Music Driven Real-Time 3D Concert Simulation

Erdal Yılmaz[1],Yasemin Yardımcı Çetin[1], Çiğdem Eroğlu Erdem[2], Tanju Erdem[2], and Mehmet Özkan[2]

[1] Middle East Technical University, Informatics Institute, nönü Bulvarı 06531, Ankara, Turkey
{eyilmaz, yardimy}@ii.metu.edu.tr
[2] Momentum Digital Media Technologies, TÜBİTAK-MAM, Tekseb Binaları A-206 Gebze 41470 Kocaeli, Turkey
{cigdem.erdem, terdem, mozkan}@momentum.dmt.com

Abstract. Music visualization has always attracted interest from people and it became more popular in the recent years after PCs and MP3 songs emerged as an alternative to existing audio systems. Most of the PC-based music visualization tools employ visual effects such as bars, waves and particle animations. In this work we define a new music visualization scheme that aims to create life-like interactive virtual environment which simulates concert arena by combining different research areas such as crowd animation, facial animation, character modeling and audio analysis.

1 Introduction

Music visualization is a way of seeing music in motion for generating an audio-visual sensation. Water dance of fountains or careographed fireworks are examples of music visualization. It became more popular after mid 1990s when PCs and MP3 songs emerged as an alternative to existing audio systems. Winamp, Windows Media Player and other similar software introduced real-time music visualization. Such systems mainly employ bar, wave and particle animations in synchronization with the beats of the music for visualization. Also there exist some other visualization approaches. Presently it is possible to download virtual dancers who perform prerecorded dance figures on the desktop. In this work, we describe a music visualization scheme, which tries to make the user feel as if s/he is in a real stadium concert. The proposed life-like interactive 3D music visualization scheme combines real-time audio analysis with real-time 3D facial animation and crowd animation.

2 Components of a Concert

In order to create a virtual concert environment, the following main visual components of a concert should be modeled and animated realistically in harmony with the incoming music:

Performer/s and band/orchestra: Concerts are mostly shows in which the performer and the band is the focus. Photo-realistic modelling and animation of the performers is desirable.

B. Gunsel et al. (Eds.): MRCS 2006, LNCS 4105, pp. 379 – 386, 2006.

Audience: Concerts are usually attended by thousands of people and modeling and animation of an audience of such size is a challenging issue.

Stage: Different kinds of shows are performed on stage. Even in a simple concert moving lights, smoke generators, video walls etc. are used. Therefore, modeling such effects and the decoration of the stage can become a complex task.

Environment: To create a realistic concert model, we should also model the environment where the concert takes place. The environment can be an outdoor place such as an open field or it can be an indoor concert hall.

These components are individually described in the next section.

3 Music Driven 3D Virtual Concert Simulation

We envision a virtual concert environment where the only input is an MP3 file and the output is the real-time simulation of a concert that might be considered as an interactive video-clip in which the user freely moves around.

Fig. 1. Level 0 Data Flow Diagram

Concert simulator uses the music file as the main input and at the initialization phase it passes the piece to the audio analyzer for extracting necessary information for concert visualization. The performers, audience, stage, concert arena etc. are all made available for rendering. Following this phase, music begins to play and concert event is rendered according to the outputs of the sub-modules. In this interactive phase, user input for camera position will be managed as explained in Figure 2.

3.1 Audio Analysis

The crowd behaviors and the activities on the stage are highly dependent on the music. Tempo and temporal energy mostly determines the crowd actions in the concert. This behavior could take forms like clubbing, dancing, cheering etc. Hence real-time analysis of the music is crucial for automatic prediction of crowd behaviors and we plan to use its output for realistic concert arena simulation. Such features of music can also be used to animate lights and smoke generators on the stage.

Current studies on audio analysis use several methods to classify music, extract rhythm or tempo information and produce summary excerpts [1]. Self-similarity is commonly used for automatic detection of significant changes in the music [2]. Such information is valuable for segmentation as well as beat and tempo analysis.

Presently MP3 is the most popular format for audio files. The metadata for MP3 provides some hints for audio analysis. Certain fields in the header such as genre, mood, and beat per minute, artist and key contain valuable information when available. The genre field can be used to determine parameters related to the audience

such as average age (kids/young/elder) or attire (casual/formal/rock etc.). Similarly, keywords like “angry”, “sad”, “groovy” can be used to determine audience and band attitudes. MIDI is an alternative audio format that has separate channel for different instruments and hence simplifies audio analysis. Certain instruments such as a drum could enable us to extract the desired information such as rhythm and beat.

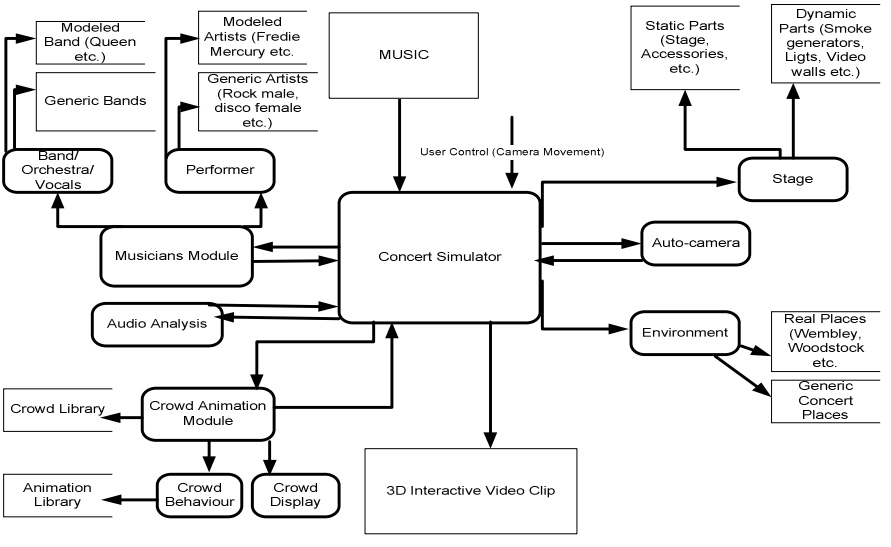

Fig. 2. Detailed Software Architecture

3.2 Modeling the Virtual Performer

The performer and the band are the main actors in concerts. Therefore, they should be modeled and animated as realistically as possible. In this work, we chose Freddie Mercury as the performer of our virtual concert. The face of Freddie Mercury is modeled using the method described in [3]. This method involves algorithms for 2-D to 3-D construction under perspective projection model, real-time mesh deformation using a lower-resolution control mesh, and texture image creation that involves texture blending in 3-D. The 3-D face model is generated using 2-D photographs of Freddie Mercury. Given multiple 2-D photographs, first, the following fourteen locations on the person’s face are specified as the feature points: the centers of the right and left eye pupils, the central end points of the right and left eyebrows, the right and left corners and the tip of the nose, the top and bottom points of the right and left ears, and the right and left corners and the center of the lips. The locations of feature points are manually marked on all images where they are visible. Given the 2-D locations of the feature points in the neutral images where they are visible, the 3-D positions of the feature points of the person’s face are calculated using a modified version of the method in [4].

The estimated 3-D positions of the feature points of the person’s face are used to globally deform the initial geometry mesh to match the relative positions of the feature points on the globally deformed geometry mesh and to match the global

proportions of the person's face. Following the global adjustments made to the face geometry mesh, each and every node of the geometry mesh is attached to, and hence controlled by, a triangle of a lower-resolution control mesh. Once the geometry mesh is attached to the control mesh, local modifications to the geometry mesh are automatically made by moving the nodes of the control mesh. The results of the above algorithm are very realistic due to the following novelties in the modeling and animation methods: (1) an iterative algorithm to solve the 3-D reconstruction problem under perspective projection, (2) a 3-D color blending method that avoids the problem of creating a single 2-D sprite for texture image, and (3) attachment of geometry mesh to a lower resolution control mesh and animation of the geometry mesh via the control mesh and actions. The created 3-D Face Model of Freddie Mercury is given in Figure 3. We can see that the 3D model is quite realistic.

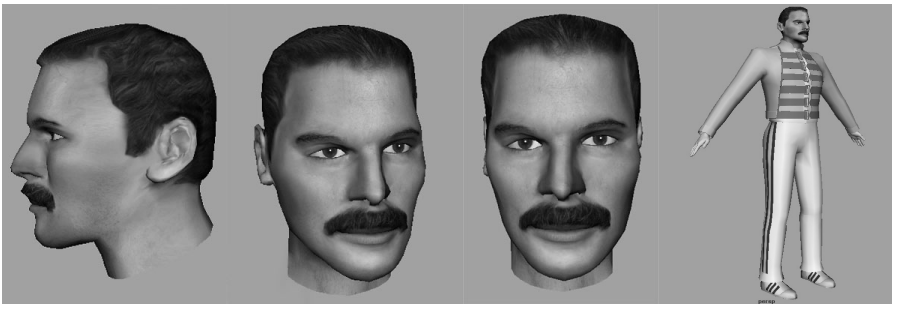

Fig. 3. The generated 3-D head model of Freddie Mercury and Full 3-D model

The generated 3D model of Freddie Mercury will be incorporated into the virtual concert environment in two phases. In the first phase, the head will be animated on a virtual video wall in the concert area, where the lips will be moving in synchronization with the lyrics of the song. In the second phase, the full model of the artist including the body model will be animated on stage. The full 3-D model of Freddie Mercury is also given in Figure 3.

3.3 Animating the Virtual Performer

Creating realistic facial animation is one of the most important and difficult parts of computer graphics. Human observers will typically focus on faces and are incredibly good at spotting the slightest glitch in the facial animation. The major factor giving the facial animation a realistic look is the synchronization of the lips with the given speech. In order to create a realistic virtual singer we will animate the lips of the computer generated 3-D face model of the singer, with the given lyrics of a song.

The approach followed for this task is to use the phonetic expansion of each spoken word in terms of phonemes and to estimate the phoneme boundaries in time for the given speech data. That is, the given speech data is aligned with its phonetic expansion. Then, each and every phoneme duration is mapped to a visual expression of the lips, which are called visemes and the 3D face model is animated using this sequence of visemes. Animation is done through mapping of every viseme to a pre-defined 3D face mouth shape and transformation between the shapes.

However, without any postprocessing, the estimated viseme animation may not be natural looking, since there may be too much jittery movement because of sudden transitions between neighboring visemes. In fact, during natural speech, there is a considerable interaction between neighboring visemes and this interaction results in certain visemes to be 'skipped' or 'assimilated'. This phenomenon is called as coarticulation. Therefore, we post-process the estimated visemes to simulate the effects of coarticulation in natural speech. In this post-processing step, which is based on a set of rules, visemes are merged with their neighbors depending on their audio-visual perceptual properties. For example, the phonemes "p,b,m" correspond to closed-lip visemes and should be estimated carefully, since incorrect animation of these phonemes is easily noticed by an observer. On the other hand the viseme corresponding to the phoneme "t" can be merged with neighboring visemes, since it is a sound generated by the tongue position with little or no motion of the lips.

3.4 Crowd Animation

Crowd animation is a popular research topic in computer graphics community since it has already reduced costs and helped adding thousands of realistic creatures or people in Hollywood productions such as "Lord of the Rings", "Narnia" and "Troy". In all such productions, crowds in the battle scenes are realized by using computer generated soldiers. This trend is expected to continue with the investment of several companies in crowd animation. In these productions, crowds are visualized on high-end workstations and they are all pre-rendered. Real-time crowd animation is the main challenging issue in this field.

Several studies in the literarture have achieved real-time visualization and animation of crowds up to few thousand [5,6]. These crowds contain few base human models and the rest of the crowd is mainly clones of these base models. Texture and color variations are used to increase the variety and decrease the sensation of duplicated avatars [7]. Another characteristic of these models is the limited animation capability since they are mostly designed to realize few actions such as walking, sitting and running.

In this work we try to visualize a crowd of up to 30.000 people in real-time by using COTS hardware and a good blend of well known techniques such as Level of detail (LOD), frustum-culling, occlusion-culling, quad-tree structure and key-frame animation. 3D human models in this study contain up to 5000 polygons and use 1024*1024 photo-realistic texture maps. In the near future, we plan to add more polygons and smoother animations to the models that are closest to the camera which can be classified as Level 0 in our LOD structure.

Figure 4 illustrates the working model of Crowd Animation Module (CAM). CAM accepts three inputs. First input is generated by the initialization module only at the initialization phase. This input covers everything about the user preferences and music meta-data such as the number of audience, concert place (stadium/ auditorium/concert hall etc.), music genre, music mood, average tempo/beat etc. CAM gets this input and determines the details about the general structure of the crowd such as age group, typical attire, gender etc. Considering these parameters, base models from human model library are chosen and related texture maps are extracted. Audience variation is realized by applying various texture maps to the same model. Also, each model is processed to be unique in shape by changing its height, width and depth properties. In order to eliminate run-time preparation of texture

mipmaps, we used commercial JPEG2000 software library and extracted lower resolution sub-textures in this phase. This action saves process time and offers better-quality texture maps by using less storage.

Second input of CAM is camera position information, which can be modified by the user or via auto-camera control. In this study, user has the capability of changing the virtual camera at any time he/she desires. This capability gives both the feeling of interaction with the scene and the freedom of moving in the concert arena. At the same time, AI controlled auto-camera mode changes the camera position if it is enabled. This camera automatically focuses on important events such as drum attack, guitar solo or attracting movements of the audience.

CAM uses camera position at every rendering time and avoids sending audience models that are not visible to the graphics pipeline. In order to save process-time human models those are away from the camera are marked with special flags which minimize future controls if the camera stays steady in the following rendering/s. Since it is possible for the audience to move and change their position people that are closest to the camera are processed at every frame even though the camera position does not change. In this study, currently we use 6 LOD for human models that are decreased by 35% at each level so that the number of polygons in the model ranges from a few hundred to 5000. These LOD models are all pre-rendered and loaded to the memory at the initialization phase to minimize the initialization time, which a typical music listener can not bear if it exceeds few seconds. Other well-known techniques such as frustum-culling, limited occlusion-culling and back-face culling are also used to increase rendering performance.

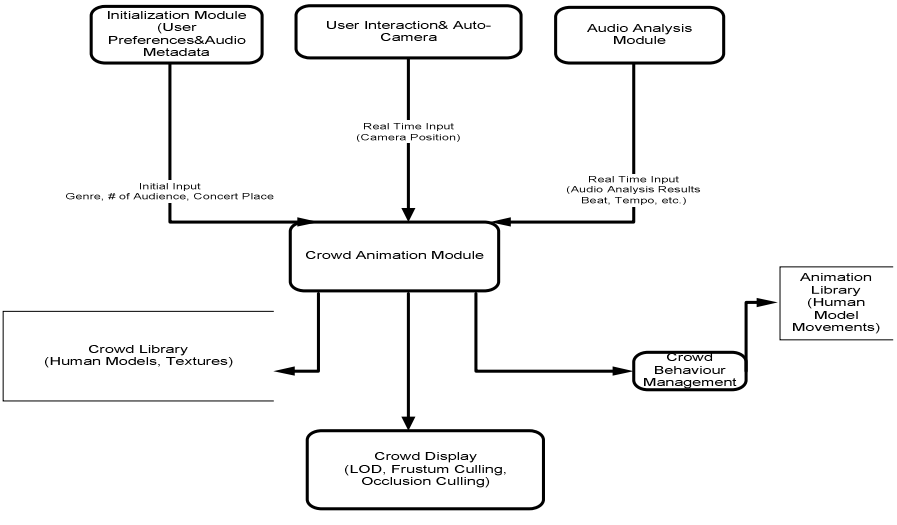

Fig. 4. Data flow diagram of the Crowd Action Module (CAM)

Virtual bounding boxes that cover audience groups, which are organized in an effective quad-tree structure decreases the processing time significantly and helps in achieving better frame rates. Our current test results give promising frame rates considering the developments in the graphics hardware.

Fig. 5. Screenshots of Crowd Animation

Third input of CAM will be the results of Audio Analysis Module such as tempo, rhythm, beat, silence etc. This information, if succesfully extracted, is planned to be use by Crowd Behavior Management (CBM) in order to visualize human models that move according to the music. This part covers group and individual behavior models in a concert event. We analyze large collection of concerts and try to extract significant and typical actions in the concerts and find relations between the music and actions. Some typical actions are moving hands slowly or raising hands and clubbing with the drum beats or jumping with the same frequency of the other people around. In fact, each individual has the potential of performing unpredictable action at any time [8]. At this stage of our, work we are only capable of relating music metadata and some actions in the animation library. We are also planning to build a human model motion library by the help of graphics artists. Although an artist-made animation library serves our principal goals, an ideal and realistic motion library of human actions in a concert should be produced by using motion capture equipment.

4 Conclusions

We completed parts of the system described above and we are currently merging these parts to complete the first phase of the virtual concert environment, which consists of the crowd, the concert arena and the virtual performer singing in the center video wall. The final system will be able to automatically convert a music file into a fully interactive and realistic concert simulation. We believe that this study will be a good basis for next generation music visualization, which will consist of real-time computer generated music videos.

Acknowledgements

This study is supported by 6th Frame EU Project : 3DTV Network of Excellence and METU-BAP “Virtual Crowd Generation” 2006-0-04-02.

References

1. Foote, J.: Automatic Audio Segmentation Using A Measure Of Audio Novelty. Proceedings of IEEE International Conference on Multimedia and Expo, Vol. I. (2000) 452-455
2. Foote, J., Cooper, M.: Visualizing Musical Structure and Rhythm via Self-Similarity. Proceedings of International Conference on Computer Music (2002)

3. Erdem, A.T.: A New method for Generating 3D Face Models for Personalized User Interaction. 13th European Signal Processing Conference, Antalya (2005)
4. Tomasi, C., Kanade, T.: Shape and Motion from Image Streams under Orthography, A Factorization Method. International Journal of Computer Vision, Vol. 9. (1992) 137-154
5. Tecchia, F., Loscos, C., Chrysanthou, Y.: Visualizing Crowds in Real-Time. Computer Graphics Forum, Vol. 21 (1996) 1-13
6. Dobbyn, S., Hamill, J., O'Connor, K., O'Sullivan, C.: Geopostors, A Real-Time Geometry/Impostor Crowd Rendering. Proceedings of the 2005 symposium on Interactive 3D graphics and games (2005) 95-102
7. Ciechomski, P.H., Ulincy, B., Cetre, R., Thalmann, D.: A Case Study of a Virtual Audience in a Reconstruction of an Ancient Roman Odeon in Aphrodisias. Proceedings of the 2005 symposium on Interactive 3D graphics and games (2005) 103-111
8. Braun, A., Musse, S.R., Oliveria, L.P.L.: Modeling Individual Behaviours in Crowd Simulation. CASA 2003 - Computer Animation and Social Agents (2003) 143-148