Shadow Volumes for MPEG-4 Applications

DIPL.-ING. HELGE DRUMM
DIPL.-ING. STEPHAN KUSSMAUL
Technische Universität Ilmenau, Institut of Media Technology (IMT)
PF 10 05 65, Helmholtzplatz 2, 98684 Ilmenau
GERMANY

Abstract: This manuscript is a short summary of the recent action researches to scene realism in a MPEG-4 environment. The special area, that was regarded here, is the creation of real time shadows based on the method of shadow volumes. Shadow volumes are well-known an used in the present computer games. The new approach in this paper is the attempt for standardization for MPEG-4 real-time applications. Additionally also refinements of this method were executed, by the detailed view of the scene management.

Key-Words: shadow, graphics, MPEG-4, virtual, reality, interactive, scene, realism

1 Introduction

The method to create shadows with shadow volumes for real-time applications, was introduced by Crow and improved by Heidmann in [1] *„Real Shadows Real Time“*. Heidmann used the Stencil Buffer for the calculation of the inner geometry of shadow volumes.

The fundamental algorithm is based on utilization of point lights, supported by OpenGL. Shadow volumes normally only can be applied on polygonal geometry. In recent time, McCool published another approach in [2] *„Shadow Volume Reconstruction from Depth Maps“*. This algorithm slightly changed the basic algorithm, so that it can be used for arbitrary objects. This caused an improvement for independence on object properties.

Nowadays, shadow volumes for real-time rendering are a used method for shadow creation. The content of this paper describes the results of recent action researches for applications with shadow volumes in MPEG-4 environments. The research project is called „**I**nteractive **A**udio**v**isual **A**pplication **S**ystems” (IAVAS) and executed on the Technical University Ilmenau (Germany) Institute of Media Technology (IMT).

Principle purpose of work was the standardisation of shadow volumes as a node in MPEG-4 systems part. Even for this where important aspects for instance performance and improvement of visual quality for shadow volumes. The results of our work show one possibility of standardisation in connection with proposals for robust implementations.

2 Algorithms for Shadow Volumes

Shadow volumes are a volumetric representation of shadows. Shadow covered objects are situated inside of this volume. The volume is created by the lightsource and shadow casting geometry as seen from the

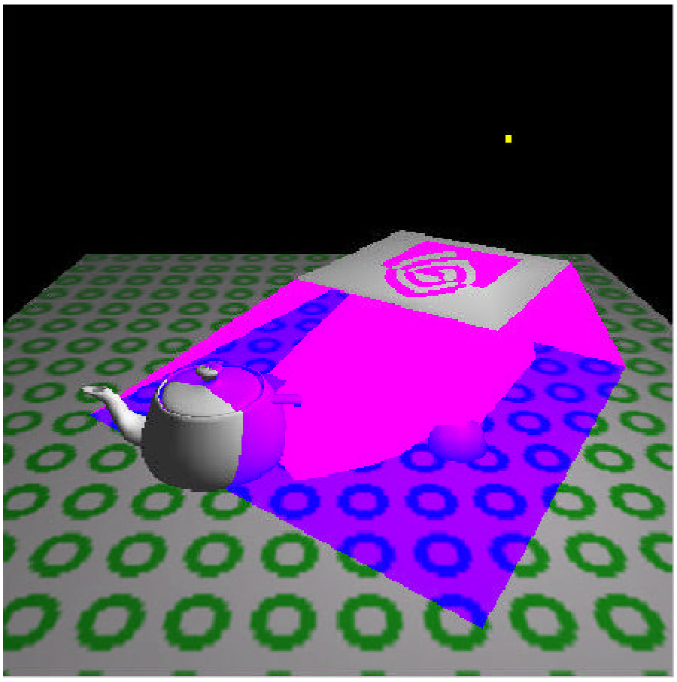

Fig. 1: NVIDIA-logo creates Shadow Volumes

lightsource. Depending upon later use, you should clip the volume on view frustum ore scene geometry. The shadow volume is a polygonal object, however not rendered into the framebuffer. Thus it is not visible for the viewer.

The precondition for the calculation of the shadow casting geometry is the calculation of silhouette. A very good and fast approach was applied in [3] by Bestimt und Freitag in *„Real-time Shadow Casting Using Shadow Volumes“*. In this case for each polygon the relationship with its neighbours is calculated. The “connectivity” of each polygon is stored in an internal list. A possible way the calculation temporally meaningfully to execute, would be during the loading procedure of geometry. The results are then stored, e.g. into the 3D-format.

For calculation the silhouettes based on connectivity of the neighbouring polygons is to do a visible test at first. Only visible polygons are able to be a part of the silhouette. If polygons are visible in opposite to their neighbours, is concluded, to get an edge of silhouette. If there's no other neighbouring polygon you get an edge of silhouette too. For objects, with exchanging geometry, its need to recalculate the connectivity for each frame, e.g. level of detail objects.

If you've got the silhouette, you are able to project their edges as seen from light into the scenery. In the next step, is to decide how deep to project the volume. One problem of shadow volumes is caused by their geometric penetration of the whole scenery toward infinity. It can happen, that non-shadow casting objects, suddenly cast shadows. So you get bad visual artefacts. A simple method to avoid this lack was introduced by clipping the shadow volumes to the viewing frustum. A better way would be a clipping to the scene geometry, but currently it's not possible. The calculations are much intensive and slow down the performance immediately, you have objects with high polygon count. A performance gain, you would get by using GPU support, e.g. calculations by vertex shader. Finally, another simple, but non the best way is the clipping to the skybox. Commonly shadows on the skybox are not wished.

Heidmann introduced in its publication [1] an important improvement of shadow volumes called „Stencilled Shadow Volumes". The stencil test fragment operation in connection with depth test is a very comfortable tool, to recognize whether a fragment is situated inside or outside of a shadow volume. The basic principle for using stencil buffer in this case, is to render the front facing polygons of shadow volume and the back facing polygons of all and to count by increment the stencil buffer for front and decrement for back facing, if fragment passed the normal Z depth test.

A detailed description for using the stencil buffer you find in Kilgards [4] *„Improving Shadows and Reflections via the Stencil Buffer"*.

The camera is inside the shadow volume, maybe is the most difficult challenge for implementation. A solution is the capping of shadow volumes to the near clipping plane. Introduced by Diefenbach, an approach for this method you can find in [5] *„Interaction and Realism through Hardware-based Multi pass Rendering"* and Bestimt, Freitag in [3]. The algorithms are not robust, finally. John Carmack developed the „Reverse Approach". The clipping is now a problem of far plane and shadow volumes are now capped in a finite distance. A detailed description and more thoughts to this topic you find in publication of Mark Kilgard [6] *„Robust Stencil Shadows"*.

3 Optimising Shadow Volumes

Fundamental possibilities for optimising shadow volumes take for granted in the scene management. Finally, shadows are not view dependent. For static objects together with static lights you only need to pre-calculate and cap the shadow volumes. If dynamic changes happens, i.e. the object itself changes its position or geometry or another object moves and intersects the shadow volume, need to be done a re-calculation.

Further optimisations are level of detail for silhouettes. Naturally, to avoid artefacts the LOD-shadows dependent on their details need to be far away from the viewpoint. This method was analysed and implemented in IAVAS-project. The results show a considerable performance gain. For reduction of silhouette resolution is used a method from Stan Melax, published in [7] *„A Simple, Fast und Effective Polygon Reduction Algorithm"*. It's a variation of *„Progressive Mesh-Algorithm"* [8] from Hoppe. This method collapses polygon edges based on the smallest loss of objects topology.

Based on this, you are able to create infinite detail levels from silhouettes. But view dependent level of details, causes a dynamic re-calculation for polygon connectivity. With such initial conditions, any kind of GPU support is absolutely necessary to enable real-time rendering for complex geometries. Further inquiries are needed, maybe by support of vertex shader.

Not only for this, vertex shader are currently best practise for hardware acceleration based on vertex calculations. The extrusion of shadow volumes is possible with GPU support too. Dave Kirk shows with its demo application [9], how to use vertex shader with DirectX 8 for generation. A complete generation with GPU support, including LOD, will realize in the next time. Same implementations are possible by using OpenGL API.

Finite shadow volumes offer an additional approach for optimisation, i.e. the influence of light source. With OpenGL command glScissor you are able to limit the influence of light source on to the screen area. This includes the area of shadow casting too.

Future GPU developments let expect further tremendous accelerations. More and more calculations will be done with by GPU support and additionally buffers give more freedom for storage of intermediary results.

4 Soft Shadows

Soft shadows are much more realistic, since they show the rendered shadows are not geometry. In reality sharp shadows are an exception and they only happen with direct sunlight. Currently two different methods are applied to create soft shadows. The first technique is the method of “Jittering light”. The light source is moved and shadow rendered as often as more soften the shadow should be. The result is added into accumulation buffer. Still better results you get with filtering of created shadow afterwards. A simpler method is to be done completely without jittering lights and only execute the filtering.

The results are not as closed to reality, since they do not consider the distances and the expansion of light source for calculations.

5 Shadow Fade- In and Out

The sphere of light source determines the spread of shadow. That means, which objects abandon the influence of light source should not lose their shadows suddenly. For this object property, it’s necessarily to fade in and out the shadows. Each concerned scene geometry, must have in addition its own brightness level for their own shadow and adding it into the memory to the final shadow.

6 Shadow Volumes based on “Shaped Video”

In the future different media will merge into a multimedia data stream, as required for MPEG-4. MPEG-4 defines a shaped video object, which is very simplified expressed a surface with a video texture and a mask channel for transparency information in its standard. The typical example would be for it a moderator before a blue wall, is integrated in virtual scenery. For this, shadow volumes seem to be completely unsuitable. But there’s a solution, based on the approach from McCool [2] for generation of shadow volumes with depth images. McCools technique can be called as a hybrid method between shadow maps and shadow volumes. We don’t use deph-buffer pictures, but alpha-channel. First the silhouette information is extracted by means of edge extraction filters and set an edge flag based on the alpha channel amounts. With these conditions, now a shadow volume can be provided.

7 Shadow Volumes in MPEG-4

The Intension for the definition of the MPEG-4 standard was situated in the standardised description of multimedia content. Today usually proprietary formats were used, in order to describe multimedia content. Further more, the standardized transfer of multimedia contents by streaming, represents a large challenge. In MPEG-4 by the standardisation of these processes, the prerequisites for it were created.

One part of the Mpeg-4 standard is the systems part, which defines the **BI**nary **F**ormat for **S**cenes (BIFS) referred to in the further. The MPEG-4 standard took itself the VRML2.0 standard as basis for the definition of the scene graph.

The “complete scene graph profile” provides the complete set of scene graph elements. The Complete scene graph of profiles wants enable applications like dynamic virtual 3D world, interactive 3D television and games. At present, in this profiles still no definition of shadows, which includes also the method of shadow maps.

In this section now a possible beginning is to be introduced for the integration of shadow volumes. The suggestion is regarding its flexibility and the necessity to change already existing nodes, e.g. the Shape node, for the implementation suitably.

7.1 Proposal for a Separate Shadow Volume Node Description

In the first beginning for the definition of shadows, it should be permitted all by this node produced shadows at one time switch on and off. This is implemented by the field `eventIn` on with the values `TRUE` or `FALSE`. With a separate shadow node you get a maximum at flexibility. Separate means, this node is not a child of an existing node. The shadow volume node itself includes a list of the shadow creating lights with identical characteristics in field `exposedField MFNode Lights`, a lists of the objects with existing properties in field `exposedField MFNode Shapes`, and in order to the shapes there shadow properties defined in the fields `eventIn MFBool ShadowCaster`, `eventIn MFBool ShadowReceiver` and `eventIn MFBool SelfShadow`. The lists of the type `MFNode` contain the defined names of the objects and light sources and are thus unique identified. By the use of `exposedField` you are able to transmit and received data. I.e. the lists of the lights and shadows can be defined also dynamically, if it is required.

The field `MFFloat radius` has the function as described in chapter 1.4 to the fade- in and out shadows. By routing the radius of influence from light source, this field is directly controlled.

The aspect of the soft shadows was integrated in the node likewise. You find the technique of „Jittering Lights“ in `eventIn SFInt jitteringLights 16 #[0,50]`. The default value is intended by 16 shifts. The filtered shadows were considered by the field, `eventIn SFFLoat filterShadow 0,1 #[0,1]`.
Even at present only with very fast GPUs and processors

is possible to calculate soft shadows, should these for future development play a role nevertheless and taken up to the standard.

The method of the reduction of the silhouettes is an important beginning to the real time ability and find its representation in `eventIn SFFloat LODShadow 1,0 #[0,1]`.

The possibility of an infinitely variable controlling for the LOD is given. Thus also a view-dependent solution of the silhouette would be possible.

The fields `eventIn SFFLoat darkness 0,5 #[0,1]` and `eventIn SFBool FALSE # strongestLight` or `nearestLight` are optionally possible. They deliver support for optimisation to avoid multiple render passes. The shadow colour and brightness should not be made accessible necessarily. These calculations should be intercepted by the used algorithm.

```
ShadowVolume {
  #Switch on/off
  eventIn on TRUE

 #Definition of lightSources and geometry
  exposedField MFNode Lights children
  exposedField MFNode Shapes children
  eventIn MFBool ShadowCaster TRUE
  eventIn MFBool ShadowReceiver TRUE
  eventIn MFBool SelfShadow TRUE

  #Parameters of Shadow
  exposedField SFFloat radius 1.0

  #or defined by ROUTE
  eventIn SFFLoat filterShadow 0.1#[0,1]
  eventIn SFInt jitteringLights 16#[0,50]
  eventIn SFFloat LODShadow 1.0 #[0,1]

  #optional parameters
  eventIn SFFLoat darkness 0.5 #[0,1]
  eventIn SFBool strongestLight FALSE
  #or nearestLight
}
```

8 Conclusion

During the research became clear that shadow volumes have a lead in the spreading for the moment opposite to shadow map algorithm, to which not least contributed the possibility of the robust implementation by the beginning of Carmack and Kilgard [9] and [10]. Since the generation of Shadow map has a technical lead by the implementation in hardware, is able to change the spreading quite fast. At the shadow volumes the strongly increased realism of a scene is positive, without having to fall back thereby to texture manipulation. On the other side one must deal with oneself at programming with the support of the stencil buffers, filling rate limitations, sharp edges, complex scene management and exceptions such as shaped video, which does not make the implementation simple even.

The integration of shadows into Mpeg-4 should be one of the next steps for standardisation. The presented suggestion for a shadow volumes node is implemented at present and is appropriate for future applications.

References:

[1] Heidmann, T.; *Real Shadows Real Time*, IRIS Universe, Number 18, 1991, pp. 28-31

[2] McCool, M.; *Shadow Volume Reconstruction from Depth Maps*, ACM Transactions on Graphics, Jan. 2001, pp. 1-25

[3] Bestimt, J. and Freitag, B.; *Real-time Shadow Casting Using Shadow Volumes*, Gamasutra.com web site, Nov. 15, 1999

[4] Kilgard, M.; *Improving Shadows and Reflections via Stencil Buffer*, Advanced OpenGL Game Developer course notes, Game Developer Conference, March 16 1999, pp. 204-253. Nvidia Webpage http://developer.nvidia.com/developer

[5] Diefenbach P.; *Pipeline Rendering: Interaction and Realism through Hardware-based Multi-pass Rendering*, Ph.D. thesis, University of Pennsylvania, 1996, tech report MS-CIS-96-26

[6] Kilgard, M.; *Robust Stencil Shadow Volumes* CEDEC 2001 presentation, Tokyo, Sept. 4, Nvidia Webpage, 2001, http://developer.nvidia.com/developer

[7] Melax, S.; *A Simple, Fast and Effective Polygon Reduction Algorithm*, Game Developer, , Nov. 1998, vol. 5, no. 11, pp. 44-49

[8] Hoppe, H.; *Progressive Meshes*, Computer Graphics (SIGGRAPH ’96 Proceedings), 1996, pp. 99-108

[9] Kirk, D.; Demo for Shadow Volumes realized with vertex shader (DirectX) Source: Nvidia Webpage, http://developer.nvidia.com/developer

[10] Carmack J.; Corresponding over Shadow Volumes with Mark Kilgard, 2000, publish by Kilgard in 2002

[11] Kilgard M.; *Practical and Robust Stenciled Shadow Volumes for Hardware-Accelerated Rendering*, Nvidia Webpage, 2002