View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Loughborough University Institutional Repository

A Tutorial on Motion Capture Driven Character Animation

Ben Bradwell and Baihua Li
Department of Computing and Mathematics, Manchester Metropolitan University, UK
bradwell24@gmail.com, b.li@mmu.ac.uk

ABSTRACT

Motion capture (MoCap) is an increasingly important technique to create realistic human motion for animation.
However MoCap data are noisy, the resulting animation is often inaccurate and unrealistic without elaborate manual processing of the data. In this paper, we will discuss practical issues for MoCap driven character animation, particularly when using commercial toolkits. We highlight open topics in this field for future research. MoCap animations created in this project will be demonstrated at the conference.

KEY WORDS: Marker-based motion capture, character animation, facial animation.

1. Introduction

Humans are very acute at recognising plausible motion. This makes the task of generating aesthetically pleasing, human-like character animation a challenging problem. Though manually key-framing continues to play an essential and complementary role in real-time animation, realism and productivity issues of key-framing are becoming critical as the complexity and volume of content increases in the fields of film, television and computer games.

Motion capture (MoCap) is a relatively new technique widely used for biomedical studies, sports analysis and entertainment industry [1, 2, 3]. MoCap is the process of measuring the movement of a performer (typically, but not always, human) in real-time and interpreting the information in a compact and computer-usable manner. Ideally, the capture of motion data should be easily available, inexpensive, and non-invasive. Various MoCap techniques have been developed, such as mechanical trackers, electro-mechanical trackers and electromagnetic systems [2]. Due to the problems of intrusive sensors, limited workspace, low accuracy and sampling rate, application of these systems has been largely restricted. The state-of-the-art technology is video-based optical MoCap, which detects motion from standard video cameras using advanced image processing and computer vision techniques [3]. In this category, markerless MoCap facilitates wide application prospect and has drawn great attention in the last decade. Though much progress has been made in this area, because of the inherent challenges of image ambiguity and noise, markerless un-encumbrance MoCap is still in its infancy in many aspects [3, 4, 5].

The most reliable and commercially available systems are marker-based MoCap [6, 7], such as Vicon [8] and Motion Analysis [9]. In a marker-based system, to reduce the difficulty of image processing, retro-reflective markers are attached on a subject at tightly clothed key sites, such as joints and limbs, as shown in Fig.1. During MoCap, the movements of performers are recorded by multiple cameras in real-time with an unrestricted background. 3D marker trajectories are extracted from the high contrast images. Various motion parameters can be derived from the concise marker data and used for motion analysis, animation, etc. The measurement accuracy of such a system is at a level of a few millimeters in a control volume spanning meters in linear extent. The application prospect of marker-based MoCap has largely motivated many excellent work in related fields, such as motion warping, blending, retargeting, mapping, synthesizing, noise filtering, motion representation and data compression [3, 6, 7, 10, 11, 12].

Fig.1 Maker-based full body MoCap

MoCap can be highly productive to generate realistic and complex character animations. However, raw MoCap data are very noisy [6, 7]. Though, significant progress has been made and many toolkits for MoCap handling have been developed, the resulting animation often contains artifacts and requires time consuming elaborate handwork of highly skilled animators. In this tutorial, we present some practical issues on creating character animation using marker-based MoCap data. We also identify open problems in this field for future research.

2. Character Animation

2.1 Character modeling

Low number polygon models, composed by triangles or quadrilaterals, are widely adopted because of the good support for efficient rendering by current graphics hardware. Meshes can be generated interactively though a graphical interface using various primitives and modifiers, such as extrusion, symmetry, tweaking, subdivision (NURMS in 3ds Max), and transformations. Joint meshes are special segments for attaching joints to an underlying skeleton. They also prevent mesh collision caused by deformation of adjacent segments.

2.2. MoCap data pre-processing

MoCap marker data are usually noisy [6, 7]. Missing data can be caused by occlusion. "Ghost" points may be involved from the workspace or caused by image analysis algorithms due to imaging ambiguity. The data point are unlabeled from 3D reconstruction. To infer the underlying physical structure, an identification procedure is often required to know which point in massive MoCap data corresponds to which marker attached to the subject. Even with the most sophisticated system [8, 9], auto-labeling may fail for complex movements [10]. In addition, motion blur and measurement error can generate jagged marker trajectories resulting in wobbly motion in animation. It is cumbersome to manipulate and very difficult to alter noisy MoCap data, especially when the key "essence" of the motion is not distinguished from the large amount of potentially irrelevant details and noise. In practice, we need to re-capture a movement until an acceptable quality is achieved. For the MoCap data used in this project, identification errors were corrected manually or using the algorithm developed in [10]. Missing data gaps were filled by interpolation or splines; marker trajectories were smoothed by various low-pass filters.

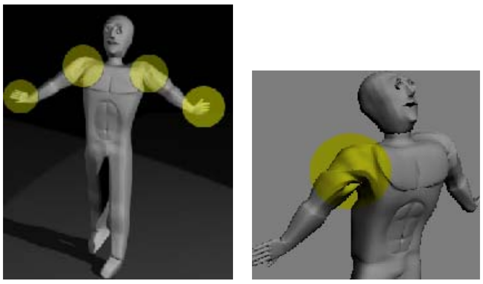

Fig. 2 Distorted mesh and gestures with CSM

Common MoCap data formats include CSM, BVH and BIP. The CSM file stores frame-by-frame 3D coordinates of identified body markers in world coordinate system. The BHV format contains hierarchy and motion data. The hierarchy section provides information on joint-to-joint connections and offsets of child-segments from their parent. The motion section describes the movements of these joints. These two formats require rescaling the rigging skeleton to conform with absolute marker or joint offset data obtained from real MoCap. Though rescaling can be done with the aid of "Talent Figure Mode" in the "Motion Capture" rollout of 3ds Max, we found it was very time consuming and could generate unexpected mesh distortion in rigging process, as shown in Fig. 2. To solve this problem, the BIP format was adopted. The BIP file stores limb rotation data which are less dependent to physical body measurement.

To preserve the fidelity of human movement in animation, MoCap data are usually captured at 60~120Hz. However, the animation generated from MoCap on a frame-by-frame basis, at a high sampling-rate, can cause very slow performance and may be unnecessary in applications. An alternative way is to reduce the MoCap data to sparse key-frames. We found this method makes the animation compact, effective, and a lot easier to edit and personalize. In 3ds Max, the "Key Reduction" in the Motion Capture Conversion Parameters menu can be used to detect MoCap keys based on the motion intensity and interpolate the in-betweens. However, the key reduction may cause artifacts like jittering movements and "foot-slide". Jittering movements can be improved by tweaking (see Section 2.4). Foot-slide can be solved via "Footstep Extraction". Footstep Extraction extrapolates footsteps so that keys that make up the foot placement and movement will be turned into footsteps to maintain correct foot-toe-ground contact.

2.3 MoCap rigging

To animate the character mesh, a complete bone hierarchy, or "skeleton", is required with full simulation of Inverse Kinematics. Most commercial animation tools have pre-built skeletons or third party plug-ins, such as Biped in 3ds Max.

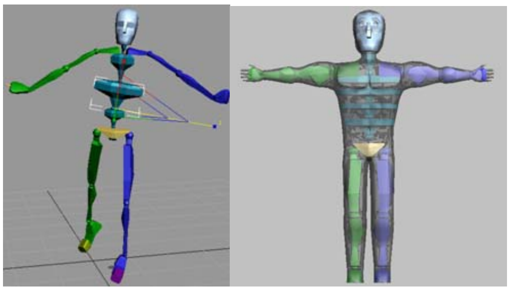

Fig. 3 Biped skeleton and posed Biped in character mesh.

The first stages of rigging are to correctly position, scale and rotate the bones/joints of the skeleton within the character mesh, as shown in Fig. 3. Subsequently, the skeleton should be linked to the polygon model, known as rigging. Rigging starts from a root node, e.g. the pelvis, and spans out through all the children in the hierarchy. In 3ds Max, rigging will generate a series of structural vertex-bone links and "Envelopes", as shown in Fig. 4. Rigging can be achieved manually in 3ds Max using two modifiers: "Skin" and "Physique". The Skin modifier provides many advanced features to aid complex rigging, such as Manually Vertex Weighting and good flexibility over "Envelope" control. The main drawback of the Skin modifier is the lack of deformation functions of its rigging

Envelopes, therefore it cannot provide realistic mesh deformations around joints. The Physique modifier is designed particularly for rigging the Biped to a humanoid character. It allows deformation models to be applied to the rigging Envelopes.

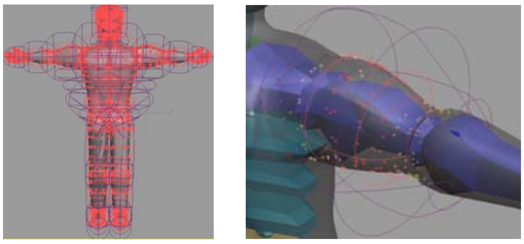

Fig. 4 Rigging links and Envelopes

Envelopes are used to define the region of motion influence of a link in the skeleton hierarchy and control deformation styles. Each envelope is comprised of an inner and an outer bound (ring). Vertices within the inner bound are assigned a weight value of 1, indicating the strongest influence. Vertices between the inner and outer bounds are assigned a fall-off value between 1~0, linearly or non-linearly related to the distance of the vertex from the inner bound. An example of a Biped's upper arm envelope created by Physique is shown in Fig.4 right. After rigging, MoCap data in the BIP format can be loaded and applied to the character using the "Motion Capture" rollout.

2.4 MoCap Tweaking

It is often necessary to correct errors and refine the original MoCap animation, such as smoothing jittering movements or correcting unrealistic limb rotations as shown in Fig.2. It may also be necessary to fine turn the blending MoCap clips, or modify the MoCap to allow the avatar to run along a wall perpendicular to the ground for example. All these can be achieved by MoCap tweaking that creates new animation layers over the original MoCap layer. Novel motion can be configured separately at each layer, and all animation layers can be stored in a BIP file for future editing or reuse in other trials. On completion of the MoCap tweaking, all layers can be merged together to create desired effects.

3. Facial animation

Character animation concerning emotion, style and intent can be enhanced using facial animation. Representative methods of facial animation include key-frame based wrapping and morphing, parameterized models, physics models, performance-based methods, and more recently image-based techniques, see reviews in [13, 14]. In this tutorial, morphing method was used.

3.1 Face modeling form range data

A neutral face scan was obtained using a structure-light based range scanner manufactured by InSpeck [15], as shown in Fig. 5 left. The scan data contain measurement errors and holes caused by light reflections during the scan process. The data were filtered and some small holes were fixed using software tools embedded in the InSpect system. The distorted vertices were fixed manually with the aid of "Soft Selection" in 3ds Max. The original scan was also very large, almost 50,000 polygons. This amount of detail is unmanageable for real-time facial animation. We used the "MultiRes" modifier in 3ds Max to reduce the polygon count to about 2200 as shown in Fig. 5 right.

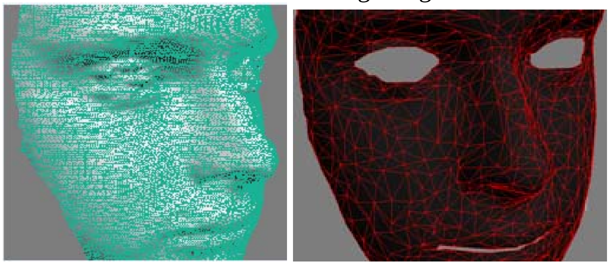

Fig. 5 Noisy dense range scan (left) and clean-up low-polygon mesh model (right).

The face texture was captured by a synchronized CCD camera with the scan. The InSpect EM software was then used to UV map the texture to the mesh. The texture coordinate information can be correctly preserved with the use of the MultiRes modifier to reduce polygon count.

3.2 Facial rigging

To animate basic movements for head, jaw and eyelid, a facial bone rig was created using "Bones System" in 3ds Max, as shown in Fig.6.

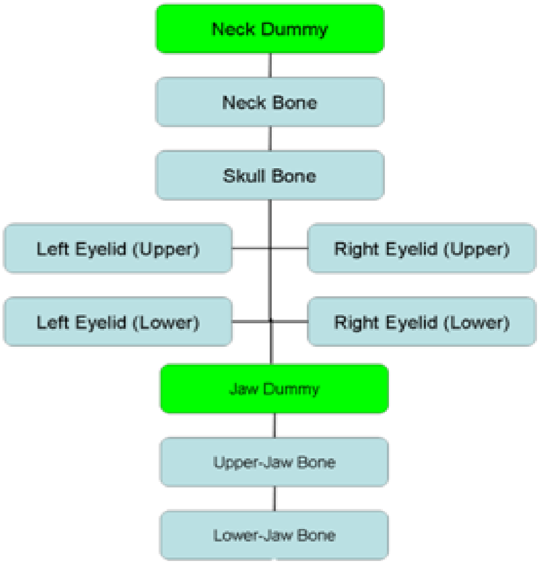

Fig. 6 Facial bone hierarchy.

To rotate the bones, various controllers can be created using circles, splines etc. The controllers are linked to the bones via orientation constraints. Fig. 7 shows the skull (red), neck (black), upper jaw (purple) low jaw (yellow) and eyelid (grey) bones, and circle controllers used to control the rotation of these bones.

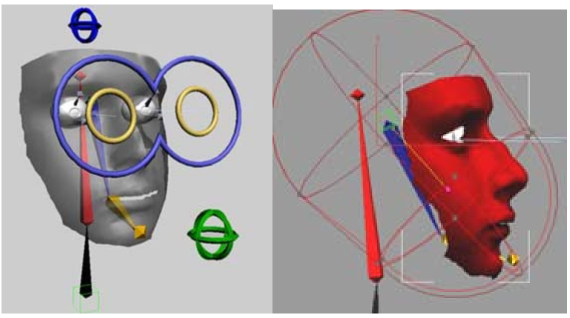

Fig. 7 Bones and controllers

The facial bone rigging was done using the 'Skin' modifier due to its advanced feature of Manual Vertex Weighting. Firstly Envelopes were used to create a rough outline of the vertices that will be influenced by the bones. The rigging was then refined by manually assigning influence weights to the vertices in the mesh. Fig. 8 shows the region of the mesh influenced by the jaw bone. Areas shaded in varying shades of red/orange/yellow indicate decreasing influence, and areas shaded in blue indicate the fall-off and weak influence.

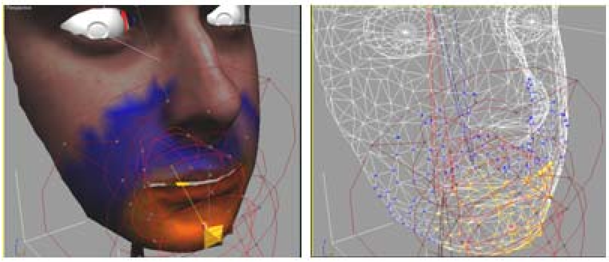

Fig. 8 Weighted rigging of the lower-jaw bone.

3.3 Morphing

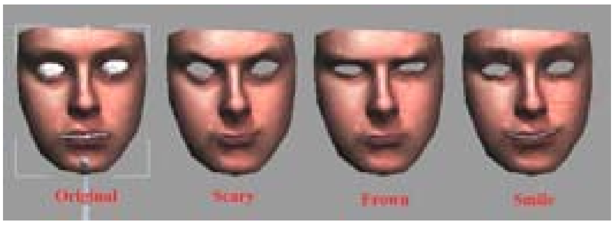

Fig.9 Neutral face with three morph targets.

To animate facial expressions, morphing technique was employed to transfer geometry between morph targets. Ideally morph targets of expressions should be created from scan data, and the scan data must be meshed using the same structure. There is still a lack of software tools for automatic and precise mesh fitting. In this case, the "Soft Selection" tool was used to modify vertices in the scanned neutral face to create scary, frown and smile targets, as shown in Fig. 9. Then with the aid of the 'Morpher' tool, we blended multiple morphing targets to create novel faces.

We attempted to use morphing to animate blinking eyelids. However it caused serious mesh distortion. Therefore, we used bones and the Skin modifier to animate eyelids, which produced a much better result.

4. Data driven character animation

Maya, 3ds Max, Blender, Autodesk and MotionBuilder are popular modeling and animation applications with integrated MoCap toolkits. We chose 3ds Max for this project because of its comprehensive industrial standard functions for low polygon modeling, handy character rigging and animation layering without using plug-ins or scripts. Maya has similar functions to 3ds Max. However its "Full Body IK System" does not have as many easy to use features as the "Biped" in 3ds Max. Blender integrates a number of advanced deformation tools good for realistic organic modeling. However it is a high-polygon modeler. It is possible to use MoCap data within Blender, but a large amount of scripting is required since there are no built-in tools for MoCap data handling. Autodesk MotionBuilder PLE is a freeware featured on MoCap. It could be a good choice for home learners.

In the past, MoCap data was limited to studio access. Nowadays free MoCap databases are available to the public [16, 17]. Some of the MoCap data used in this project were captured from a Vicon system installed at (omitted for blind review). Some are obtained from freely available MoCap databases [16]. We have applied a wide range of MoCap data to a number of avatar models. Some examples are present below.

4.1 MoCap animation 1: Humanoid Robot

The intention of this animation was to evaluate: 1) how MoCap data can be used on a mechanical character with limb proportions completely diffident from those of the MoCap performers. 2) how MoCap clips, captured at different location/orientation and from different subjects, can be blended seamlessly.

The humanoid robot is shown in Fig. 10. Physique modifier was used for rigging. Several MoCap clips, e.g. walking, running, jumping and head scratching, were downloaded from [16]. MoCap blending was achieved in 3ds Max using "Motion Flow Mode". MoCap clips were linked with specified transitions and/or interpolated in-betweens. These transitions were saved as scripts for reusing in other animations. When MoCap data was applied, by default the character was positioned at the origin. We must reset a start location and rotation, especially if scene objects have already been created or MoCap clips need to be blended at different locations. To avoid un-characteristic deformations of the robot, we need to carefully refine the rigging process.

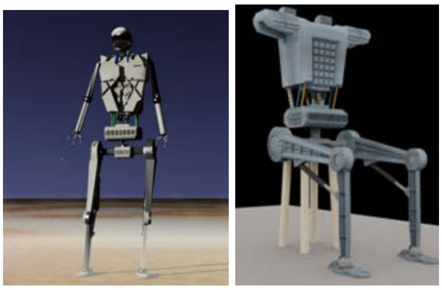

Fig. 10 Rendered still robots from animation.

4.2 MoCap animation 2: Male Dancer

The intention of this animation was to: 1) animate a downloaded, pre-rigged mesh, 2) modify the mesh with procedural hair and shading, and 3) create facial expressions using key-framing official bones.

Modeling: The character mesh used in this animation was freely downloaded [18], as shown in Fig.11 left. It was fully rigged with a Biped and additional bones to control facial movements. Modifications were performed including creation of procedural hair and using shaders for the character's skin and clothing, as shown in Fig.11 right.

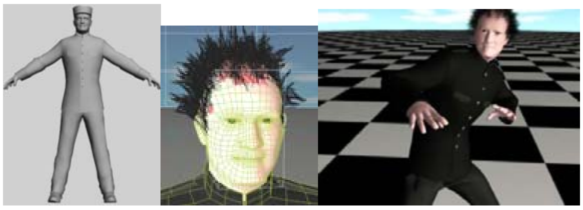

Fig.11 Original model (left), procedural hair (middle) and Male Dancer (right)

Body and facial animation: The animation was created using the combination of MoCap, keyframe and procedural animation techniques. Two MoCap clips were blended together to animate the character walking and then dancing. Multi-layer tweaking was applied to fix incorrect limb rotations and to prevent limbs from passing through one another. Keyframing of facial bones was used to generate various face gestures. Expression animation was created via morphing.

Dynamic hair: A procedural modifier: "Hair and Fur" was used to produce realistic looking hair, and to simulate hair dynamics caused by gravity and body movement. However, it was found that the use of procedural modeling is computationally expensive and extremely slow for rendering, about 11 hours on a 3.2 GHz Pentium Core2Duo with 2 GB of RAM, and an Nvidia 8800 series graphics accelerator card.

4.3 MoCap animation 3: Female Dancer

The aim for this animation was to: 1) rig a downloaded mesh using the Skin modifier, instead of Physique, 2) create hair animation without using procedural techniques, and 3) simulate non-rigid motion for cloth.

Modeling: The character mesh used in this animation was also freely downloaded [19]. We rigged the mesh with Biped for MoCap animation. We added hair and cloth features to the mesh model. To speed up rendering, the hair and dress were modeled using polygons rather than procedural methods. The hair geometry was created from several small mesh segments, attached together to produce a layering effect, as shown in Fig 12 top. Transparency mapping [20] was then used to merge layered textures of diffuse, alpha and specular to produce a realistic look for the hair, as shown in Fig 12 bottom.

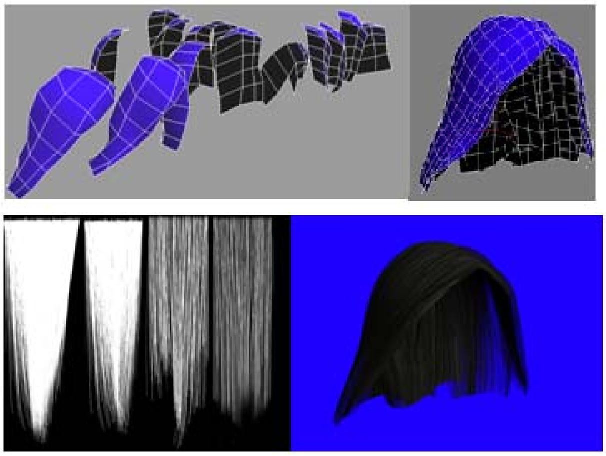

Fig. 12 Polygon hair (top) and fully transparency mapped hair mesh (bottom)

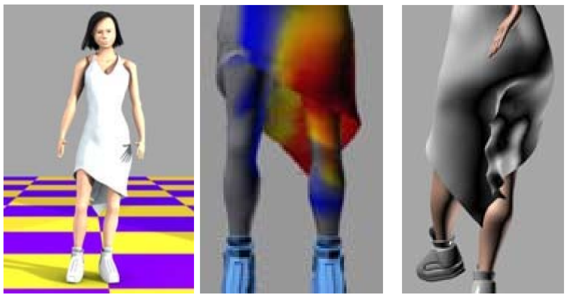

Fig. 13 Rendered character in motion (left), dress rigging (middle) and collapsing dress (right)

Dress dynamics: Cloth dynamics was generated by rigging the dress to the thigh bones of the Biped using the Skin modifier and Manually Vertex Weighting feature of the Skin modifier to provide more control over individual vertices. Rigging of the lower dress to the left thigh is shown in Fig.13 middle. However the dress mesh may incorrectly collapse into itself due to fast and close leg movements, as shown in Fig.13 right. To solve this problem, we rigged parts of the dress to the Biped's spine and calf bones with small weightings. A smooth, fluid motion simulation was achieved. The dress moves realistically corresponding to character's legs without using complex deformation modifiers. However, at some points in the animation, the character's legs still pass through the dress due to ineffective collision detection, which would require further fine turning to rectify.

Body motion: The animation was created by blending walking and dance MoCap clips, similar to the Male Dancer. Keyframing and tweaking techniques were applied to the MoCap trials to remove MoCap errors and to incorporate some personalized features. A rendered image from the animation is shown in Fig. 13 left.

Another important topic of MoCap driven animation is lip-syncing facial animation, which has not been addressed in this project. We will work on this in the future.

4.4 Environmental modeling

The ground was modeled using simple flat polygons and textured with a diffuse map. In 'Humanoid Robot' trial differed from the other two, a fog atmospheric effect was generated on the surface of the ground plane. This was achieved using a thin layer of mist, which both adds depth to the surface and compliments the rest of the scene. The sky was constructed from a hollow sphere and then UVW mapped with a sky. The lighting effect was provided from an omni-directional light source. To make the light appear more like a sun, a lens flare effect was applied to the camera.

5. Conclusion

MoCap has been proven an effective way to generate expressive character animation. MoCap technique has been widely recognized by animators. Handy toolkits for MoCap handling have been integrated within animation software by the developers. However, many issues still exist. Currently, most MoCap systems are very expensive and are not affordable for average users. MoCap data standardization (e.g. marker attachment protocol, format) and usability need to be addressed. Comprehensive MoCap databases including a wide range of motion categories and configurations are still very limited. Motion synthesis from a restricted number of MoCap samples is highly expected to extend applications of MoCap. Great strides on the development of new and improved methods for capturing and processing MoCap data, motion retargeting, tweaking, deformation handling and real-time rendering, are demanded to make MoCap a more viable tool for animation production. In the future,

Acknowledgements

Some MoCap data used in this tutorial were captured by a optical MoCap system — Vicon 512, installed at the Dept. of Computer Science, Univ. of Wales, Aberystwyth.

References

[1] C. Bregler, Motion capture technology for entertainment, *IEEE Signal Processing Magazine, 24*(6), 2007, 158-160.

[2] D. Sturman, A brief history of motion capture for computer character animation, *SIGGRAPH 94: Course 9*.

[3] M. Gleicher & N. Ferrier, Evaluating video-based motion capture, *Proc. of Computer Animation*, 2002, 75-80.

[4] T. Moeslund, A. Hilton & V. Kruger, A survey of advances in vision-based human motion capture and analysis, *Computer Vision and Image Understanding, 104*(2), 2006, 90-126.

[5] L. Wang, W. Hu & T. Tan, Recent developments in human motion analysis, *Pattern Recognition, 36*(3), 2003, 585-601.

[6] J. Richards, The measurement of human motion: a comparison of commercially available systems, *Human Movement Science, 18* (5), 1999, 589–602.

[7] D. Washburn. The quest for pure motion capture. *Game Developer*, *8*(12), 2001, 24–31.

[8] Vicon Motion System, *www.vicon.com*.

[9] Motion Analysis Corporation, *www.motionanalysis.com.*

[10] B. Li, Q. Meng & H. Holstein. Articulated pose identification with sparse point features. *IEEE Trans. Systems, Man and Cybernetics, 34*(3), 2004, 1412-1422.

[11] A. Kirk, J. O'Brien & D. Forsyth. Skeletal parameter estimation from optical motion capture data. *IEEE Conf. on Computer Vision and Pattern Recognition*, 2005, 782-788.

[12] H. Shin, J. Lee, M. Gleicher & S. Shin. Computer puppetry: an importance-based approach. *ACM Transactions on Graphics, 20*(2), 2001, 67-94.

[13] J. Noh & U. Neumann. A survey of facial modelling and animation techniques. Technical report, Univ. of Southern California, 1998.

[14] Z. Deng & J. Noh, Computer Facial Animation: A Survey, *Data-Driven 3D Facial Animation*, Springer London, 2007

[15] Inspeck Inc, http://www.inspeck.com

[16] Universidade Lusófona MovLab, http://movlab.ulusofona.pt/FilesMocap.php?categoria=1

[17] CMU Graphics Lab Motion Capture Database, *mocap.cs.cmu.edu.*

[18] Highend3d, "Character Rig With Facials and Morphs",http://www.highend3d.com/3dsmax/downloads/character_rigs.

[19] 3dtotal, http://www.3dtotal.com

[20] Ordix Interactive Tutorial, Transparency Mapping, www.ordix.com/pfolio/tutorial/hair/transpmap01.ht