Streaming VRML

Jaroslav Krivánek

Overview

- MPEG4 Binary Format for Scene (BIFS)
- MPEG4 version 2 Extensions - Progressive 3D Meshes
- Guéziec, Taubin, Horn: A Framework for Streaming Geometry in VRML, 1998

MPEG4 Binary Format for Scene (BIFS)

Introduction

- BIFS = Binary Format for Scene
- A coded representation of a parametric scene description format similar to VRML
- Scene is
  - 2D & 3D Geometry
  - Texture
  - Audio
  - Video
  - Text

Introduction (contd.)

- Uses slightly modified VRML grammar
  - ROUTE specifications at the end of the scene
  - There must be one top level grouping node
- Specifies binary coding for all elements of the language

Introduction (contd.)

- Nodes
  - Those known from VRML97 + some new nodes
- Nodes for geometry specification = VRML97
- No direct support for progressive triangular meshes
- No direct support for view dependent LOD
- View dependent transmission of textures and geometry is handled by the “View Dependent Object” (Part 1, Annex C)

Binary Syntax Overview

- Binary encoding of the scene graph
- Nodes are encoded depending on the context in which they appear
- Fields may be quantized
- Quantization may be controlled by the **`QuantizationParameter`** node
- ROUTE specification is at the end of the scene

Binary Syntax Overview (contd.)

- Coding follows the depth first order
- Each node may have a **`nodeID`**
  - ~ DEF in VRML
  - Number of bits specified in the terminal configuration (which is conveyed before the scene description itself)

Terminal Configuration

- Configuration information is sent before the scene description
- Specifies
  - Number of bits for **`nodeID`**
  - Number of bits for **`routeID`**
  - Whether the stream is Command Stream or Animation Stream
  - Metric (pixel x meter)
  - …

Node Identification

- Is context dependent
- Each field that accepts nodes as children has a **Node Data Type** (NDT) assigned to it
- Exact node identification is computed as **`currentNDT[node_ident]`**
- Example: **`Anchor`**
  - 5bit code 0b00001if the parent field is SF2DNode
  - 7bit code 0b0000001 if the parent field is SFWorldNode

Field Indexing

- The code used to identify a node's field depends on the way the field is referenced
- Referencing modes
  - **`defID`** – fields with values set at initialization
  - **`inID`** – can be modified from outside
  - **`outID`** – can be output from the node
  - **`dynID`** – can be animated using BIFS-Anim
  - **`allID`** – all events and fields of the node ???

Example of node encoding

Anchor	SFWorldNode SF3DNode SF2DNode					0000001 000001 00001		
Field name	***Field type***	***DEF id***	***IN id***	***OUT id***	***DYN id***	***[m, M]***	***Q***	***A***
addChildren	MF3DNode		000					
removeChildren	MF3DNode		001					
children	MF3DNode	00	010	00				
description	SFString	01	011	01				
parameter	MFString	10	100	10				
url	MFURL	11	101	11				

- Lengths of fields' codes is given by the number of fields, which can be referenced by a given mode

Quantization

- Field values may be quantized
- The quantization can be controlled by the **`QuantizationParameter`** node
- Each field belongs to one of the "Quantization categories" (e. g. 3D position, normal, rotation, …)

`QuantizationParameter`

- **`QuantizationParameter`** node sets the quantization parameters for subsequent nodes in the scenes
- For each quantization category, following parameters are set:
  - On / Off
  - Min / Max values
  - Number of bits

TermCap Node

```
TermCap {
  eventIn     SFTime   evaluate
  field       SFInt32  capability   0
  eventOut    SFInt32  value
}
```

- Values can be ROUTEd to **`Switch`** node
- Scene presentation dependent on device capabilities

MPEG4 version 2 Extensions - Progressive Meshes

`Hierarchical3DMesh`

```
Hierarchical3DMesh {
 eventIn      SFInt32  TriangleBudget
 exposedField SFFloat  level
 field        MFString url []
 eventOut     SFBool   doneLoading
}
```


`Hierarchical3DMesh` (contd.)

- Multi-resolution polygonal model
- Smooth transition between LODs (Geomorph)
- Hierarchical transmission through independent elementary stream

3D Mesh Coding (3DMC)

- Basic Method
  - manifold model
  - incremental representation of single resolution 3D model
- Options
  - support for degradation control
  - support for nonmanifold model
  - support for error resilience
  - hierarchical transmission of levels of detail

3D Mesh Coding (contd.)

- Single resolution Mode
  - Based on *Topological Surgery*
- Hierarchical Mode
  - Based on *Progressive Forest Split*

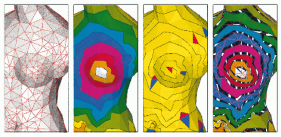
Topological Surgery

- Vertex positions
  - delta coding + quantization
- Delta coding
  - predict the vertex position
  - store the correction value
- Prediction the position
  - weigted sum of ancestors in the VERTEX TREE

Topological Surgery

- Vertex tree

- "Orange peeling"



Progressive Forest Split

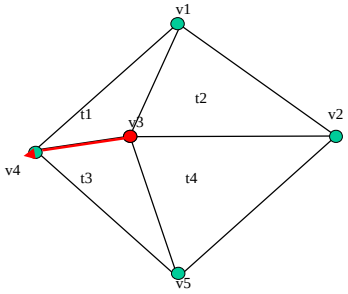
- *base 3D mesh* followed by a sequence of *forest split* operations.
- base 3D mesh is encoded as a single resolution 3D mesh using the Topological Surgery scheme.
- supports smooth transition between levels of detail (geomorph)

Guéziec, Taubin, Horn: A Framework for Streaming Geometry in VRML, 1998

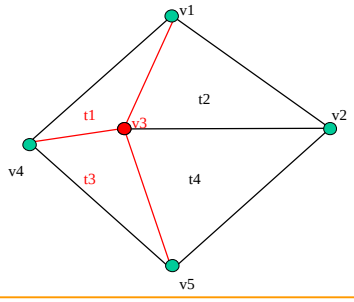
Overview

- Data stucture for representation of progressive LOD
- Can use any polygon reduction algorithm
- Imlementation using VRML PROTO and script in Java is given
- Basic term: vertex representative

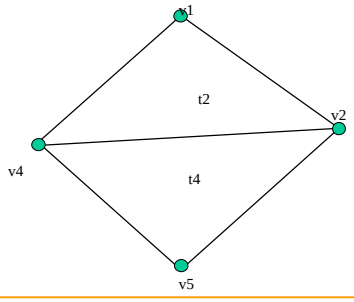
Simplification



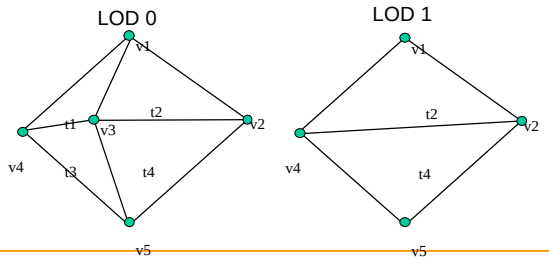
Simplification



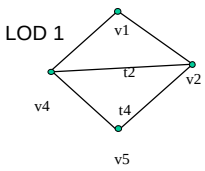
Simplification



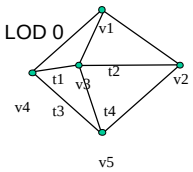
Simplification



LOD representation



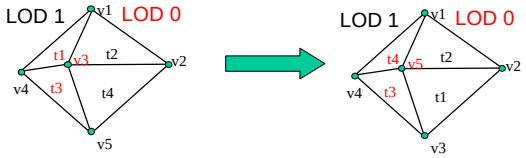
v1, v2, v4, v5
t2, t4



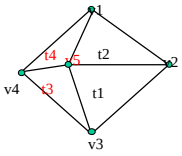
LOD 1 + v3
LOD 1 + t1, t3

LOD representation

- Sort the triangles and vertices of the original mesh according to the LOD they belong to
- Rename the vertices and triangles



Simplification



Representation of LOD 1:
vertices:
v1=(0,1), v2=(1,0), v3=(0,-1), v4=(-1,0)
triangles:
t1=(v2,v3,v5), t2=(v1,v2,v5)
representatives:
v5 is represented by v4

Representation of LOD 0:
vertices:
v5=(0,0)
triangles:
t3=(v5,v3,v4), t4=(v1,v5,v4)
no trepresentatives

Simplification

Representation of LOD 1:
vertices:
v1=(0,1), v2=(1,0), v3=(0,-1), v4=(-1,0)
triangles:
t1=(v2,v3,v5), t2=(v1,v2,v5)
representatives:
v5 is represented by v4

Representation of LOD 0:
vertices:
v5=(0,0)
triangles:
t3=(v5,v3,v4), t4=(v1,v5,v4)
no trepresentatives

In ascii:

```
# vertices for LOD 1
{ 0,1,   1,0,   0,-1,   -1,0}
# triangles and representatives
# for LOD 1
{2,3,5,   4,   1,2,5}

# vertices for LOD 0
{ 0, 0 }
# triangles and representatives
# for LOD 0
{5,3,4,    1,5,4}
```


References

- ISO/IEC JTC1/SC29/WG11, *CODING OF MOVING PICTURES AND AUDIO*, March 2001, (MPEG 4 Standard).
- G. Taubin and J. Rossignac: *Geometric Compression through Topological Surgery*, ACM Transactions on Graphics,1998.
- Taubin et al. :*Progressive Forest Split Compression*, SIGGRAPH, 1998.
- Guéziec, Taubin, Horn: *A Framework for Streaming Geometry in VRML*, IEEE CG And Applications,1998 (see demos at http://www.gueziec.org/).

This is the end!

Thank you for attention.