3-D Model Coding

2003. 10. 15.

Yo-Sung Ho

Kwangju Institute of Science and Technology

Contents

- Introduction
- Representation of 3D Models
- 3D Model Encoder
- Adaptive Optimal Quantization
  - Optimal Uniform Quantization of r
  - Subdivision Quantization of (θ,ϕ)
  - Multiple Component Coding
- Simulation Results
- Conclusions

Introduction

K-JIST

MPEG-1 : Storage of A/V Information on CD-ROM
MPEG-2 : Generic Coding for Digital TV and HDTV

⇩

Multimedia Applications
(Internet Service, Virtual Reality)
2D/3D Computer Graphics

⇩

MPEG-4 SNHC(Synthetic/Natural Hybrid Coding)

Representation of 3D Models

K-JIST

Connectivity Information
(Identifier List of Vertices)

$C_l = \{v_i, v_j, v_k\}$

i, j, k : Vertex Index for Each Triangle

Vertex Position
(3D Floating Vectors)

$V_i = \{x_i, y_i, z_i\}$

Photometry Information
- Normal Vectors
- Colors
- Texture Values

Geometry Compression

K-JIST

- Surfaces of a 3D model are described by very dense triangular meshes in most applications (1 million vertices/model)
- Storage, Manipulation, Transmission and Rendering need large memory, bandwidth, processing time and resources.

⇩

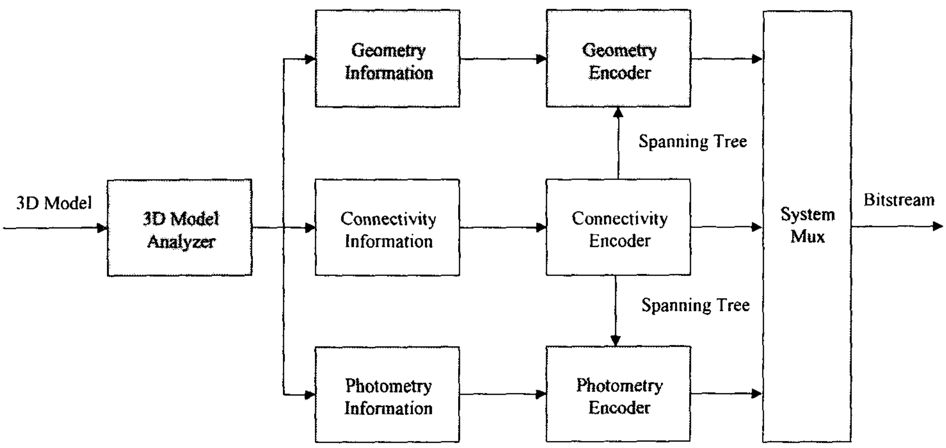
Geometry data contains a large amount of data
(32bits x 3 = 96 bits/vertex)

⇩

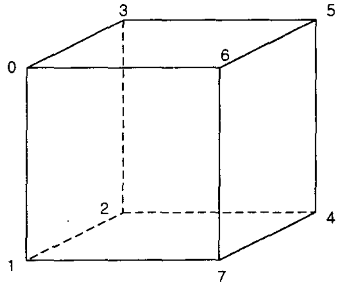
Geometry Data Compression

3D Model Encoder

K-JIST



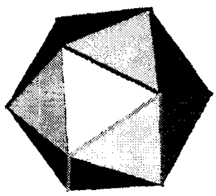
VRML File Format



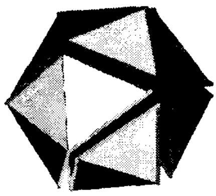

```
Coordinate3{
  point[
    1.0  -1.0   1.0,  #0
    1.0  -1.0  -1.0,  #1
   -1.0  -1.0  -1.0,  #2
   -1.0  -1.0   1.0,  #3
    1.0   1.0  -1.0,  #4
   -1.0   1.0  -1.0,  #5
   -1.0   1.0   1.0,  #6
    1.0   1.0   1.0,  #7
    ]
}
```

```
IndexedFaceSet{
  coordIndex[
      2, 1, 0, -1,
      2, 0, 3, -1,
      4, 1, 2, -1,
      4, 2, 5, -1,
      5, 2, 3, -1,
      5, 3, 6, -1,
      6, 3, 0, -1,
      6, 0, 7, -1,
      4, 5, 6, -1,
      4, 6, 7, -1,
      1, 4, 7, -1,
      1, 7, 0, -1,
         ]
}
```

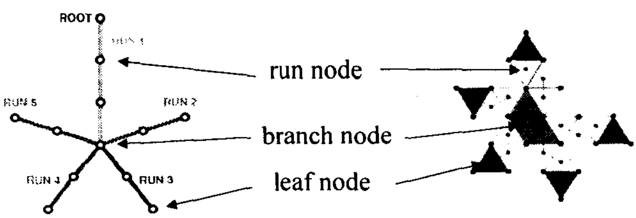
Topological Surgery



(a) 3D Model



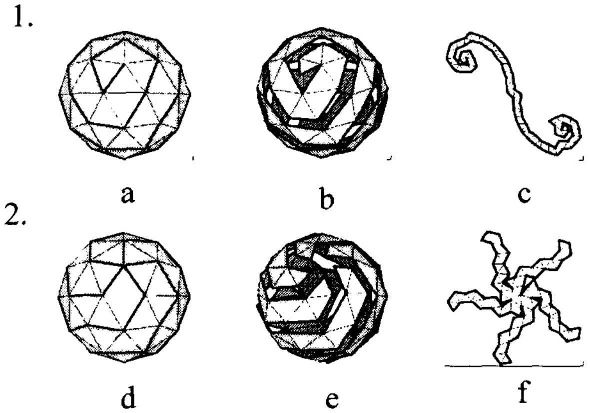
(b) Cut Model



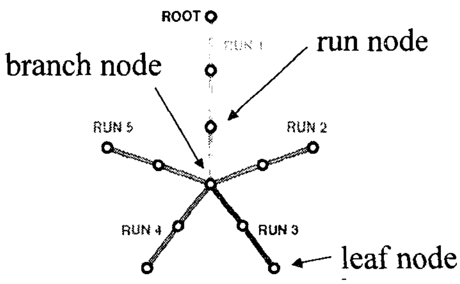
(c) Vertex Spanning Tree (d) Triangle Spanning Tree

Surface Peeling

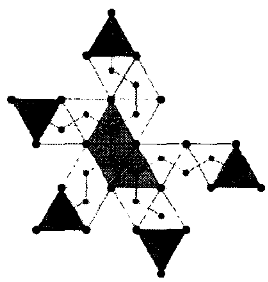
Cutting strategy should be effective to minimize the number of triangle and vertex runs



Spanning Trees

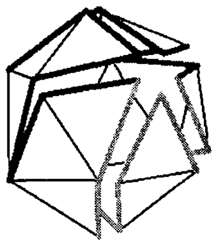


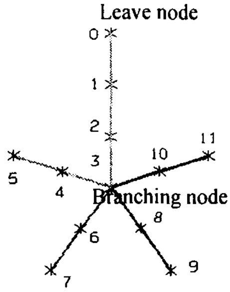
Vertex Spanning Tree



Triangle Spanning Tree

Vertex Spanning Tree





- The mesh is first cut through a subset of its edges (cut edges).
- The branching nodes and the leave nodes of the vertex spanning tree are inter-connected by vertex runs.

Encoding of Vertex Tree

vertex run (length) + 2 bits (branching bit + leaf bit)

- A leaf is chosen as the root node, and the tree is traversed in pre-order.
- Branching bit indicates if current run is the last one of the same parent node.
- Leaf bit indicates if the run ends in a leaf node.

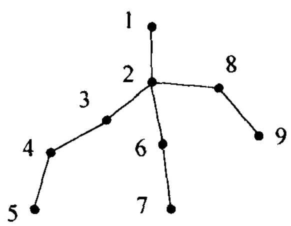
☞ Applying to the tree of previous figure, the vertex tree structure table is generated

(3, 0, 0), (2, 1, 1), (2, 1, 1), (2, 1, 1), (2, 0, 1)

Vertex Spanning Tree

- Five Binary Strings
  - Vrun, Vleaf, Vchild, Loopstart, Loopend
- Three Types of Nodes
  - Vrun : a single node
  - Vleaf : no children node
  - Vchild : two or more children nodes
- For each traversed node z,
  - If z is a run node : 1 -> Vrun
  - If z is a leaf node : 0 -> Vrun , 1 -> Vleaf
  - If z is a branching node : 0 -> Vrun , 0 -> Vleaf,
    0 of # (k - 2) + 1 -> Vchild
    where k is a number of children

Example



Vrun : 1 0 1 1 0 1 0 1 0
Vleaf : 0 1 1 1
Vchild : 01
Loopstart: 0 0
Loopend :

Adaptive Arithmetic Coding →

	# 0's	# 1's	Total
Vrun	# runs (4)	# vertex + # loop - # runs (9 + 0 – 4 = 5)	# vertex + # loop (9)
Vleaf	# runs - # leaf (4 – 3 = 1)	# leaf (3)	# runs (4)
Vchild	# runs - # leaf (4 – 3 = 1)	2*# leaf - # runs + # loop– 1 (2*3 – 4 + 0 – 1 = 1)	# leaf + # loop– 1 (3 + 0 – 1 = 2)
Loopstart	# leaf – 1 (3 – 1 = 2)	# loop (0)	# leaf + # loop – 1 (3 + 0 – 1 = 2)

Arithmetic Coding

K-JIST

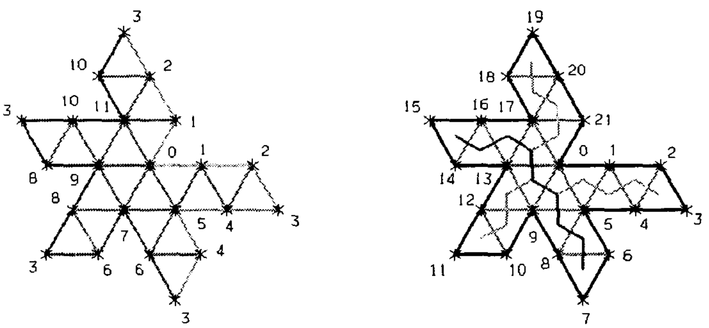
- String S: a = number of 0, b = number of 1
- For coding of the first symbol probability

$$P(0) = \frac{a}{a+b}, \quad P(1) = \frac{b}{a+b}$$

- After encoding/decoding of each symbol, the probability model is then recomputed.
- This procedure is repeated for each symbol in S.

Triangle Spanning Tree

K-JIST



- marching edges: The edges that connect triangles within a run or that bound branching triangles.

- branching triangle: Connect three triangle runs.

Coding of Triangle Tree

Root ID, triangle run (length) + leaf bit

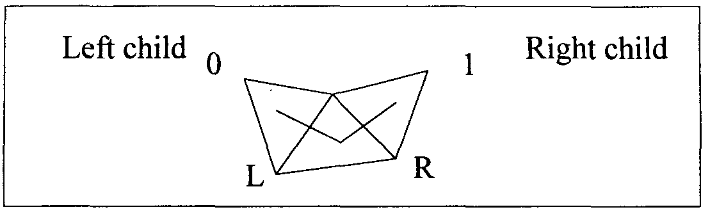
- A leaf of the triangle tree is chosen as the root triangle.
- This triangle has two edges on the bounding loop. The bounding loop index common to those two edges is the root vertex for the triangle tree, or triangle tree Root ID.
- The run length is the number of edges in the run. (number of triangles in the strip + one)
- The leaf bit represents whether the run ends at a leaf triangle or not.

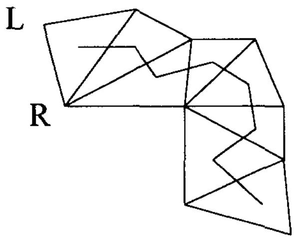
☞ The triangle tree table is:
19, (4, 0), (3, 1), (1, 0), (3, 1), (1, 0), (3, 1), (4, 1)

Triangle Spanning Tree

- Four Binary Strings
  - Trun, Tleaf, Tmarching, Polygonedge
- For each traversed node z,
  - If z is a leaf node : 0 -> Trun, 1 -> Tleaf
  - If z is branching node : 0 -> Trun, 0 -> Tleaf
  - else : 1 -> Trun
    - If z has a left child : 0 -> Tmarching
    - If z has a right child : 1 -> Tmarching

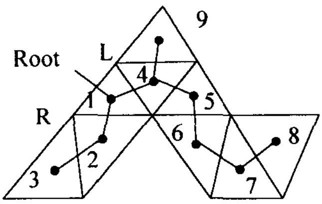
Tmarching Pattern





Marching Pattern: 11011101

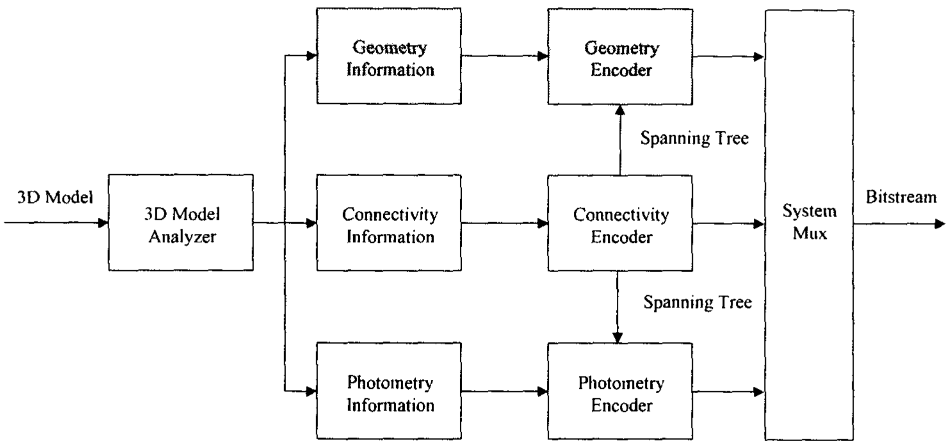
Example



Trun :0 1 0 0 1 1 1 0 0
Tleaf :0 1 0 1 1
Tmarching : 1 1 0 0
Polygonedge: 0 0 0 0 0 0 0 0 0

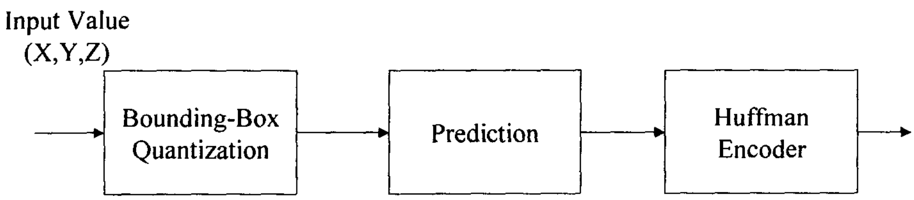
	# 0's	# 1's	Total
Trun	2*# branches+1 (2*2 + 1 = 5)	# triangle–2*#branches– 1 (9 – 2*2 - 1 = 4)	# triangle (9)
Tleaf	# branches (2)	# branches + 1 (2+ 1 = 3)	2*# branches+1 (2*2 +1 = 5)
Tmarching	# leftchild (1)	# triangle–2*#branches- #leftchild-1 (9-2*2-1-1=3)	#triangle- 2*#branches-1 (9 – 2*2-1 = 4)
Polygonedge	#triangles- #polyedge (9-0=9)	#polyedge (0)	#triangles (9)

3D Model Encoder



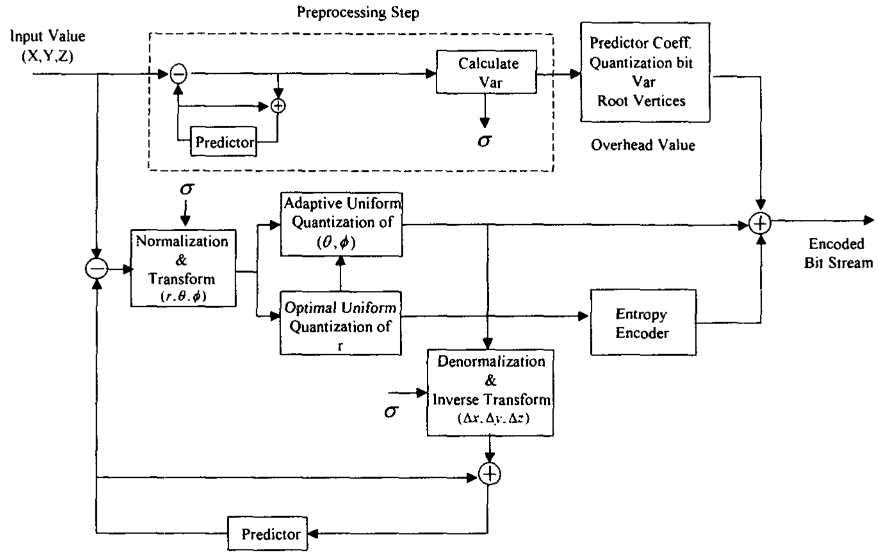
Previous Work

• Bounding Box Approach (IBM)



• Problems
- Large Quantization Errors
- Vertex Merging Problem

Encoder Block Diagram



Prediction

- Use the father-son relationship defined by the vertex spanning tree to trace all ancestors
- Find prediction coefficients by LMSE method
- Perform linear prediction

$$\hat{x}_n = \sum_{i=1}^{P} \lambda_{x_i} x_{n-i} \qquad \hat{y}_n = \sum_{i=1}^{P} \lambda_{y_i} y_{n-i} \qquad \hat{z}_n = \sum_{i=1}^{P} \lambda_{z_i} z_{n-i}$$

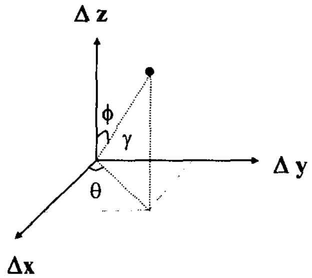
- Obtain prediction errors

$$\Delta x_n = x_n - \hat{x}_n \qquad \Delta y_n = y_n - \hat{y}_n \qquad \Delta z_n = z_n - \hat{z}_n$$

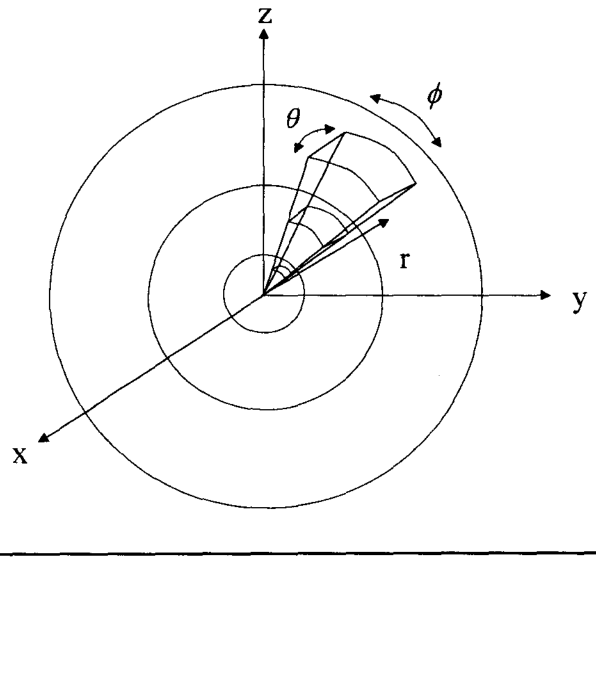
Coordinate Transform

- The residual components (Δx, Δy, Δz) are normalized by their means and variances.
- The residue vector is represented in the spherical coordinate system.

$$r = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}, \quad \theta = \tan^{-1} \Delta y / \Delta x, \quad \phi = \cos^{-1} \Delta z / r$$



Relationship of (r,θ,ϕ)



Probability Model for r

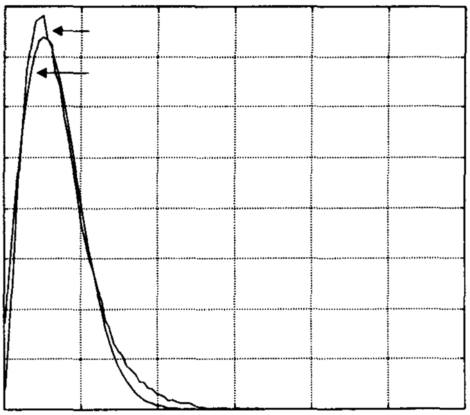
- Assume that (Δx, Δy, Δz) are of independent Laplacian distribution
- Gaussian variables → Chi-square distribition
- Difficult to analyze the distribution precisely.
- Distribution of r is fitted to curve

$$f_r(r) = re^{-\alpha r^2}, \quad r \geq 0$$

- Find α which gives the minimum fitting error

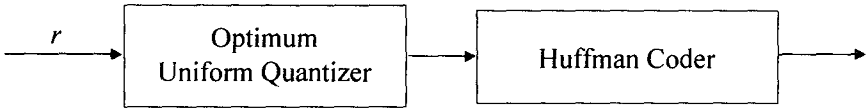
Approximation of r

$$f_r(r) = re^{-0.372r^2}, \quad r \geq 0$$



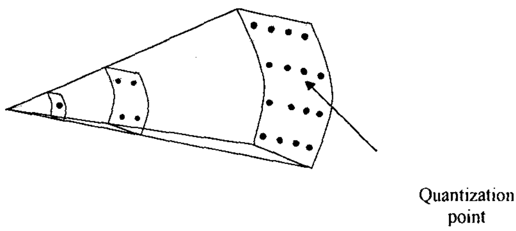
Optimal Coding of r

- The radius r is quantized using an optimum uniform quantizer for the distribution of r
- Huffman table is not transmitted
  - ☞ Distribution of r is fixed.



Surface of Sphere

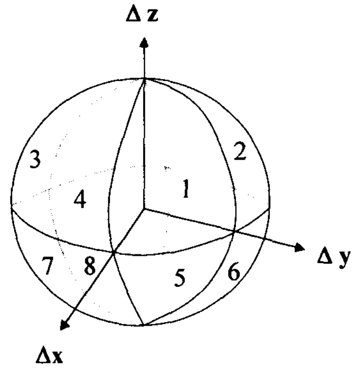
- Surface of the sphere is expanded n^2 time as r is increased n times
- Allocate n^2 quantization points over the sphere



Surface Subdivision (1)

K-JIST

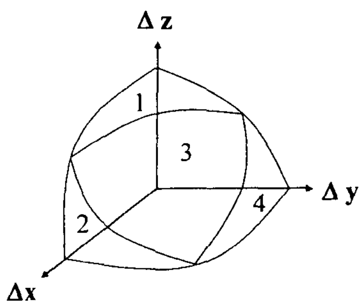
- A pair (θ, φ) is a point on the unit sphere
- Surface of the sphere is partitioned into small 8 regions.



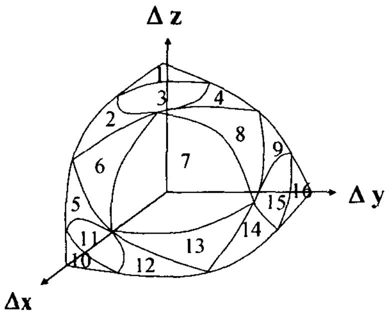
Surface Subdivision (2)

K-JIST

- All eight regions are identical
- When r is large, a region is subdivided into four small regions for each patch.



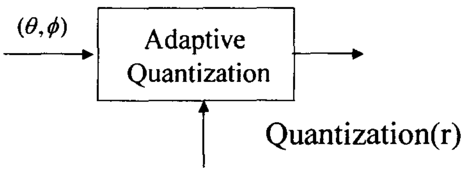
< Four Patches >



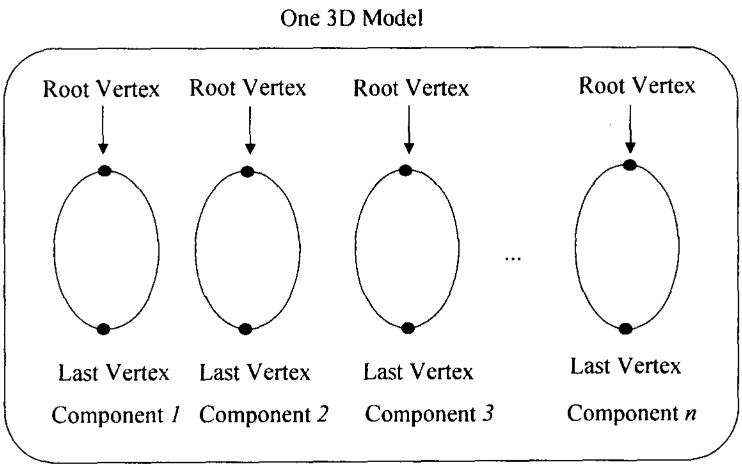
< Sixteen Patches >

Adaptive Quantization of (θ,ϕ)

- Encode the corresponding pair of (θ,ϕ) adaptively according to the magnitude of the quantized value of r
- Distribution of the radius r is highly skewed toward zero
  - ☞ Adaptive bit allocation for (θ,ϕ) reduces the total number of bits for residue vectors.



Multiple Component Coding



Root Vertex: 16 bit coding

Error Metric

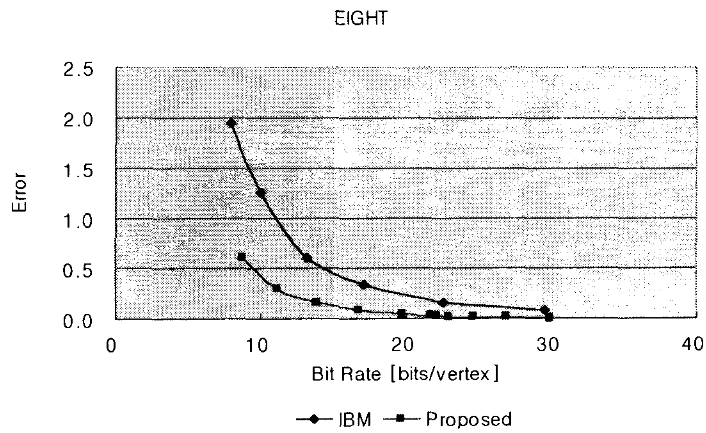
$$mean\_error = \frac{1}{2}\sum_{i=1}^{n}(dist_1(i) + dist_2(i))$$

Where,

A: Original Model
B: Reconstructed Model
n: Number of Vertex
$dist_1(i)$: Distance between vertex *i* of A and closest vertex of B
$dist_2(i)$: Distance between vertex *i* of B and closest vertex of A

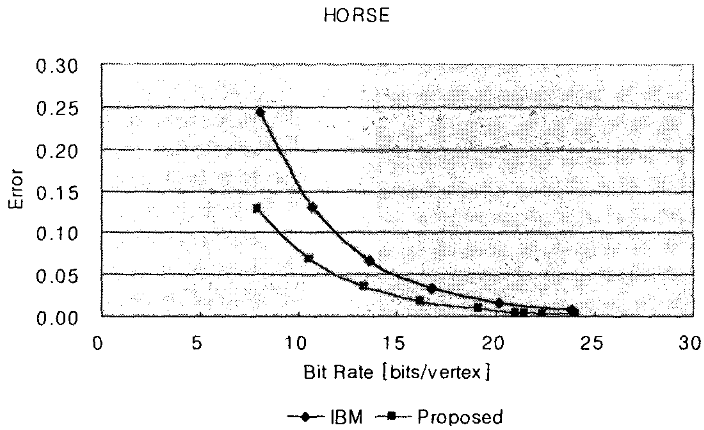
Result: Model EIGHT

Number of Vertices: 766

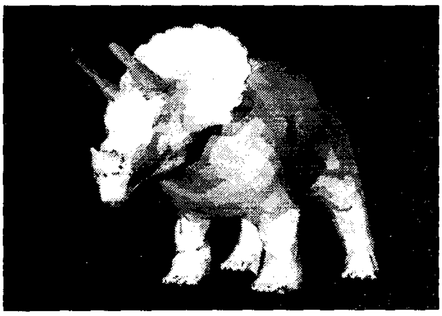


Result: Model HORSE

Number of Vertices: 11135



Triceratops : VBR = 8

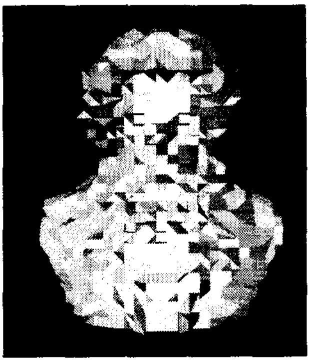


(a) Proposed Scheme
Error = 0.1922

(b) IBM Scheme
Error = 0.5339

Beethoven: VBR=8

(a) Proposed Scheme
Error = 0.5180



(b) IBM Scheme
Error = 1.3074

Conclusions

- 3D Model Coding
  - Mesh Topology/Connectivity Coding (M1)
  - Mesh Geometry/Vertex Coding (M2)
  - Progressive/Scalable 3D Mesh Coding (M3)
  - Attribute Coding and Tool Integration (M4)
  - Incremental Rendering and Error Resilience (M5)
- Geometry Coding
  - Prediction and Quantization
  - Huffman Coding vs. Arithmetic Coding
  - Coding of Root Vertices