Compression, Processing, Indexing and Retrieval of 3D Objects and Data:

How to extend image/video processing to graphics?

Tsuhan Chen
Carnegie Mellon University
tsuhan@cmu.edu

Joint work with Howard Leung, Masa Okuda, and Cha Zhang

(Mis)Understanding

- To graphics and vision communities
  - Video is just low-level processing
- To the video community
  - Graphics is just some fancy tools
  - Vision is things that don t work in practice

First Attempt...

- MPEG-4
  - Started out as model-based coding
  - Analysis and synthesis
  - Using vision/graphics for video coding
- That didn t happen (not completely)
  - Settled with 2D shape-based coding
  - Model-based coding for limited content, e.g., faces

Modeling and Coding

MODELS	CODED INFORMATION	EXAMPLES
Pixels	Color of pixels	PCM
Statistically dependent pixels	Prediction error or transform coeffs	Predictive Coding Transform Coding
Moving blocks	Motion vectors and prediction error	Block-based coding H.261/263, MPEG-1/2
Moving regions	Shapes, motion, and colors of regions	Region-based coding H.263+, MPEG-4
Moving objects	Shapes, motion, and colors of objects	Model-based coding
Facial models	Action units	MPEG-4
A/V objects	Description	MPEG-7

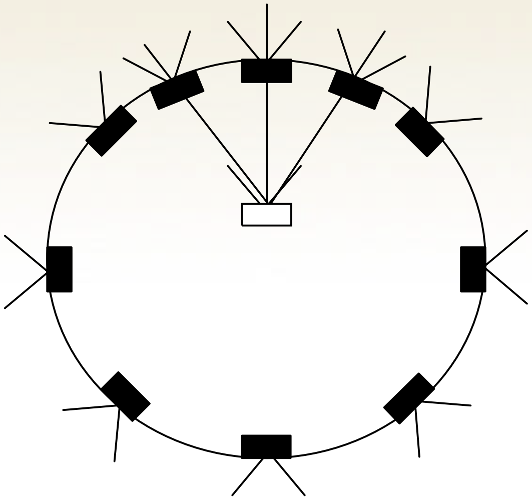
Modeling and Coding (cont.)

- Better modeling implies
  - Higher compression
  - More content accessibility
  - More complexity
  - Less error resilience
- Video and vision/graphics do go hand-in-hand all along
- Video research is evolution of vision and graphics techniques

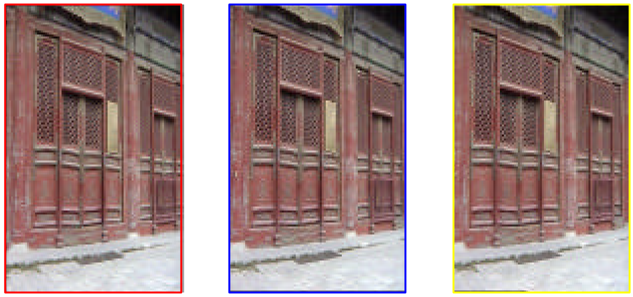
Topics

- Compression for image based rendering
- Compression for 3D meshes
  - Streaming in texture and geometry jointly
- Indexing and retrieval of 3D objects
- Building immersive environments

Image-Based Rendering



[Shum et. al]

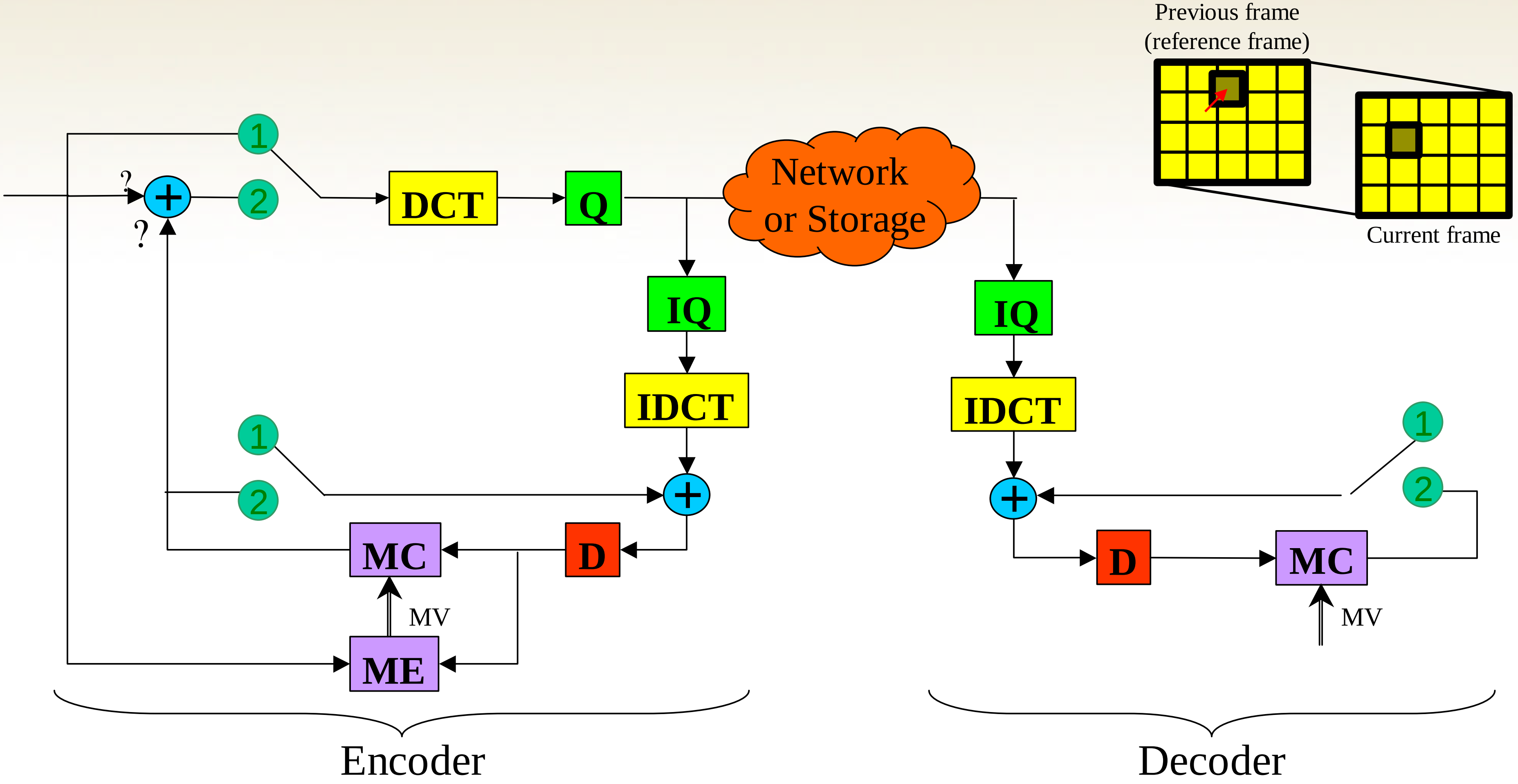


......

Compression

- The number of images is large, so we need compression
- Good to have fewer samples
  - Does not guarantee fewer bits
- Consider these as a video sequence
  - General video coding applies

Video Codec

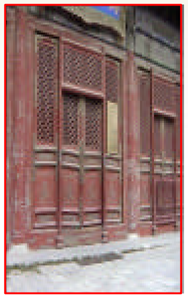


1 Intra coding

2 Inter coding

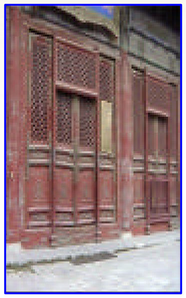
Tsuhan Chen

Intra Coding



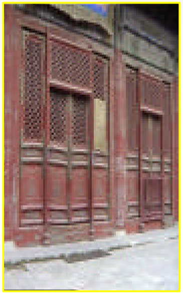
1

I-frame



2

I-frame



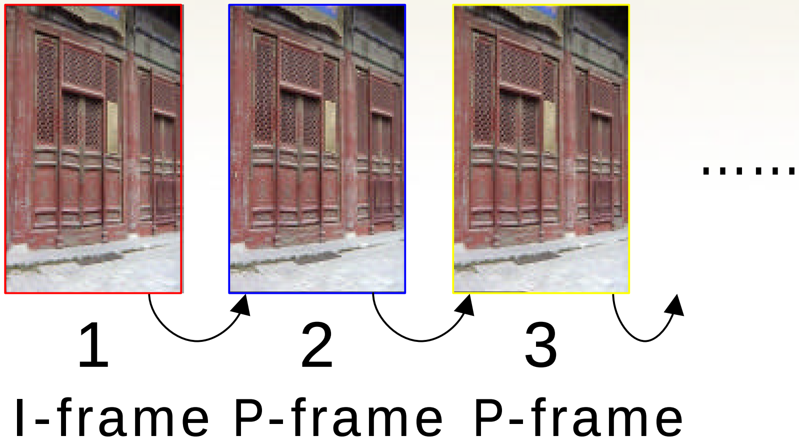
3

I-frame

......

Disadvantage: Does not exploit the correlation between images

Inter Coding



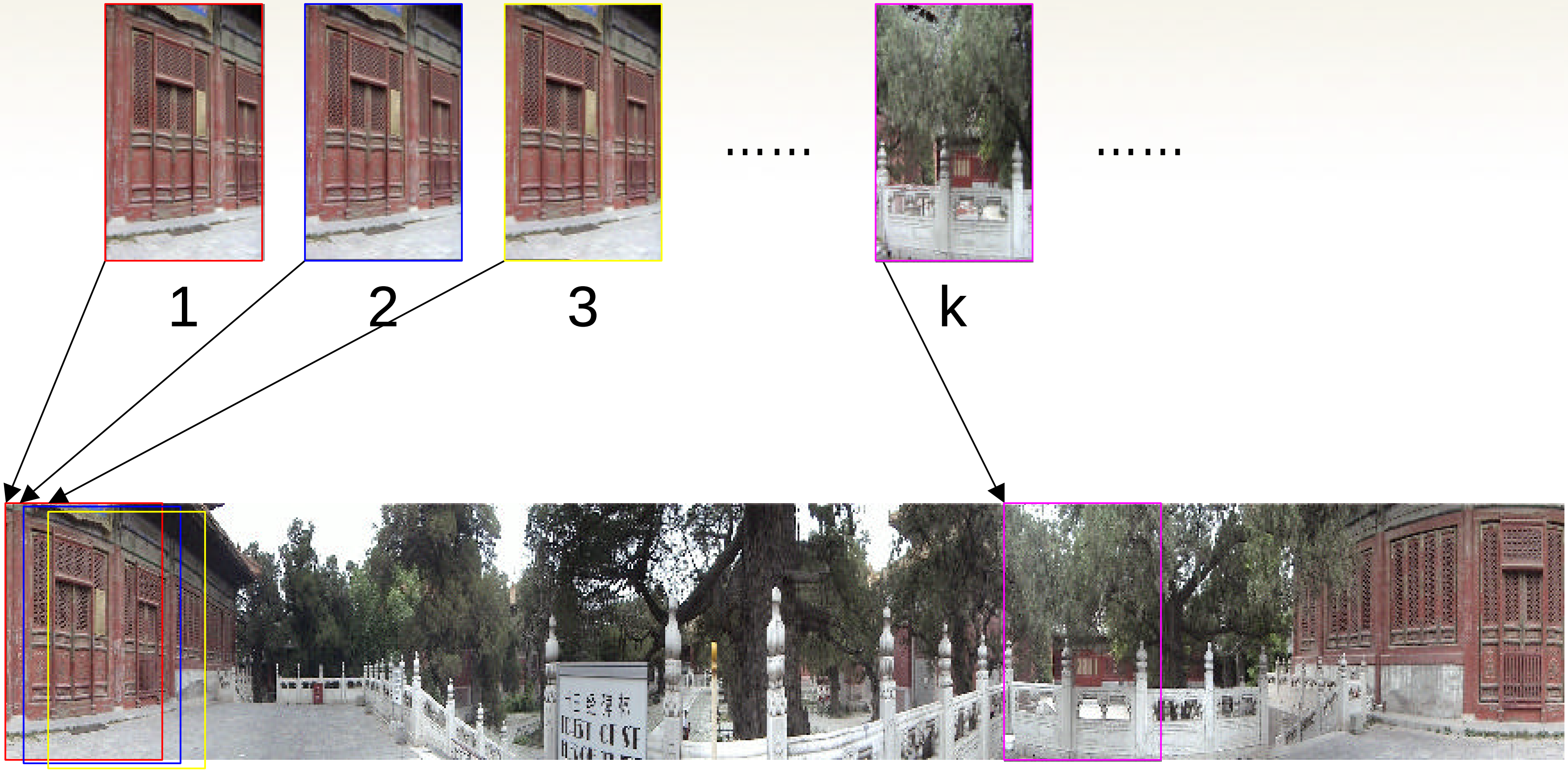
1 2 3 ......

I-frame P-frame P-frame

Disadvantage: Does not provide random access

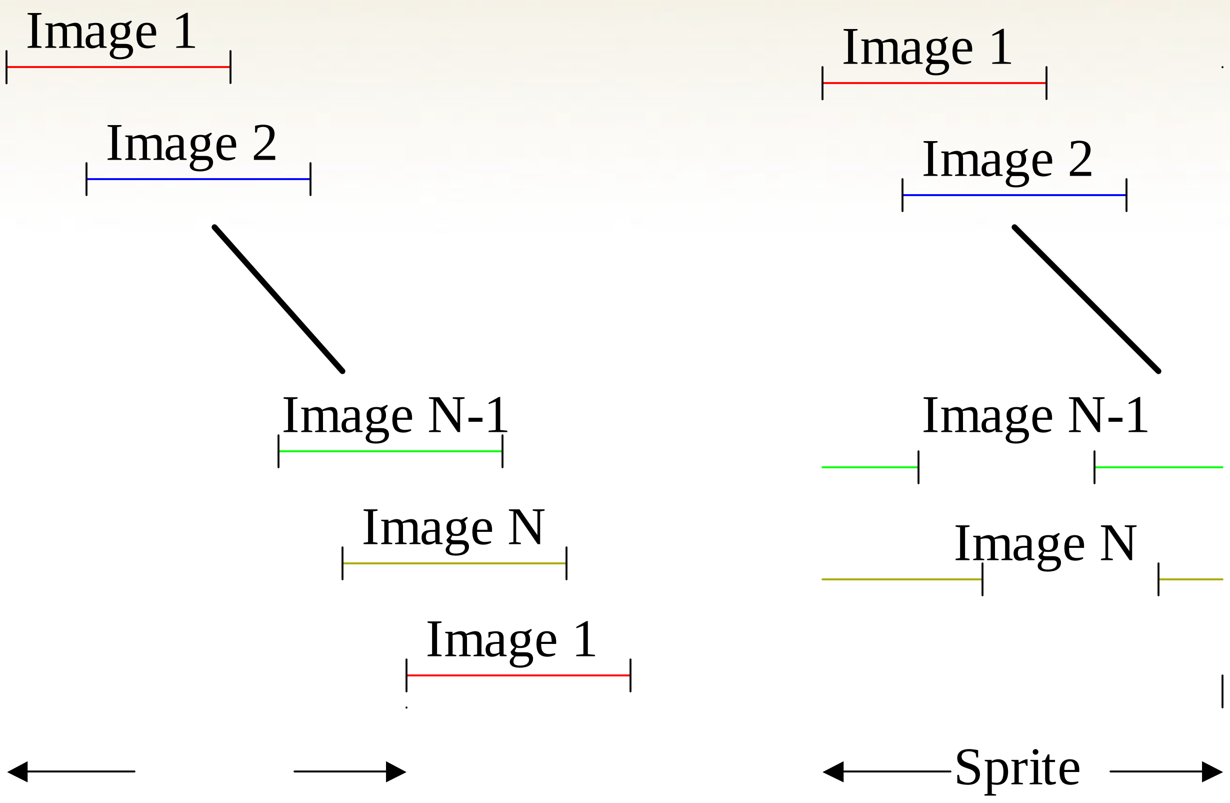
i.e., frame N depends on frame N-1

Prediction from Sprite



[cf. Anandan et. al]

Generation of Sprite

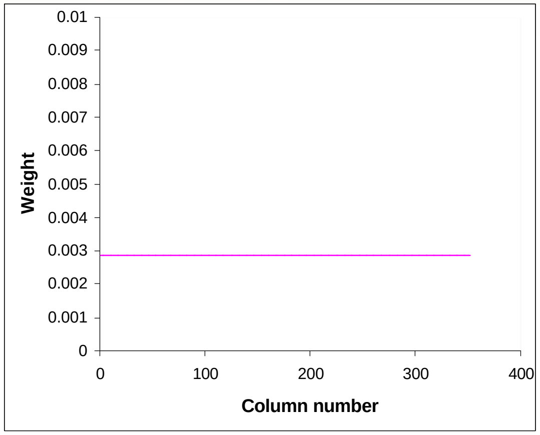


Step 1: Finding the offset

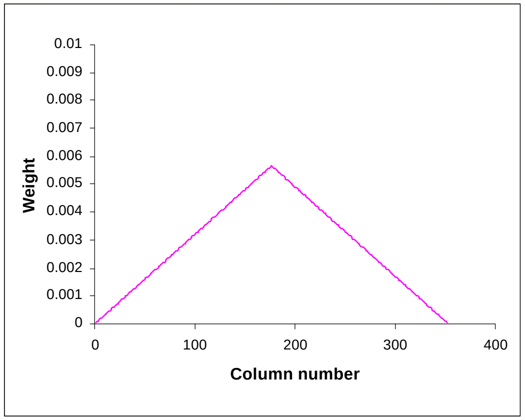
Step 2: Generating the sprite

Weighting

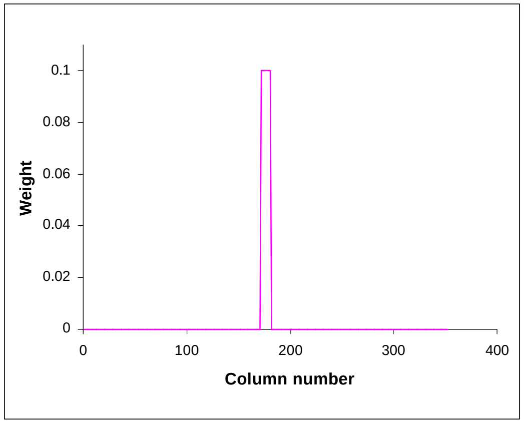
- need to find a weighting function to blend the images to form the sprite



Constant weighting

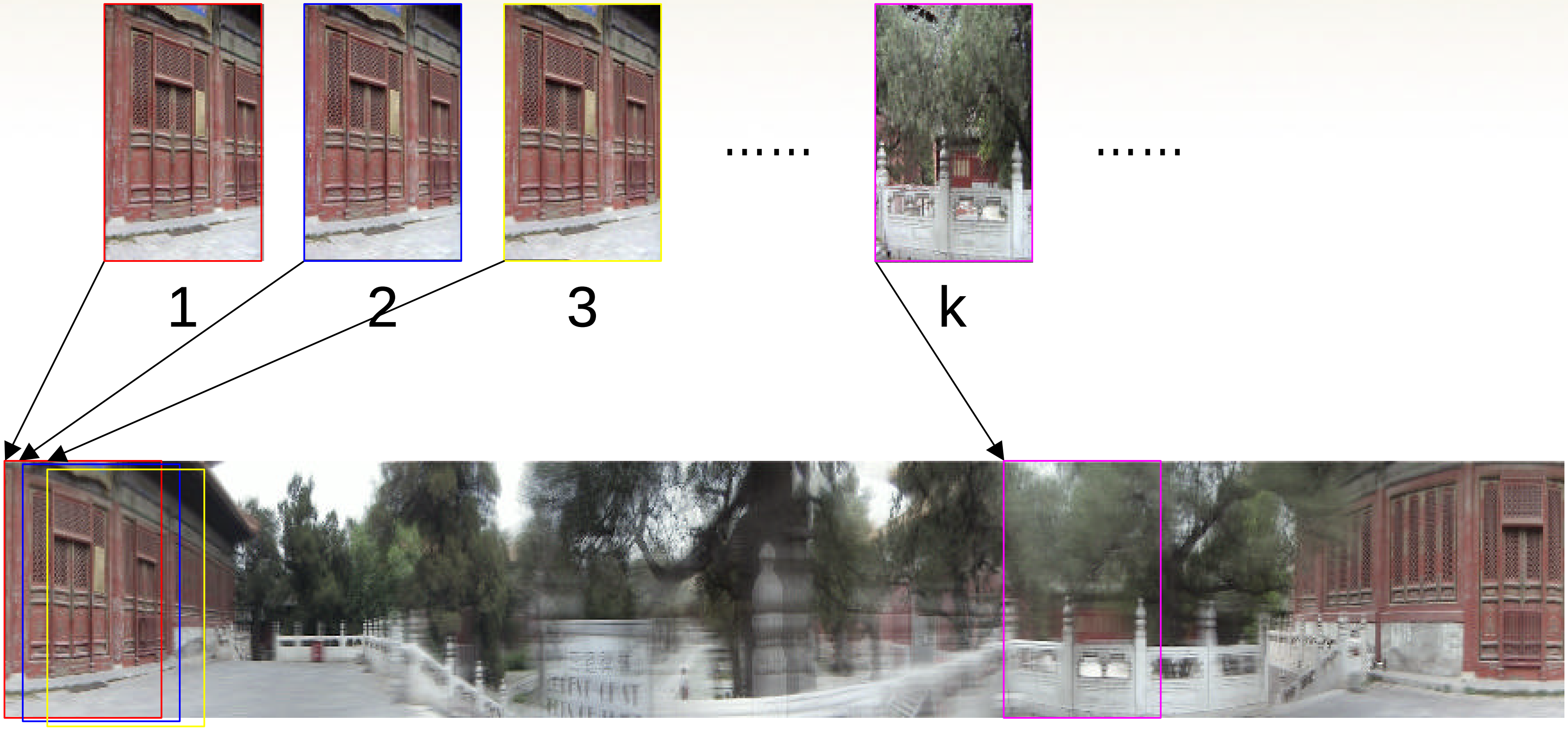


Triangular weighting

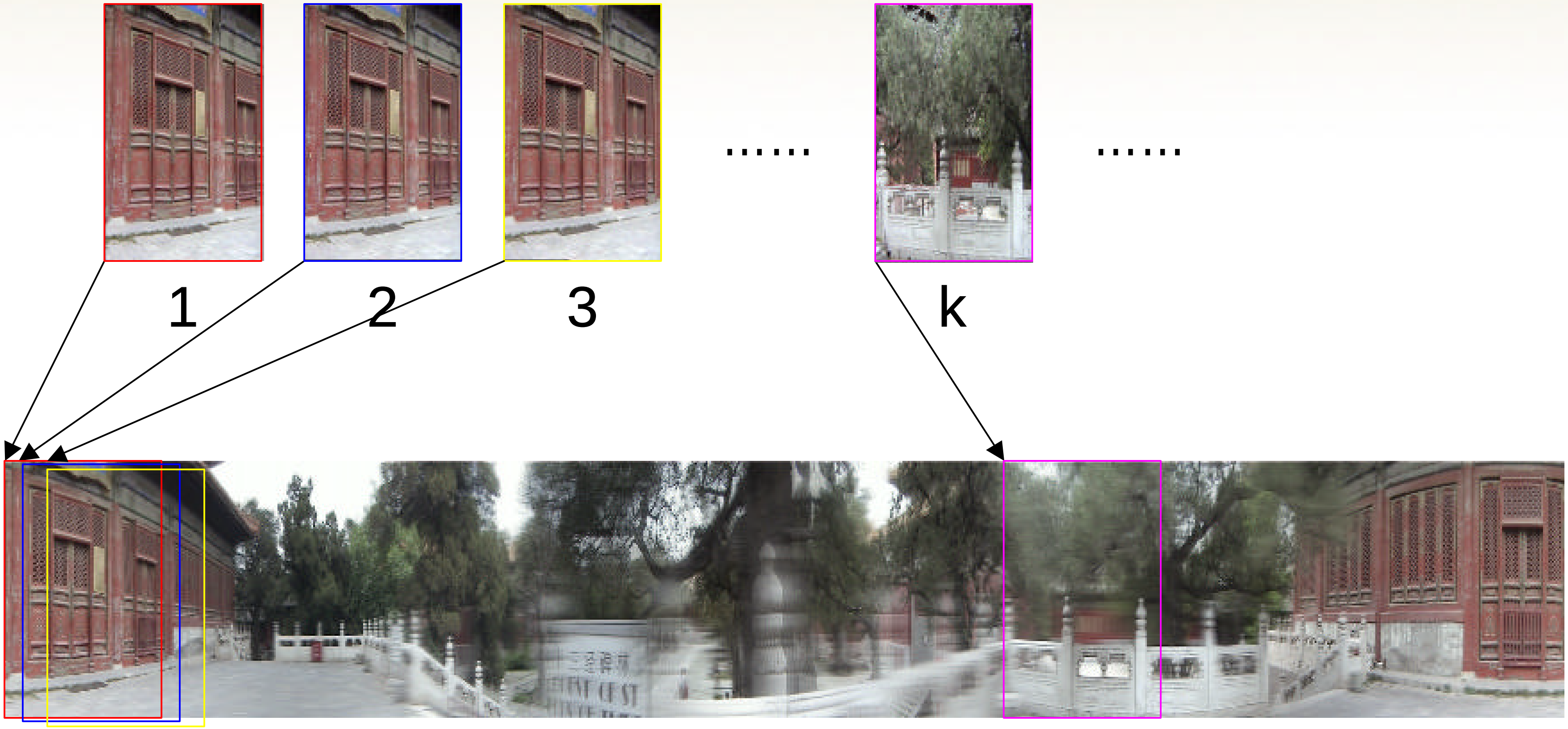


Delta weighting

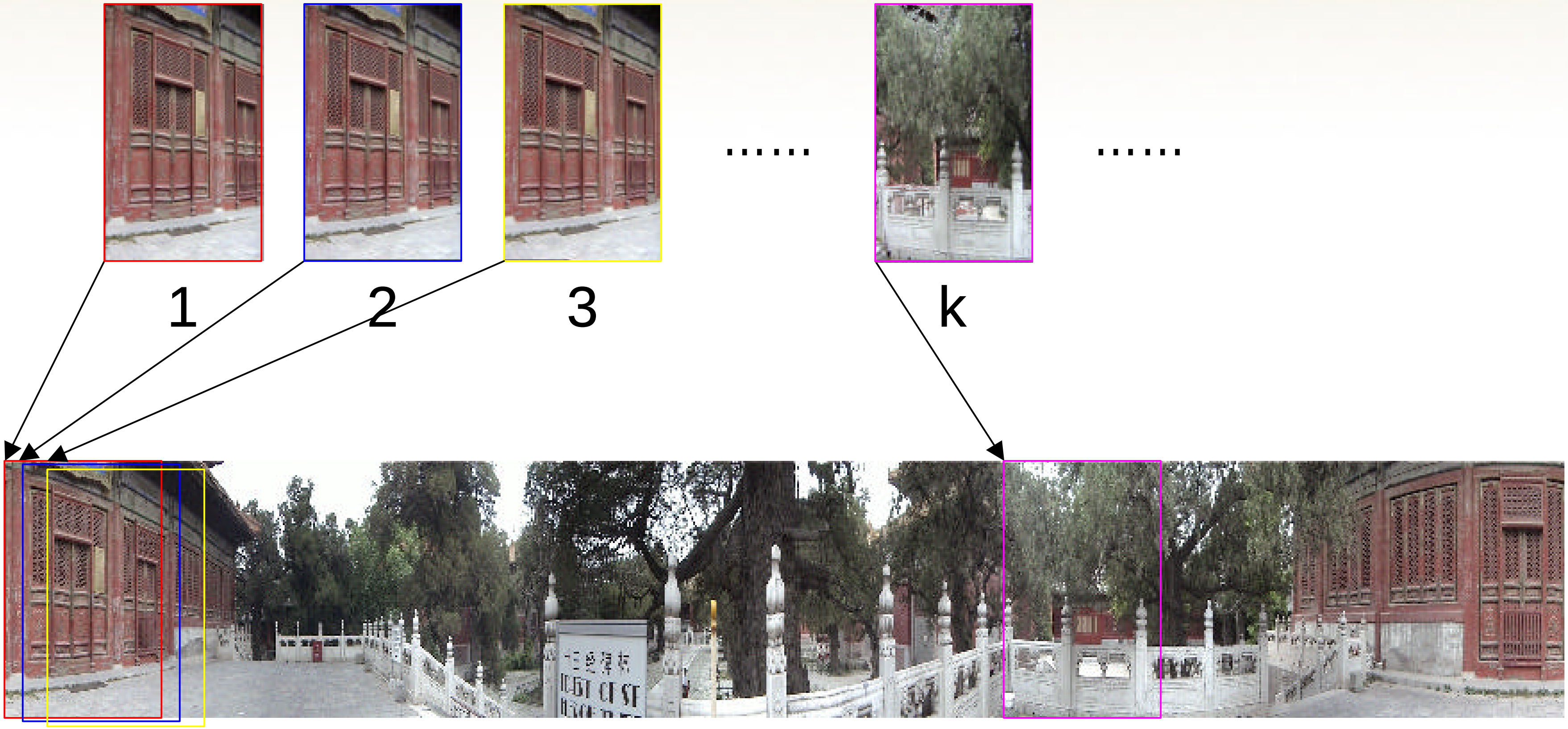
Constant Weighting



Triangular Weighting

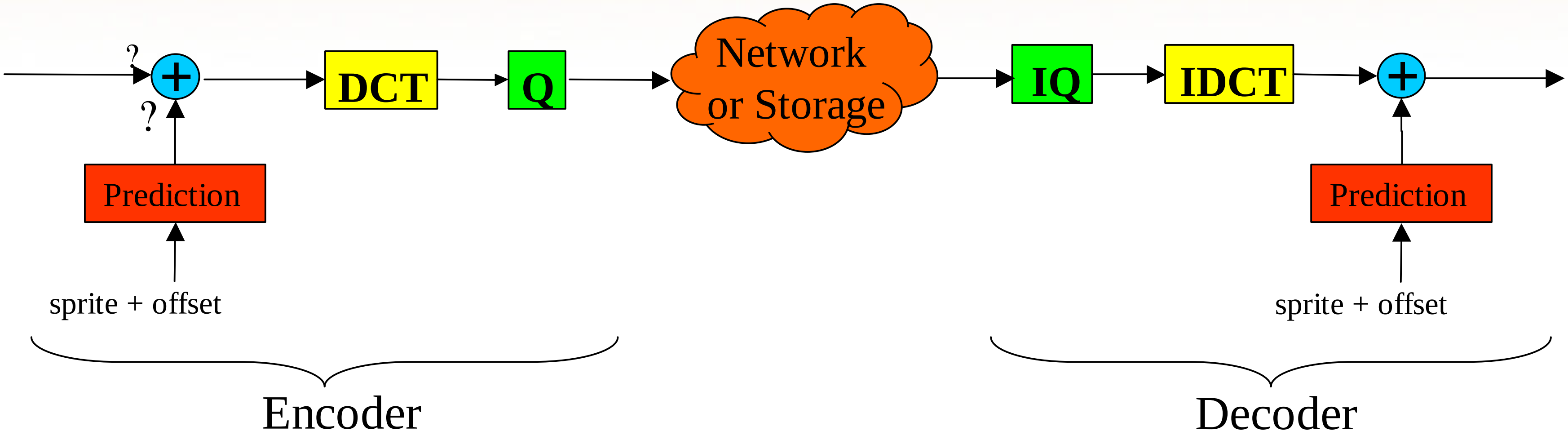


Delta Weighting



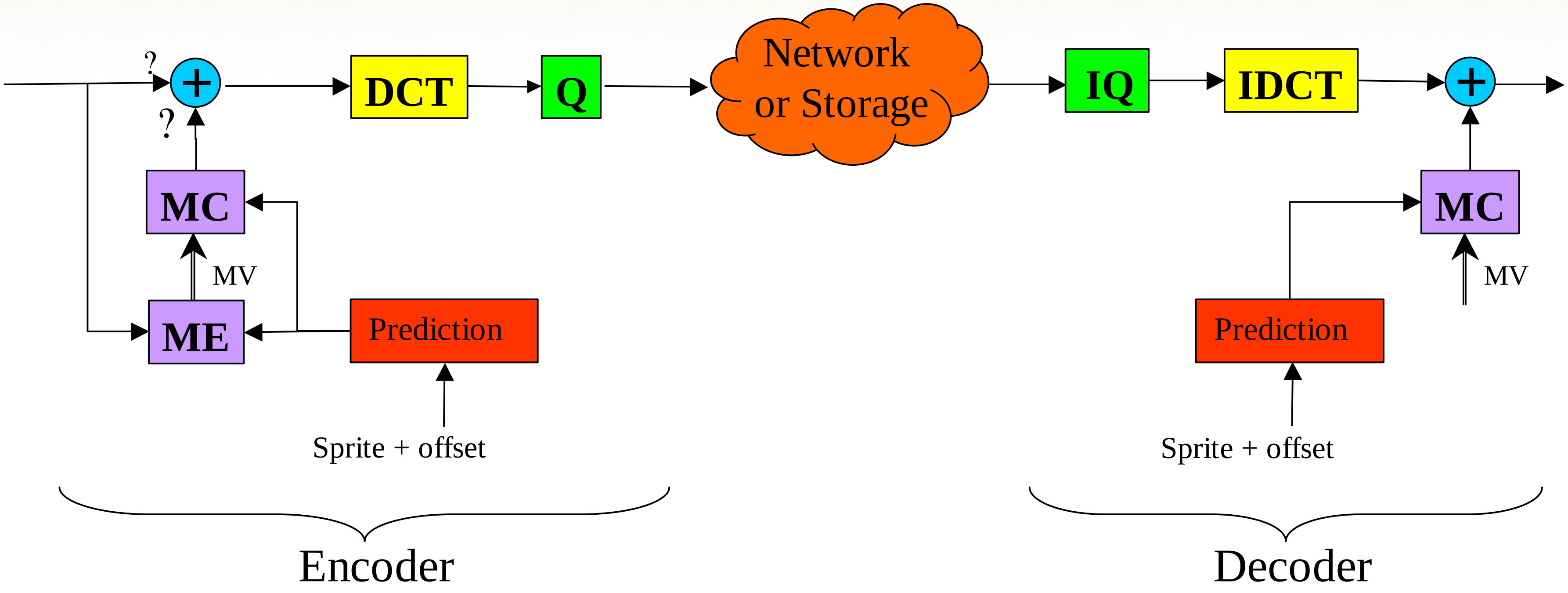
Modified Codec

3 Prediction from sprite image without MC

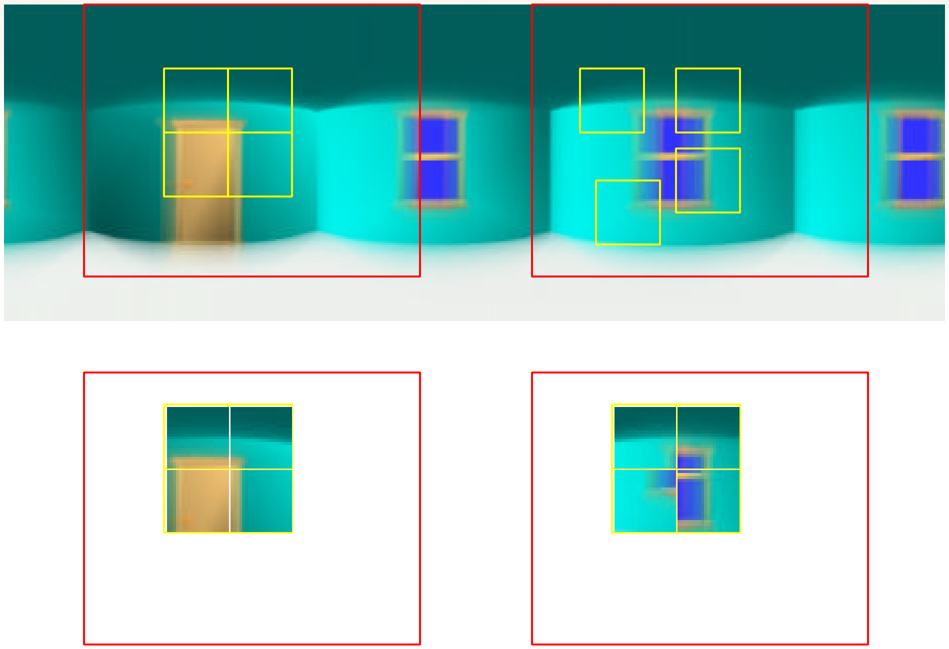


With Motion Compensation

4 Prediction from sprite image with MC



With vs. Without MC

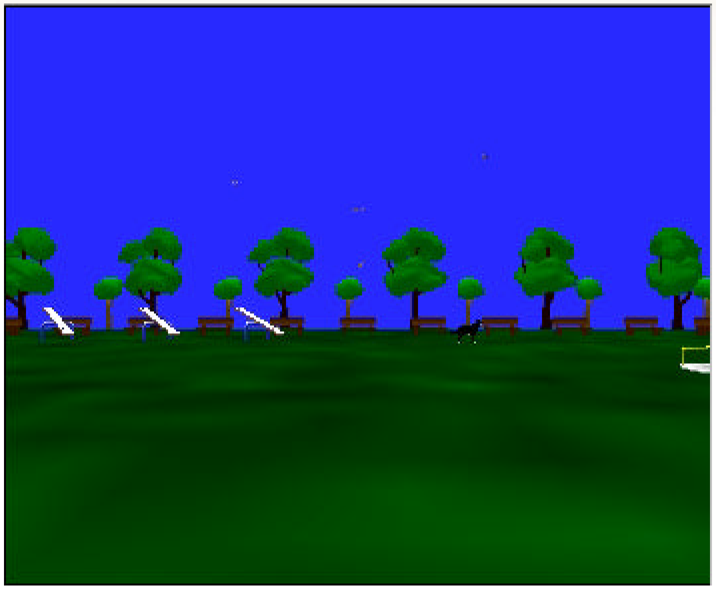


3 without MC

4 with MC

Test Sequences (1)

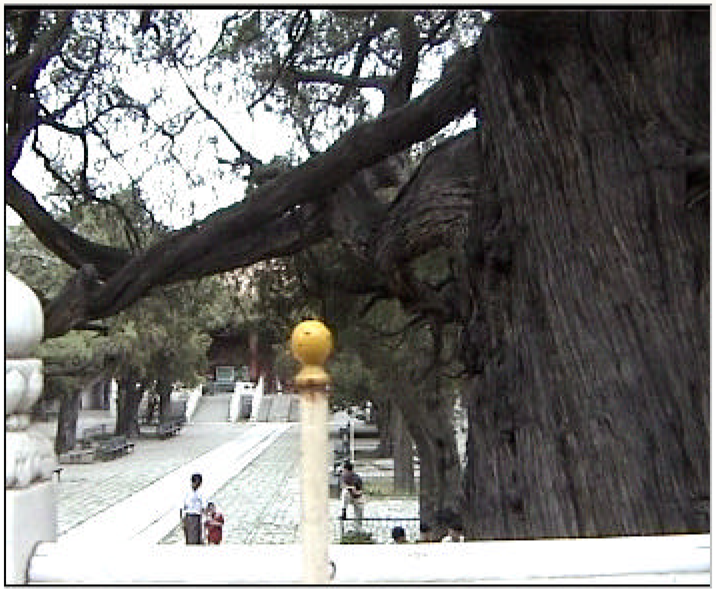
Synthetic sequence 1: NetICE room



Synthetic sequence 2: Park

Test Sequences (2)

Real sequence 1: Kids

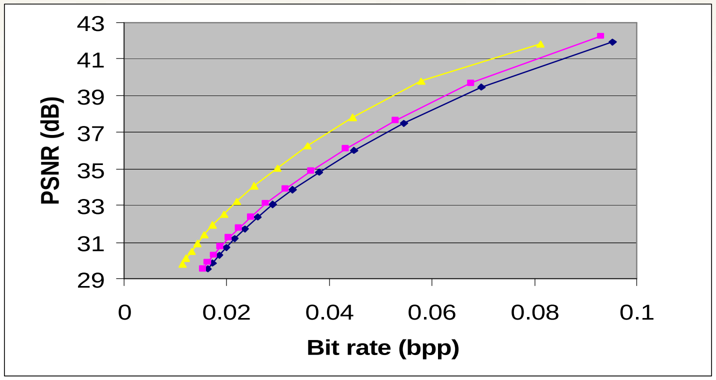


Real sequence 2: Kongmiao

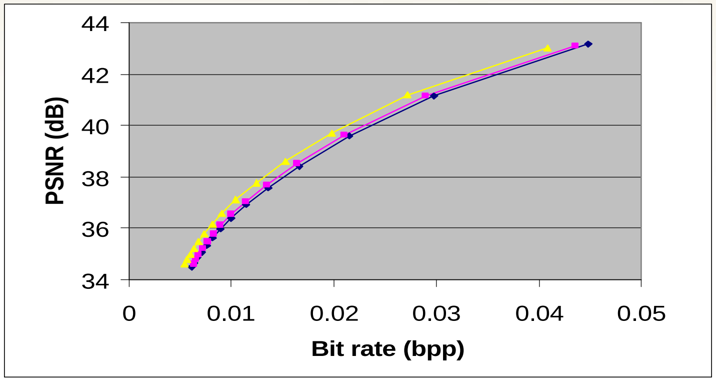
[Shum, et al]

Weighting function Results

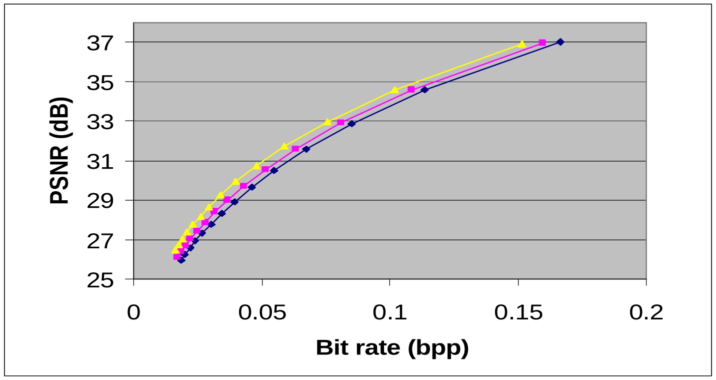
◆ Constant weighting ■ Triangular weighting ▲ Delta weighting



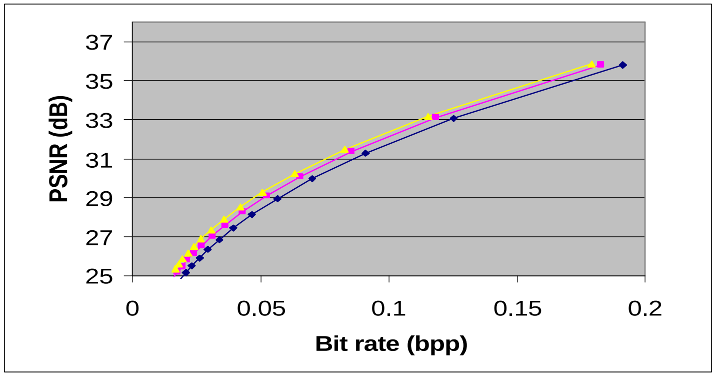
Synthetic sequence 1



Synthetic sequence 2

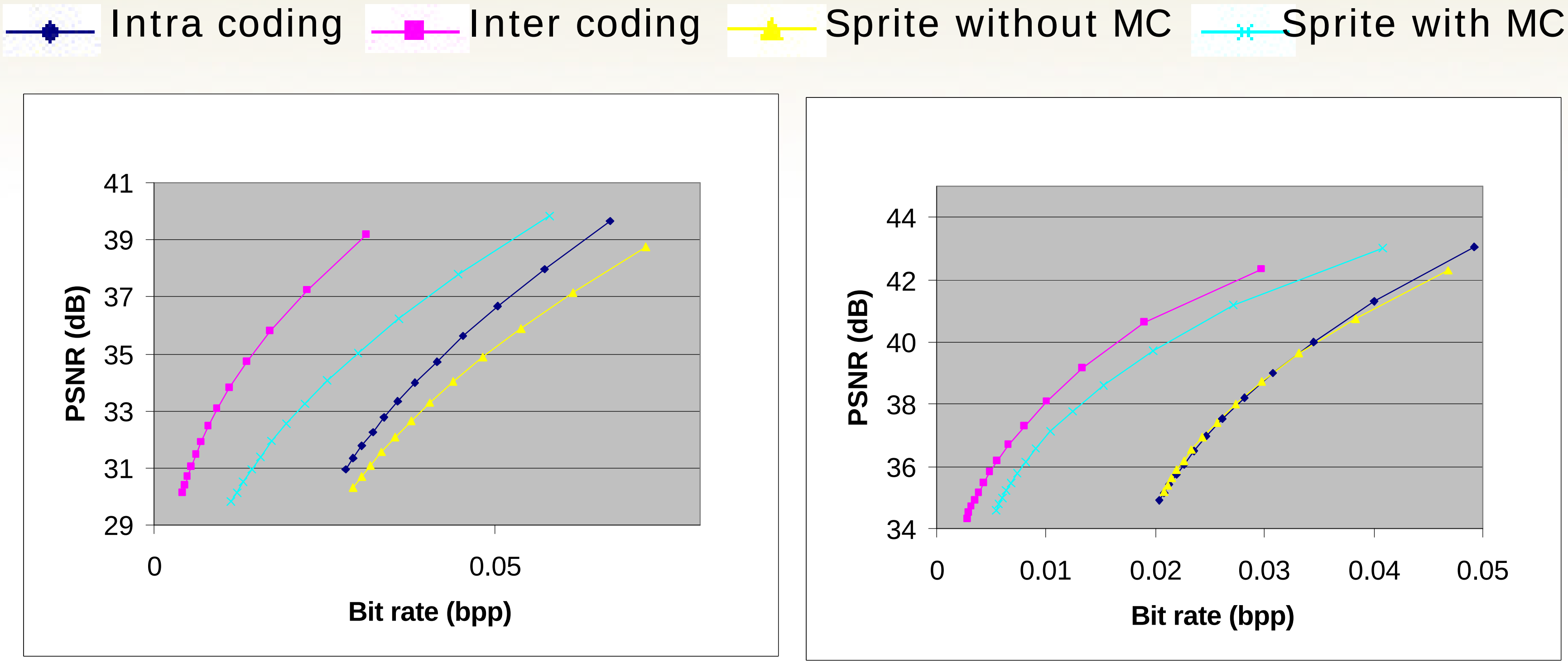


Real sequence 1



Real sequence 2

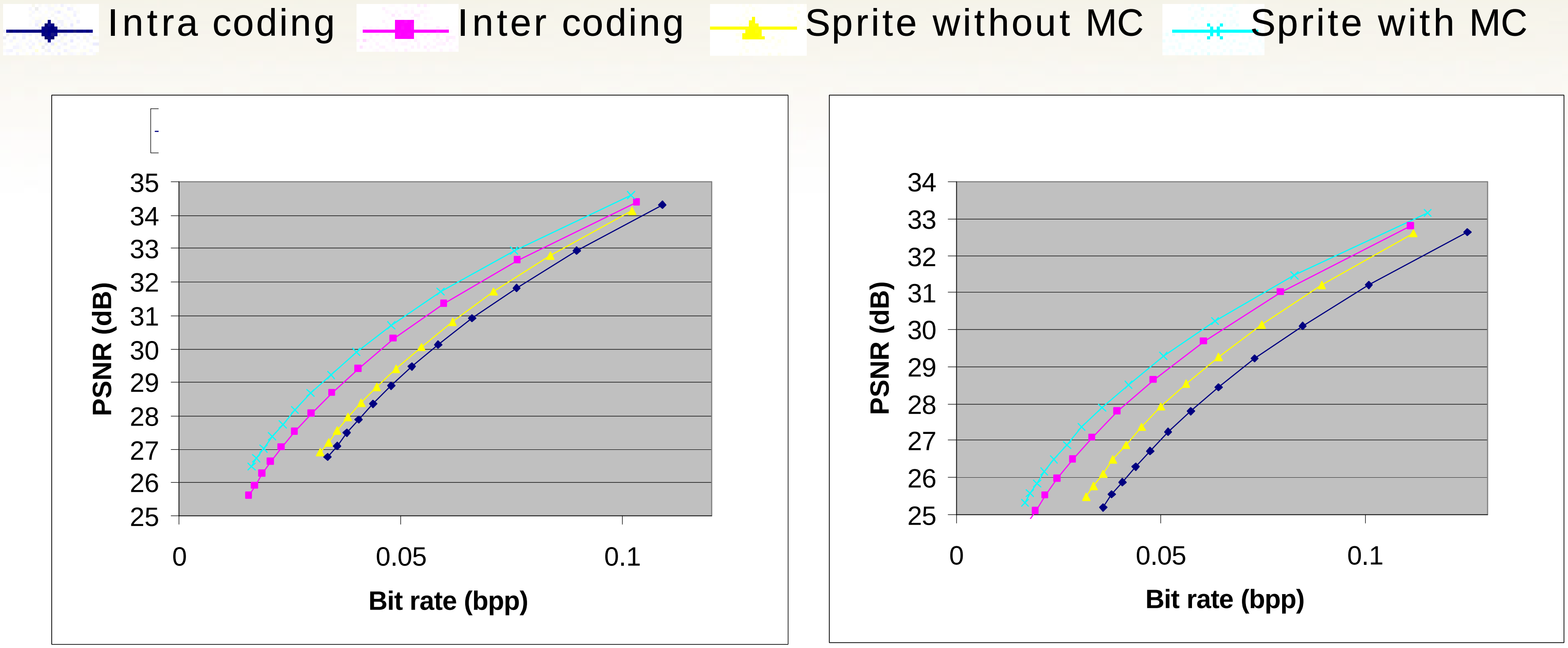
Compression Result



Synthetic sequence 1

Synthetic sequence 2

Compression Result

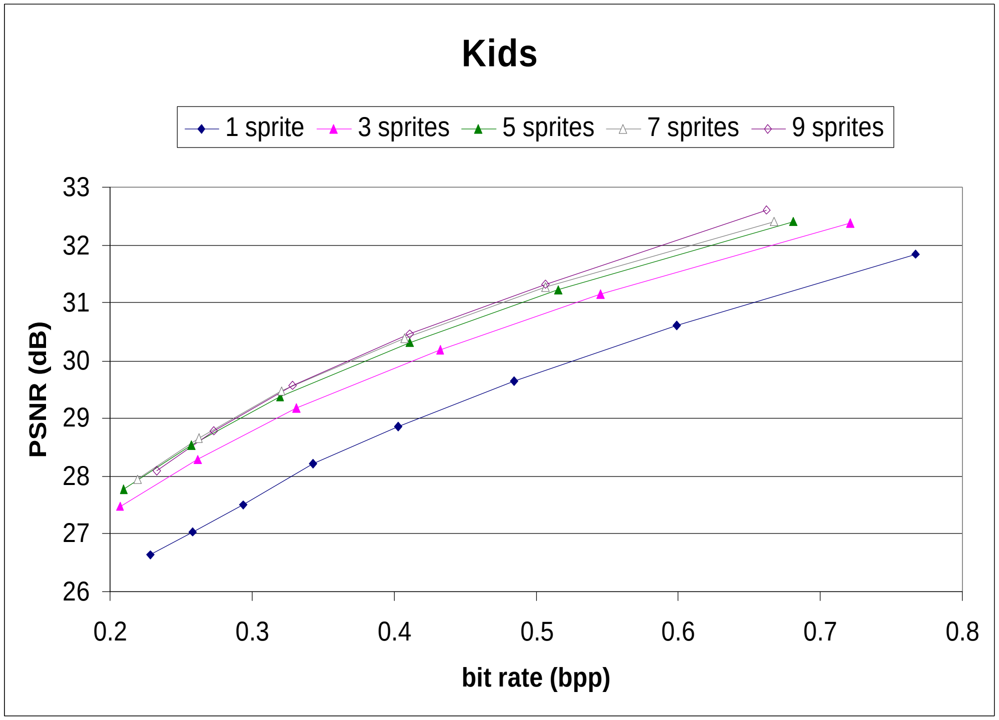


Real sequence 1

Real sequence 2

Enhancements

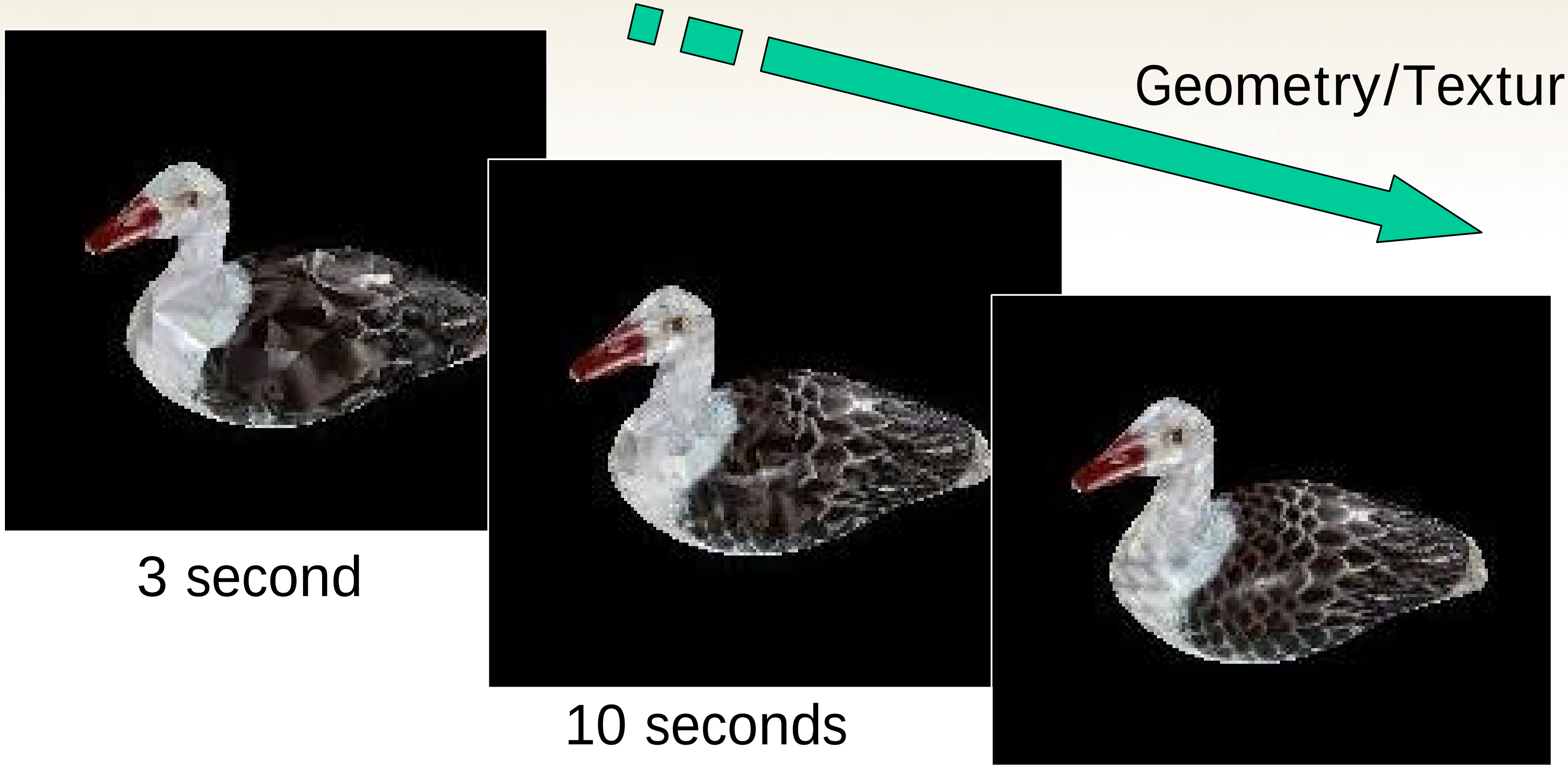
- Window size for searching offsets
- Stripe motion compensation
- MC using a large reference frame
- Multiple sprites



Recap...

- Sprite prediction with MC better than Intra coding
  - Sprite prediction with MC is preferred for random access
  - Better than Inter coding for real data
- Delta weighting is the best for constructing the sprite
- Can be extended to higher dimensions
  - Lumigraph, lightfield, etc.

Streaming 3D

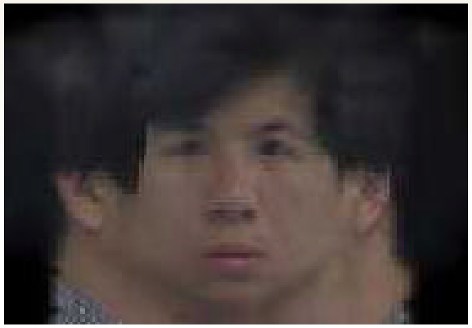


3 second

10 seconds

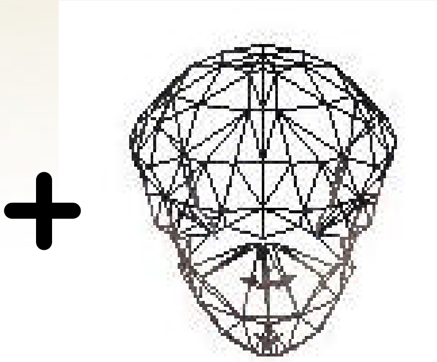
40 seconds

Texture + Geometry = 3D Object

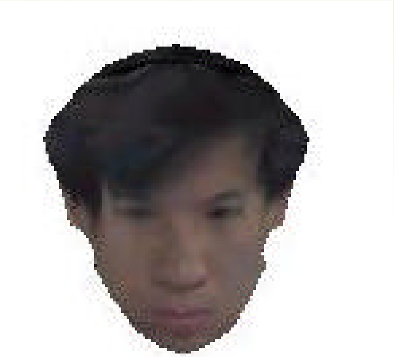


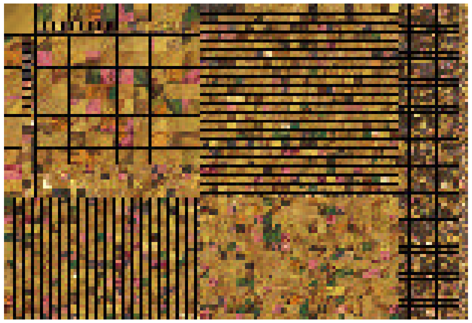
Vertex-Based

+



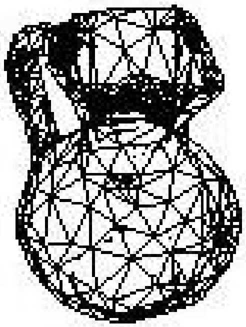
=



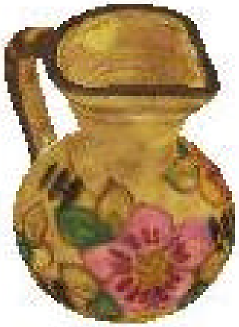


Corner-Based

+



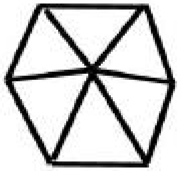
=



Why Compression?

- Each vertex: three floating-point numbers
- If each vertex shared by 6 triangles, and max number of vertices per model is 2^{20}

?   108 bits/triangle needed



$$\left(\frac{1}{6}\right)*\left(\frac{3\,vertices}{triangle}\right)*\left(\frac{32\,bits*3}{vertex}\right)\;?\;\left(\frac{3\,vertex\,IDs}{triangle}\right)*\left(\frac{20\,bits}{vertex\,ID}\right)\;?\;\frac{108\,bits}{triangle}$$

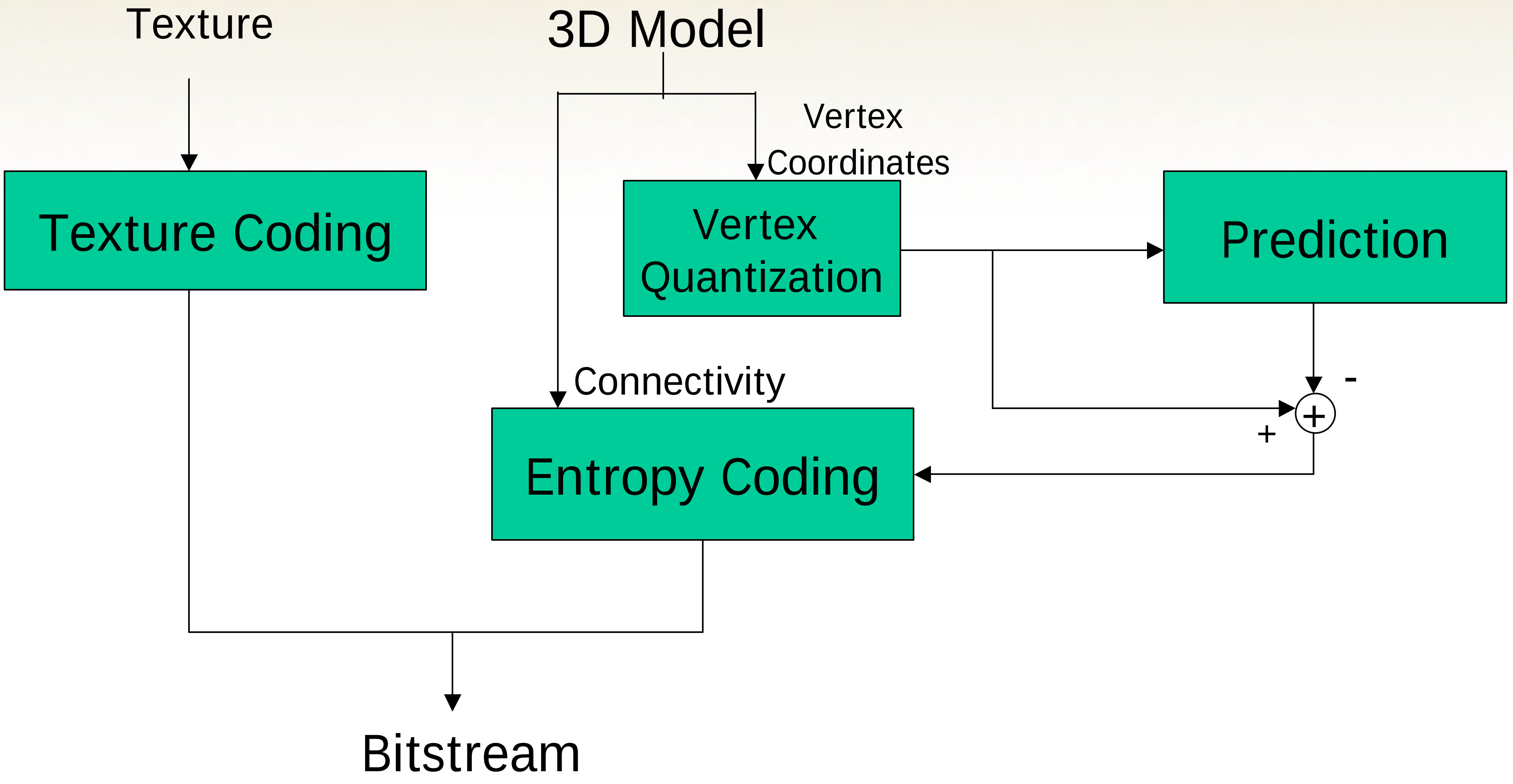
? 100KB~1MB for an average model + texture

Compression of 3D Objects

- Texture compression
  - Static textures: JPEG or JPEG 2000
  - Dynamic textures: MPEG or H.263
- Geometry compression
  - Quantization of vertex coordinates
  - Predictive coding
  - Entropy coding
- Granular/stable progressive coding
- Mesh optimization/simplification

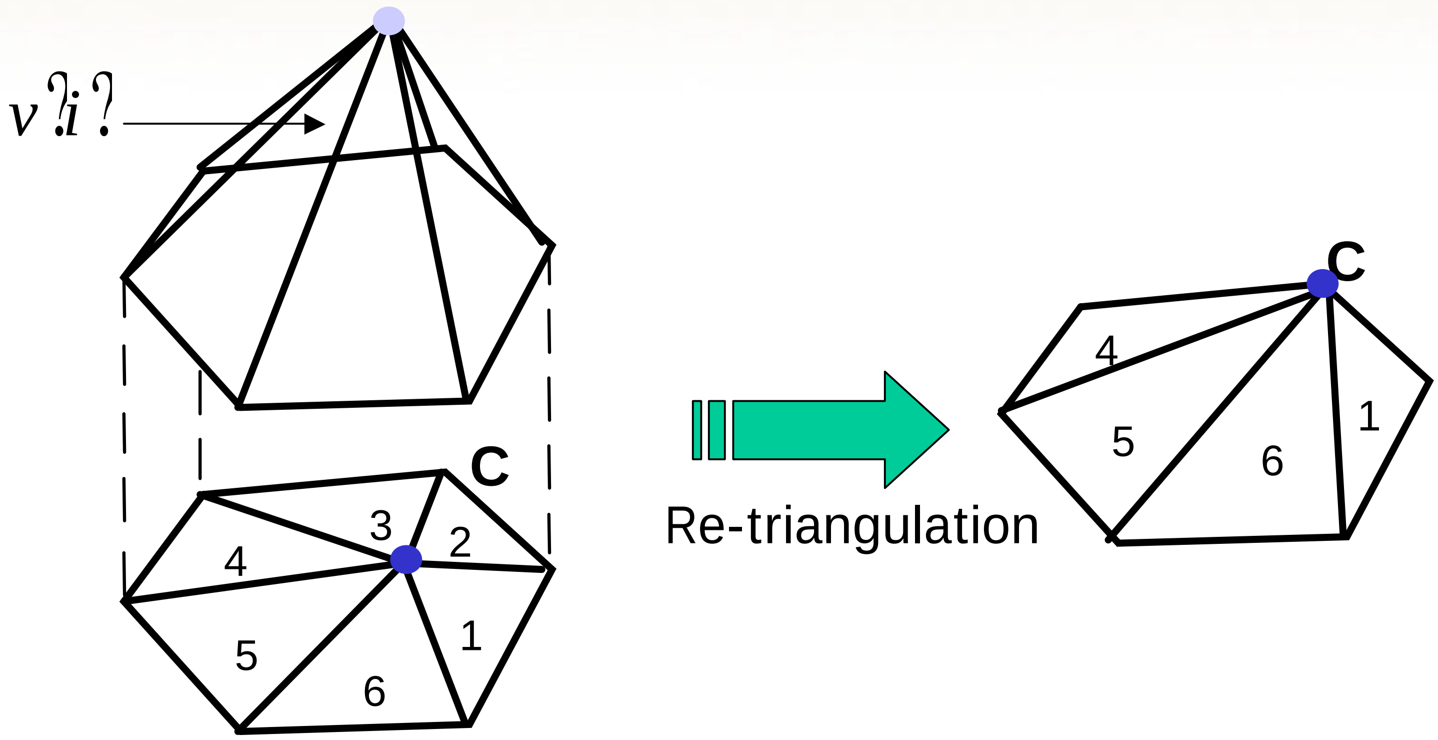
[Hoppe et al][Heckbert et al][Schroder et al][Taubin et al.]

Block Diagram



Encoding

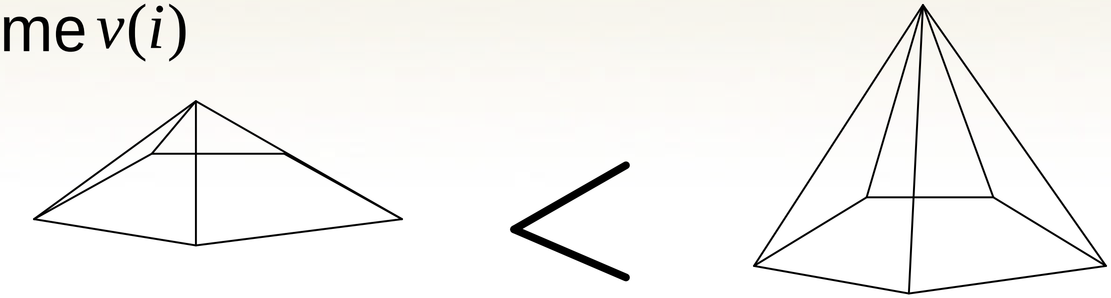
- Vertex decimation



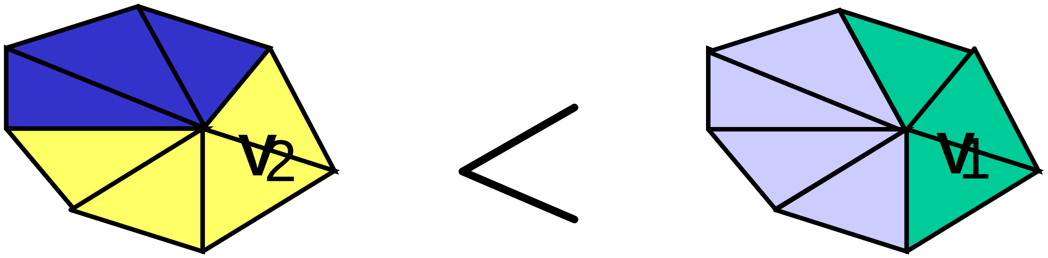
Re-triangulation

Importance of Vertices

(1) Volume $v(i)$



(2) Color $c(i)$



- Rank all vertices from high to low based on a cost function:

$$m(i) ? ? v(i) ? (1 ? ?)c(i)$$

v (i) is the geometry cost

c (i) is the texture/color/normal cost

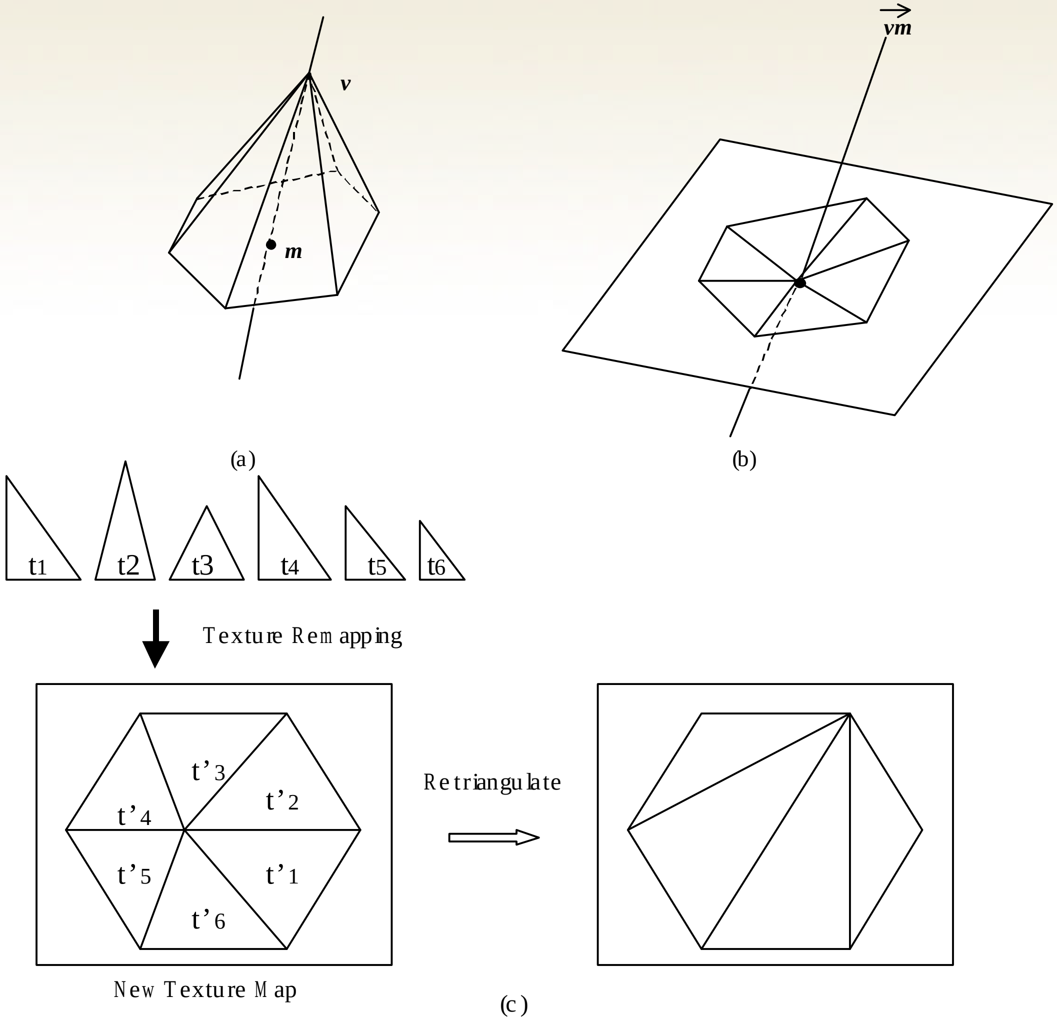
? is an user-specified parameter

- Decimate the vertices with low cost first
- Transmit the vertices with high cost first

Coding of Texture

- Vertex-based
  - Wavelet (SPIHT) + entropy coding
- Corner-based
  - Padding + DCT + run-length coding + entropy coding
  - Texture re-mapping needed

Texture Re-Mapping



(a)

(b)

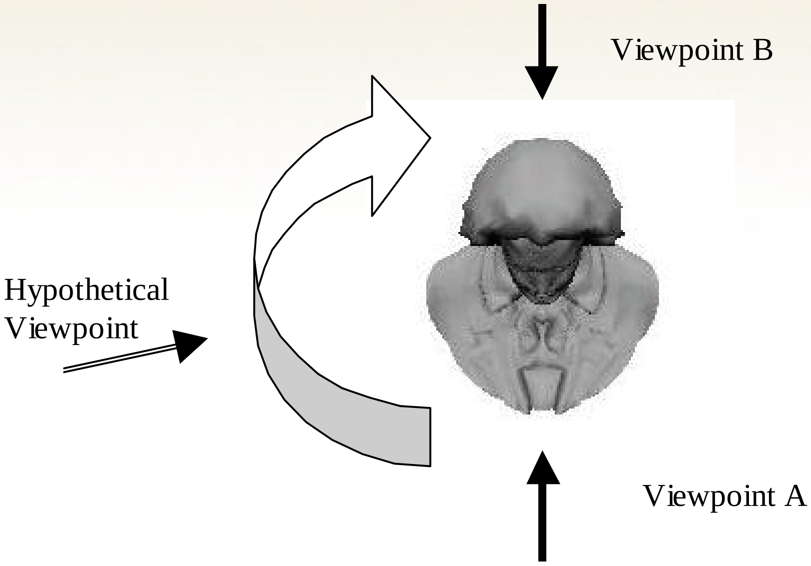
(c)

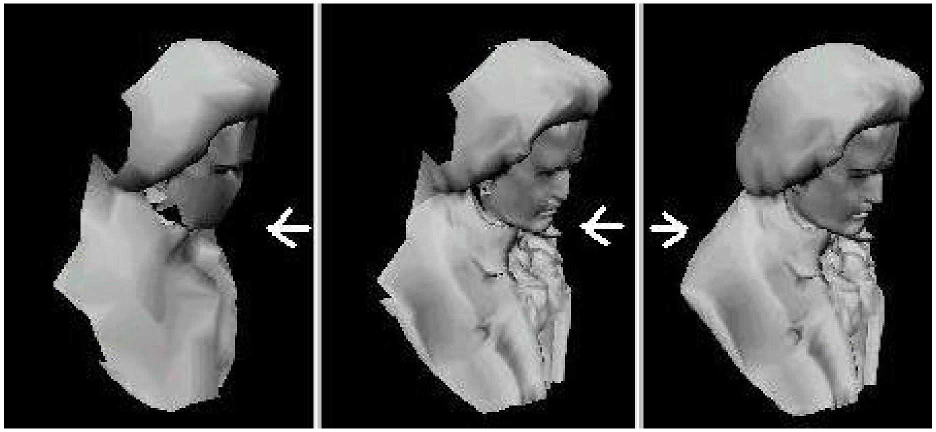
Comparison

	Our Algorithm	MPEG-4	VRML + gzip	Attributes
Beethoven	10	14	50	None
Cow	12	15	67	None
Crocodile	36	41	234	none
Horse	40	48	266	none
Pieta	14	18	79	none
Duck	15	-	605	texture
Vase	126	-	651	texture
Totem	160	-	683	texture

(in KBytes)

View-Adaptive Transmission





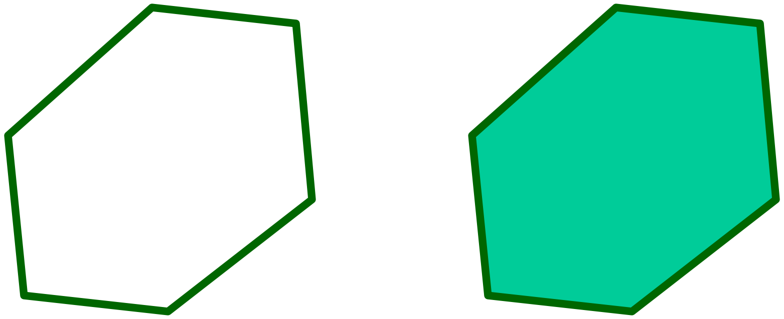
Demo

Retrieval of 3D Objects

- Indexing and retrieval
  - Much is done for images
    - [Huang et al][Cox et al]
  - Recent work for 3D objects
  - Related to MPEG-7
- Feature extraction
- Feature matching

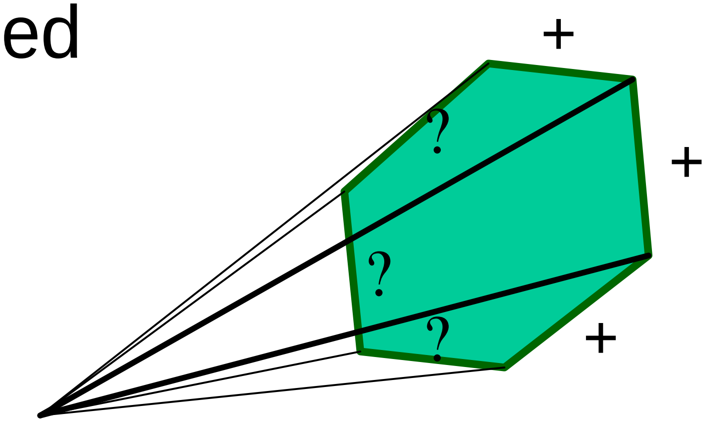
Feature Extraction

- Feature extraction
  - Traditionally vertex/surface-based
  - New region-based features
    - moment invariants, Fourier transform coefficients, etc.
  - Preprocessing to close the model

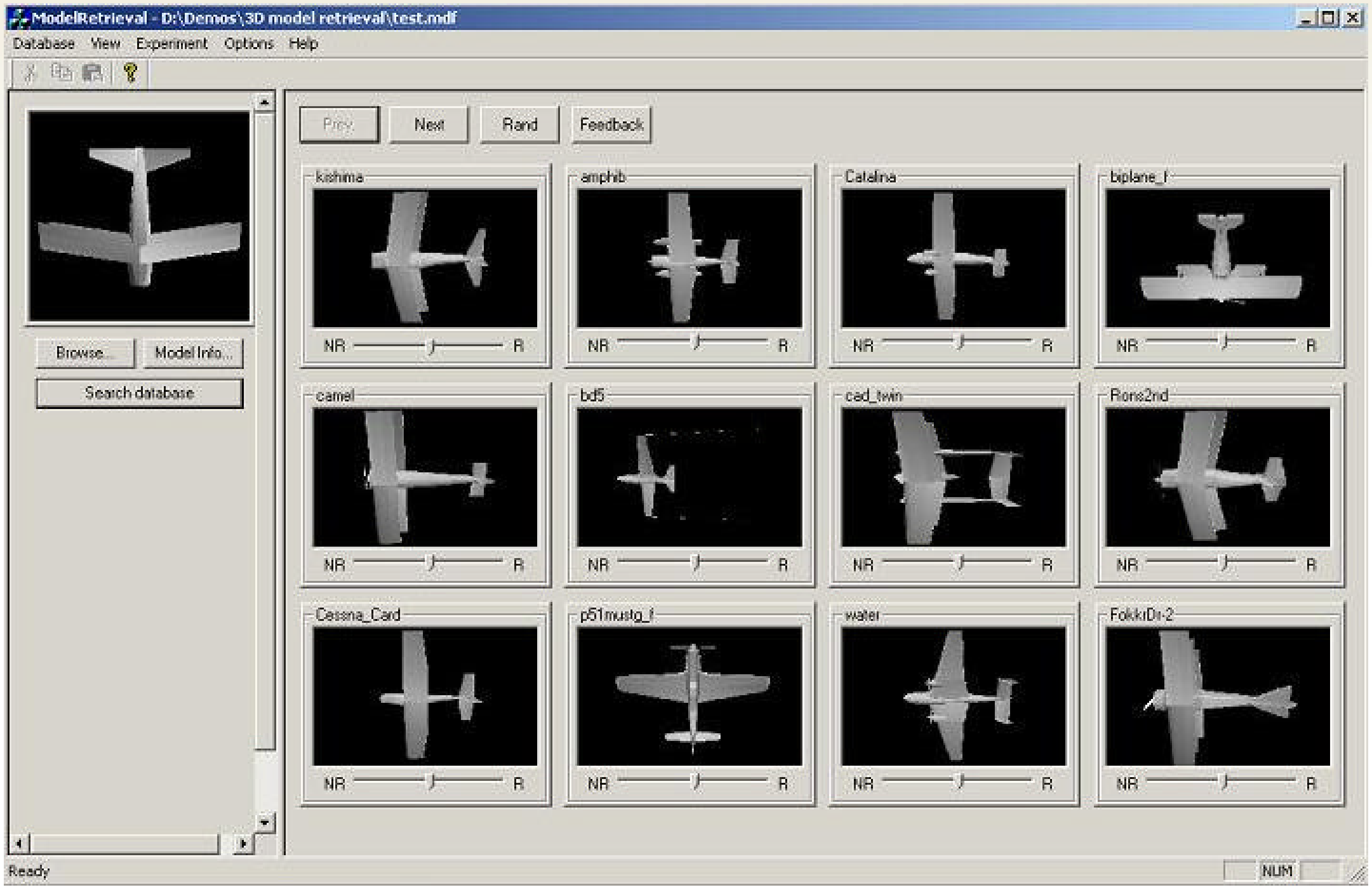


Feature Extraction (cont.)

- Efficiently calculate region-based feature directly from mesh
- Signed feature for each mesh element
- Robust to triangulation
- Applies to any feature that can be decomposed to each mesh element



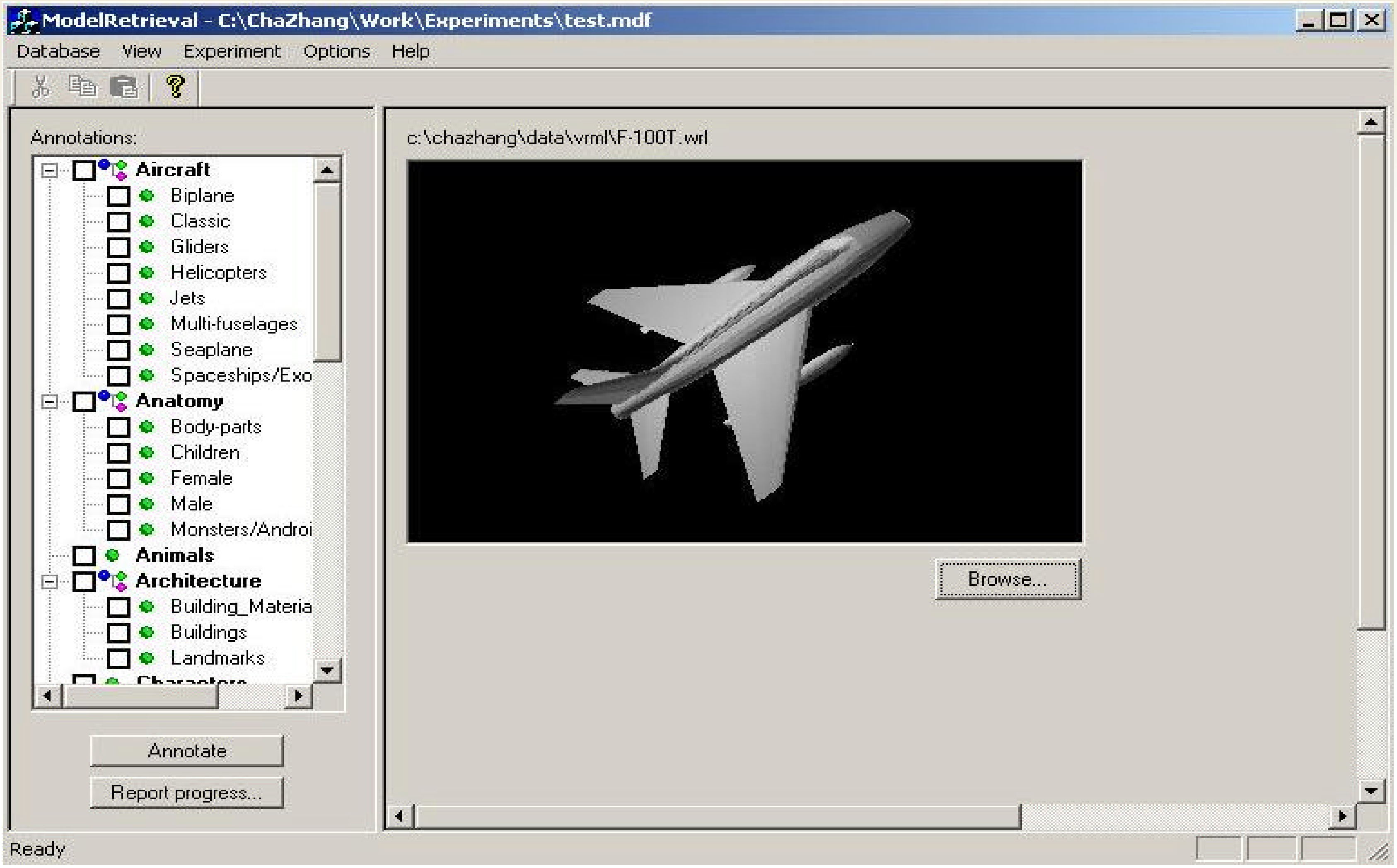
3D Model Retrieval



Annotation and Active Learning

- Semantic thru annotation is needed
  - Low level features not enough
  - Hierarchical annotation
  - Compatible concepts in annotation
- Active learning
  - Complete annotation is impractical
  - Select the object most uncertain for annnoation

Annotation



Annotations:

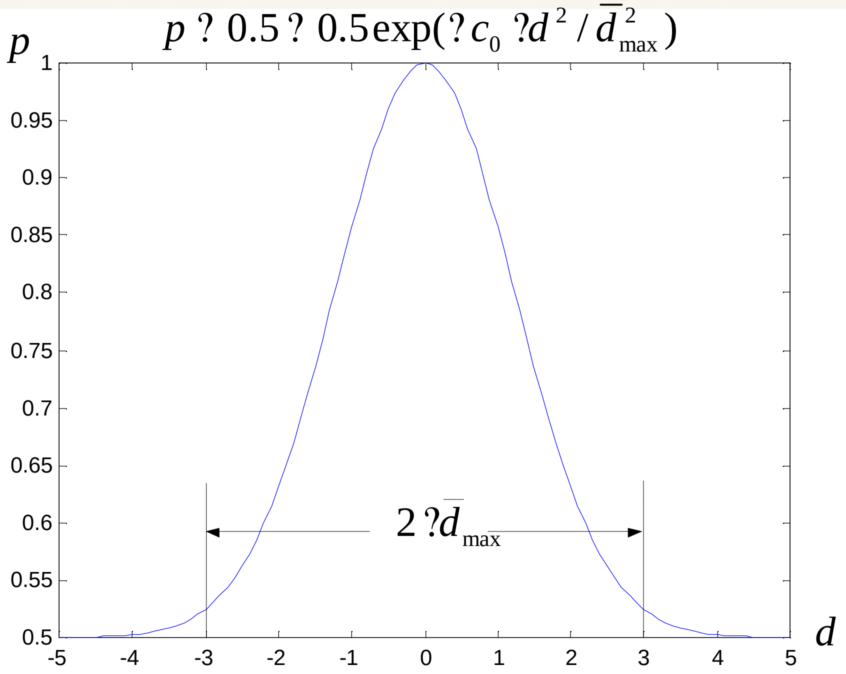
- □ Aircraft
  - □ Biplane
  - □ Classic
  - □ Gliders
  - □ Helicopters
  - □ Jets
  - □ Multi-fuselages
  - □ Seaplane
  - □ Spaceships/Exo
- □ Anatomy
  - □ Body-parts
  - □ Children
  - □ Female
  - □ Male
  - □ Monsters/Androi
- □ Animals
- □ Architecture
  - □ Building_Materia
  - □ Buildings
  - □ Landmarks
- □ Characters

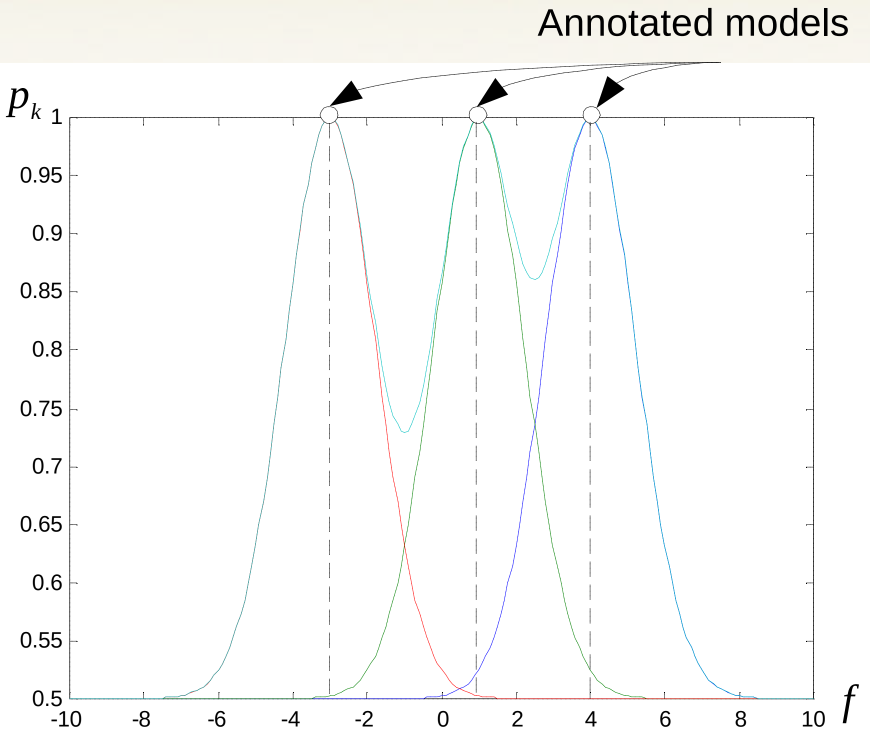
c:\chazhang\data\vrml\F-100T.wrl

Active Learning

- For each model, each concept, we maintain a probability of this model belonging to this concept
- Set the probability to 1 or 0 if annotated
- Estimate probabilities of the unlabeled objects with *potential function*
- Use the probabilities to estimate uncertainty and to measure the semantic distance

Active Learning





One annotated neighborhood

Multiple annotated neighborhoods

The potential function

Estimate the Uncertainty

- The general criterion:

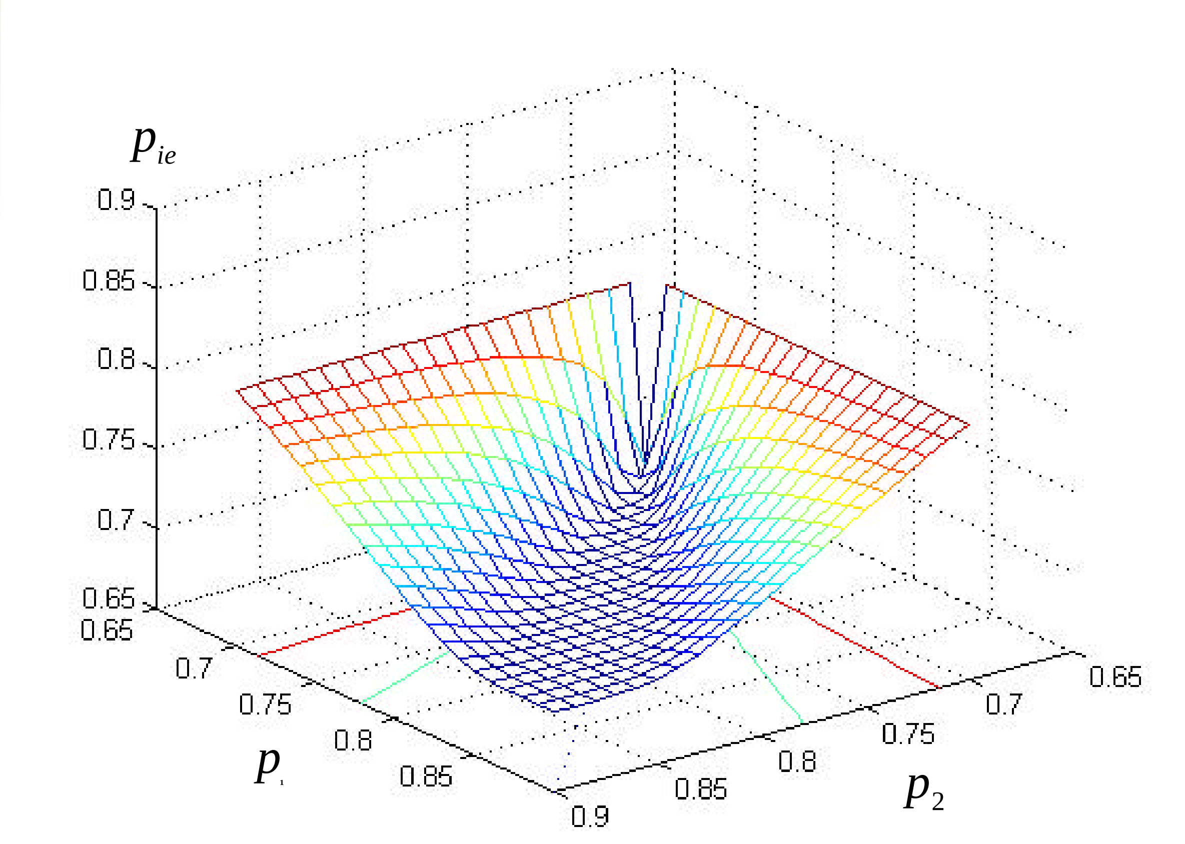
$$G_i ? f_{Di} ?U_i ? f_{Di} ?? (p_{i1}, p_{i2}, ..., p_{iK}), \qquad i ? 1, 2, ..., N.$$

- f_{Di}: local density function
- U_i: uncertainty measurement

$$p_{i\max} ? \max_k (p_{ik}), k ? 1, 2, ..., K$$

$$U_i ? ? (p_{i1}, p_{i2}, ..., p_{iK}) ? ? p_{i\max} \log p_{i\max} ? (1 ? p_{i\max}) \log(1 ? p_{i\max}).$$

Estimate the Uncertainty



Semantic Distance

- Cannot use Kullback-Leibler convergence
- Our semantic disance is defined as:
  - Annotated models

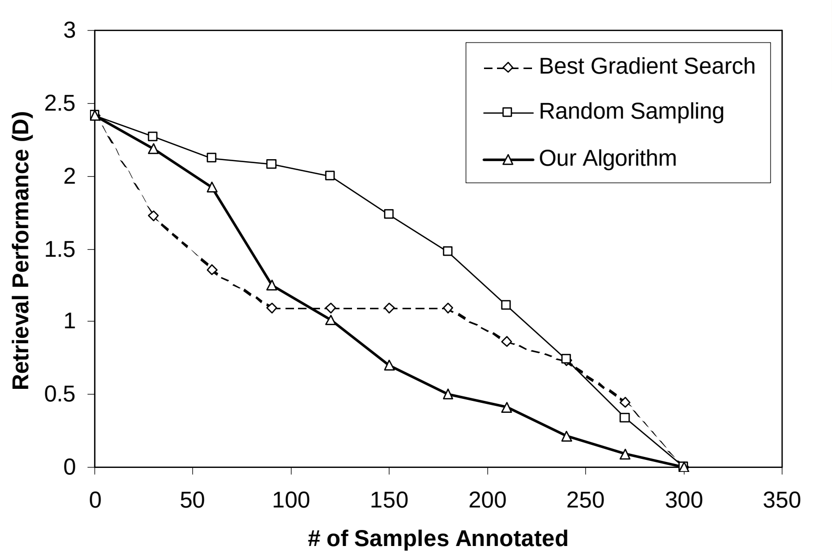
$$LowestLevel \; ? \; \max_k ?l ? 1 : p_{ik} \; ? \; p_{jk} \; ? \; 1, \textit{and} \text{ concept } k \text{ is at level } l \text{ in the concept tree} ?,$$

$$d_s \; ? \; ?^{LowestLevel}.$$

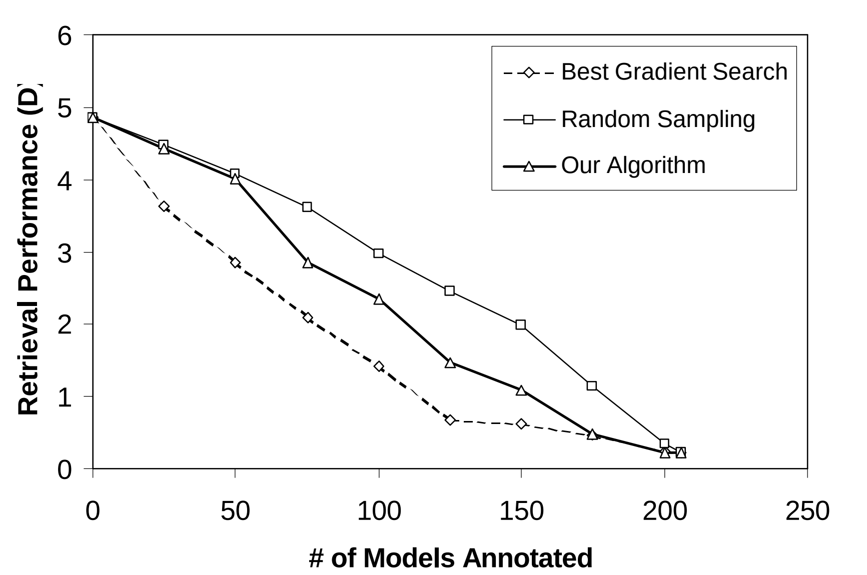
  - Unannotated modes

$$\begin{array}{l} ? \; d_{sk} \; ? \; p_{ik}(1 ? \; p_{jk}) ? \; p_{jk}(1 ? \; p_{ik}), k \; ? \; 1,2,\ldots,K; \\ ? \\ ? \; k_0 \; ? \; \arg\max_k ? \begin{array}{l} ?l : d_{sk} \; ? \; T_2, ?p_{ik} \; ? \; 0.5 \textit{ or } p_{jk} \; ? \; 0.5?, \\ ?\textit{and} \text{ concept } k \text{ is at level } l \text{ in the concept tree} ? \end{array} ?; \\ ? \; LowestLevel \; ? \; ?l \; ? \; 1 : \text{concept } k_0 \text{ is at level } l \text{ in the concept tree} ?; \\ ? \; d_s \; ? \; ? \begin{array}{ll} 1 ? \min_k d_{sk}, & \text{if } LowestLevel \; ? \; 0 \\ (1 ? \; d_{sk_0}) ??^{LowestLevel}, & \text{if } LowestLevel \; ? \; 0 \end{array} . \end{array}$$

Results

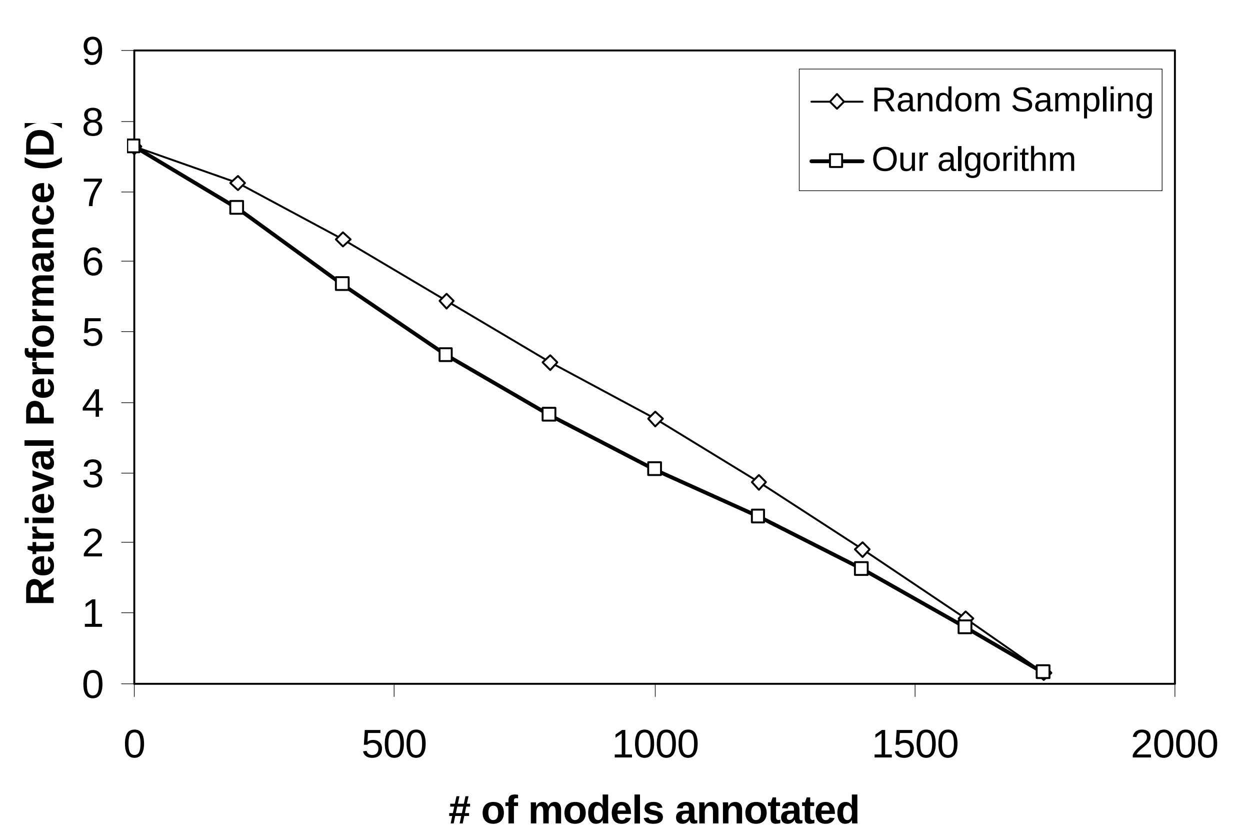


Synthetic database



A small database

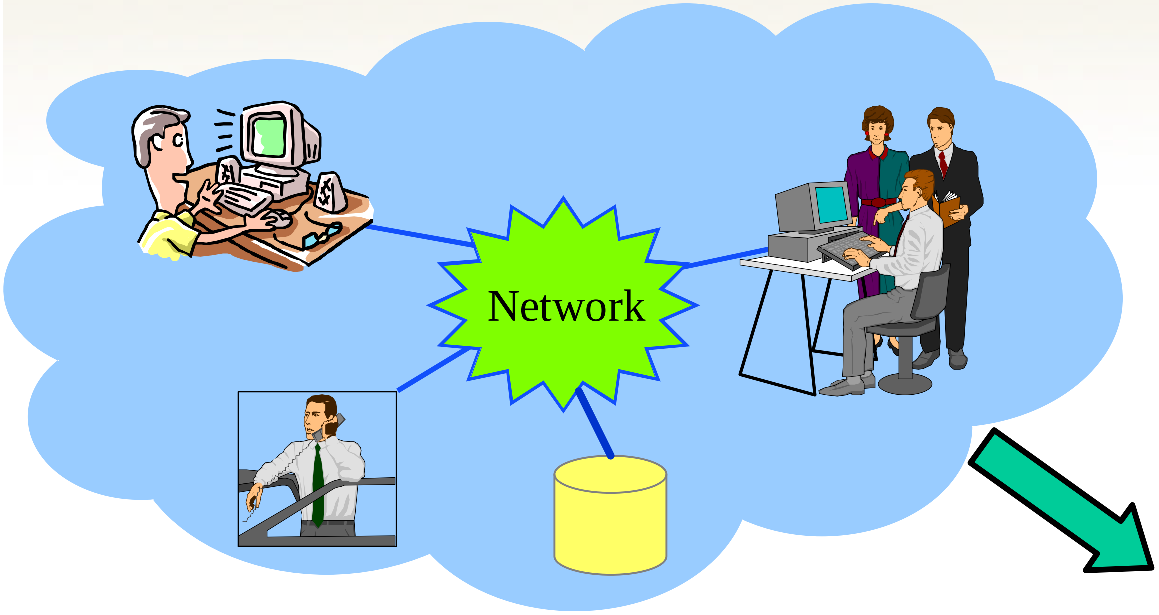
Results (cont.)



Recap...

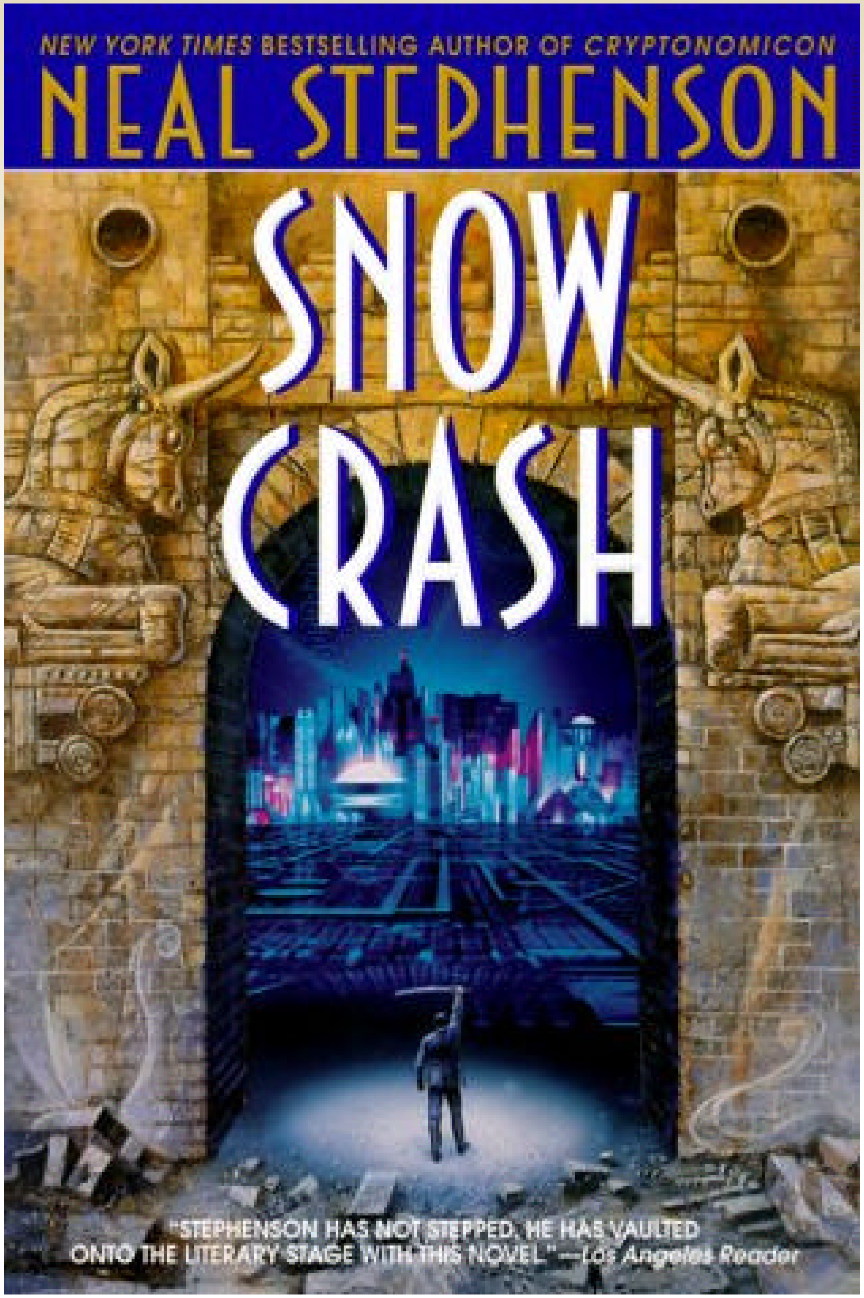
- New feature set for 3D models
- Active learning to improve annotation efficiency
- Compatible concept tree for annotation
- Probability for both uncertainty estimation and semantic distance

Immersive Environments



"Transparent" Communication

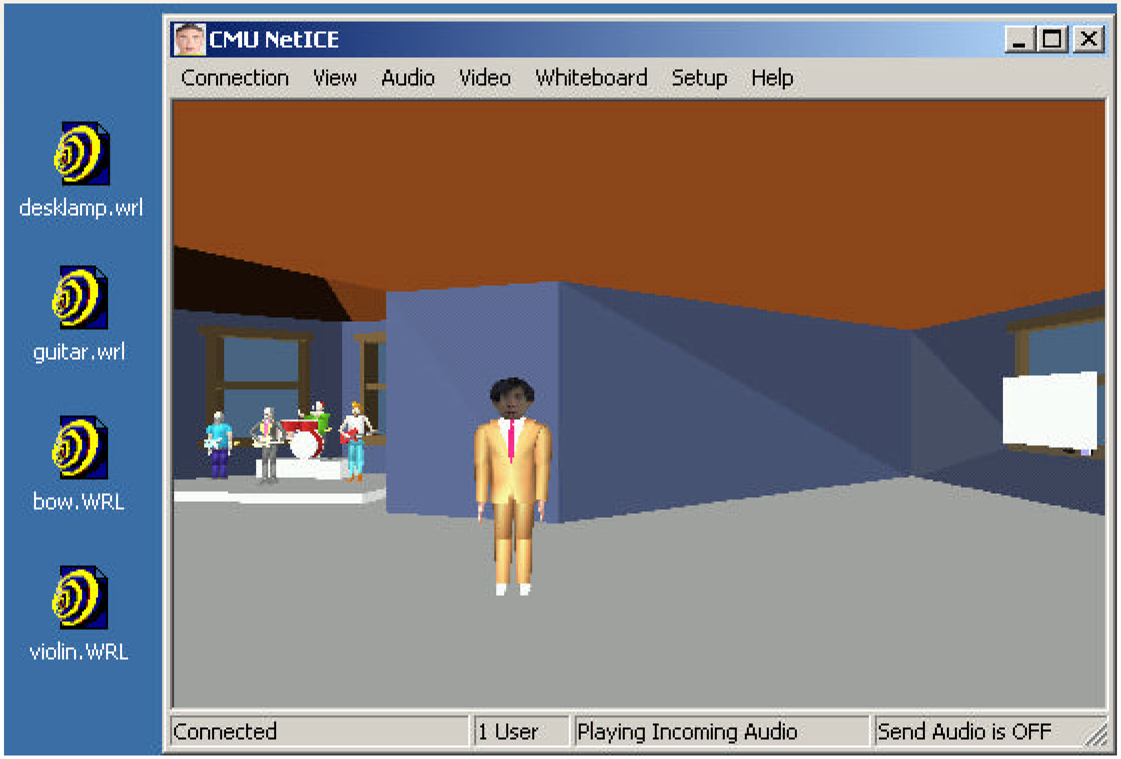
" Collaboration from anywhere, through any media, as if face-to-face in one room"

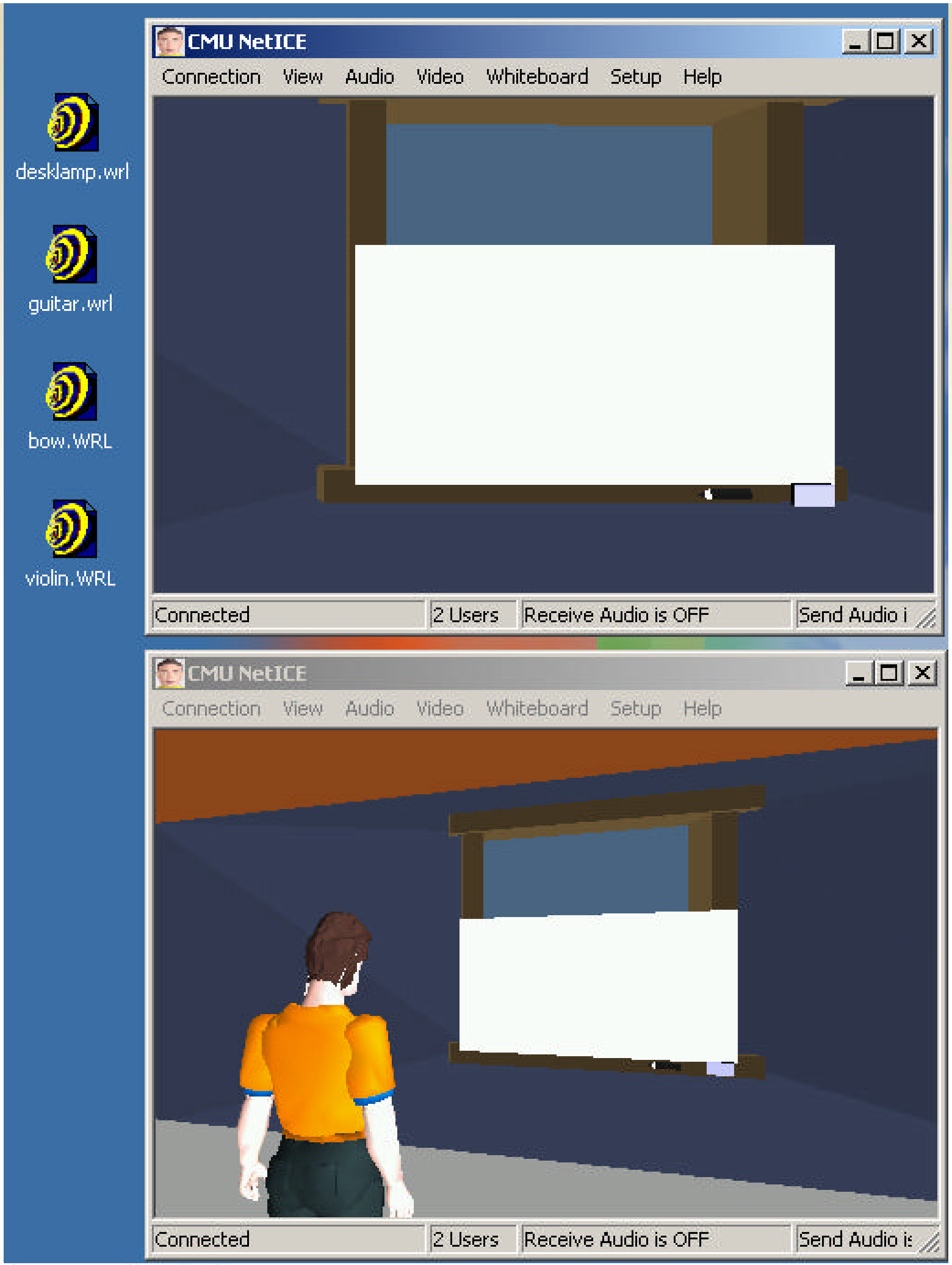

NEW YORK TIMES BESTSELLING AUTHOR OF CRYPTONOMICON
NEAL STEPHENSON
SNOW CRASH
"STEPHENSON HAS NOT STEPPED, HE HAS VAULTED ONTO THE LITERARY STAGE WITH THIS NOVEL." —Los Angeles Reader

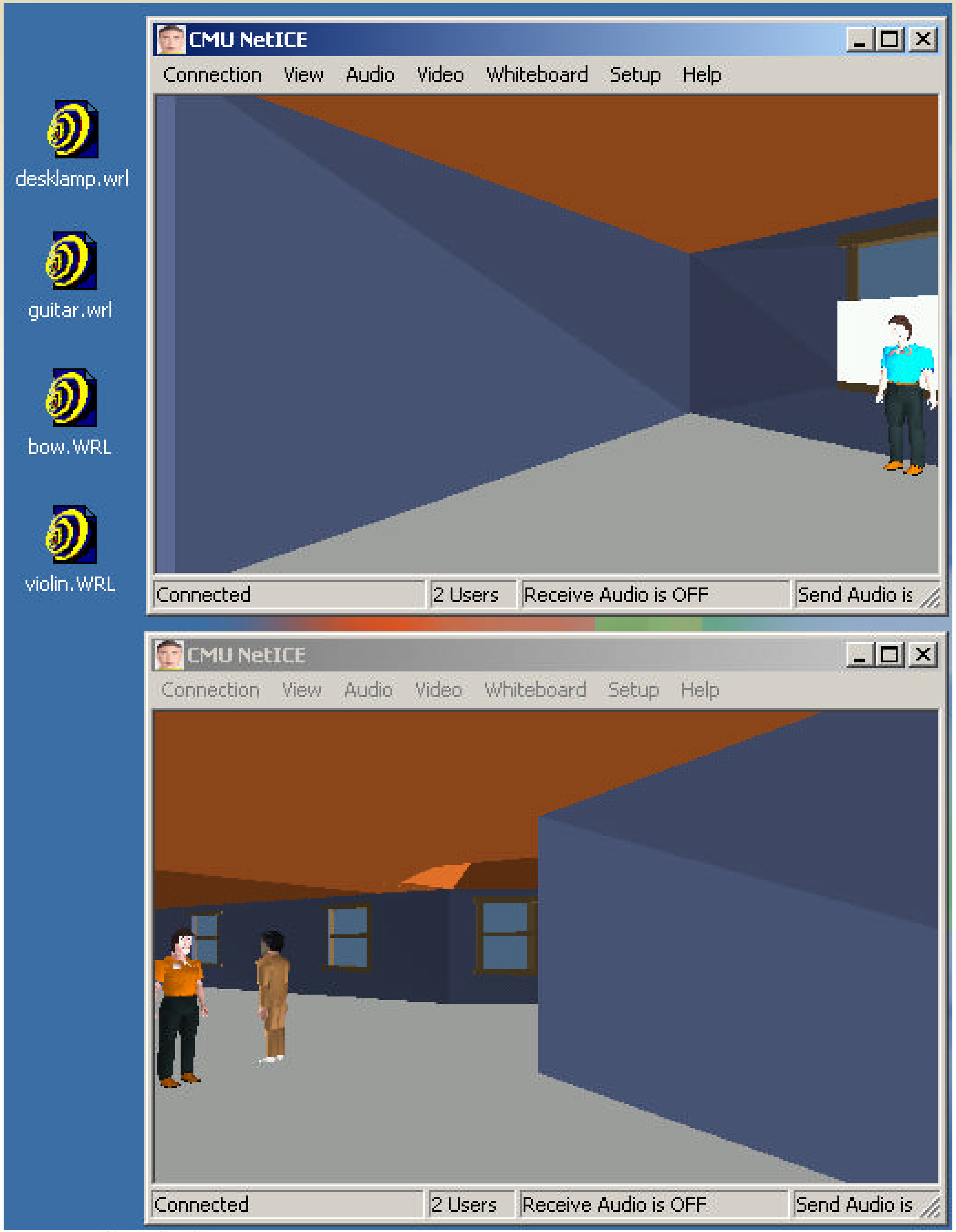
A Prototype

- NetICE: Networked Intelligent Collaborative Environment
- Lip-sync facilitates speech understanding
  - Who is speaking and what is being said
- Consistent spatial relationship with eye contact
  - Whom is spoken to
- Facial expressions and voice-driven hand gestures
- Directional sound give sense of distance and direction
  - Who is where; Who is speaking
  - Enable small-group interaction in a room full of people
- Information sharing
  - Shared whiteboard
  - Streaming 3D objects
  - Enable collaborative design, e.g., cars, buildings, etc.

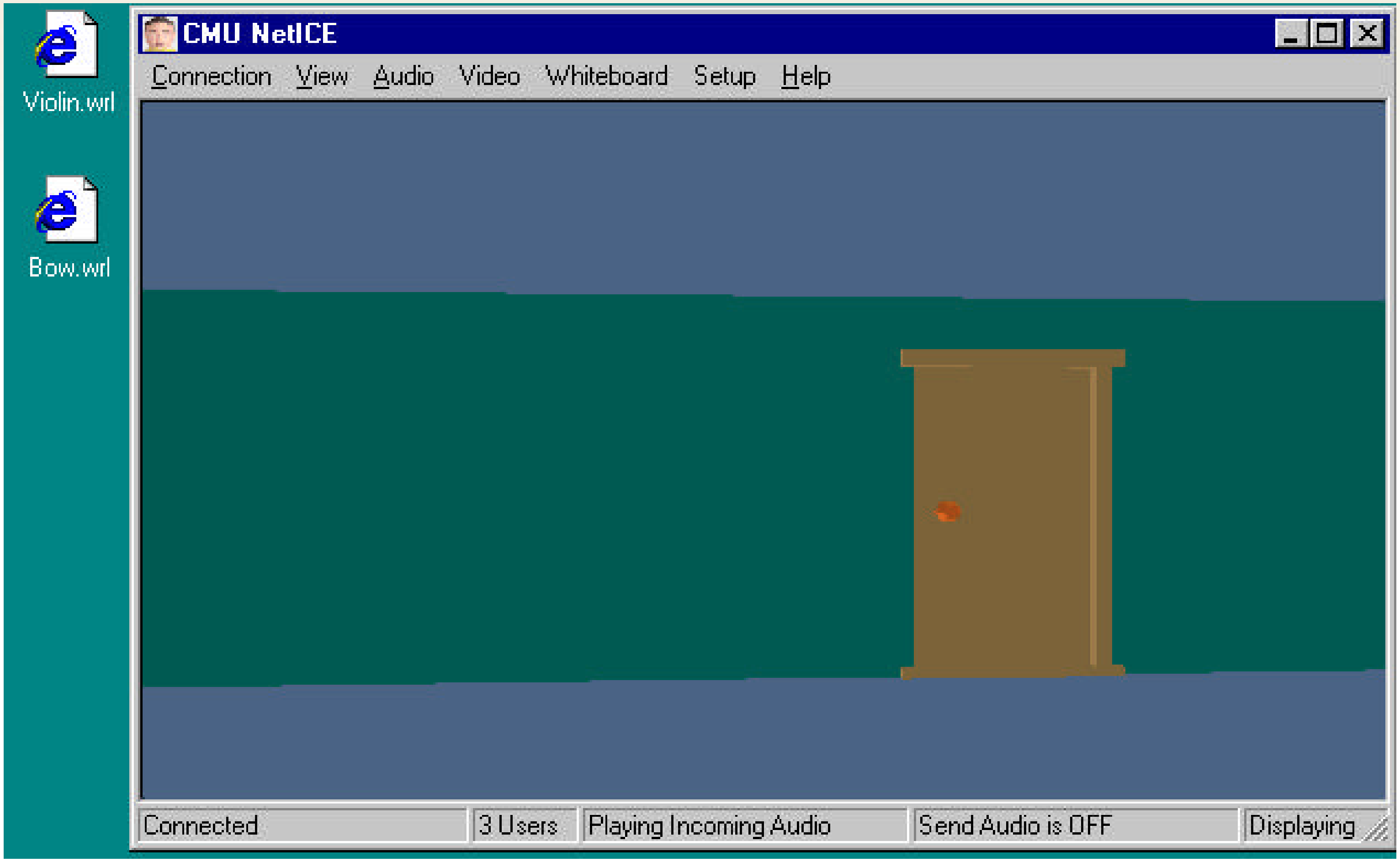
NetICE


desklamp.wrl
guitar.wrl
bow.WRL
violin.WRL
CMU NetICE
Connection
View
Audio
Video
Whiteboard
Setup
Help
Connected
1 User
Playing Incoming Audio
Send Audio is OFF

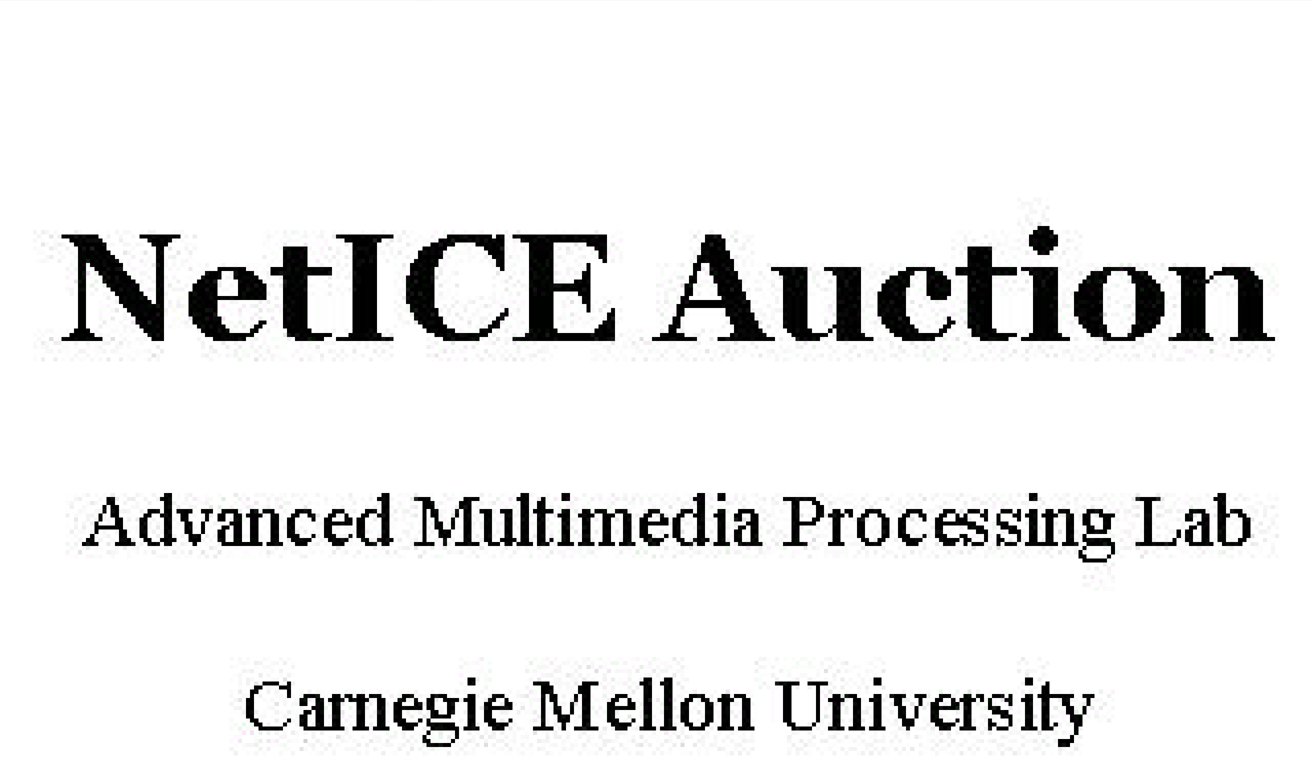

desklamp.wrl
guitar.wrl
bow.WRL
violin.WRL
CMU NetICE
Connection View Audio Video Whiteboard Setup Help
Connected
2 Users
Receive Audio is OFF
Send Audio i
CMU NetICE
Connection View Audio Video Whiteboard Setup Help
Connected
2 Users
Receive Audio is OFF
Send Audio is


desklamp.wrl
guitar.wrl
bow.WRL
violin.WRL
CMU NetICE
Connection View Audio Video Whiteboard Setup Help
Connected
2 Users
Receive Audio is OFF
Send Audio is
CMU NetICE
Connection View Audio Video Whiteboard Setup Help
Connected
2 Users
Receive Audio is OFF
Send Audio is

NetICE


Violin.wrl
Bow.wrl
CMU NetICE
Connection View Audio Video Whiteboard Setup Help
Connected
3 Users
Playing Incoming Audio
Send Audio is OFF
Displaying

Case Study: Online Auction



NetICE Auction

Advanced Multimedia Processing Lab

Carnegie Mellon University

Ongoing Work

- Use IBR for background rendering
- User study
  - Together or on-location
- Tracking for rendering
  - Head tracking for head orientation
  - Gaze tracking for eye contact
  - Hand tracking for hand gestures

Summary

- Compression for IBR
- Compression for 3D meshes
- Indexing and retrieval of 3D objects
- Immersive environments

Advanced Multimedia Processing Lab

Please visit us at:

http://amp.ece.cmu.edu