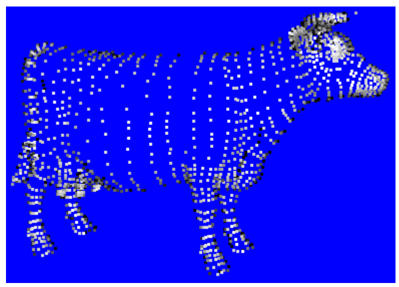
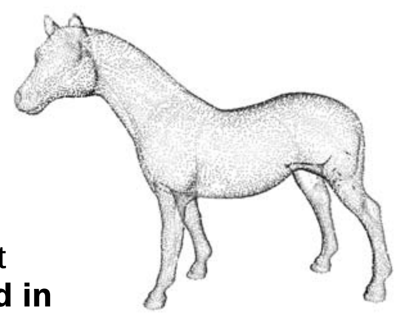
Mesh Geometry Coding



University of
British Columbia

The Geometry

- Vertex coordinates (x, y, z) are
  - Floating point values
  - Almost unrestricted in:
    - range
    - precision
  - **Uniformly** spread in 3D



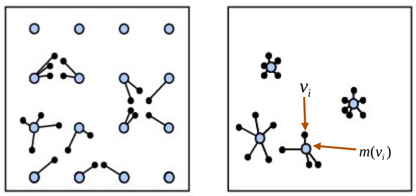
- Compression exploits input redundancy **- hard to find in raw geometry data**

- ***Lossy* compression is OK !!**

University of
British Columbia

Quantization

- Map n values v_i to k<<n values $m(v_i)$, without losing **too much** information



Uniform Non-uniform

University of British Columbia

Quantization

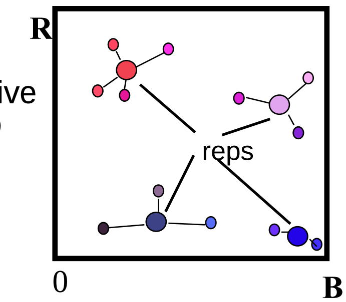
- Applications:
  - Image and voice compression
  - Voice recognition
  - Color display
  - Geometric compression

University of British Columbia

Example: color quantization

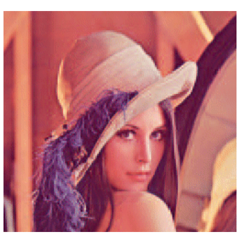
- Used for limited dynamic-range displays (e.g. an 8 bit display can display only 256 different colors)
- Reducing number of colors
  - Choosing set of representative colors (*colormap* or *palette)*
  - Map rest of colors to them
- Usually uses 256 colors

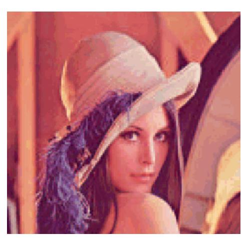
quantization to 4 colors



University of British Columbia

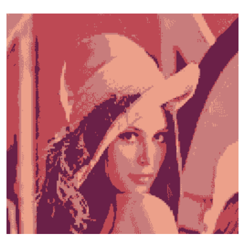
Color quantization examples


256 colors


64 colors


16 colors


8 colors

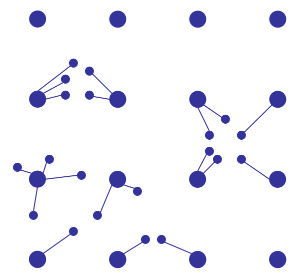

4 colors


2 colors

University of British Columbia

Uniform quantization

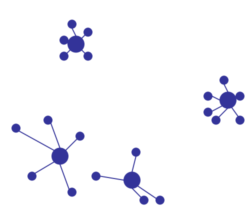
- Quantization space partitioned into equal sized regions (e.g. grid) – colors in each region mapped to its center
- Input independent
- Some representatives may be *wasted*
- Common way for 24->8 bit color quantization: retain 3+3+2 most significant bits of R, G & B components



University of British Columbia

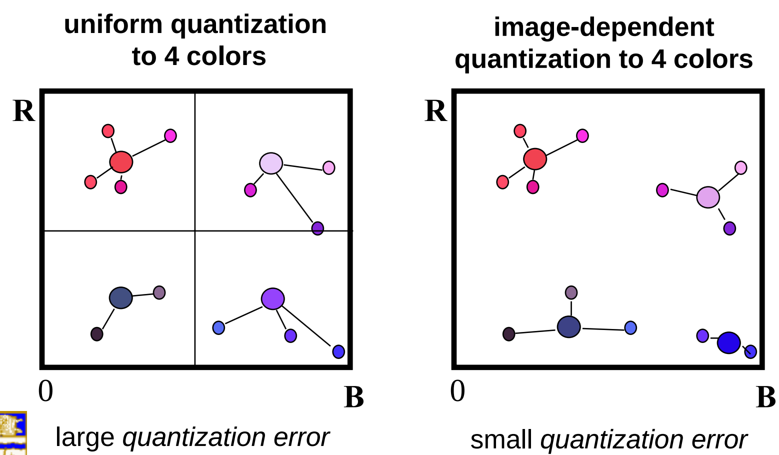
Non-uniform quantization

- Quantization space partitioned according to input data
- Goal: choosing "best" representatives
  - Minimal distance error (if "distance" is defined)



University of British Columbia

Examples



University of
British Columbia

Quantization & *Lossy* Coding

- Quantization used as lossy coding method when there is notion of distance between symbols to be coded
  - Coordinates
  - Colors
  - Not good for characters

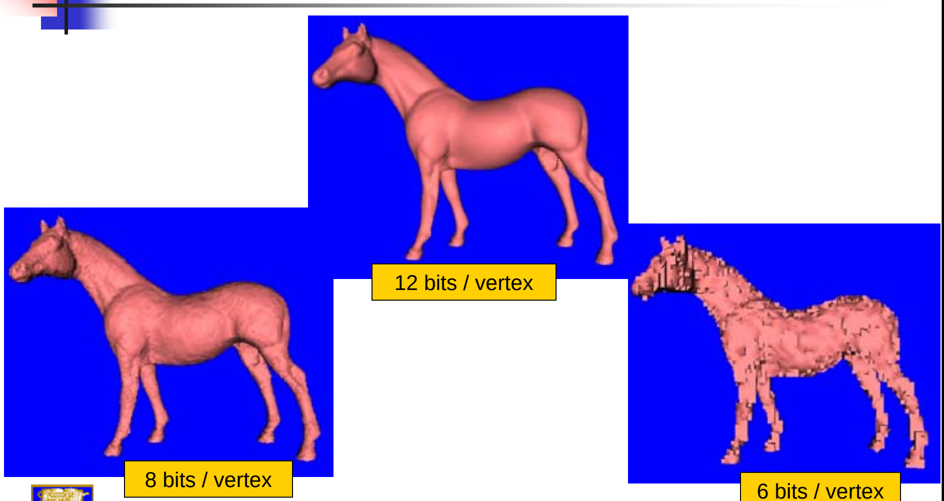
University of
British Columbia

Uniform geometry quantization

- Coordinates can be considered integers in a finite range after quantization
- Quantization is done on the data bounding box/cube
- Geometry quantization to n bits:
  - All integer values in $[0, 2^n\text{-}1]$ can be used
  - Scale/transform coordinates to be maximal over given range
  - Quantize each coordinate (rounding to nearest integer)

University of British Columbia

Uniform geometry quantization - results



University of British Columbia

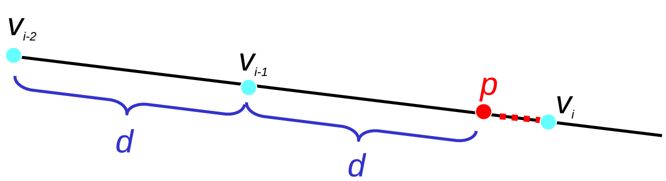
Prediction

- Quantization alone gives poor compression – vetex positions evenly (more or less) distributed in range
- To improve use *prediction*
  - Assume vertex positions can be estimated based on neighbours
  - Store & encode (quantize) prediction error
- Good prediction
  - small error range
  - uneven error distribution

University of British Columbia

Prediction – History Repeats Itself

- Linear 2D predictor:



- Prediction rule: $v_i-1 - v_i-2 = p - v_i-1$
  or: $p = 2 v_i-1 - v_i-2$
- Prediction error: $e_i = v_i - p$

University of British Columbia

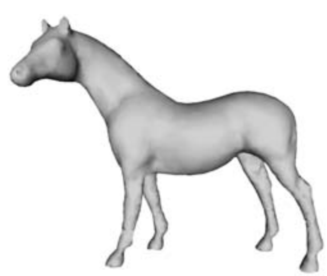
Using Predicted Geometry

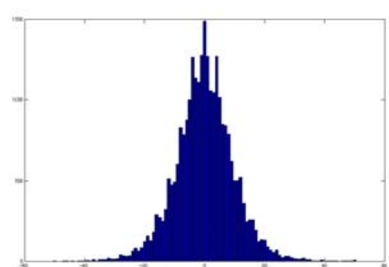
- $(v_1\ v_2\ v_3\ \ldots)$ - vertex coordinates
  $(e_3\ e_4\ e_5 \ldots)$ - prediction errors
- Naive geometry coding: $v_1\ v_2\ v_3\ \ldots$
- Coding using prediction: $v_1\ v_2\ e_3\ e_4\ e_5\ \ldots$
- Decoding: $v_1\ v_2$

$$v_i = 2\ v_i\text{-}1 - v_i\text{-}2 + e_i \qquad i>2$$

University of
British Columbia

Good Prediction Reduces Entropy

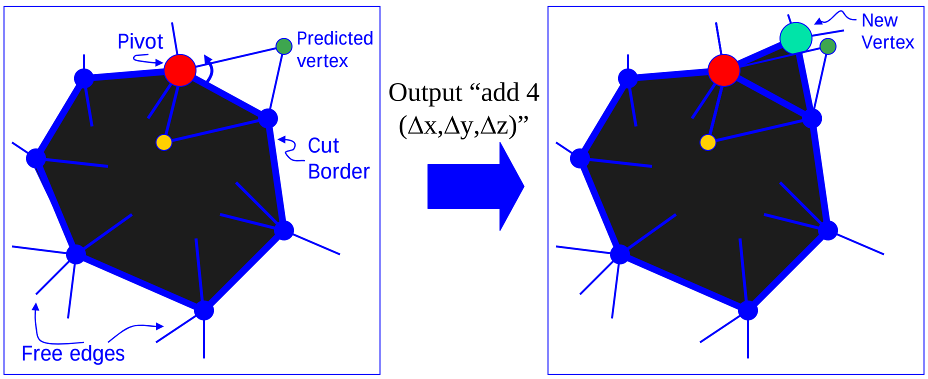




0
Distribution of prediction errors

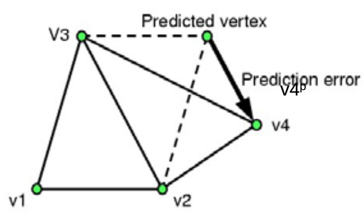
University of
British Columbia

Surface-Based Prediction - TG

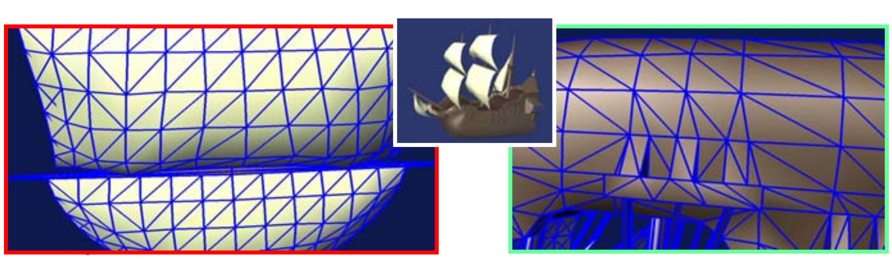


University of British Columbia

Parallelogram Prediction



- Use connectivity to predict geometry: $V_{4p} = v_2 + v_3 - v_1$
- (-1, 1, 1) in *barycentric coords*
- **Can** be applied to integers



British Columbia

Some Results

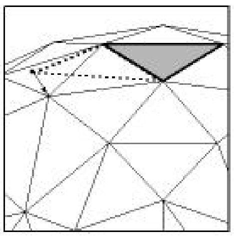
Raw quantized data = 10 bits/coord = 30 bits/vertex

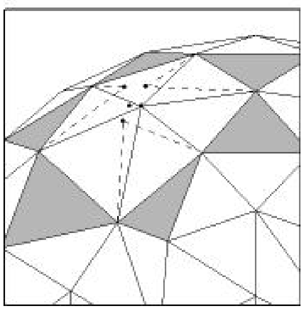
Model	vertices	line predictor	parallelogram	ratio
Eight	766	18.8	14.0	1.3
Triceratops	3100	18.4	14.1	1.3
Cow	3078	18.9	14.6	1.3
Beethoven	2847	22.7	17.3	1.3
Dodge	10466	19.8	12.4	1.6
Starship	4468	19.2	13.2	1.5
Average		19.6	14.3	1.4

University of British Columbia

K-way prediction

predict that vertex is average of its neighbors

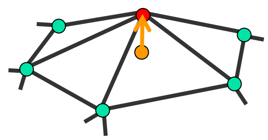




University of British Columbia

Correction

- Detail = surface – *smooth*(surface) = surface – predicted(surface)



$$\boldsymbol{\delta}_i = \mathbf{v}_i - \frac{1}{d_i}\sum_{j\in N(i)} \mathbf{v}_j$$

$$\boldsymbol{\delta}_i = \sum_{j\in N(i)} \frac{1}{d_i}\left(\mathbf{v}_i - \mathbf{v}_j\right)$$

University of British Columbia

Laplacian matrix

- Transition between δ & xyz is linear:

$$\begin{pmatrix} \mathrm{L} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \delta_1^{(x)} \\ \delta_2^{(x)} \\ \vdots \\ \vdots \\ \delta_n^{(x)} \end{pmatrix}$$

$$A_{ij} = \begin{cases} 1 & i \in N(j) \\ 0 & otherwise \end{cases}$$

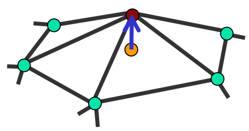
$$D_{ij} = \begin{cases} d_i & i = j \\ 0 & otherwise \end{cases}$$

$$\mathrm{L} = \mathrm{I} - \mathrm{D}^{-1}\mathrm{A}$$

University of British Columbia

Laplacian matrix

- Transition between δ & xyz is linear:



$$\boldsymbol{\delta}_i = \sum_{j \in N(i)} w_{ij} \left(\mathbf{v}_i - \mathbf{v}_j \right)$$

$$L \, \mathbf{v}_x = \delta_x$$

$$L \, \mathbf{v}_y = \delta_y$$

$$L \, \mathbf{v}_z = \delta_z$$

University of British Columbia

Basic properties

- Rank(L) = n-c (n-1 for connected meshes)
- Can reconstruct *xyz* geometry from delta up to translation

$$L\mathbf{x} = \boldsymbol{\delta}$$

$$\mathbf{x} = L^{-1}\boldsymbol{\delta}$$ (almost....)

University of British Columbia

Quantizing differential coordinates

- Note: even if xyz are integer δ won't be
- Quantize δ-coordinates?
  - Can we still go back to xyz ?
  - How does reconstruction error behave?

$$L\mathbf{x} = \boldsymbol{\delta}$$

$$\boldsymbol{\delta} \rightarrow \boldsymbol{\delta}' = \boldsymbol{\delta} + \varepsilon$$

University of British Columbia

Quantizing differential coordinates

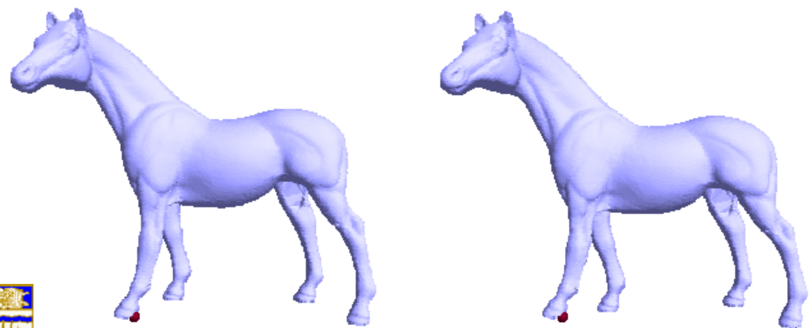
How does reconstruction error behave?

$$\mathbf{x}' = L^{-1}\boldsymbol{\delta}' = L^{-1}(\boldsymbol{\delta} + \varepsilon)$$

University of British Columbia

Quantizing differential coordinates

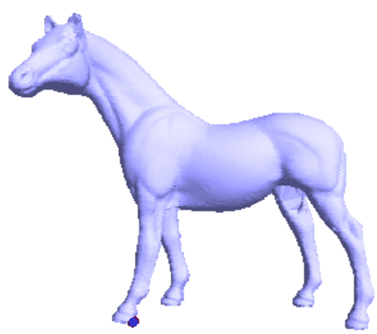
- Find the differences between the horses…



University of British Columbia

Quantizing differential coordinates

- Original horse model



University of British Columbia

Quantizing differential coordinates

- Quantizing δ to 8 bits/coordinate
- One anchor point (front left leg)

University of British Columbia

Quantizing differential coordinates

- Original model

University of British Columbia

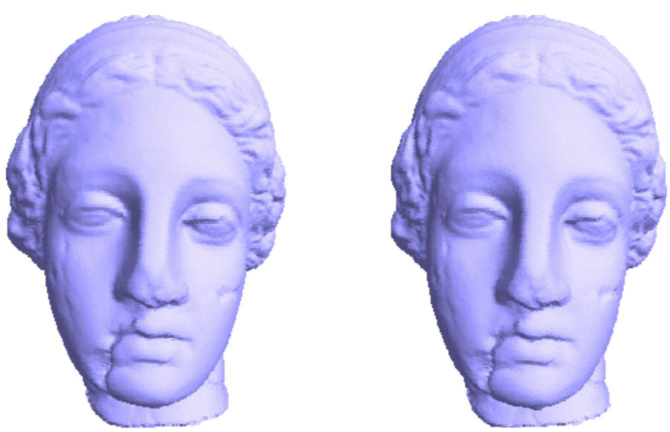
Quantizing differential coordinates

- Quantizing δ to 7 bits/coordinate, one anchor

University of British Columbia

Quantizing differential coordinates

- Can reduce error by adding more anchors



University of British Columbia