VIRTUAL CRIME SCENE RECONSTRUCTION WITH INTEGRATED ANIMATED CHARACTERS

Nicholas Davies, Qasim Mehdi, and Norman Gough

Games Simulation and AI Centre
The University of Wolverhampton
School of Computing and Information Technology
Lichfield Street, Wolverhampton, WV1 1SB West Midlands, UK
N.P.Davies2@wlv.ac.uk

KEYWORDS
3D Scene Construction, DirectX, Skinned Meshes, Computer Graphics.

ABSTRACT

This paper proposes some new techniques for the development of virtual 3D environments for crime scene reconstruction. Current techniques were investigated, which concentrate on the production of photo-realistic scenes. However, these were considered to be too specialist and expensive. The Crime Scene Creator developed here provides a cheaper and quicker alternative that allows quasi-accurate scenes to be constructed and populated with animated characters to act out crimes events, with some very encouraging results. The paper also describes some initial investigations for implementing voice recognition and natural language processing as a basis for further development.

INTRODUCTION

This paper describes research on developing an application for constructing a 3D virtual scene that can be manipulated and viewed from any angle in real time without having to be pre-rendered. This enables investigators to evaluate and eliminate different hypotheses much more rapidly. Current research has developed techniques for the creation of high definition photo-realistic models (Howard *et al* 2000, Little *et al* 2001). However, these techniques are very process intensive and highly technical. The primary goal of the present research was to develop faster methods for creation of quasi-accurate scenes that can be produced with very little training or expensive hardware. Other techniques investigated were the introduction of animated intelligent characters that could act out scenes described to them via Natural Language (NL). This would enable witnesses and victims of crime to develop scenes from descriptive text rather than complicated user interfaces which may require the involvement of a system administrator who could inadvertently influence the final scenes. The paper is structured into 6 sections: Introduction: this section; Background: Current crime scene reconstruction systems; Requirements and Methodology; Implementation; Example: An example of how the system was used to create a scene; Conclusions and Further Work.

BACKGROUND

It is now possible to develop photorealistic 3D models of a scene that can be viewed from multiple angles. The first method investigated was created by Howard and Murta (2000) in collaboration with the Greater Manchester Police, UK. Data was collected via a set of forensic photographs. These were taken from four corners of a room to give as much coverage of the scene as possible. Photographs were also taken of individual objects with police rulers placed next to them to give an indication of scale. An initial floor plan was produced from measurements taken at the scene and entered into a scene builder in the form of line segments. These were then extruded to produce the 3D room. The height was judged visually rather than using exact measurements. Once the room had been built, objects were placed within the scene depending upon their position gathered from the forensic photographs and measurements. Existing libraries of models that matched objects from the scene were used. These were objects the police had already developed for other applications, as well as some public domain objects. Although some of these items did not match the objects within the scene exactly, as the objects were not of critical importance it was decided that this was of little significance. The completed model allows the viewer to navigate around the scene and view objects from any angle. However, the scene may not have been completely accurate due to estimation of scales, and the inclusion of third party models.

Little *et al* (2001) developed a portable automated system for gathering information from crime scenes. Their technique uses a robotic system with a combination of cameras and laser range finders. The machine automatically moves around the room to get comprehensive coverage. The information gathered by the laser produces a set of points in 3D space. These are combined to form a mesh, and images taken from the cameras are mapped onto it. A model is available for preview as the machine makes its way around the room so a human operative can check for errors. For instance, due to laser technology, the system can have problems with mirrors or other reflective objects. Once the room has been accurately reconstructed, data can be exported to a number of different platforms for rendering, including PC, Unix and Silicon Graphics Machines. The results obtained were far more accurate than scenes developed from photographs; however, the resolution of the models was not as high as that obtained from Howard's technique, and as such was not as visually impressive.
Sequeria *et al* (2001) outline a method very similar to that of Little *et al*. It also involves a robotic system that uses camera and laser range finders to produce a 3D reconstruction of the environment. Its approach is slightly more methodical as it plans its next capture point automatically to guarantee complete coverage. Again, the results generated were highly accurate, although no animation was implemented.

Proceedings 18th European Simulation Multiconference
Graham Horton (c) SCS Europe, 2004
ISBN 3-936150-35-4 (book) / ISBN 3-936150-36-2 (CD)

Hu and Brown (2002) investigate a method of recreating scenes using mosaic photographs. A mosaic is a single picture of a scene created by taking multiple pictures at regular intervals. This is particularly useful where there is not enough space in a location to get a complete picture from one angle. A camera was placed in three corners of a room and rotated by regular amounts. An algorithm was then employed to determine the 3D structure of features within the room. The resulting scene was of a high quality, although in the conclusion of the report it was noted that the technique had only been tested on simple environments. It is unclear how useful this technique would be in a larger complex environment.

Dyer (2001) outlined a method of scene recreation that uses a volume-based approach. Typically, scene recreation attempts to model shapes within an environment by making a triangular representation of its surfaces. This has some disadvantages. It can be difficult to reconstruct a surface that has sections obscured from certain viewpoints. There must be correspondence across multiple views that indicate that surfaces are linked. Surface patches must be fused together to form a single consistent model. Dyer's technique differs in that it starts with a volume of 3D space and splits it into regular sized cells known as voxels. It is then uses multiple views to indicate whether a space is filled or not. When the room has been completely generated, the voxels can either be transparent, indicating empty space, or opaque indicating occupied space. The filled voxels are then coloured to match the surface colour of the scanned object. The advantages of using this technique are that complete coverage of the scene is not required and it is not necessary for the camera to be calibrated exactly. The results of this method were accurate but the images did appear to be quite pixelated. A reason for this may be that the volume was not broken down into sufficiently small voxels to produce high definition results, although it can be assumed that the smaller the voxel size, the greater would be the processing power required to render the image.

In addition to investigating methods for environment recreation, work based on modeling artificially intelligent characters was also examined, and several potential agent architectures were identified. Prendinger and Ishizuka (2004) outline the creation of life-like characters for uses such as synthetic actors, teammates in games and tutors. Particular attention is paid to the benefits of implementing a model of the internal emotional state of synthetic entities and applying social constraints, which it is hoped will make the characters appear more believable allowing viewers to gain a feeling of empathy towards them. They go on to describe two tools they developed that can be used to control and model the internal emotional state 'SCREAM' (**SCR**ipting **E**motion-based **A**gent **M**inds) which contains the emotional model, emotion generation, emotion regulation and importantly emotion expression. Giles et al (2003) explain the importance of expressing the emotional state. Failure to do so can make complicated processes seem like a sequence of arbitrary actions, which fails to produce a sense of empathy from the viewer. The second tool MPML (**M**ultimodal **P**resentation **M**arkup **L**anguage) is an XML style markup language that allows characters actions to be scripted using a combination of tags. This is an important feature, as it will allow the freer scripting of characters without having to rely on pre-render graphics or set animation sequences.

Going one step further Anastassakis et al (2002) describe the mVITAL Intelligent Agent System that can be used to model an intelligent environment populated with multiple agents that interact with each other. An overall world view is maintained, and agents work on a sense-decide-act cycle. An agent can request a view of the current world state, and update the world state by performing actions. Agents do not directly communicate with each other; instead they send a special speech action to the world, which other agents pick up. To decide whether to act upon the speech action, each agent has reasoning capacity based on Belief–Desire–Intention (BDI) architecture, and an agent personality module called Virtual Agent Language (VAL). It is hoped that this method will eventually merge intelligent multi-agent systems with intelligent virtual environments to produce an integrated platform for areas such as education, presentations and entertainment.

REQUIREMENTS AND METHODOLOGY

While the scene creation techniques investigated produced some impressive results, it was considered that the amount of technical skill required to produce the scenes was so complex it would require a team of specialists, and such resources are rarely available. We preferred to have a system that would be fast, efficient and relatively cheap to use. Therefore an alternative had to be found. This was achieved with the use of a pre-created DirectX graphics engine for scene rendering, custom graphics developed in 3D Studio Max, and the development of a graphical application to orientate the objects when creating the scene. While this method is not as accurate as other techniques, it does offer some advantages. It is not always possible, or necessary, to create a photorealistic scene to evaluate and document the circumstances of a crime, and it would be very expensive to produce a virtual scene for every crime committed. In certain situations a method is required to quickly and cheaply create and render scenes. Examples of this are where the scene has been destroyed *e.g.* in the case of fire, or in areas that are in constant public use. It also appeared that the systems created previously had been over reliant on producing static photo-realistic scenes and had neglected the introduction of virtual animated characters. This is a drawback when the actions of the individuals are more important than the environment in which the incident occurred. Vital information and hypotheses can be evaluated when there are virtual characters acting out the crimes in front of the investigator rather than just motionless images. Therefore the following requirements for the system were identified:

A simple to use graphical system was required that would allow information to be gathered from various data sources to create a virtual crime scene. The system was required to render the scenes that would accept input generated from the scene creator and render it to screen. It should be possible to explore and view the scene from difference angles in real time. To show the committed crime, the rendering system should be able to display animations. A set of graphic objects that could be used within the generated scene needed to be developed. For example these should include floor, wall and ceiling tiles used to create the room as well as objects for use within the scene such as furniture, lights and plants. A set of realistic animated characters and a set of complete scenes were also needed.

The methodology used to complete the design and development tasks was based around the waterfall method. After the initial requirements were investigated, and a design introduced, the system was implemented. However, the design was not rigid, and with the use of waterfall method, the requirements and implemented system could be updated constantly, with the result affecting subsequent sections. In this way it was possible to add a degree of flexibility to the design. This methodology was implemented due to the learning curve required to investigate all the systems and applications needed for this system. It was not possible at all stages to know whether an implementation would be achievable. The waterfall method allowed the freedom to go back and update the design based on new information.

IMPLEMENTATION

The crime scene creation system implemented (Fig. 1) was developed in Visual Basic 6 and allows users to develop scenes in a graphical environment. The scene was split into several layers each containing a grid of cells that could be populated with a library of pre-created objects. There are levels for wall, floor, object and characters. Each item can be scaled and orientated in any direction. When a scene is produced, first wall panels are drawn out and scaled, and then a floor and ceiling panels are selected and positioned. Finally objects are placed within the created room. The scene creator produces a set of files containing coordinate data and a list of objects, which are loaded by the graphics engine for the scene to be rendered. Once the scene has been viewed in the graphics engine, it can be modified instantly and re-rendered as required until the user is satisfied with the results.

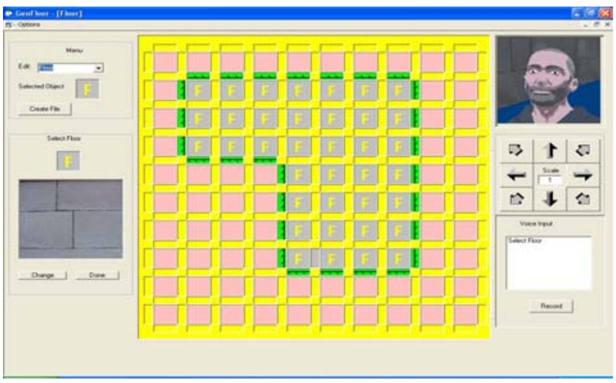

Figure 1 Crime Scene Creator

Before the implementation of intelligent agents is commenced, as an interim process, high definition characters with pre-created animation sequences were produced. According to Herda *et al* (2000), one of the greatest challenges in computer graphics is the realistic modeling of the human body. This is because of the brain's familiarity with common objects. If a character is not animated and reconstructed accurately, the viewer will pick up on it, even if they cannot tell exactly what is wrong. Hence, the first step of the process was to develop an accurate graphical representation of a human figure.

The introduction of data collection methods such as laser scanning have made it possible to create realistic human characters to a high resolution. However, this is just a surface representation. The human body is a very complex structure, which is made up of a skeletal frame, fat and muscle and covered with skin, which stretches and folds over this volume. Algorithms that allow realistic body movement including the fluid motion of skin have been developed to add a sense of realism to models. The current standard technique based on a moving hierarchical skeleton is known as a skinned or skeletal mesh, which is the technique implemented in 3D Studio Max. Skinned meshes are based on the idea that an object's shape can be controlled by an underlying bone structure. The bones positions can be changed to alter the stance of the object, and animations can be produced via the manipulation of the position and orientation of the bone structure over a regulated time period. The surrounding layers react in an appropriate manner simulating muscle contraction and tension.

Again due to budget and time constraints it was not possible to develop a character via laser scanning techniques, although it was possible to develop a full skinned mesh with animation in 3D Studio Max following a method outlined by Steed (2002). This also has some advantages. It is possible to control the resolution of the created character. This is important, as smooth animation of high definition objects is still a demanding task on today's graphics hardware. With the use of 3D Studio Max, the created model could be optimised to remove unneeded faces to aid faster real-time rendering. Another advantage of 3D Studio Max is the ability to export models as various file types. In this case models were exported as .x files which are a native file type used in DirectX. This aided development of the graphics engine, as within the DirectX development framework there exists many libraries specific to the manipulation and animation of .x files.

The first stage of the process of creating the animated characters was to gather photographs of a real-life person that were used as a guide when modeling. The photographs were also used to make materials and textures to colour the model. Two initial sets of photographs were taken. One set of the head in isolation, and the other for the full body. Both sets were taken from head on, and from the side. Unneeded and distracting information was then removed from the bitmapped image, and the images scaled so the proportions were equal. The images were then imported into 3D Studio Max, and set as a background image. Lines were drawn over prominent features to produce a set of guide lines for the front and side of the head. The side view was then rotated through 90˚ and moved to the centre of the image to produce a 3D shape. With the guidelines in place, one side of the head was modeled with the use of a geodesic sphere (a spherical object made up of 128 faces), which was placed over the guidelines and cut down the middle to produce a semi sphere. This was done as it is more efficient to only model half the head as once it is completed it can be mirrored to produce the complete head. The exosphere was manipulated to the shape of the head by moving, adding, and dividing faces. Making complex elements such as the mouth, eyes and ears proved to be very time consuming. Because of the 3D nature of the object being modeled, and only having a 2 dimensional image on the screen it can difficult to know exactly where a point is being moved. It is possible to use a multiple panel view that shows the object from four different perspectives; however, with the limited screen size it is a trade off between scale and perspective (Burford & Blake 2001).

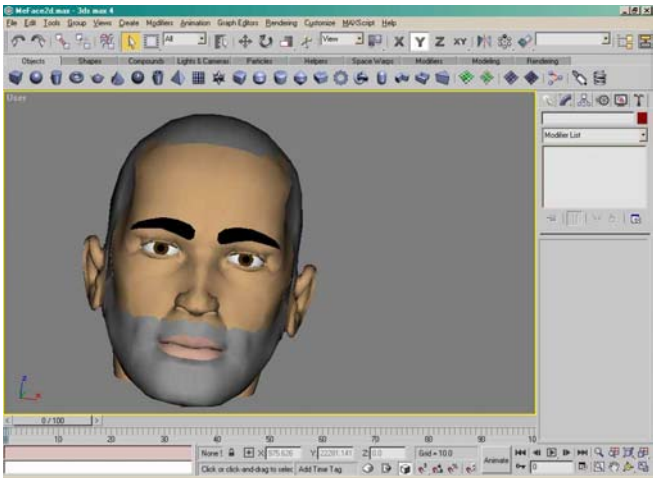

Figure 2 Character Developments

Next, the face was split into separate areas, namely the mouth, beard, eyes and eyebrows and hair. Textures generated from the original photographs were then added to appropriate areas. Finally, the left head segment was copied and mirrored, and both halves joined together to form the complete head shown in Figure 2.

Once the shell of the character was completed, a biped object was developed to act as the internal bone structure, and attached. Each point on the biped had to be positioned exactly to the corresponding bone on the original object for a successful attachment, which again is a very labour intensive task. When the character was completed, animation sets could be imported and used to animate the model by moving the internal bone structure to create fluid motion (Aubel & Thalmann, 2000).

After some initial experimentation and investigation it was decided that the rendering system used to display the scene would be developed using the graphics library DirectX 8.1. However, this is an arbitrary choice as in many ways as OpenGL is just as powerful. Even VRML would be sufficient for the rendering of the scenes. The choice was made easier with the availability of a pre-developed graphics engine created by Adams & La Mothe (2002) that was, after some modification, suitable for this project's purposes. The graphics engine works by loading all mesh and position data generated in the crime scene creator, and the graphics created in 3D Studio Max. The resulting scene is then rendered to the screen in real time. It is possible to view the scene from any angle by moving the point of view via the use of the keyboard and mouse. It was found the most convenient method of doing this was to use the keyboard to change position (up, down, left, right, forward, back) and to change the viewing angle via the mouse. Further research is required to indicate whether this is the optimal method, but it is a control system used in many first person computer games so users may be familiar with it.

EXAMPLE

Figure 3 shows two views of a scene created in the crime scene generator, and rendered by the graphics engine. The room was composed of 8 * 8 floor and ceiling panels and 24 wall panels. 5 objects including 3 chairs, 1 wall cabinet, and a sofa. 1-point light is located in the scene to illuminate it. Finally the scene shows an animated character performing a kick.

The example shows how a scene could be developed with the use of a witness and a system expert who would utilise the software. First, measurements and photos of the scene were taken. The textures used on the floor, wall and ceiling tiles were created directly from these photos and incorporated into the system. Next, the chair and wall cabinets were modelled in 3D Studio Max. Finally, the character was developed. In this case they were made from photographs taken of a real person, although it is not always possible to get this amount of detail about suspects. It could be feasible to develop the characters in a similar way to current photo fit images, but this was beyond the scope of this project. Once the basic scene has been created, the witness of the crime could sit in with the software user and describe the witnessed crime, in this case one person kicking another person. Animations are developed until the user is happy with the sequence of events and the characters are incorporated into the scene. Finally, the scene is rendered and viewed from multiple angles – including the point of view from the witness.

Figure 3 Created Scenes

CONCLUSION AND FUTURE WORK

After researching current crime scene recreation techniques it appeared that the main aim was the creation of photo-realistic models. However, the limitations they all face are that the scenes are static; once the scene has been developed it is difficult to alter it by adding objects without generating an entirely new model. Also, the initial process of creating the scene is a very expensive and labor-intensive task. The technology is only suitable for crime scenes that are still in existence, so that if the location has been destroyed through a crime such as arson, the current techniques would not allow the recreation of these scenes. Also, sometimes it may not be necessary to produce a completely photo-realistic model when the actions that took place there are more important than the location. Finally, and most importantly, the major disadvantage with these techniques is that the actual crimes themselves cannot be recreated and rendered. The main advantages of using the system described in this paper are that scenes can be developed very quickly and effectively, without the need for expensive and technical equipment. Using the Crime Scene Creator, it is now possible to alter the created scene rapidly by changing the components and re-rendering the scene. The original scene does not even have to exist and a scene could easily be developed from blueprints, or even the memory of witnesses. Animated characters can be incorporated into the scene, which adds an important dimension to static scenes created previously. Finally, the scenes are rendered in real time and therefore there is no need to wait while an animation sequence is pre-rendered.

The system does have some limitations. The creation of characters is a major drawback, as it requires an extensive knowledge of 3D studio max to develop them, and a level of artistic ability of the creator. An improvement would be to create the characters from within the crime scene generator. This may be achieved with the use of a photo-fit type system that allows users to select appropriate interchangeable body parts. The animation sequences are limited to a few sample actions, and it may be impossible to create a character that is able to perform any potential action described by individuals by the simple use of animation sets. No collision detection has been introduced. At present, in the interests of simplicity, the characters can wander round the scene, walking through objects, walls and other characters. The animation of characters could be improved with the use of a collision detection algorithm and some form of AI that allows characters to interact with each other. Work is currently being undertaken to produce a system similar to Prendinger et al (2002) and Anastassakis (2002) that will allow characters to become emotional, socially aware, and have the ability to reason about tasks. In this way, actions could be ascribed and acted upon, and with the use of realistic emotion expression a very realistic scene reconstruction could be produced.

In addition to character modelling, the level of detail that can be achieved in the crime scene generator is low. Objects can only be placed in a grid area of 500 * 500 pixels; also the generator only allows one item to appear in one grid area. This limits the ability to place multiple small items close together, which may be an important aspect of some crime scenes. Also, the scene creator is incapable of generating uneven terrain *i.e.* steps or slopping surfaces. A better, if more processor-intensive solution may have been to use dynamic triangular points rather that a 2D grid. This would allow a degree of freedom in the creation of the scene; however, it would add a level of complexity to the process.

The most important enhancement that could be adopted in the future would be a Natural Language Processing (NLP) system that allows the users to explain crimes to the system in their own words to increase speed and accuracy of the scene development process. This would involve the use of a speech recognition engine, NLP, and a text-to-speech (TTS) system. The first stage would be to gain a textual representation of the spoken input to the system as Gauvain *et al* (1994) describe. However such systems are still in development. It is still not possible to implement a system that can achieve perfect recognition levels that accepts input from multiple speakers. Therefore, recognition applications have been split into two types: speaker dependent and non-speaker dependent (Lea 1980; Rowden 1992). The former achieve very high accuracy results over large grammars, but they have to be trained by the user before use, and once trained, will only work at acceptable levels for the trainer. Non-speaker dependent systems dos not require users to train the system, but have the disadvantage of only supporting limited grammars and vocabulary. An attempt was made to implement this type of system into this application, with mixed results. The recognition engine used was only capable of accepting American English pronunciation and as a result, recognition levels were low and the system was not fully implemented. However, this type of system has been implemented before. Pausch & Leatherby (1990) created a system that accepted a combination of voice and mouse input to develop technical drawings, which achieved an average of 21.23% speed increase over mouse and keyboard input. The system developed would have to be more sophisticated than a simple command entry system, as a level of understanding would be required. This would be in the form of an NLP application, which would attempt to gain an understanding of naturally spoken text. As the users would enter data in their own words, a level of complexity is added. Spoken text differs in many ways from written text, as it does not always comply fully with grammatical rules. Methods need to be investigated as to which NLP system would be most appropriate. Finally, a TTS system such as that described by IBM (2000) could be introduced to take speech in a textual format and output a synthesised natural sounding voice. To make the speech sound as natural as possible IBM uses a system called Concatenative TTS, that has a dictionary of unique phonemes. When the speech is generated in the synthesiser, the appropriate phonemes are chosen and combined (concatenated) according to linguistic rules generated by a text pre-processor. Theoretically, it is possible to produce any phrase using only these sounds and rules. In this way, the system could be adapted so the characters performing the crimes could also have a voice

REFERENCES

Adams, J. and La Mothe, A. 2002 *Programming Role Playing Games with DirectX*, USA: Premier Press.

Anastassakis, G. Panayiotopoulos, T. Ritchings, T. 2002. *Virtual Agent Societies with the mVital Intelligent Agent System* Knowledge Engineering Lab, Department of Informatics, University of Piraeus, Piraeus, Greece

Aubel, A. and Thalmann, D. 2000 Realistic Deformation of Human Body Shapes, *Proc. 11th Eurographics Workshop on Computer Animation and Simulation*, Interlaken, Switzerland.

Burford, D. and Blake, E. 2001 *Real-Time Facial Animation for Virtual Characters*, Collaborative Visual Computing Laboratory, Department of Computer Science, University of Cape Town.

Dyer, C. 2001 *Volumetric Scene Reconstruction from Multiple Views*, University of Wisconsin, 469-489.

Gauvain, J., Lamel, L., Adda, G. & Adda-Decker, M. 1994 *Speaker-Independent Continuous Speech Dictation* cedex, France.

Giles, M., Ballin, D., Dodsgon, N., 2003 *Intergrating internal behavioural models wih external expression* University College London in Partnership with BTexact Technologies, Ipswich

Herda, L., Fua, P., Plankers, R., Boulic, R. & Thalmann, D. 2000 *Skeleton-Based Motion Capture for Robust Reconstruction of Human Motion*, Computer Graphics Lab (LIG) In: Computer Animation, Philadelphia, PA. http://citeseer.ist.psu.edu/article/herda00skeletonbased.html.

Howard, T. and Murta, S. 2000 *Virtual Environments for Scene of Crime Reconstruction and analysis* Advanced Interfaces Group, Manchester.

Hu, B. and Brown, C. 2002 *Interactive Indoor Scene Reconstruction From Image Mosaics Using Cuboid Structure* The University of Rochester Computer Science Department.

IBM. 2000 *Introduction to TTS.*

Lea, W. 1980 *Trends In Speech Recognition* 1st ed. New Jersey: Prentice-Hall, Inc.

Little Q., Small, E. and Ralph, R. 2000 *Forensic 3D scene Reconstruction*, Sandia National Laboratories.

Murta, A., Gibson, S., Howard, T & Hubbold, W. 2000 *Modelling and Rendering for Scene of Crime Reconstruction: A Case Study*, Advanced Interfaces Group, University of Manchester, Department of Computer Science.

Pausch,R. and Leatherby, J.H. 1990. A Study Comparing Mouse-Only Input vs. Mouse-Plus Voice Input for a Graphical Editor, *Proc. AVIOS `90 Voice I/O Systems Applications Conf.*, 227-231

Prendinger, H. and Ishizuka, M. 2004. *Introducing the Cast for Social Computing: Life-Like Characters*, Department of Information and Communication Engineering, Graduate School of Information Science and Technology, University of Tokyo

Prendinger, H. Descamps, S. Ishizuka, M. 2002. *Scripting the Bodies and Minds of Life-like Characters*, Department of Information and Communication Engineering, Graduate School of Information Science and Technology, University of Tokyo

Rose, C., Bodenheimer, B. & Cohen, M.F. 1998 Verbs and Adverbs: Multidimensional Motion Interpolation Using Radial Basis Functions, *IEEE Trans. Computer Graphics and Applications*, 32-40.

Rowden, C. 1992 *Speech Processing* 1st ed. Maidenhead: McGraw-Hill.

Sequeira, V., Gonçalves, J. & Ribeiro. I. 2001 *3D Reconstruction of Indoor Environments* European Commission, Joint Research Centre, Ispra (VA), Italy.

Steed, P. 2002. *Modeling a Character in 3DS Max*: Texas: WordWare Publishing.