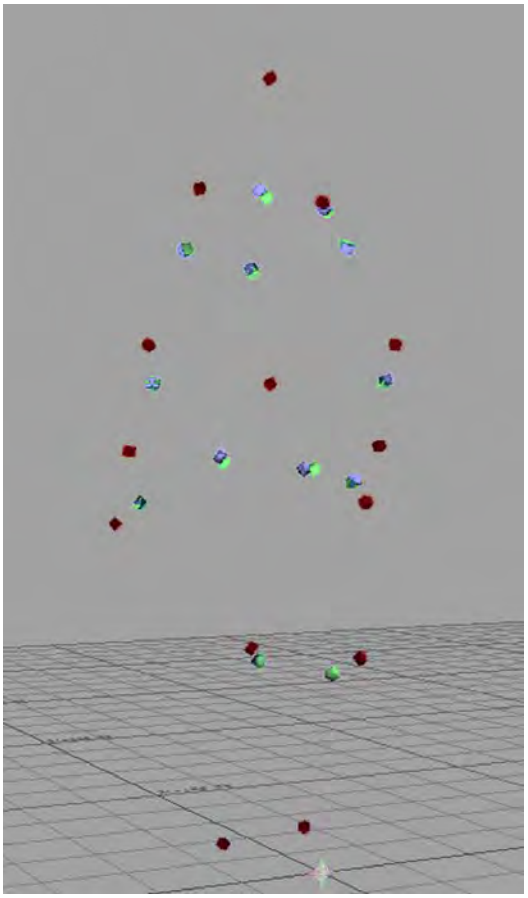

Fig. 11. Estimation of the joints. (dark-motion capture marker, brighter-parent to child, child to parent.

Thanks to our approach that uses rigid structures, the targets are always assigned correctly in real-time. If a marker is lost for a short period, it is not necessary for the actor to return to a T-position as is the standard practice when using commercial systems that do not use this technique.

7.5 Calibration of the body

The calibration of the body was done in Motionbuilder. It roughly consists of the following steps:

- create an actor
- scale and position the whole actor to fit the targets
- scale and orient single joints if necessary
- assign a joint to every target and define the targets as 6-DOF
- start the calibration routine

Now the actor can be used as an input source for any virtual character.

We also used the algorithms described in [38] to fit the actor automatically. We estimate the joint locations by analyzing a range of motions from the capture session. For that we used a predefined skeleton hierarchy, a simple version of our character skeleton.

The translation and rotation of each marker given by our tracking system is used as input. We evaluate the possible joint position from two directions: parent to child and child to parent. Both results differ marginally (see Fig. 11). By connecting the joint locations we also derive the joint length and thus our simplified skeleton. In the future we need to implement the final step, i.e., fitting the Motionbuilder actor to this estimated skeleton.

7.6 The capture session

As motion capture is a technique that includes the subtleties of motion specific to the manner of movement of each person, we selected an actor whose physique corresponds to that of our virtual character. In less than 5 hours we captured more than 135 takes. We spent about 3 hours with preparations and calibration (morning and afternoon).

7.7 Postproduction of the hands

The CyberGloves calibration in Motionbuilder is based on two values for each sensor, i.e., the maximum and the minimum bending of the sensor during finger movements. They are fitted to the maximum and minimum bending of the joint of the model. Values outside of this range do not change the bending of the joint, while values inside are interpolated. The limits of the bending of the sensor can be determined by a quick calibration with 5 hand poses or by assigning values manually.

Even after an extensive calibration of the gloves, the captured results for the finger motions were not satisfactory. Other groups have tried to improve the calibration and the mapping of the cybergloves by taking into account the interferences between the different bending sensors [39]. However, this does not resolve the problem of the gloves shifting on the hands over time. To avoid time-consuming recalibration we developed a method to correct this artifact in postproduction.

Every recorded movement started and ended with the same posture. A standard hand pose was modeled and saved. We correct every key frame to normalize the rotation of the first and last frame of a single take to the standard rotation. The key frames in between are calculated by linear interpolation of the offset.

Additionally we added constraints to the model that restrict the possible rotations to fixed limits. This procedure results in very appealing, realistic-looking motions. Of course it does not result in the exact original joint rotations or finger tip positions.

VIII. CONCLUSION AND OUTLOOK

In this paper, we describe our pipeline to create photorealistic virtual humans step by step. Because the realistic appearance of the human head is important, we use a 3D-scanner to achieve an optimal mesh. Furthermore, with our improved texture creation pipeline we can acquire very high resolution textures. Simplification of the high resolution scan is done with an interactive tool that manipulates a generic animatable mesh onto the surface of the model. Therefore, the blend shapes of the generic mesh could be reused for the creation of facial expressions. Overall we generate with our face editor 19 expressions, thereof 8 emotions. The rest of the character is modeled using pictures of the person, combined with photos of cloth details. A skeleton is attached with smooth binding to the 3D model. Body and hand animations are captured in our optical motion capturing studio and visualized in real time. Altogether we produced 93 animations, which are mainly dialog gestures with highly articulated hand and body motion. We also performed two VICON captures sessions in cooperation with a professional motion capture studio, Metricminds, in order to compare the different approaches with each other. Although to that time the A.R.T. system had a sampling rate of 60 Hz, we achieved also good results from gestures which required a higher sampling, e.g. for instance “clap hands”.

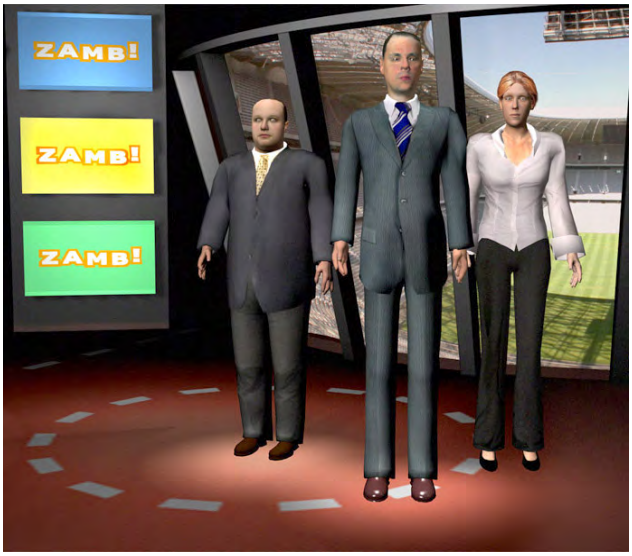

Fig. 12. Our three different characters in our “ZAMB!” sport studio. See Color Plate 4.

For the project we created three different 3D-models using this pipeline, a female and two male characters (see Fig. 12, 13, 14). In order that our project partners could use the characters, we developed some H-Anim/VRML [40] exporters for Maya and for the fbx-format [19]. In several demonstrations of our system, e.g., Cebit 2006, we demonstrated the functionality of our characters.

Fig. 13. Close-up of one male character. See Color Plate 5.

Fig. 14. Our female character.

Creating virtual humans remains a challenging issue, especially when real-time animation must be supported. More articulation in face modeling is needed before highly realistic animation can take place. Algorithms based on free-form

animation, muscle-based animation and tissue simulation are now maturing rapidly and GPU-based acceleration of these algorithms have already started to show impressive results in commercial gaming which were formerly not applicable for real-time animation and rendering. Mesh sizes used the real-time animation have increased considerably during the last years, and GPU-based shading and deformation have made it possible that more and more realistic characters start to appear even in real-time applications. However, manual work from engineers and artists is still required. The technical aspects of synthesis are no longer a problem, what is missing are ways to captures the human body realistically so that animations and shapes can be combined, synthesized and reused cleverly without relying on artists to optimize every vertex position and animation curve afterwards. We have shown in this article how realistic virtual humans can be created with moderate afford and investigated places where manual inspection and optimization can be avoided by using more intelligent algorithms and sophisticated hardware. Nevertheless, despite our progress, reconstructing the human body and capturing simultaneously full-boy motion with highly complex hand and facial articulation remains a challenge.

Our future work will focus on getting rid of some limitations which still hinders us to produce virtual humans effortlessly: In motion capturing, our actor was still prevented to move freely since the data gloves are still connected though wires to the capture PC. Despite the fact that this limitation could be lifted by using Immersion's wireless CyberGlove-II, the general need to (re)calibrate the data gloves continuously remains problematic. Some research has already been initiated in our group in collaboration with Advanced Realtime Tracking by performing active optical tracking of human hands simultaneously with passive 6DoF-full-body capture. First results are promising. Realistic facial modeling and animation is another field of very active research in which more statistical modeling may help to overcome the problem of blending, motion editing and the syntheses of realistic facial expression.

REFERENCES

[1] D. Thalmann and N. Magnenat-Thalmann. Handbook of Virtual Humans, *John Wiley & Sons*, 2004.

[2] J. Osipa. Stop Staring. Facial Modeling and Animation Done Right, Sybex Inc., London, 2003.

[3] F. Pighin and J. P. Lewis. Introduction, in ACM *SIGGRAPH 2005 Courses (Los Angeles, California, July 31 - August 04, 2005). J. Fujii, Ed. SIGGRAPH '05. ACM Press*, New York, NY, 2005.

[4] C. Bregler, M. Covell and M. Slaney. Video Rewrite: Driving Visual Speech with Audio, in Proceedings of the *24th Annual Conference on Computer Graphics and interactive Techniques International Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, pp.353-360, 1997.

[5] F. I. Parke. Computer Generated Animation of Faces, In Proceedings of the *ACM Annual Conference* - vol. 1 (Boston, Massachusetts, United States, August 01 - 01, 1972), ACM'72. ACM Press, New York, NY, pp. 451-457, 1972.

[6] M. Rahman. Dynamic Interesting Region Searching and Real Time Image Matching Between Different View Angle Image Sequences of an Object, *University of Applied Sciences*, Bonn, 2004.

[7] P. Ekman and W. Friesen. Facial Action Coding System, *Consulting Psychologists Press*, Palo Alto, CA. 1978.

[8] V. Blanz and T. Vetter. Face Recognition Based on Fitting a 3D Morphable Model, *IEEE Trans. Pattern Anal. Mach. Intell.* vol. 25, no. 9, September 2003, pp. 1063-1074, 2003.

[9] A. Menache. Understanding Motion Capture for Computer Animation and Video Games, *Morgan Kaufman*, 2000.

[10] NDI, http://www.ndigital.com/.

[11] VICON, http://www.vicon.com, 2007.

[12] L. Herda, P. Fua, R. Plänkers, R. Boulic and D. Thalmann. Skeletonbased Motion Capture for Robust Reconstruction of Human Motion, in *Proceedings of the Computer Animation*, IEEE, 2000.

[13] K. Dorfmüller-Ulhaas. Robust Optical User Motion Tracking Using A Kalman Filter, Technical Report of Augsburg University, Tech. Rep., 2003.

[14] R. van. Liere and A. van. Rhijn. Search Space Reduction in Optical Tracking, in *Eurographics Workshop on Virtual Environments*, 2003.

[15] PhoeniX Technologies Incorporated (PTI), http://www.ptiphoenix.com, 2007.

[16] A. Sementille, L. Lourenco, J. Brega and I. Rodello. A Motion Capture System Using Passive Markers, in Proceedings of the *2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, 2004.

[17] A. Hornung, S. Sar-Dessai and L. Kobbelt. Self-Calibrating Optical Motion Tracking for Articulated Bodies, in *IEEE Virtual Reality 2005*, Bonn, Germany, March 12-16, 2005.

[18] D. Lowe. Distinctive Image Features from Scale-invariant Key Points, *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[19] Autodesk, http://www.autodesk.com, 2007.

[20] M. Bogen and R. Kuck. Reconstructing and Presenting Bernini's Borghese Sculptures, in *J. Trant and D. Bearman (eds.). Museums and the Web 2005: Proceedings*, Toronto: Archives & Museum Informatics, published March 31, 2005.

[21] P. Besl and H. McKay. A method for registration of 3D Shapes, *SIGGRAPH*, IEEE Transactions on vol. 14, no. 2, pp. 239-256, February 1992.

[22] B. Curless and M. Levoy. A Volumetric Method for Building Complex Models from Range Images, *ACM Siggraph1996*, pp. 303-312.

[23] M. Garland and P. Heckbert. Surface Simplification Using Quadric Error Metrics, *Computer Graphics (SIGGRAPH '97 Proceedings)*, 1997.

[24] J. Davis, S. Marschner, M. Garr and M. Levoy. Filling Holes in Complex Surfaces Using Volumetric Diffusion, *First International Symposium on 3D Data Processing*, Visualization and Transmission, Padua, Italy, June 19-21, 2002.

[25] P. Hanrahan and W. Krueger. Reflection from Layered Surfaces Due to Subsurface Scattering, in *Computer Graphics (SIGGRAPH '93 Proceedings) (Aug. 1993)*, *ACM SIGGRAPH*, J. T. Kajiya, Ed., vol. 27, pp. 165-174.

[26] A. Krishnaswamy and G. V. G. Baranoski. A Study on Skin Optics, Technical Report CS-2004-01. School of Computer Science, University of Waterloo. Canada, January 2004.

[27] M. J. C. van Gemert, S. L. Jacques, H. J. C. M. Sterenborg and W. M. Star. Skin Optics, *IEEE Transactions in Biomedical Engineering*, no. 36, pp. 1146-1154, 1989.

[28] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin and M. Sagar. Acquiring the Reflectance Field of A Human Face, *Computer Graphics (SIGGRAPH)*, 2000.

[29] Y. Su, W. Wang, K. Xu and C. Jiang. The Optical Properties of Skin, Proceedings of the *SPIE-The International Society for Optical Engineering*, vol. 4916, pp. 299-304, September, 2002.

[30] L. Zhang, B. Curless and S. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming, The *1st IEEE International Symposium on 3D Data Processing*, Visualization, and Transmission, Pedova Italy, pp.24-36, June 2002.

[31] Y. Lee, D. Terzopoulos and K. Walters. Realistic Modeling for Facial Animation, in proceedings of the *22nd Annual Conference on Computer Graphics and interactive Techniques S. G. Mair and R. Cook*, Eds. SIGGRAPH '95. ACM Press, New York, NY, pp. 55-62, 1995.

[32] A.R.T., http://www.ar-tracking.de, 2007.

[33] I. Borg and P. Groenen. Modern Multdimensional Scaling: Theory and Applications, Springer, New York, 2005.

[34] R. Duda, P. Hart and D. Stork. Pattern Classification, John Wiley and Sons, 2nd edition, 2001, chapter 10.14, pp.573-575.

[35] Fifth Dimension Technologies, http://www.5dt.com/hardware.html#glove, 2007.
[36] Alliance Distributors, http://www.alliancedistributors.com/AllianceBrand/Products.php, 2007.
[37] Immersion, http://www.immersion.com/, 2007.
[38] J. F. O'Brien, J. Robert, E. Bodenheimer, G. J. Brostow and J. K. Hodgins. Automatic Joint Parameter Estimation from Magnetic Motion Captures Data, in Proceedings of *Graphics Interface 2000*, 2000.
[39] F. Kahlesz, G. Zachmann and R. Klein. Visual-Fidelity Data Glove Calibration, in *Computer Graphics International (CGI)*, Crete, Greece: *IEEE Computer Society Press*, June 16-19, 2004.
[40] H-Anim-Humanoid Animation, http://www.h-anim.org.

Frank Hülsken is a mathematician with background in pattern recognition. He is working as a research fellow at the Fraunhofer Institute for Intelligent Analysis and Information Systems. His research interests include speech recognition, image processing, 3D reconstruction and animation.

Christian Eckes received the diploma degree in Physics from the University of Dortmund in 1995. From 1996 to 2000 he worked as a research assistant at the Institute for Neural Computation at the Ruhr-University Bochum and, as a visiting scholar, at the University of Southern California (USC), Los Angeles, U.S.A.. In 2001 he joined the Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS) in the NetMedia group as a research fellow and project manager working in the area of multimedia analysis. His research interests include human-computer interaction, multimedia analysis, pattern recognition and biological inspired computer vision.

Roland Kuck is a computer scientist with a background in computer graphics. He is working as a research fellow at the Fraunhofer Institute for Intelligent Analysis and Information Systems. His research interests include visualization of large meshes, global illumination and shading.

Jörg Unterberg studied media technology at the Technical University of Ilmenau, where he focussed on media production and interactive computer graphics. He completed his diploma thesis in close collaboration with the Fraunhofer Institute for Media Communication (IMK) and was participating in the Project Virtual Human since then. In October 2005 he took on a postgraduate course Animation Technical Director at the Institute of Animation, Film Academy Baden-Württemberg and in 2006 he received a Karl-Steinbuch Stipendium.

Sophie Jörg joined the Graphics, Vision and Visualisation Group at Trinity College Dublin in October 2006, where she is pursuing a PhD in the field of real-time character animation. She completed her diploma in media technology in 2005 in collaboration with the Fraunhofer Institute for Media Communication. After this, she continued to work on the Virtual Human project for one year. Her current research interests include virtual human animation, perceptually adaptive graphics and crowd simulation.