BBC

R&D White Paper

WHP 045

September 2002

Real-time production and delivery of 3D media

M. Price[1], J. Chandaria[1], O. Grau[1], G.A. Thomas[1], D. Chatting[2], J. Thorne[2], G. Milnthorpe[3], P. Woodward[4], L. Bull[5], E-J. Ong[6], A. Hilton[6], J. Mitchelson[6], J. Starck[6]

[1]BBC R&D, [2]BT Exact Technologies, [3]De Montfort University, [4]Queen Mary, University of London, [5]University College London, [6]University of Surrey

Research & Development
BRITISH BROADCASTING CORPORATION

BBC Research & Development
White Paper WHP 045

Real-time production and delivery of 3D media

M. Price, J. Chandaria, O. Grau, G.A. Thomas, D. Chatting, J. Thorne, G. Milnthorpe, P. Woodward, L. Bull, E-J. Ong, A. Hilton, J. Mitchelson and J. Starck

Abstract

Television production is increasingly making use of 3D models, in applications including animation and virtual production. These models are rendered to produce 2D images during the production process. However, with the ever increasing power of 3D graphics processors in home PCs, and new developments in 3D display technology, the time is right to consider how the broadcaster can maintain content in its 3D form all the way through the programme chain. This paper presents the results of PROMETHEUS, a UK 'LINK' project, whose aim is to examine the various issues related to developing an end-to-end 3D programme production chain using MPEG-4.

This document was originally published in the Conference Publication of the International Broadcasting Convention (IBC 2002) Amsterdam, 12-17 September 2002.

White Papers are distributed freely on request.

Authorisation of the Chief Scientist is required for publication.

REAL-TIME PRODUCTION AND DELIVERY OF 3D MEDIA

M. Price[1], J. Chandaria[1], O. Grau[1], G.A. Thomas[1],
D. Chatting[2], J. Thorne[2], G. Milnthorpe[3], P. Woodward[4], L. Bull[5]
E-J. Ong[6], A. Hilton[6], J. Mitchelson[6], J. Starck[6]

[1]BBC Research and Development, UK, [2]BT Exact Technologies, UK,
[3]De Montfort University, UK, [4]Queen Mary, University of London, UK,
[5]University College London, UK, [6]University of Surrey, UK

ABSTRACT

Television production is increasingly making use of 3D models, in applications including animation and virtual production. These models are rendered to produce 2D images during the production process. However, with the ever increasing power of 3D graphics processors in home PCs, and new developments in 3D display technology, the time is right to consider how the broadcaster can maintain content in its 3D form all the way through the programme chain. This paper presents the results of PROMETHEUS, a UK 'LINK' project, whose aim is to examine the various issues related to developing an end-to-end 3D programme production chain using MPEG-4.

INTRODUCTION

Current trends in consumer technology suggest that future digital television sets will probably contain a computer with processing power exceeding that of today's high-end workstations. Coupled with the increasing availability of 3D rendering hardware, and new developments in 3D display technology, it is reasonable to assume that TV sets in 2010 would be capable of delivering full 3D television. The broadcast industry must therefore prepare for this inevitable, yet exciting, evolution from 2 to 3 dimensions.

The 'Prometheus' project (1), a UK-based collaborative project led by the BBC, and involving AvatarMe, BTexact Technologies, De Montfort University, Queen Mary University of London, Snell and Wilcox, University College London, and University of Surrey, was set-up in September 1999 to examine the practical feasibility of such an end-to-end 3D programme production chain. Early results of the work were reported in Price and Thomas (2).

This paper overviews the real-time 3D programme production demonstrator that the project has developed, highlighting the key features and limitations.

3D CONTENT

Virtual production techniques have been developed to a level where they can be used to replace some, or all, of the scenery in conventional (2D) television production. The 3D virtual set is, therefore, our starting-point along the road to 3D television. However, the central elements in broadcast programme content are actors, and their interactions with the world around them. Hence, in order to achieve 3D television production, these elements must also be 'virtual' – i.e. the actors must be realistically modelled in 3D, and placed into the virtual set.

However, we wish to keep the production process itself as close as possible to that of normal TV, to ease the transition to 3D content production and ensure that the tools we develop have applications in more conventional forms of production as well. Hence, the

model and motion capture processes should avoid the use of special scanning equipment, and the mounting of markers or transducers onto the actors' faces and clothing. Instead, we restrict ourselves to the use of vision-based techniques.

ACTOR-MODEL (AVATAR) GENERATION

As humans are more sensitive to visual artefacts in the face than in the rest of the body, the level-of-detail required for a model of an actor's face is higher than for the body. We have therefore chosen to model the head and body of the actor separately. The actor model (avatar) is formed by merging the resulting head and body models.

Head Modelling Process

Sufficient information to create a realistic head model can be obtained from two photographs of the actor's face - one from the front and the other from the side. BTexact are applying image processing techniques to these photos to automatically locate important feature positions and the shape of the head. A generic 3D head model is then deformed to match this shape and the photos of the actor are wrapped around to form a seamless texture, extending the work by Machin et al (4,5). The resulting heads are easy to animate because we already know the locations of the important features.

Deformation of the generic head is based upon the use of Radial Basis Functions (RBF) as described by Ju and Siebert (7). The Feature points (chosen to match the 84 MPEG-4 Facial Definition Parameters (6)) are displaced from their locations in the generic head to their locations in 3D space described by the two orthogonal photos. Other vertices map to smoothly weighted points in between.

Body Modelling Process

Researchers at the University of Surrey have developed a model-based computer vision framework for reconstructing articulated models of actor shape and appearance from multiple camera views. The model-based approach extends previous research by Hilton et al (8), which utilised orthogonal camera views of the subject, to projective reconstruction from multiple camera views. Simultaneous capture of images of the actor from multiple camera views enables metric reconstruction of actor shape, together with appearance matching between views to reconstruct a photo-realistic texture map. Here we present an overview of the reconstruction technique. Further details are given in Hilton et al (9).

Initially the multiple-camera studio is calibrated using a wand-based technique (Mitchelson and Hilton, (10)) to estimate the camera intrinsic (focal length, centre-of-projection) and extrinsic (position, orientation) parameters. Multiple-view calibrated images of the actor are captured with the actor in a fixed pose, as illustrated in Figure 1(a) for three views of a six camera studio. Chroma-key techniques are used to separate the actor from the background in each camera view. The resulting silhouettes are then used to compute the visual hull as shown in Figure 1(c) (note the large ambiguity in the chest area due to the limited number of views). Manual initialisation is then performed to register the articulated structure of the generic model with the visual hull and identify key feature points on the data. A shape constrained fitting procedure (11) is used to reconstruct the approximate shape of the actor in the presence of visual ambiguities. The resulting reconstructed shape model is shown in Figure 1d, note the correct approximation of surface shape in the chest area. A single texture map is then derived for the model by matching between camera views.

Integration of Head and Body Models

For correct integration of face and body models, it is important that the same lighting conditions are used by both modelling processes. Integration is then achieved by applying the face conformance process to the deformed body model. This refines the shape of the head and aligns FDPs with features in the face shape and texture so that the face can be animated. The newly-created head texture is blended with the existing body texture to conceal the join.

Actor Modelling using Simple 3D Shape

A simple but effective alternative method of creating an approximate 3D actor model is to use a simple shape model and to texture-map it with a live video image of the actor. An alpha signal, derived using chroma-key, prevents the background around the actor from appearing. The shape must be placed in 3D so that it covers the actor completely from the viewpoint of the studio camera. To do this, the position of the actor within the studio is tracked, and this information is used to control the position of the shape within the scene. This process is described in (2).

We have found that the use of a planar polygon for the transparent shape is sufficient if the viewpoint is close to the same position as the studio camera. However, the image of the actor shows significant distortions when the viewpoint moves away from the real camera position. Hence, an extension was developed that creates pseudo depth of the actor using the assumption that an object is usually more elevated in the middle of its silhouette. This depth information is transferred onto a VRML ‘elevation grid’ object, as included in the MPEG-4 standard. The elevation value of any given point on the elevation grid is a function of the distance of a point in the 2D silhouette to the closest edge as described by Grau et al (3). Figures 3 (a) to (c) show the results of this process.

ANIMATING THE AVATAR

In order to animate the modelled avatars, we need to capture the motion of the actors, and apply it accordingly. The face and body animation data are captured by separate systems, each using images from conventional cameras.

Face Motion Capture and Animation

Traditionally facial motion has been captured using markers placed on the face. However, with the use of computer vision techniques, marker-less systems are becoming viable (12,13). The aim of the Prometheus face tracker is to create an MPEG-4 compliant stream of facial animation data that reflects the actors performance, in real-time, using a single conventional camera.

The tracking algorithm is based on earlier work in BTexact by Machin (14). The subject's eyes are first found by template matching, from which is determined the likely position of the mouth and the sides of the head. We also track the brow, nose and corners of the eyes to find signs of wrinkling as the muscle is contracted under the skin, using a similar technique to Lien et al (15). These wrinkles are mapped into a FACS (Facial Action Coding System) (Ekman and Friesen (16)) representation of expression and rendered into MPEG-4 FAP (Facial Animation Parameter) descriptors using a modified version of the CANDIDE-3 (Ahlberg (17)) model. In this way motion of features is calculated in three-dimensions from a single view.

An extended set of 257 FAP descriptors is used internally. This allows the movement of every (and one additional) FDP in the x, y and z-axis to be fully described, thus preserving

as much information as possible prior to encoding. The standardised set of 68 FAPs is a subset of our extended set and can be generated by the system.

The facial animation must convey the same emotions as the actor wishes to portray. In order to achieve this, we have introduced some secondary behaviour (Gillies et al (18)) and natural-looking interpolation between expressions. We are also investigating the addition of noise in the animation to make the face more 'alive' (Perlin (19)).

Body Motion Capture

Researchers at the University of Surrey have developed techniques for capturing actor movement within a multiple camera studio. The objective of this research is video-rate reconstruction of the actor movement without the placement of markers on the subject. Model-based techniques have been developed which enable real-time reconstruction of actor movement (21). Tracking is performed based on both colour (21) and edge features.

Two approaches have been investigated to reconstruct the movement of the articulated model. Inverse kinematics using gradient descent enables real-time performance, but it is not robust to large changes in pose between frames. Particle filter techniques, which maintain multiple hypotheses for the pose at a particular frame, have been used to reconstruct a large range of movement but are computationally expensive. To overcome the limitations of both techniques and obtain a robust, computationally efficient approach, a hierarchical particle filter technique has been developed. Results of the multiple view tracking of a pirouette movement are illustrated in Figure 2. The current challenge is to scale this technology to achieve robust real-time reconstruction of arbitrary movements.

CLOTHING MODEL

Higher realism can be achieved if the actor's clothing is modelled separately rather than simply being texture-mapped onto the surface of the avatar. Recently, high-end motion picture productions such as Monsters Inc. by Disney/Pixar have included physical simulations of clothing for characters. Physical simulations of this type are highly compute intensive and a fast, robust solution is difficult to realise. Hence, animations of this kind are normally processed offline.

University College London’s Virtual Environments and Computer Graphics Group have developed a real-time cloth simulation system for Prometheus. The system is a custom rendering, animation and cloth simulation tool, capable of running stand-alone.

The cloth simulation technique uses mass-point representations connected by conceptual springs. To do this, it uses an approximate derivation from finite element methods given by Anderson (22), rather than the more common rectangular grid mass-point system used in many systems. This allows suitable 3D meshes of arbitrary connectivity to be simulated as cloth, so that modellers can design one-piece items of clothing around the avatar.

In each simulation time step, the forces that are exerted on each mass-point are calculated. The resulting behaviour over a very short millisecond timescale is then solved as an initial value ordinary differential equation problem. Forces which are exerted on each mass-point include internal tensional and flexional forces which keep the cloth together, motion and frictional forces exerted by the moving character, and environmental forces such as gravity, viscous drag and wind. The current system uses an Euler solver which is typically evaluated 30 times per frame for a clothing item consisting of less than a thousand polygons. More stable numerical methods are currently undergoing investigation.

One of our primary challenges has been to develop cloth-body collision detection systems that are efficient and scalable such that they can be evaluated at the required 750Hz rate for

each object the cloth may collide with. During a single time step, the body moves with a definite motion that is realised, whilst the cloth has an intended motion, which may be stopped short by collision response dynamics. The collision detection system uses *distance fields* (Jones and Satherley (23)) resulting from the voxelization of a segmented form of the avatar. The intended motion of the mass-points are tracked at high speed within these distance fields, allowing a vertex to surface collision detection within a specified tolerance envelope which surrounds the original avatar geometry.

Real-time performance of approximately 25 frames per second has been achieved for single characters with items of clothing consisting of 400 polygons on a 1.4Ghz Intel Pentium 4 system. Figure 4 (a-h) shows example frames from a short animation. Note the cloth deformation due to the body motion which is particularly noticeable in animation (h) on the back leg. Issues such as stability, robustness and recovery from unusual states are topics for current and future research.

SCENE COMPOSITION

The various elements of the 3D scene (animated avatars, clothing, actors represented as textue-mapped video, and environment model) need to be brought together under the control of the director to assemble the required scene. This task is carried out by the studio controller, a collection of real-time software tools that allow the 3D scene to be manipulated and rendered. Virtual camera viewpoints can be specified, and the relative position of avatars and background can be controlled. The studio controller manages the timing of the various asynchronous data streams, making use of the Prometheus stream buffering functionality described below under ‘Real-time Issues’.

DELIVERY USING MPEG4

In order to deliver the 3D content to the viewer, it needs to be encoded in a way which preserves the model-based nature of the content - allowing the viewer to independently control viewpoint, and even view it with a 3D display. The MPEG-4 coding standard, and in particular the Binary Format for Scenes (BIFS) provides a way of achieving this (Woodward et al (24)).

Queen Mary, University of London has developed a system that encodes the 3D models and associated animation data into an MPEG-4 BIFS stream. This is based on the MPEG-4 encoder/decoder pair (codec) developed by Hosseini and Georganas (25).

The incoming data from the studio controller is converted into Java Media Framework (JMF) *DataSource* buffers. These buffers are then driven through the encoder and renderer engine by the JMF. The encoder engine parses the model file and passes the 3D content to a Java3D renderer, which displays a local visual representation of the scene.

The parsed model file – now in *MPEG-4 BIFS* format – is written to a JMF channel, which can be either a file or a network connection. Subsequently, as the scene updates are received, the encoder converts the updated geometry into MPEG-4 BIFS update commands and writes the updates to the file/network.

The MPEG-4 decoder attaches to the network or opens the file, and decodes the 3D content. It then converts the decoded content to Java3D format and renders it.

REAL-TIME ISSUES

The Prometheus system has functionality for recording the captured 3D data to file for subsequent post-processing. However, our primary goal was to demonstrate a live end-to-end chain. This means that although the underlying 3D models can be delivered as files,

the corresponding animation parameters must be streamed to the MPEG-4 encoder in real-time. As the Prometheus system consists of a network of distributed data capture and processing modules, we have adopted a CORBA-based framework to meet this requirement.

The streams of animation data must be buffered, as they arrive asynchronously at differing and varying frame-rates from their respective sources. Each stream buffer is implemented as a CORBA object. The CORBA objects implement single-input, multi-output buffers, which are self-managing, distributed processes, allowing a degree of load balancing across the available resources on the network.

3D DISPLAY

Although the animated 3D content could be adequately displayed on a standard 2D display with viewpoint control functionality, its full value can only be appreciated when viewed on a 3D display. However, the 3D display used must not negatively impact upon the TV experience to which viewers are accustomed. It must allow freedom to sit anywhere, with no need for special glasses, and it must be suitable to view for several hours without inducing eyestrain.

The Integral Photographic system, originally devised by G.Lippmann in 1908 (26), fulfils these requirements. It records and displays non-coherent autostereoscopic images using a microlens array. Integral Images are by their very nature viewed as continuous images as each part of the scene is imaged in adjacent lenslets. This enables the encoded intensity distribution to contain within it enough directional information to display a correct integral image from arbitrary viewpoints.

The production of a computer generated Integral Image requires specialised rendering techniques to generate a suitable intensity distribution, which can then be ‘decoded’ by placing a suitable microlens array over the image. It also requires a very high resolution display. Various rendering techniques and displays have been investigated in the project. The most promising approach appears to be the use of a PC cluster with a hardware compositor, the output of which is displayed on a high resolution LCD fitted with a decoding microlens array.

CONCLUSION

The Prometheus project has extended the state-of-the-art in several areas required to support future 3D content generation, transmission and display. The remaining challenges include improving the robustness of marker-free motion capture, and optimising the hardware, software and display configurations needed for real-time generation of Integral Images. The project has also provided the opportunity to expose programme-makers to some of the possibilities that technology will offer in the near future.

ACKNOWLEDGEMENTS

The authors would like to thank all their colleagues in the Prometheus project for their contributions to this paper. The project was supported by the UK DTI and EPSRC under the Link Broadcast Technology Programme.

REFERENCES

1. http://www.bbc.co.uk/rd/projects/prometheus
2. Price, M., Thomas, G., 2000. 3D Virtual Production and Delivery Using MPEG-4. IBC'00 Conference Publication, pp 616-621.

3. Grau, O., Price, M., Thomas, G., 2001. Use of 3-D Techniques for Virtual Production. Proceedings of SPIE Photonics West Symposium 2001, Volume 4309.
4. Morlock, A. , Machin, D., McConnell, S., Sheppard, P., 1999. Telepresence, Kluwer Academic Publishers, 1999.
5. Mortlock, A., Sheppard, P., Wallin, N., 1998. W09858351A1: Generating An Image Of A Three-Dimensional Object, International Patent. (1998).
6. Tekalp, M., Ostermann, J., 2000. Face and 2-D Mesh Animation in MPEG-4, Image Communication Journal, Tutorial Issue on MPEG-4 Standard, Elsevier.
7. Ju, X., Siebert, J., 2001. Conformation from generic animatable models to 3D scanned data, Proc. 6th Numérisation 3D/Scanning 2001 Congress, Paris, France. 2001.
8. Hilton, A., Beresford, D., Gentils, T., Smith, R., Sun, W., Illingworth, J., 2000. Whole-body modelling of people from multi-view images to populate virtual worlds, Visual Computer: International Journal of Computer Graphics, Volume 16, pp.411-436.
9. Hilton, A., Starck, J., Collins, G., 2002. From 3D Shape Capture to Animated Models, IEEE Conference on 3D Data Processing, Visualisation and Transmission, June 2002.
10. Mitchelson, J., Hilton, A., 2002. Wand-based Calibration of Multiple Cameras, British Machine Vision Association workshop on Multiple Views, May 2002.
11. Starck,J. and Collins,G. and Smith,R. and Hilton,A. and Illingworth,J. Animated Statues, To Appear Journal of Machine Vision Applications, 2002
12. FaceStation, Eyematic Interfaces - http://www.eyematic.com/
13. faceLAB, Seeing Machines - http://www.seeingmachines.com/
14. Machin, D., 1996. Real-time facial motion analysis for virtual teleconferencing, IEEE International Conference on Automatic Face and Gesture Recognition, October 1996, pp 340-344.
15. Lien, J., Kanade, T., Cohn, J., Li, C., 1998. Automated Facial Expression Recognition Based on FACS Action Units, IEEE International Conference on Automatic Face and Gesture Recognition, April 1998, pp. 390-395.
16. Ekman P. & Friesen W. V.: 'Facial action coding system: A technique for the measurement of facial movement'. Palo Alto, California. Consulting Psychologists Press (1978).
17. Ahlberg J: 'CANDIDE-3 -- an updated parameterized face', Report No. LiTH-ISY-R-2326, Dept. of Electrical Engineering, Linköping University, Sweden (2001). http://www.bk.isy.liu.se/candide/
18. Gillies, M., Dodgson, N., Ballin, D., 2002. Autonomous Secondary Gaze Behaviours, AISB'02 Symposium on Animating Expressive Characters for Social Interactions, April 2002.
19. Perlin, K., 1997. Layered Compositing of Facial Expression, SIGGRAPH 1997 Technical Sketch.
20. Thorne, J., Chatting D., 2002. The Prometheus Project - the challenge of disembodied and dislocated performances, BT Technology Journal, Volume 20, No 1, pp 85-90.
21. Li, Y., Hilton, A., Illingworth, J., 2002. A Relaxation Algortihm for Real-time Multiview 3D-Tracking, To Appear Journal of Image and Vision Computing, 2002.
22. Anderson, J., 1998. Fast Physical Simulation of Virtual Cloth Based on Multilevel approximation strategies, Ph.D. Thesis, University of Edinburgh.

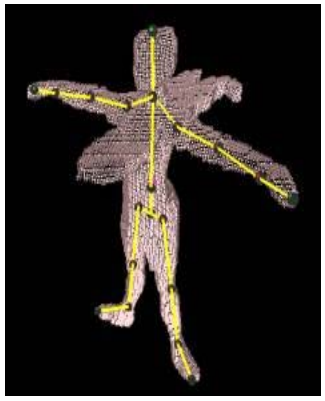
23. Jones, M., Satherley, R., 2001. Using Distance Fields for Object Representation and Rendering. Eurographics UK 19'th Annual Conference, pp 37-44.

24. Woodward, P., Paker, Y., Pearmain A., 2001. Implementing virtual studios in MPEG-4, Proceedings of Workshop on MPEG-4 (Sponsored by IEEE Circuits and Systems Society), June 18-20 2001, pp25-28.

25. Hosseini, M., Georganas, N.D., 2002. MPEG-4 BIFS streaming of large virtual environments and their animation on the Web, ACM 7th International Conference of 3D Web Technology (WEB3D 2002), February 2002.

26. Lippmann, M., 1908. Eppeuves Reversibles Donnant La Sensation Durelief, J. Phys, Paris 821.

(a) Multiple view images (3 of 6 views)

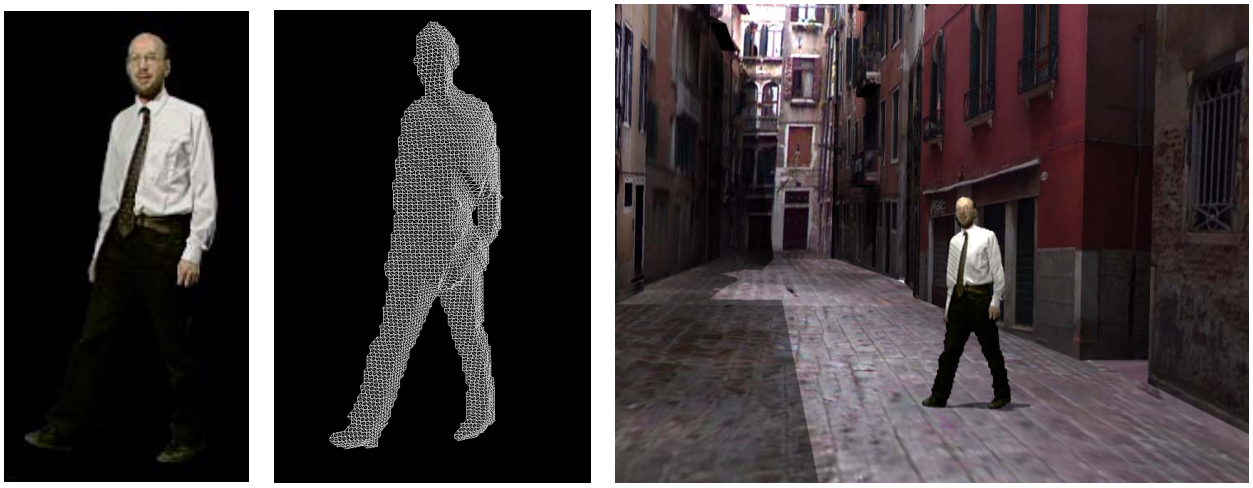


b) Generic Model (c) Visual Hull (d) Reconstructed shape

(e) Animated Dancer in Virtual Street Scene

Figure 1 - Dancer Model Reconstruction from Multiple View Images

Figure 2 - Movement reconstruction from multiple views for a Pirouette Sequence



(b) Original Video Image

(a) Resulting Transparent Shape

(c) Resulting Object Integrated into Scene

Figure 3 – Simple 3D Shape Actor Modelling

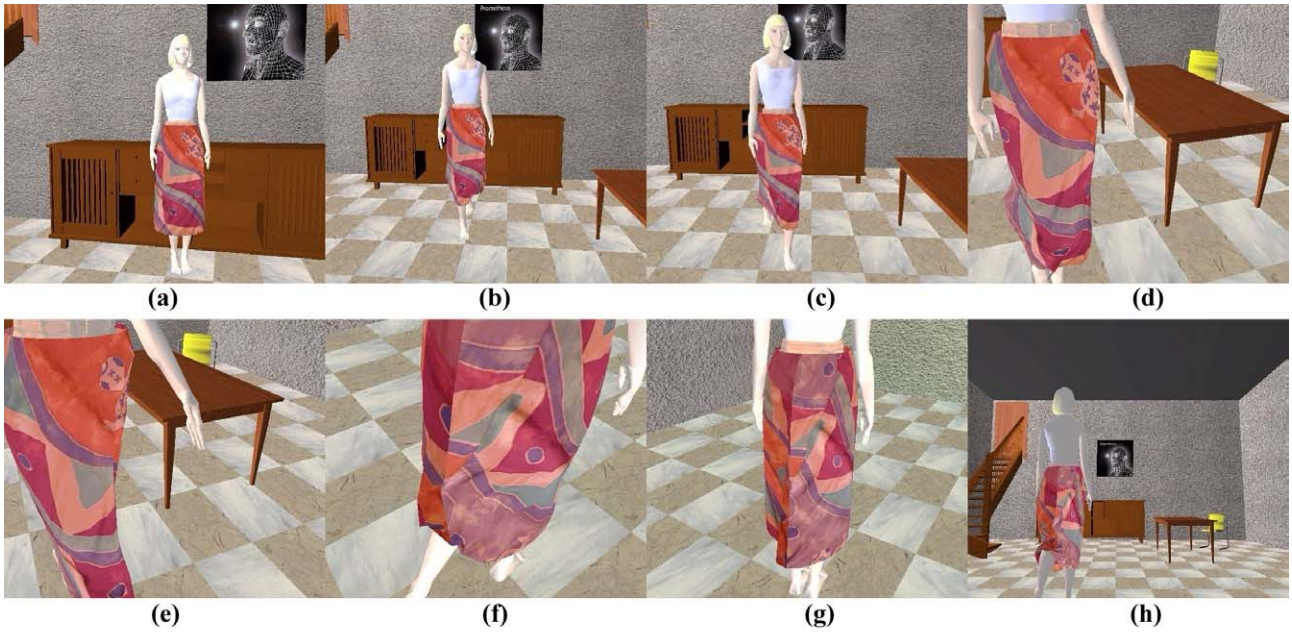


Figure 4 – Example Cloth Simulation Sequence