Coding for Virtual and Augmented Reality

Philip A. Chou, Microsoft Research
Packet Video Workshop, 15 July 2016

VR puts *you* in a Virtual World

AR puts virtual objects in *your* world

HMDs for Virtual Reality are Enclosed

Oculus Rift

HTC Vive

Playstation VR

Cardboard VR

Gear VR

Daydream

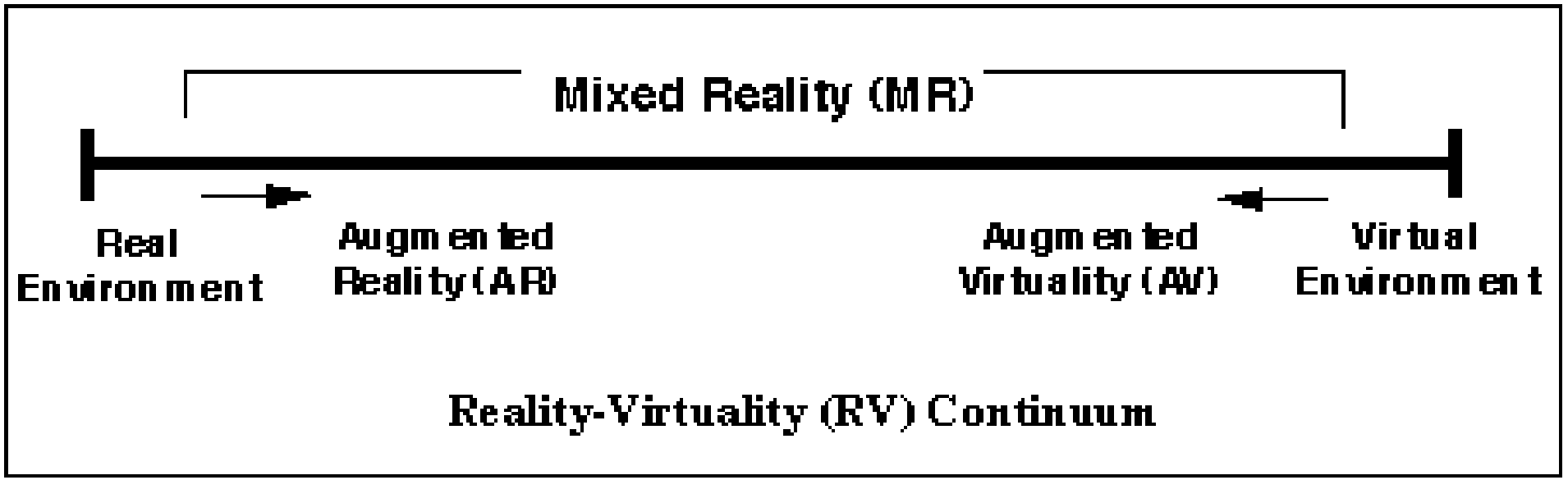
HMDs for Augmented Reality are See-Through

HoloLens

Meta

Daqri

Mixed Reality: from AR to VR

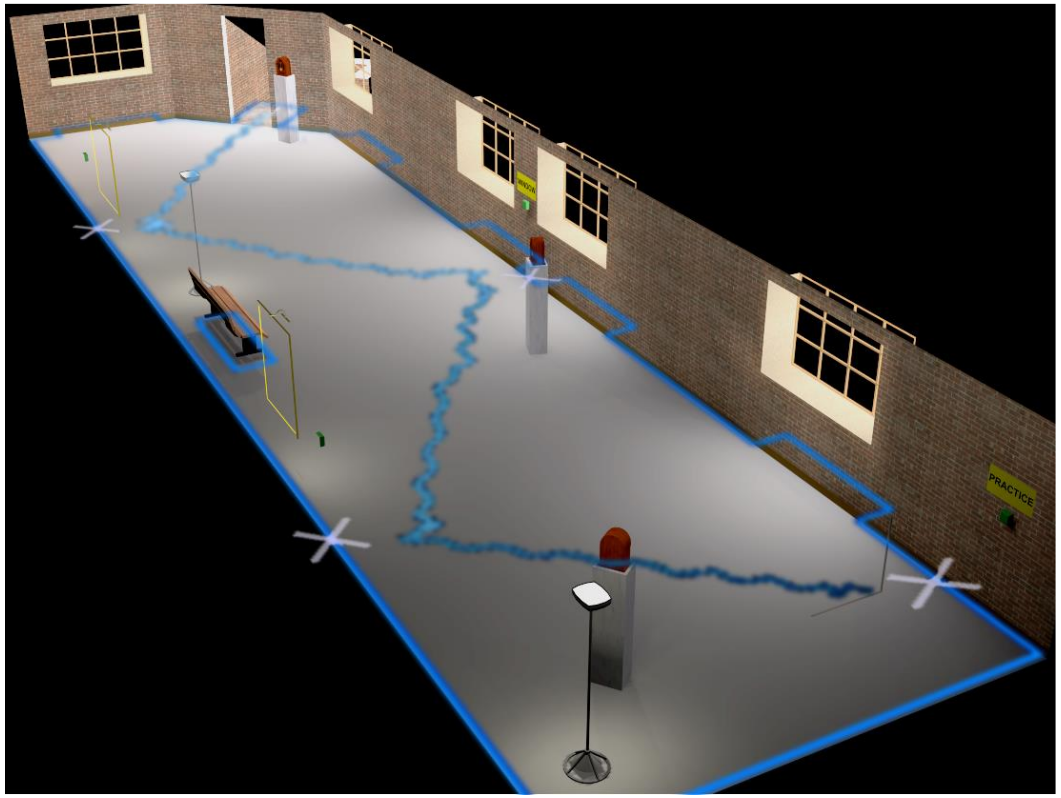


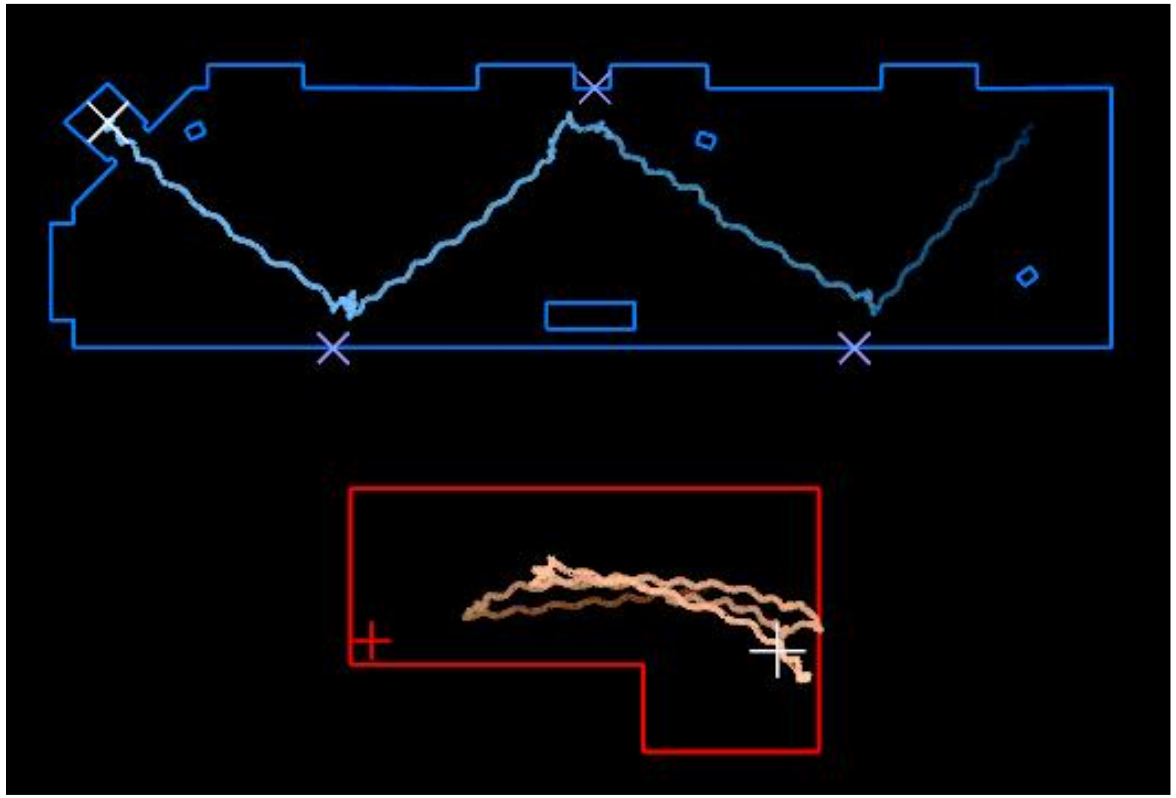
Paul Milgram, **Haruo Takemura**, **Akira Utsumi**, **Fumio Kishino**
Augmented reality: a class of displays on the reality-virtuality continuum
Proc. SPIE Telemanipulator and Telepresence Technologies, Dec 1995

Eventual Merging of VR and AR

- Video-see-through will be on VR devices
- Full immersion (high contrast and FOV) will be on AR devices

Redirected Walking in VR





Razzaque, Kohn, Whitten (2001)

“Cinematic” Content for VR and AR

VR

AR

VR Capture: “Inside-Out”

Lytro Immerge

Jaunt VR

Nokia Ozo

GoPro Odyssey

Samsung Beyond

Facebook Surround

AR Capture: “Outside-In”


JAUNT
X
THE NORTH FACE
NEPAL

High-Quality Streamable Free-Viewpoint Video

Alvaro Collet, Ming Chuang, Pat Sweeney,
Don Gillett, Dennis Evseev, David Calabrese,
Hugues Hoppe, Adam Kirk, Steve Sullivan

holoportation

http://research.microsoft.com/holoportation

Interactive 3D Technologies

http://research.microsoft.com/groups/i3d

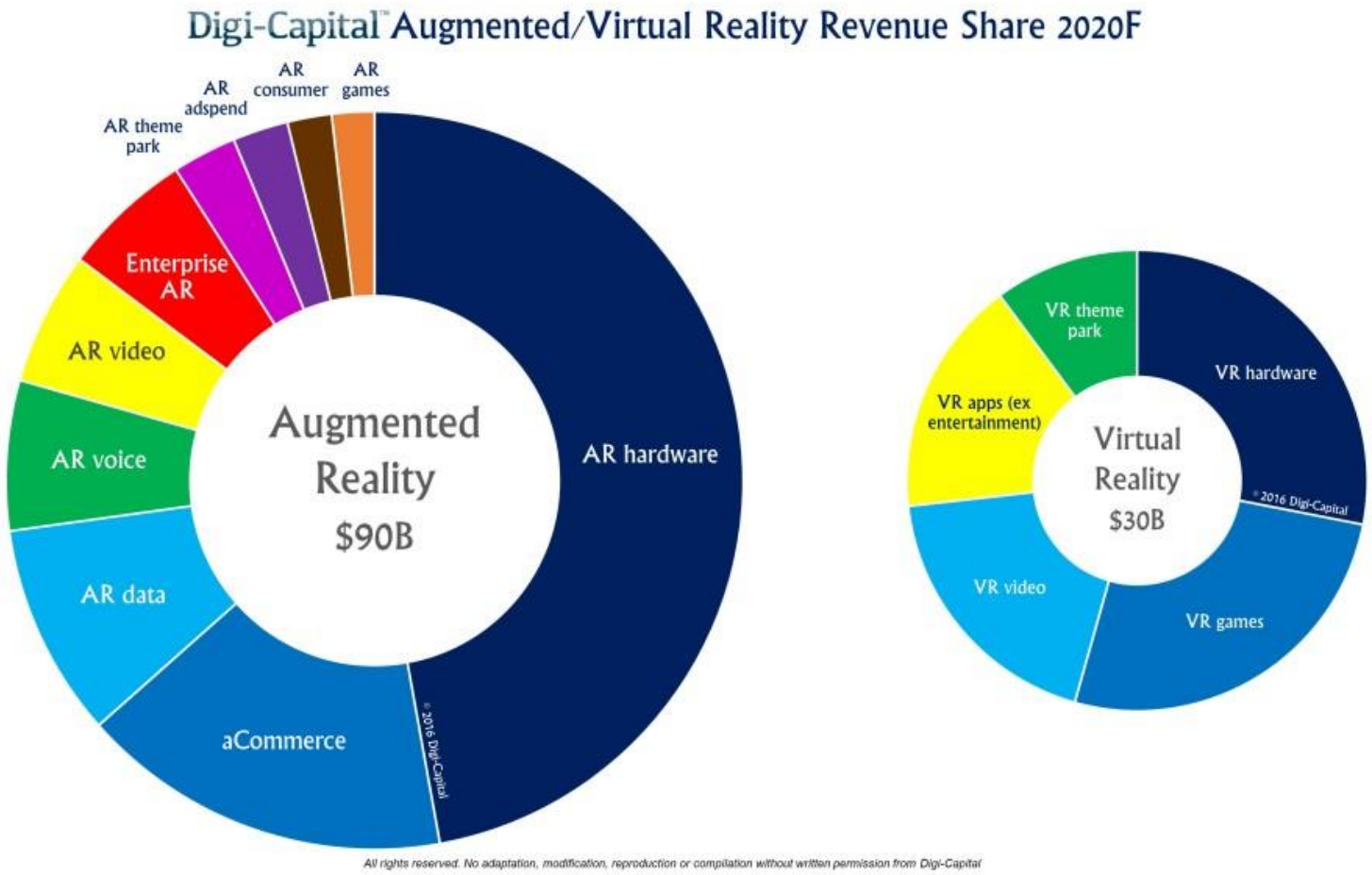
Microsoft Research

VR/AR: Third Generation of Immersive Communication

Immersive communication is the real time exchange of the natural social signals between people who are geographically separated, as if they were in co-located.

1G: TELEPHONE (1876) → 50 years → 2G: TELEVISION (1926) → 90 years → 3G: HOLOPORTATION (2016)

VR/AR: Fourth Generation Computing Platform
(after PC, web, and mobile)



http://www.digi-capital.com/news/2016/01/augmentedvirtual-reality-revenue-forecast-revised-to-hit-120-billion-by-2020/

Goldman Sachs: $80B by 2025

http://www.goldmansachs.com/our-thinking/pages/technology-driving-innovation-folder/virtual-and-augmented-reality/report.pdf

Coding

Degree of Motion Parallax determines Coding

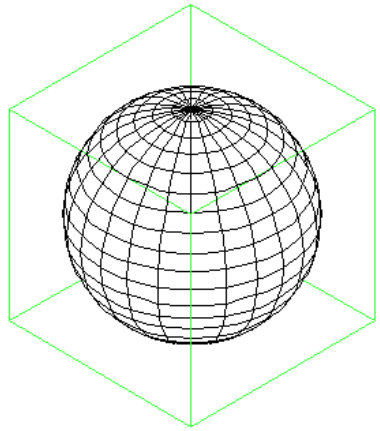
Virtual Reality (panoramic)

- Motion Parallax is limited because of limited movement
- Requires yaw, pitch, roll but not (much) x, y, z translation
- Stationary camera with 360° view makes sense (inside-out)
- Representation can be spherical video (possibly stereo, depth)
- Coding: map spherical video to rectangular video

Augmented Reality (volumetric)

- Motion Parallax is unlimited because roaming is allowed
- Requires x, y, z translation as well as yaw, pitch, roll (6DoF)
- Object capture makes sense (outside-in)
- Representation must be light field, mesh, point cloud, etc.
- Coding: novel compression techniques

Omnidirectional Mappings



Equirectangular projection

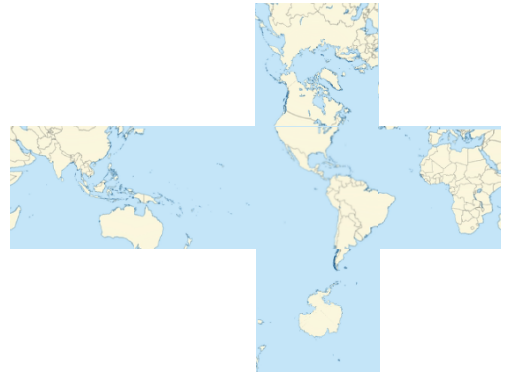
- Simple, popular
- Large horizontal oversampling near poles

Equal-area projection

- Decreased vertical sampling near poles

Cube Map

- Popular in graphics community



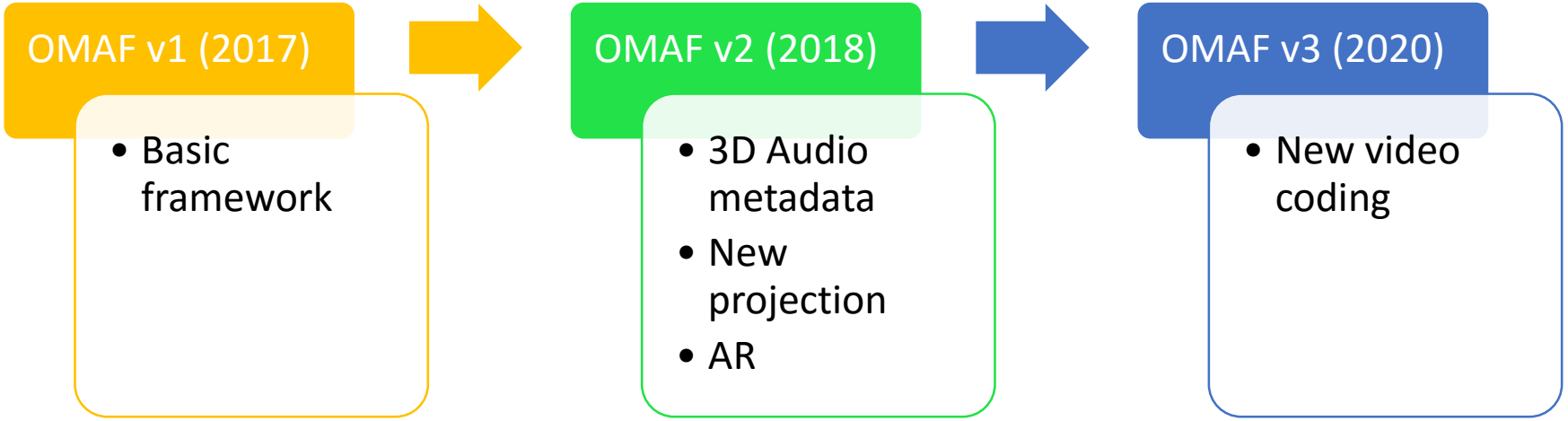
Courtesy Hari Lakshman and MPEG/JPEG

Omnidirectional Mapping Application Framework (OMAF)

- MPEG activity
- Phases
- Timeline

Regular Geometry	Irregular geometry

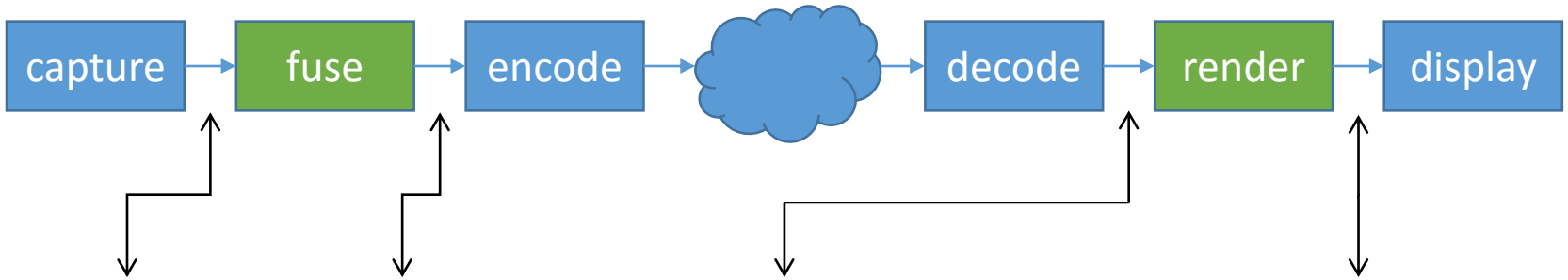
• ERP (complete/partial sphere)
• Cylinder
• Platonic Solids
  • Cube
  • Octahedron
  • Icosahedron
	• Squished sphere
	• Truncated pyramid



Issues for VR

- Mapping
- Audio
- Video coding may need adjustment (e.g., motion comp)
- Evaluation criteria
- Streaming on demand
  - Directional or spatial random access, ROI coding
  - Interactivity (much more rapid than trick modes)
  - Latency
- Broadcasting

Pipeline for Video / VR / AR

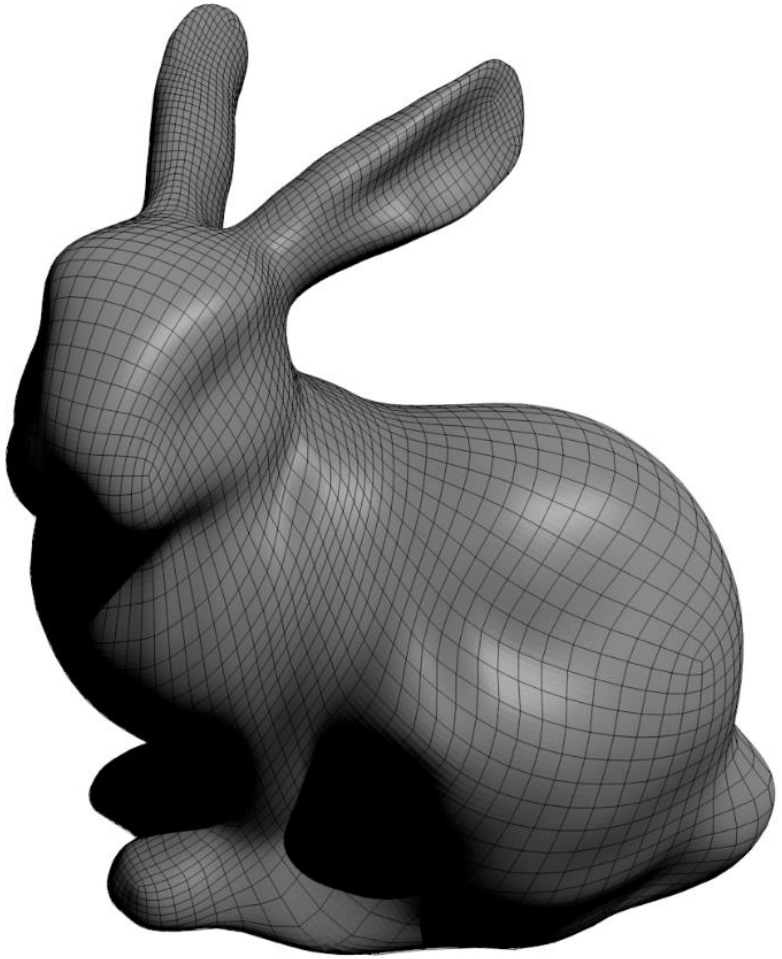


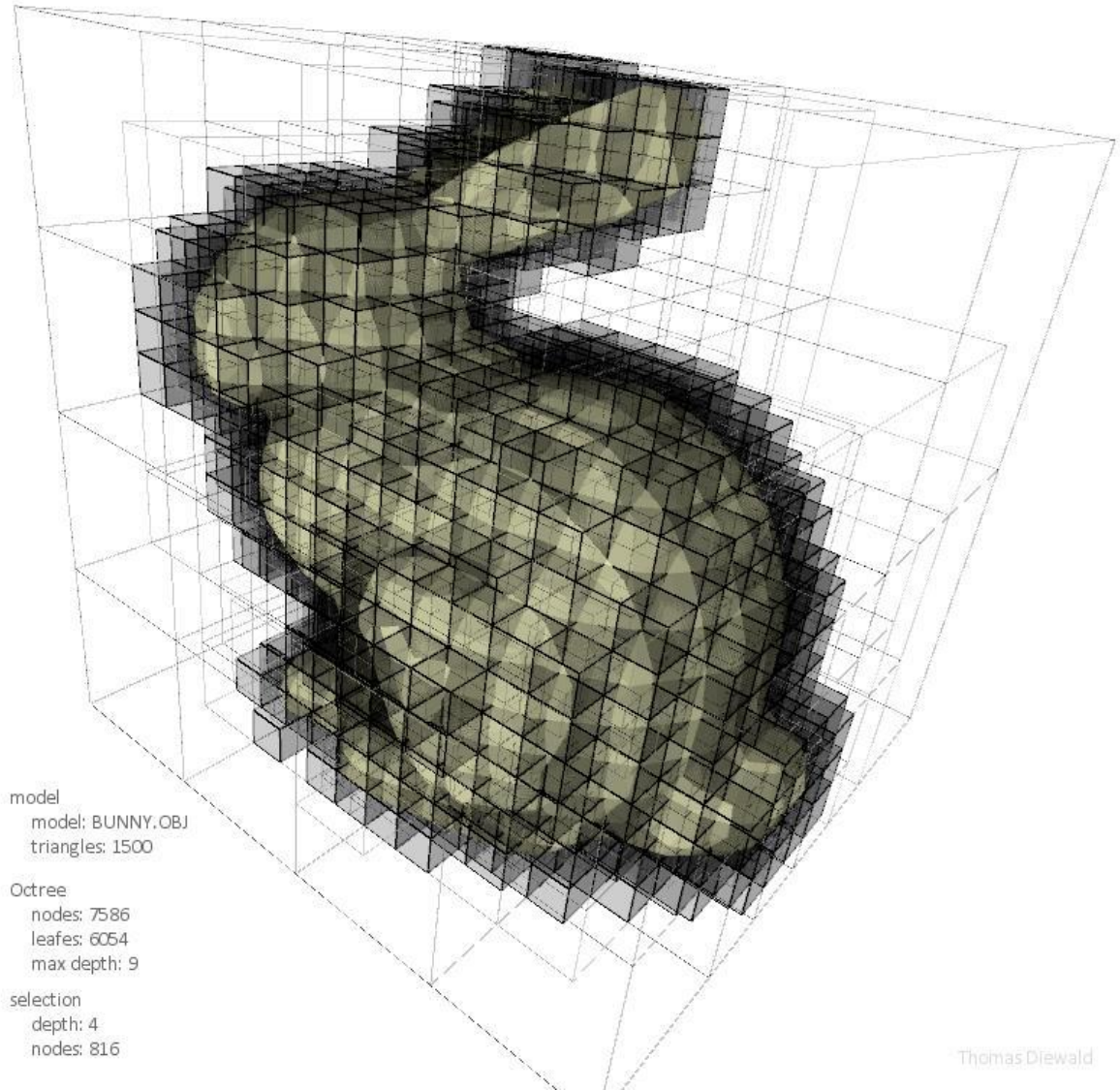
Video	Video	Digital video	Digital video	Video
VR	Array of video	Digital video (may be stereo, may have depth)	Digital video (may be stereo, may have depth)	• Browser • Phone, tablet app • Low-end VR device (e.g., Cardboard VR, Gear VR) • High-end VR device (e.g., Oculus Rift, HTC Vive, Playstation VR)
AR	Array of video + depth	Volumetric representation	Volumetric representation	• Browser • Phone, tablet app • Low-end VR device (e.g., Cardboard VR, Gear VR) • High-end VR device (e.g., Oculus Rift, HTC Vive, Playstation VR) • High-end AR device (e.g., HoloLens)

Volumetric Representations

- At present, not even the representation is agreed upon
- Some candidates:
  - Dynamic Meshes
    - Advantages: typical pipeline, easy to interpolate as it represents a surface
    - Disadvantages: does not handle noise well, or non-surfaces
  - Dynamic Point Clouds
    - Advantages: easily processed in parallel, handles noise and non-surfaces
    - Disadvantages: hard to interpolate
  - Hybrids
  - Light Fields
  - Holograms
  - Dense volumetric functions and implicit surfaces

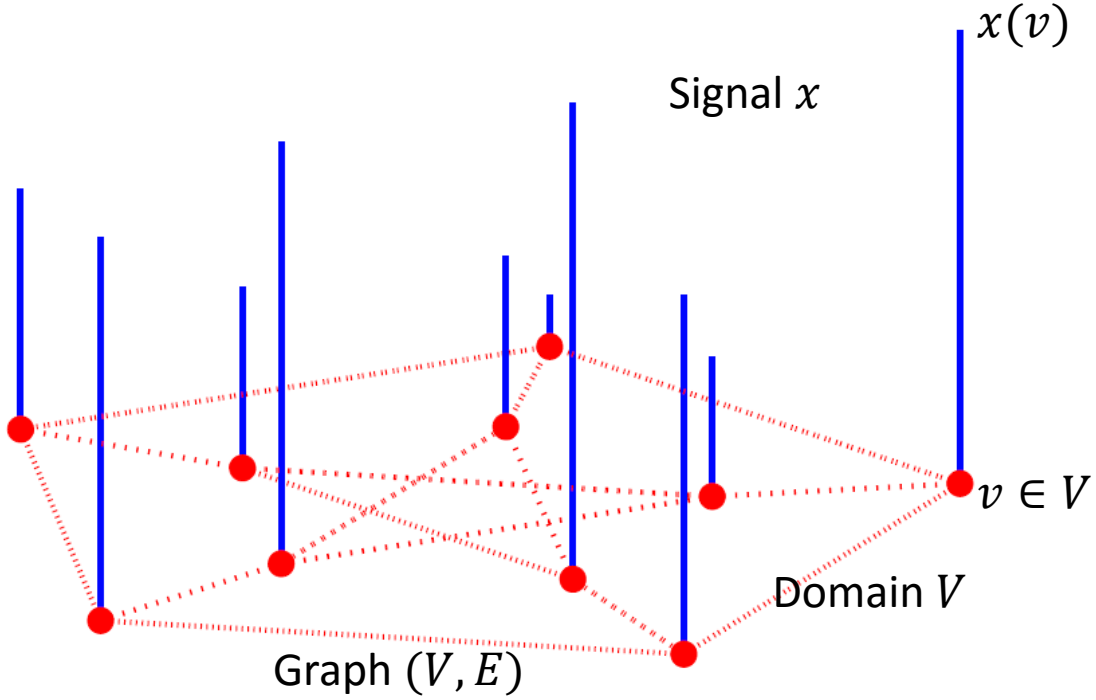
Meshes and Point Clouds: Irregular Domains





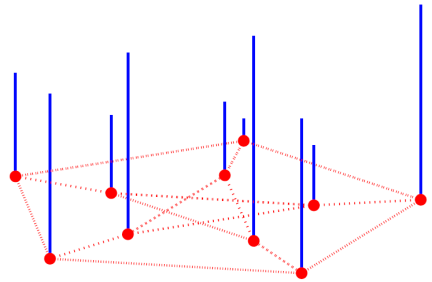
Graph Signal Processing (GSP): Framework for Signals on Irregular Domains

- Generalizes processing of real-valued signals defined on a regular domain (such as an image grid) to real-valued signals defined on a discrete graph.
- Applications to networks: social, sensor, communication, energy, transportation, neuronal; and *geometry*.
- Generalizes linear transform, impulse response, shift invariance, spectrum, frequency response, filtering, smoothing, interpolation, denoising, convolution, sampling, translation, etc.

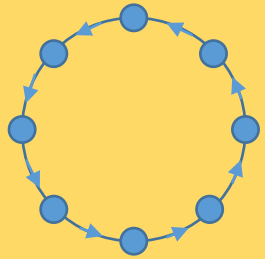
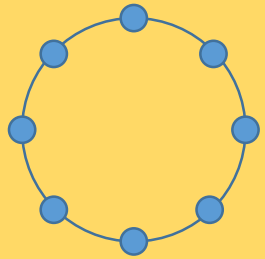
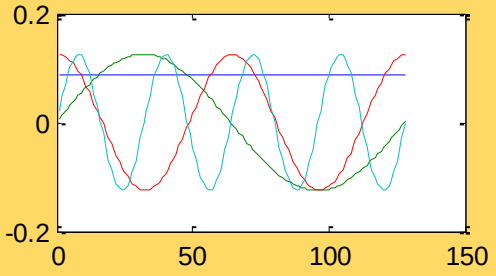


Reference: Shuman, Narang, Frossard, Ortega, and Vandergheynst, "Signal Processing on Graphs," IEEE Signal Processing Magazine, May 2013.

Adjacency Matrix as Shift Operator



- A graph (V, E) is represented by an (optionally weighted) adjacency matrix $A = [a_{mn}]$, with $a_{mn} > 0$ being the weight on edge $(m, n) \in E$.

Example 1	Left Shift operator on N points on a circle		$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$	Eigenvectors are $[e^{imn/N}]$ since $A[e^{imn/N}] = [e^{imn/N}]diag(e^{in/N})$
Example 2	Symmetric Random Walk operator on N points on a circle		$A = \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & \ddots & 0 \\ 0 & \ddots & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$	

A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs,"
IEEE Transactions on Signal Processing, vol. 61, no. 7, pp.1644-1656, 2013

Linear Shift Invariant operators and the GFT

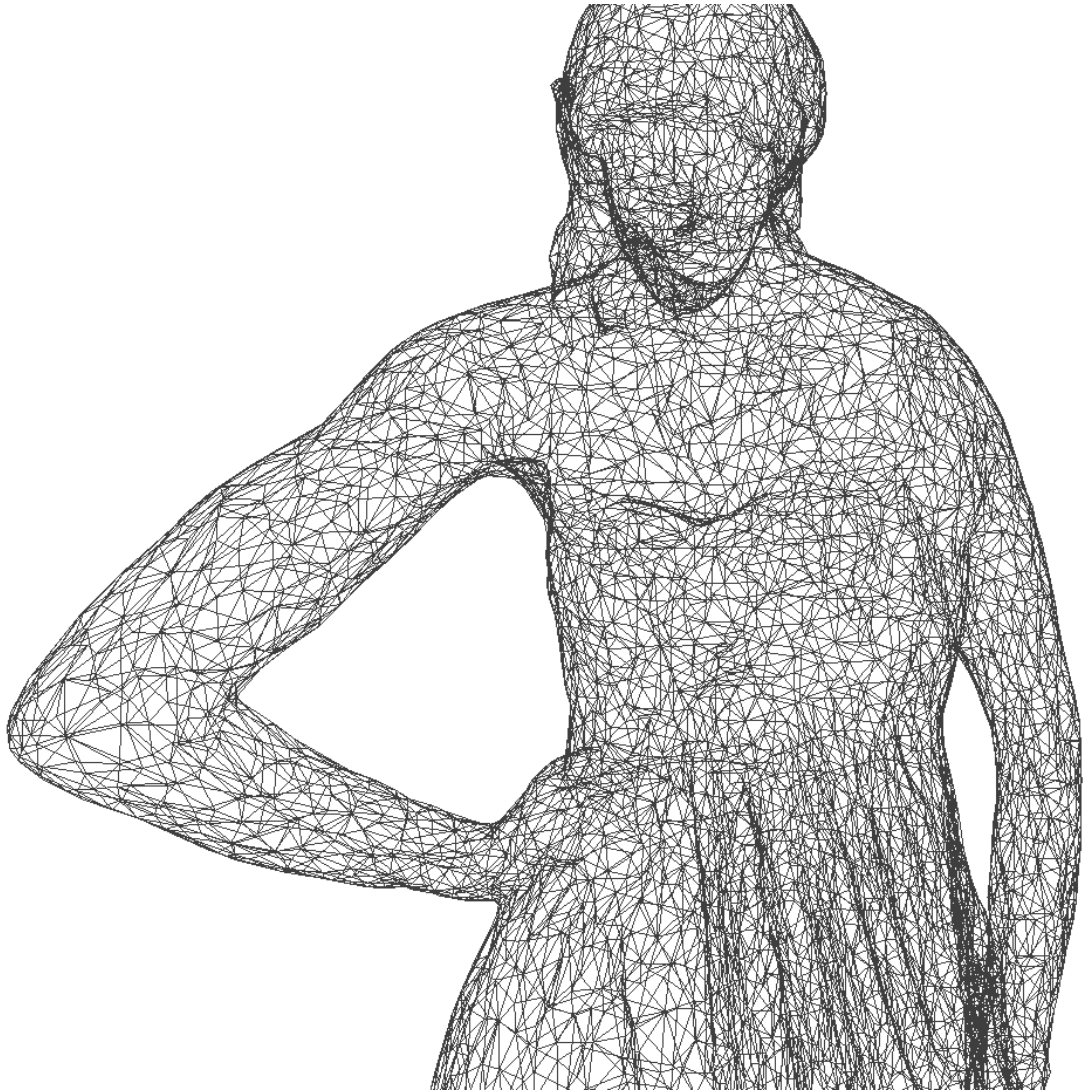
- {LSI operators} $\supseteq$ {analytic functions $f(A)$} $\supseteq$ {rational functions $q^{-1}(A)p(A)$} $\supseteq$ {polynomials $p(A) = p_0 I + p_1 A + \cdots + p_M A^M$}
  - $f(A)A = (\sum p_m A^m)A = \sum p_m A^{m+1} = A(\sum p_m A^m) = Af(A)$
- If shift operator A has eigenvectors Ψ and eigenvalues Λ (i.e., $A\Psi = \Psi\Lambda$) then any LSI operator $f(A)$ has eigenvectors Ψ and eigenvalues $f(\Lambda)$ (i.e., $f(A)\Psi = \Psi f(\Lambda)$)
  - $A = \Psi\Lambda\Psi^{-1}$
  - $f(A) = \sum p_m A^m = \sum p_m (\Psi\Lambda\Psi^{-1})^m = \Psi \sum p_m \Lambda^m \Psi^{-1} = \Psi f(\Lambda)\Psi^{-1}$

  Therefore $f(A)x = \Psi f(\Lambda)\Psi^{-1}x$.
- $\Psi^{-1}x$ is known as the Graph Fourier Transform of x.

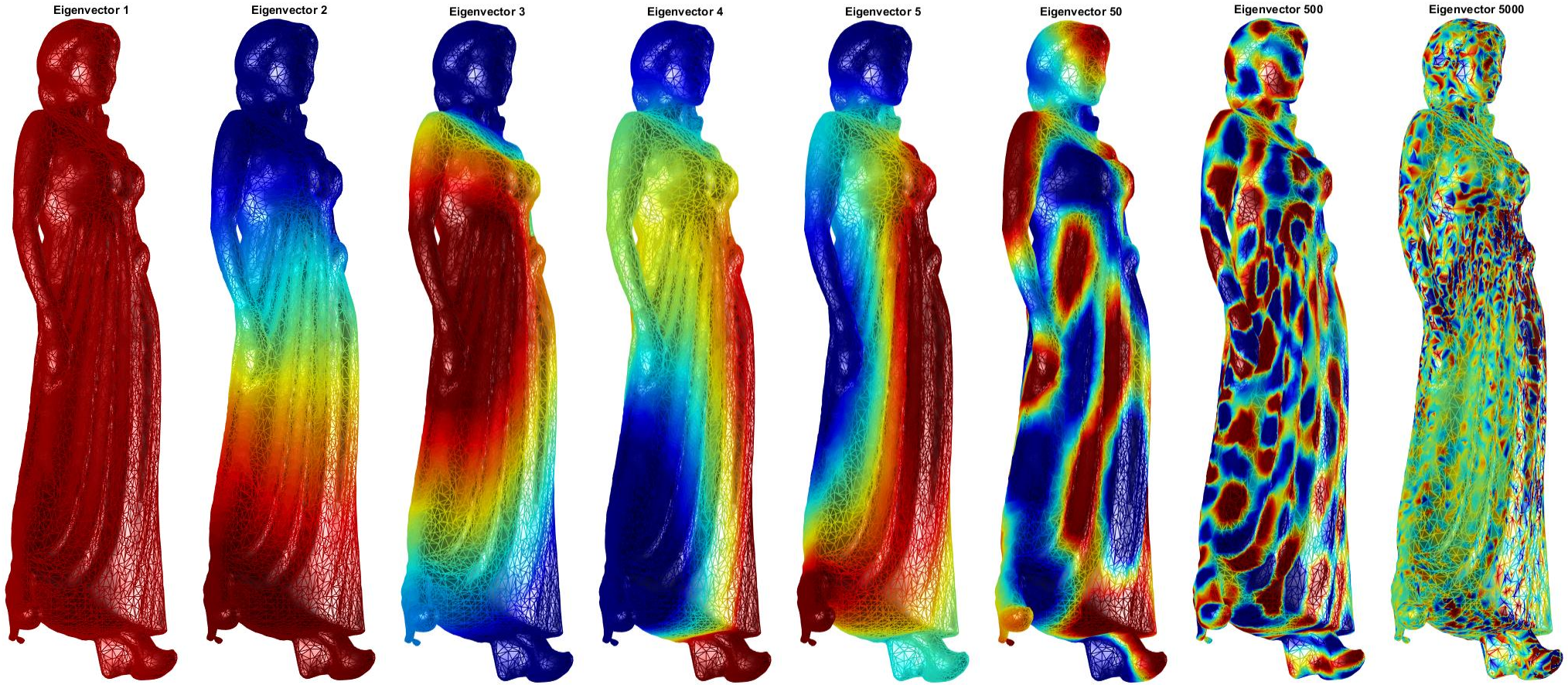
Graph Laplacian

- Often shift A is stochastic (rows or columns sum to 1)
  - $L = I - A$ is the Graph Laplacian (same eigenvectors as A, eigenvalues $I - \Lambda$)
- More generally
  - $L = D - A$ is the Graph Laplacian ($D = diag(d_i)$, $d_i = \sum_j A_{ij}$)
- Normalizations
  - $L = I - D^{-1}A$ is the "Random Walk" Graph Laplacian
  - $L = I - D^{-\frac{1}{2}}AD^{-1/2}$ is the "Normalized" Graph Laplacian
- Frequently $L = \Psi\Lambda\Psi^{-1}$ is the starting point instead of A, and real PSD
  - Eigenvalues $diag(\Lambda)$ are real non-negative (the graph spectrum)

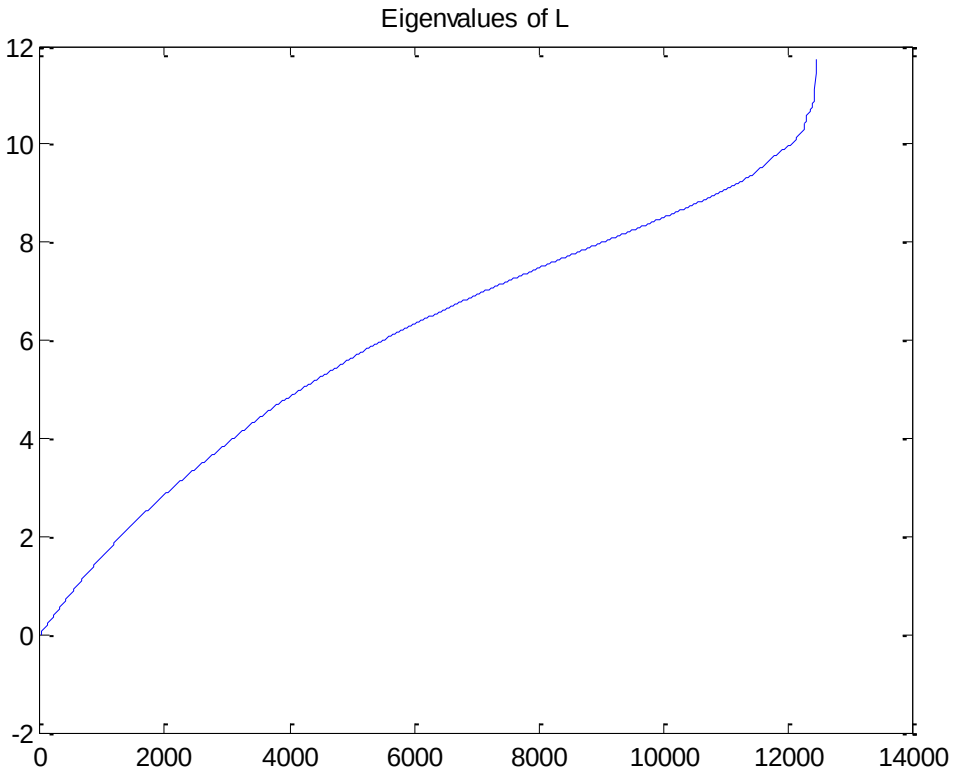
Mesh as a Graph

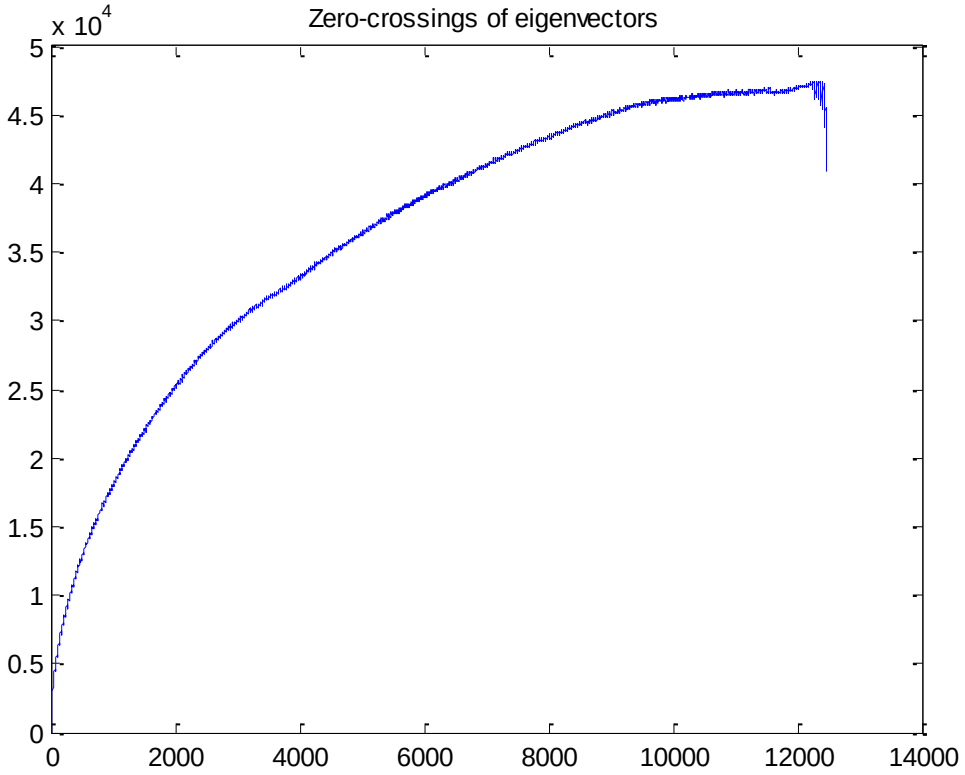


Eigenvectors of a Mesh

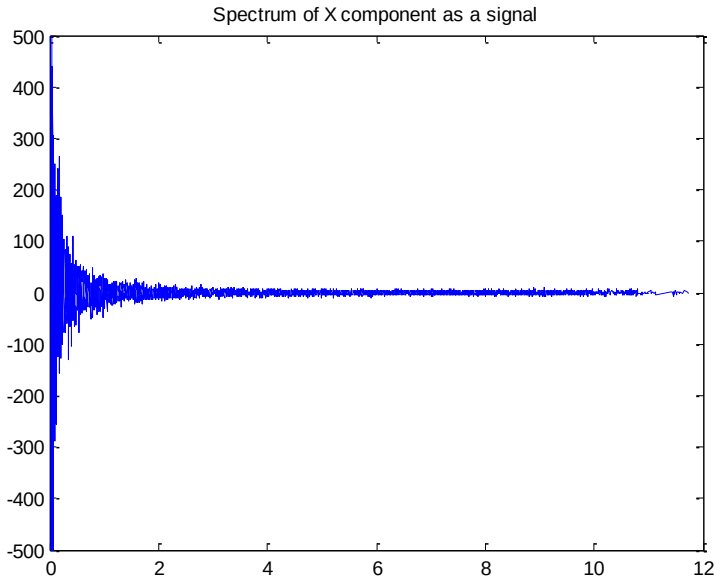


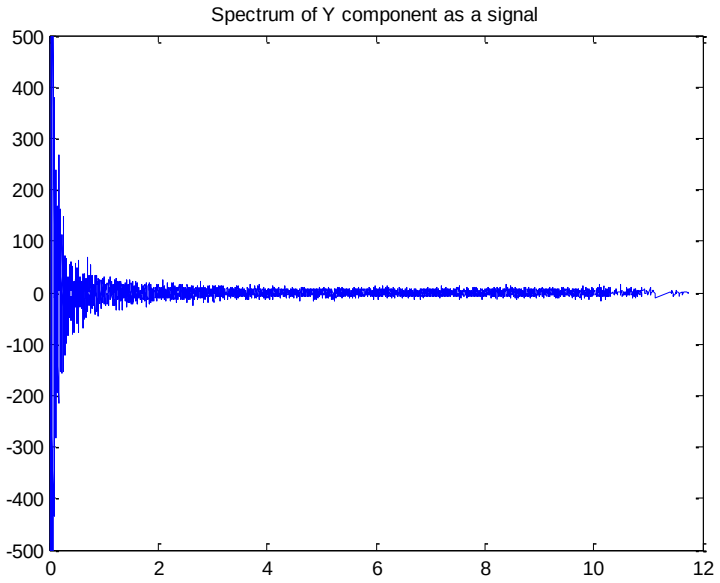
Eigenvalues (Spectrum) as “Frequencies”

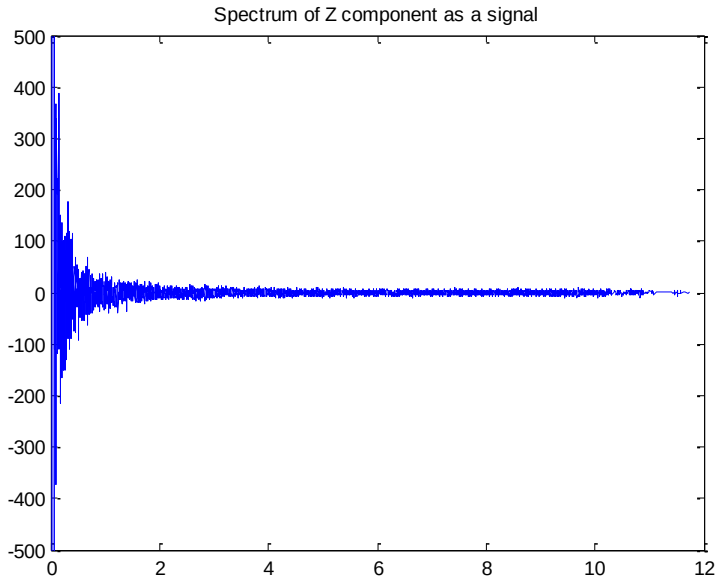




Spectrum of Geometry as a Signal







Energy Compaction

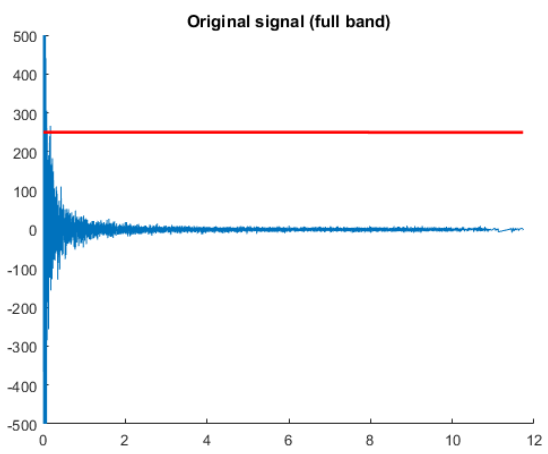
100% of coefficients

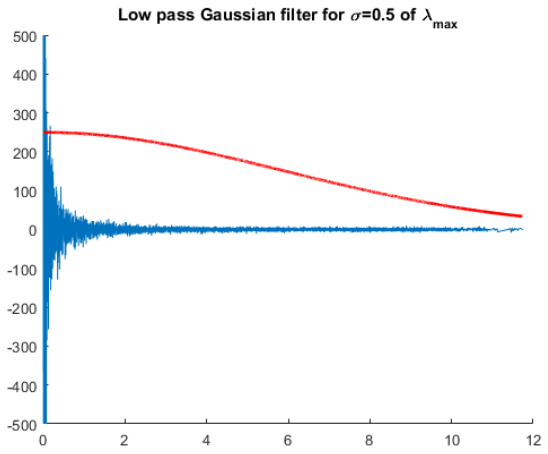
50% of coefficients

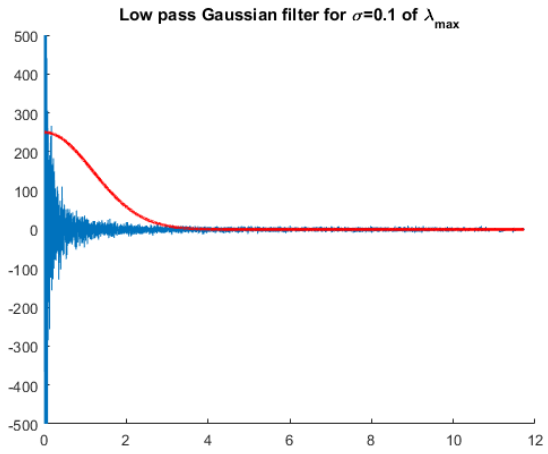
10% of coefficients

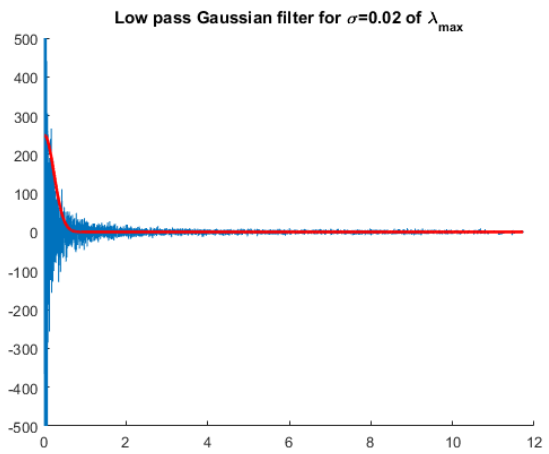
2% of coefficients

Spectral Domain Filtering / Denoising









Spectral Domain Filtering / Denoising

Original signal (full band)

Vertices smoothed by σ=0.5 of λ_{max}

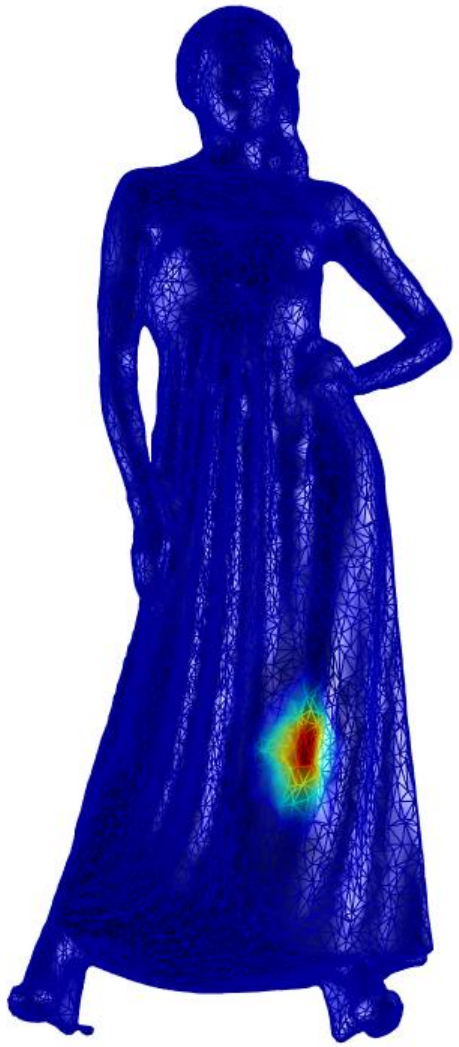
Vertices smoothed by σ=0.1 of λ_{max}

Vertices smoothed by σ=0.02 of λ_{max}

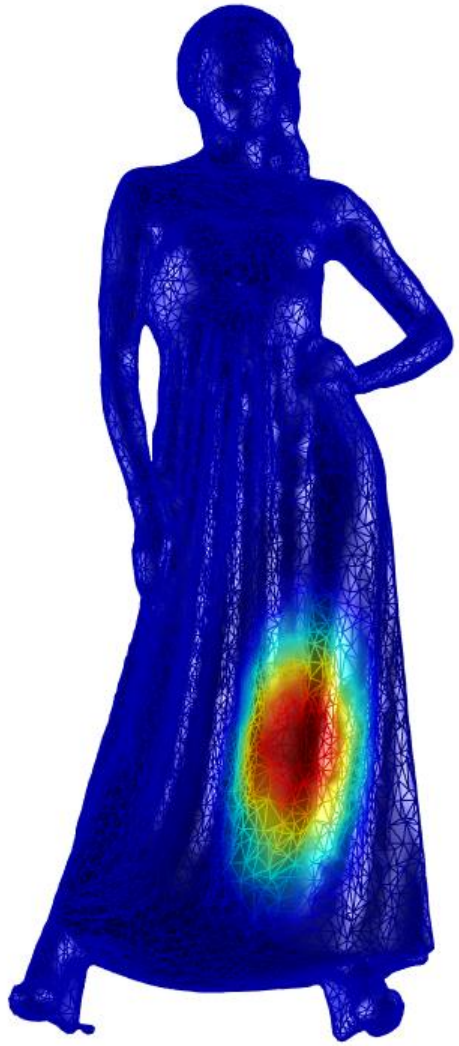
Impulse Response is Localized

Impulse vertex 11100 filterd with Gaussian σ=0.25

Impulse vertex 11100 filterd with Gaussian σ=0.05



Impulse vertex 11100 filterd with Gaussian σ=0.01



Polynomials and Vertex-Domain Filtering

- Spectral filters do not generally have compact vertex-domain support.
- Spectral filters $f(\lambda)$ correspond to LSI transforms $f(L) = \Psi f(\Lambda)\Psi^T$.
- Polynomial spectral filters have compact support
  - $y = p(L)x = \sum_{m=0}^{M} p_m L^m x$
  - If $x(i) = \delta(i, i_0)$ then $y(i) \neq 0$ only if $d(i, i_0) \leq M$, where d is geodesic distance
- Polynomial filters can be efficiently evaluated in vertex domain
  - Evaluate Lx by message passing, then L^2x, …, and lastly $L^M x$.
  - Finally sum $\sum_{m=0}^{M} p_m L^m x$. All can be done in parallel (e.g., on GPU).

Prediction and Interpolation

- Problem: Find $x_{unknown}$ such that

$$x = \begin{bmatrix} x_{known} \\ x_{unknown} \end{bmatrix}$$

minimizes the norm

$$\left\| L^{1/2} x \right\|^2 = x^T L x$$

of the high-pass signal $y = L^{1/2} x$
(using high-pass filter $f(\lambda) = \lambda^{1/2}$)

- Solution:

$$x_{unknown} = L_{22}^{-1} L_{21} x_{known}$$

- Remark:

Same as $E[X_{unknown} | X_{known} = x_{known}]$
if $X \sim N(0, L^{-1})$

Prediction and Interpolation

- Problem: Find $x_{unknown}$ such that

$$x = \begin{bmatrix} x_{known} \\ x_{unknown} \end{bmatrix}$$

minimizes the norm

$$\left\| L^{1/2} x \right\|^2 = x^T L x$$

of the high-pass signal $y = L^{1/2} x$
(using high-pass filter $f(\lambda) = \lambda^{1/2}$)

- Solution:

$$x_{unknown} = L_{22}^{-1} L_{21} x_{known}$$

- Remark:

Same as $E[X_{unknown} | X_{known} = x_{known}]$
if $X \sim N(0, L^{-1})$

Prediction and Interpolation

- Problem: Find $x_{unknown}$ such that

$$x = \begin{bmatrix} x_{known} \\ x_{unknown} \end{bmatrix}$$

minimizes the norm

$$\left\| L^{1/2} x \right\|^2 = x^T L x$$

of the high-pass signal $y = L^{1/2} x$
(using high-pass filter $f(\lambda) = \lambda^{1/2}$)

- Solution:

$$x_{unknown} = L_{22}^{-1} L_{21} x_{known}$$

- Remark:

Same as $E[X_{unknown} | X_{known} = x_{known}]$
if $X \sim N(0, L^{-1})$

Smoothing

- Smoothing is similar, with a loose constraint:

  Find $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ minimizing $\|x_1 - x_{known}\|^2 + \alpha x^T L x$

- Special case:

  Find x minimizing $\|x - x_{known}\|^2 + \alpha x^T L x$

- Solution:

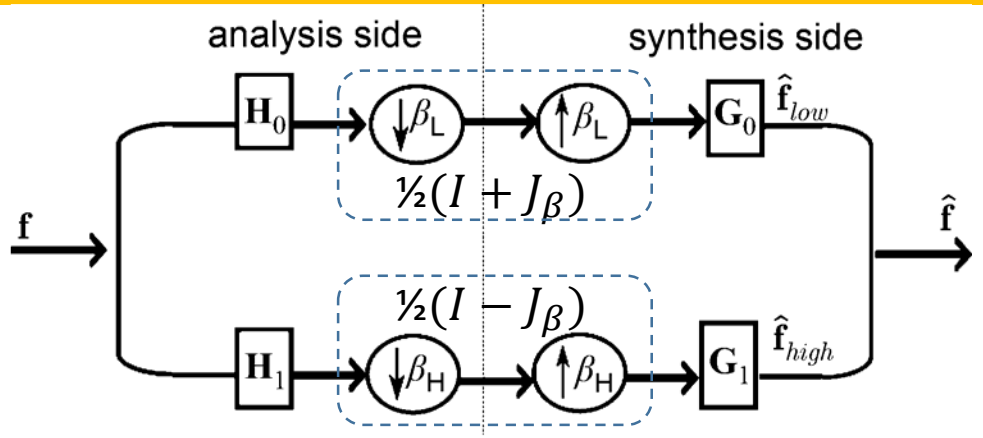
  $$x^* = \left(\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} + \alpha L \right)^{-1} \begin{bmatrix} x_{known} \\ 0 \end{bmatrix}$$

  (equivalent to low-pass Tikhonov filtering with $f(\lambda) = \frac{1}{1+\alpha\lambda}$ in special case)

Perfect Reconstruction Critically Sampled 2-Channel Filter Banks with Compact Support

Narang and Ortega, "Compact Support Biorthogonal Wavelet Filterbanks for Arbitrary Undirected Graphs," TSP 2012

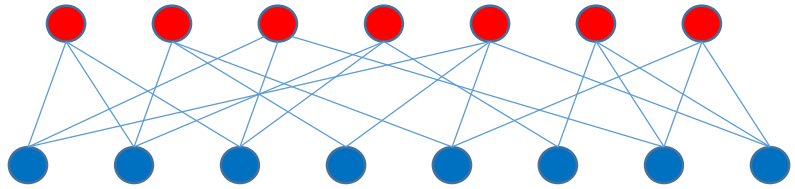
Graph Wavelet with Compact Support



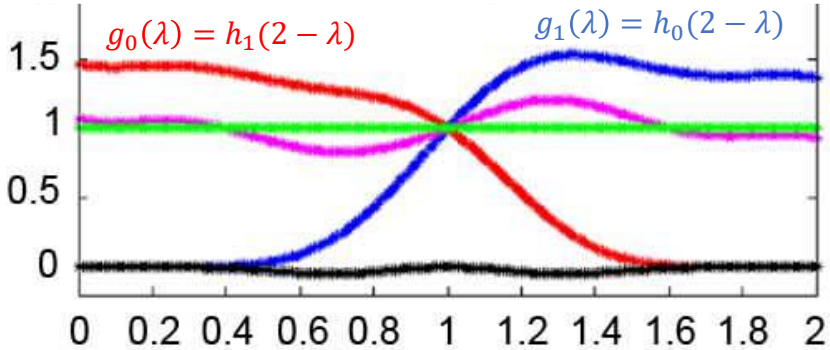
$J_\beta = diag(\beta_i), \beta_i = +1/-1$ if $i \in L/H$

$$I = \frac{1}{2}G_0(I + J_\beta)H_0 + \frac{1}{2}G_1(I - J_\beta)H_1$$
$$= \frac{1}{2}(G_0H_0 + G_1H_1) + \frac{1}{2}(G_0J_\beta H_0 - G_1J_\beta H_1)$$

Design for Bipartite Graph

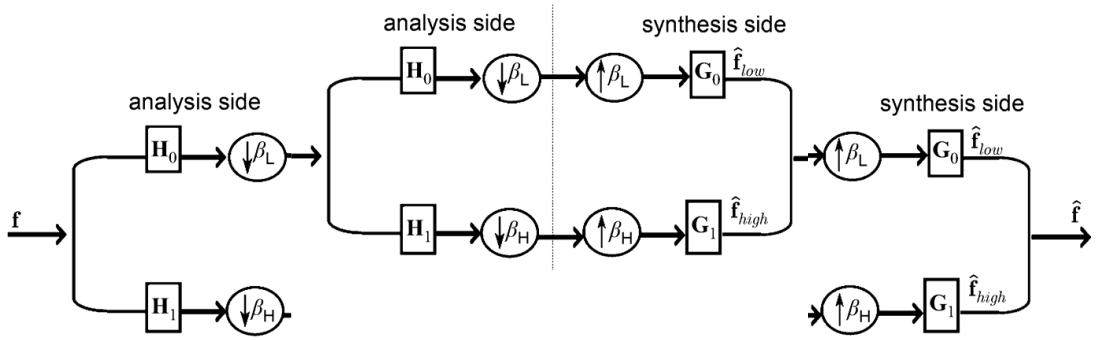


- $\psi_{2-\lambda} = J_\beta\psi_\lambda$ (spectral folding)
- Perfect Reconstruction if
  - $g_0(\lambda)h_0(\lambda) + g_1(\lambda)h_1(\lambda) = 2$
  - $g_0(\lambda)h_0(2-\lambda) - g_1(\lambda)h_1(2-\lambda) = 0$



Multi-Resolution Processing

Iterate Low-Pass Filtering

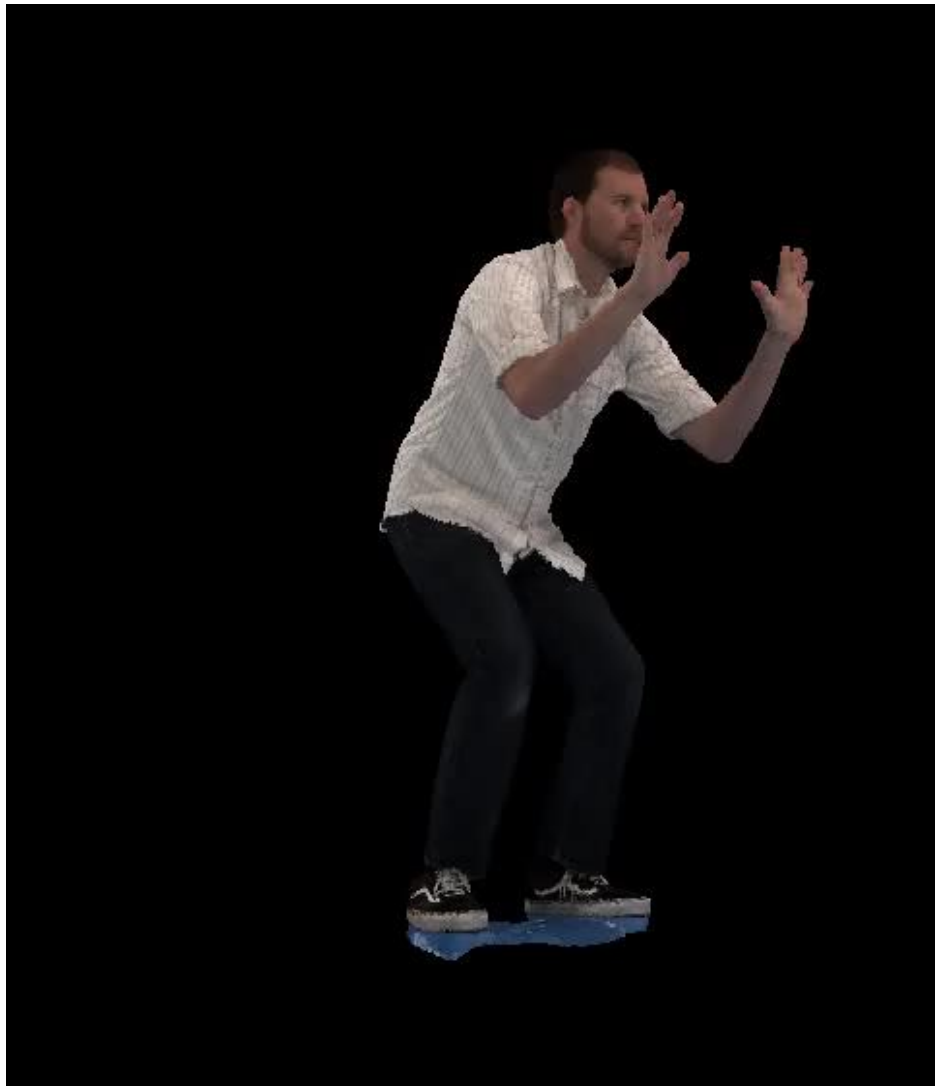


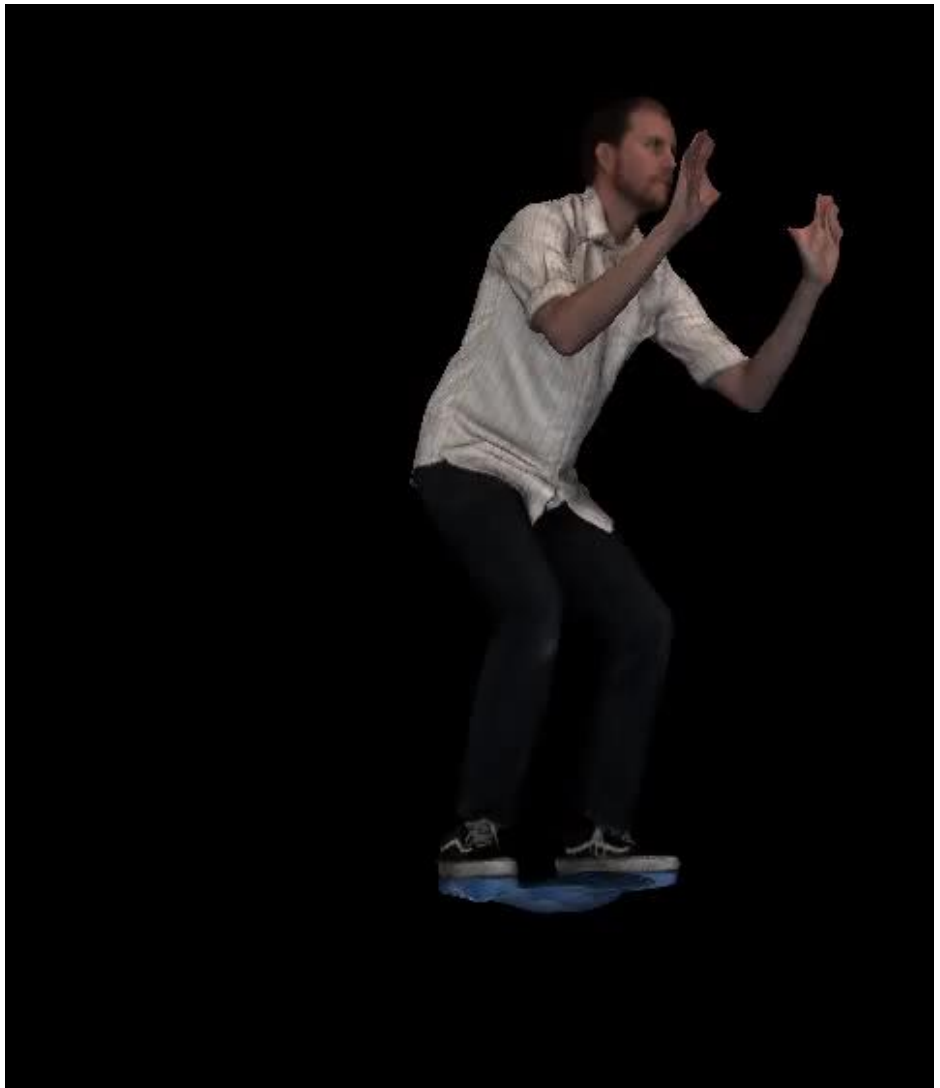
Procedure

- At level L, use some rule to color graph with only two colors (🔴 = low pass, 🔵 = high pass)
  - Easy if graph is bi-partite
  - Otherwise separate graph into union of bi-partite graphs
- Filter to get coeffs at L/H vertices
- At level $L - 1$, use some rule to reconnect low pass vertices
- Repeat

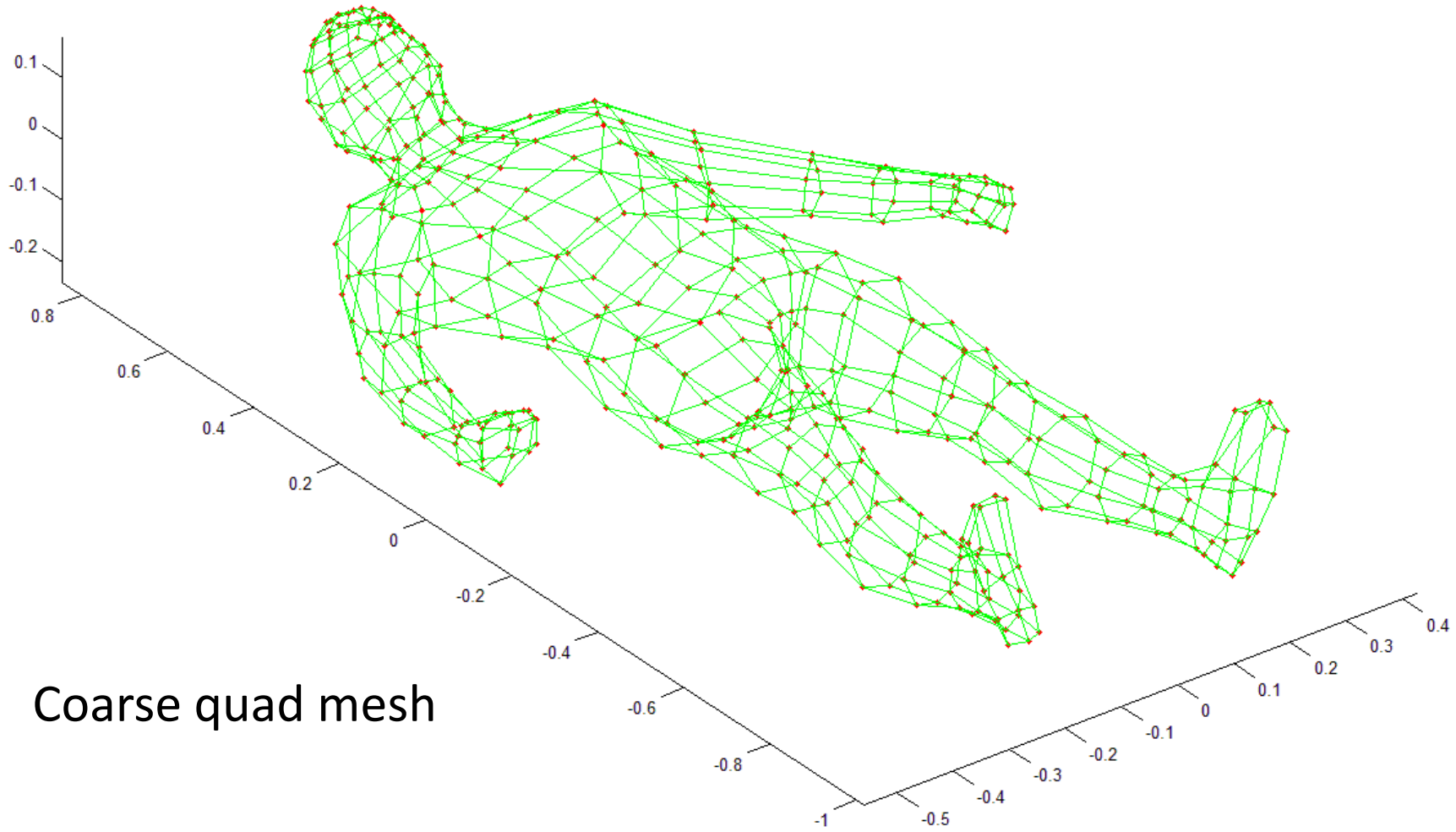
Application to Mesh Geometry and Color Compression

Nguyen, Chou, and Chen, “Compression of Human Body Sequences Using Graph Wavelet Filter Banks,” ICASSP 2014
Anis, Chou, and Ortega, “Compression of Dynamic 3D Point Clouds using Subdivisional Meshes and Graph Wavelet Transforms,” ICASSP 2016



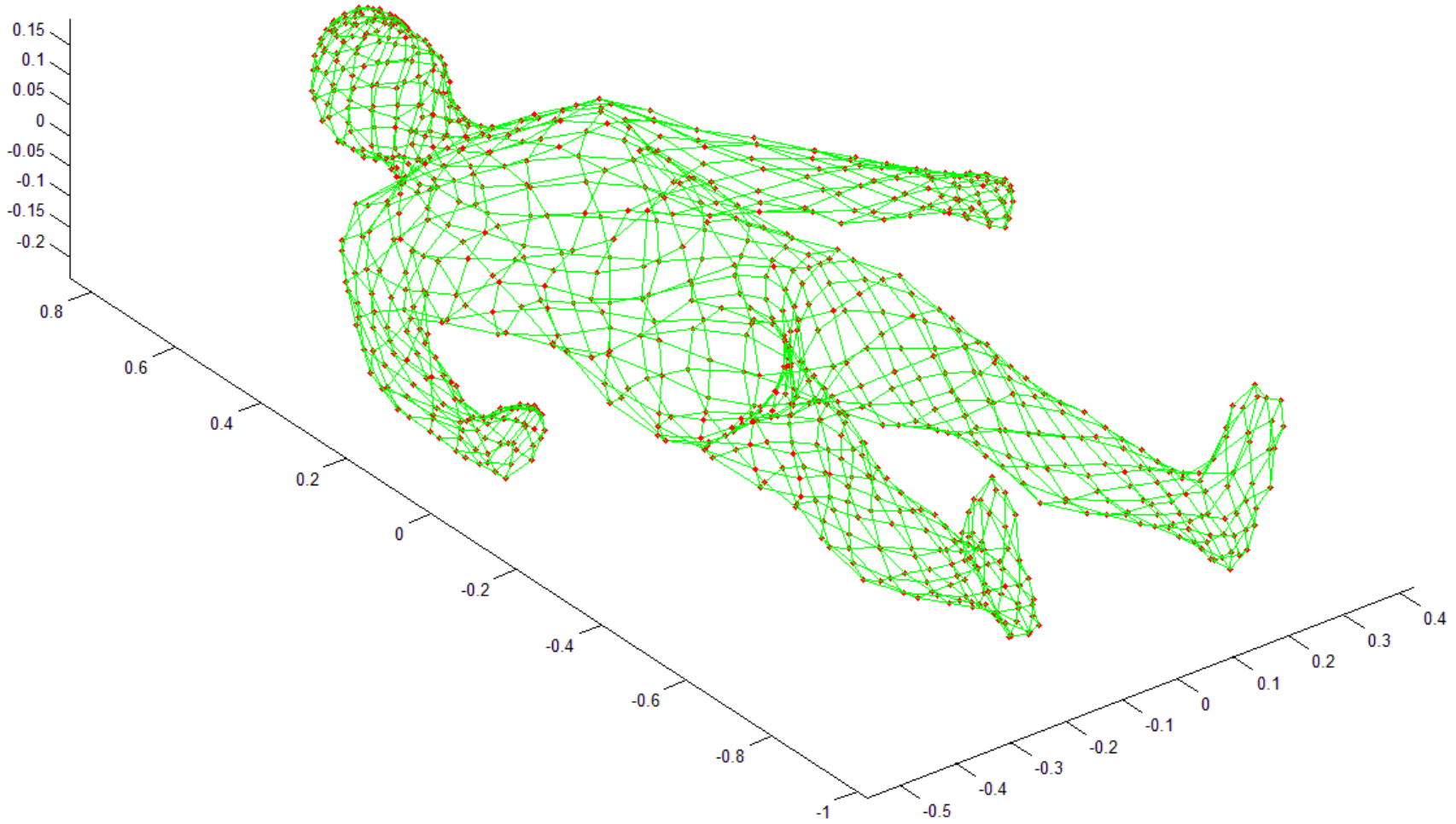


Quad Mesh Subdivision

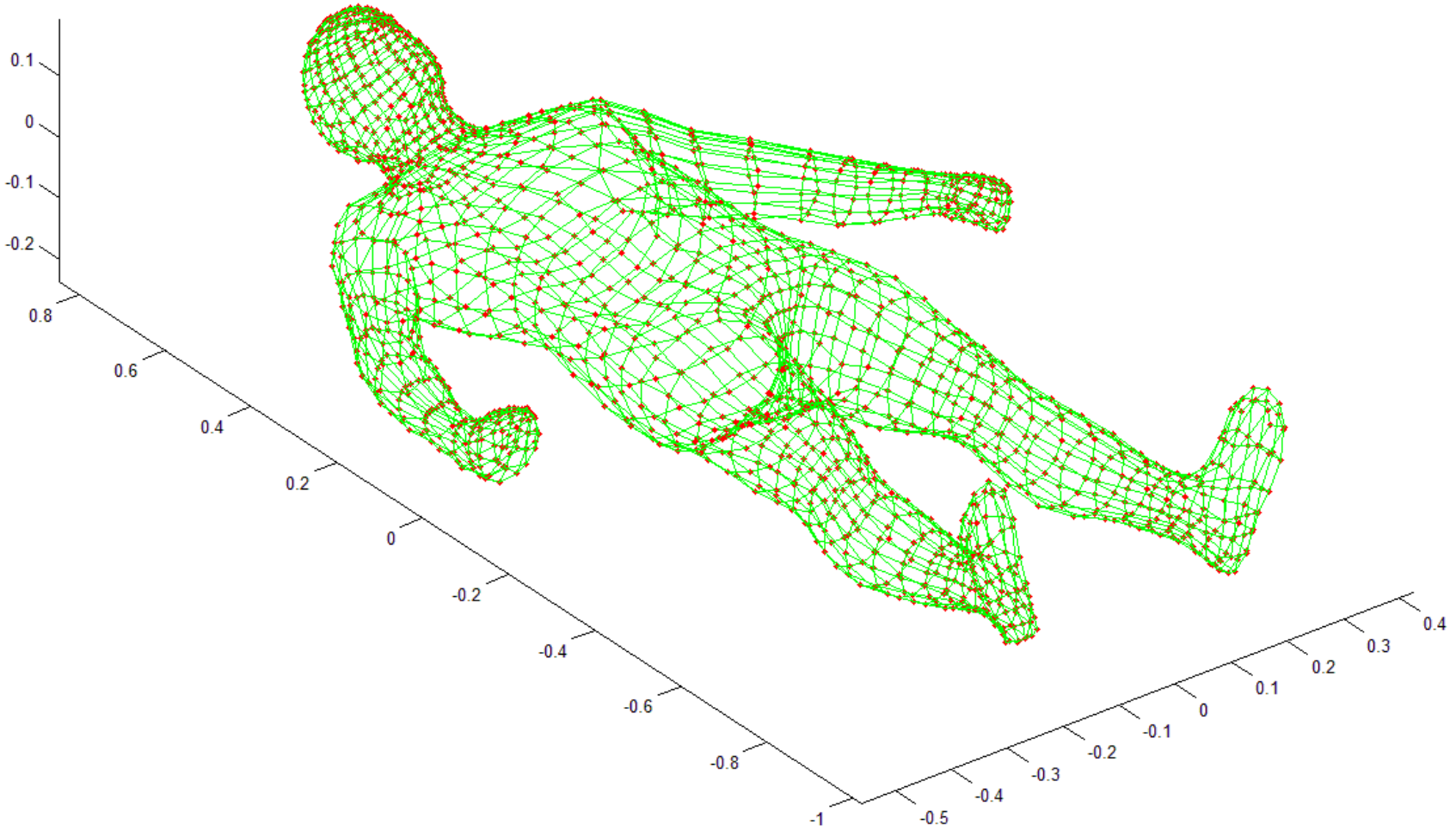


Coarse quad mesh

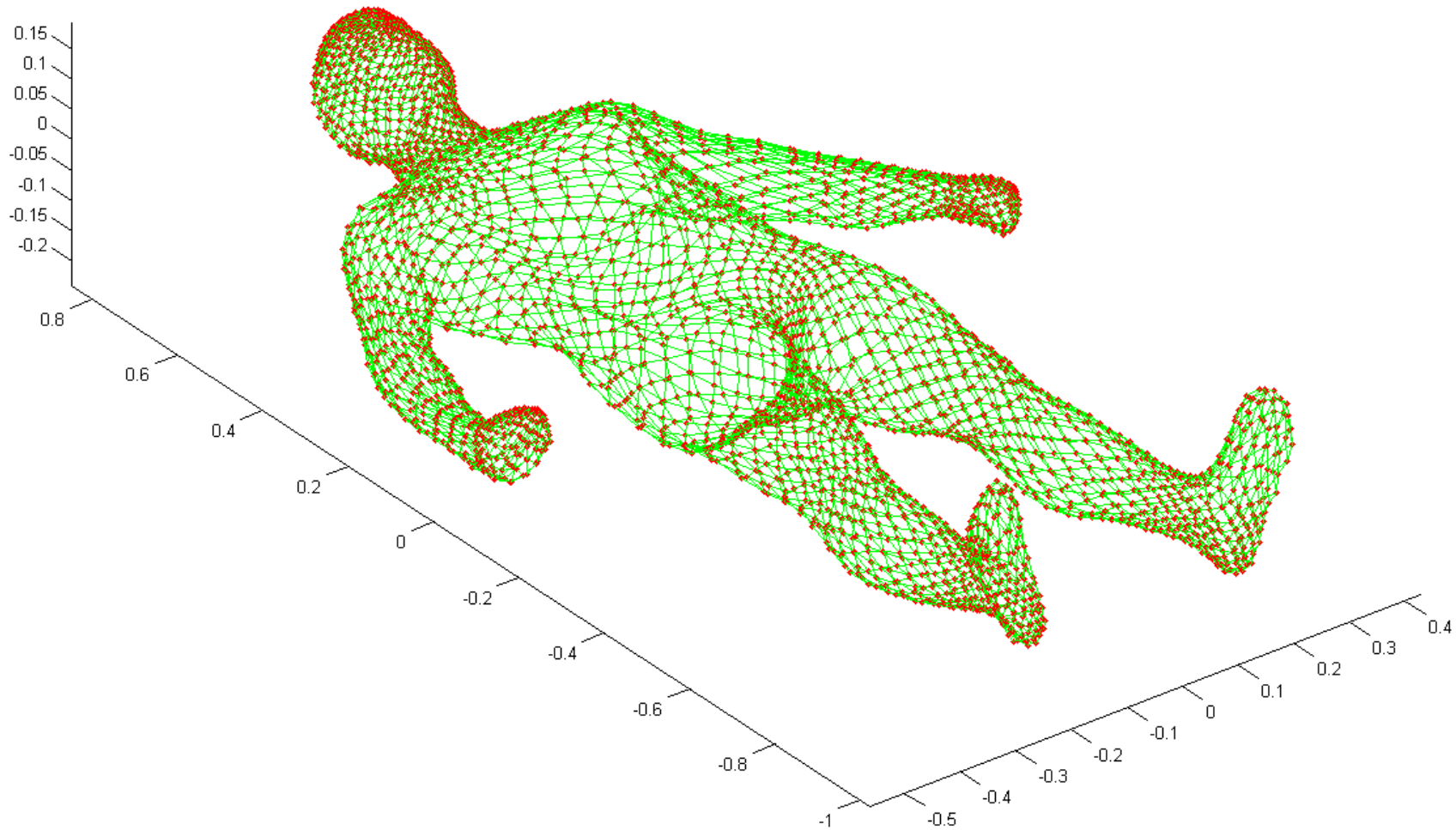
Quad Mesh Subdivision



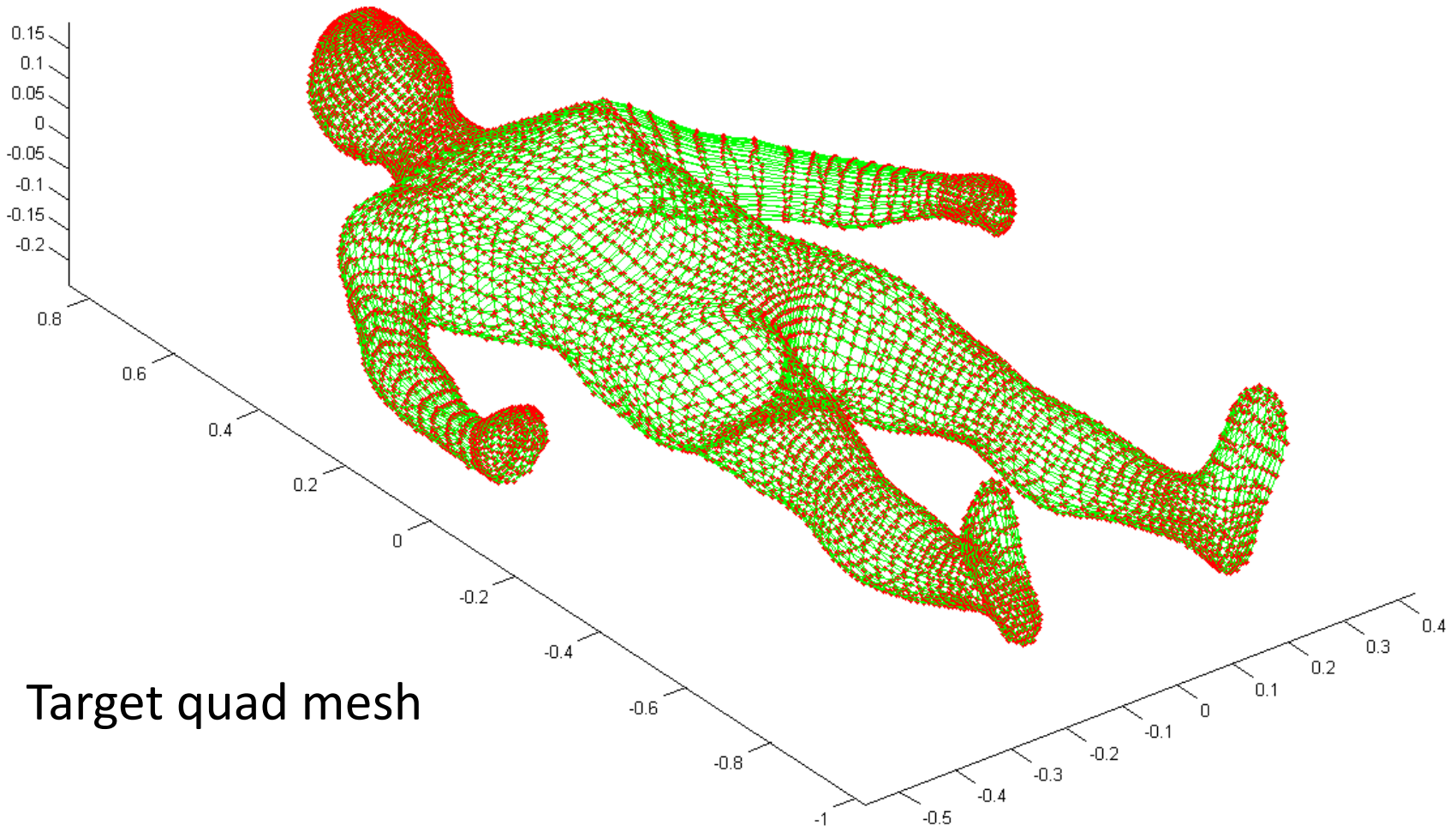
Quad Mesh Subdivision



Quad Mesh Subdivision

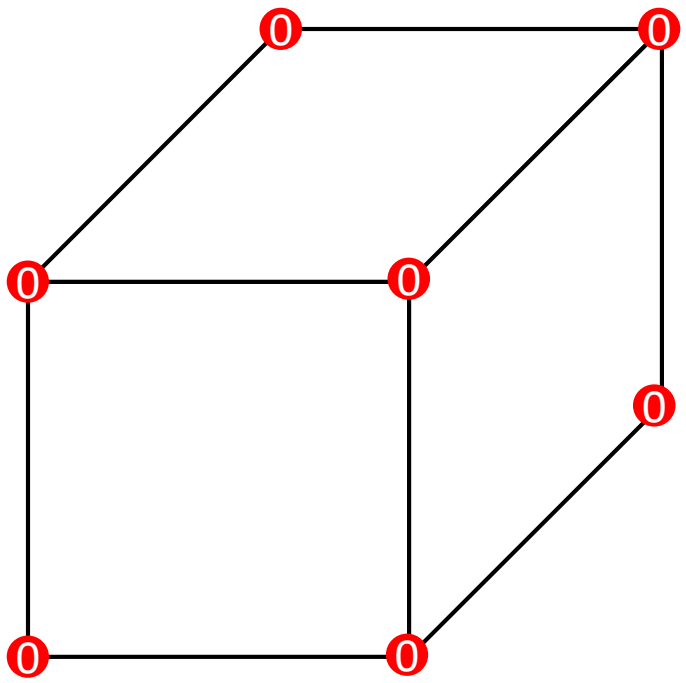


Quad Mesh Subdivision



Target quad mesh

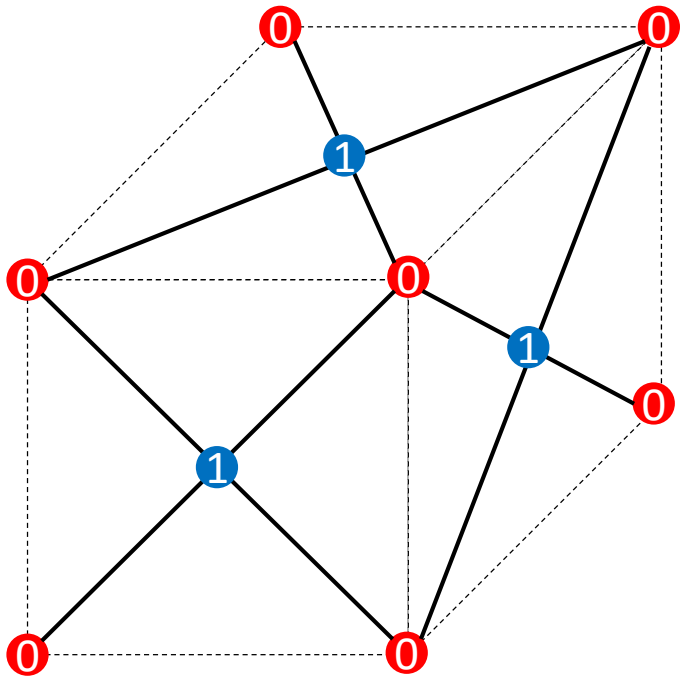
Quad Mesh Subdivision



Lowpass subband

Highpass subband

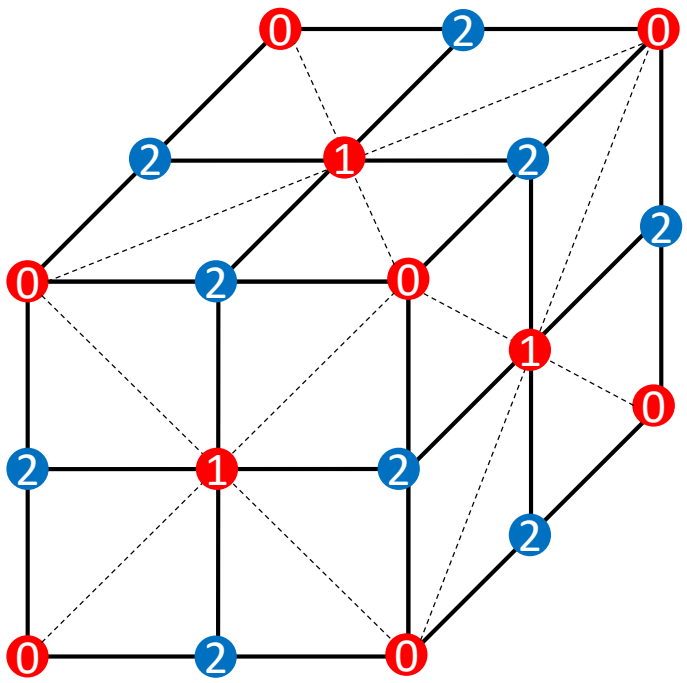
Quad Mesh Subdivision



Lowpass subband

Highpass subband

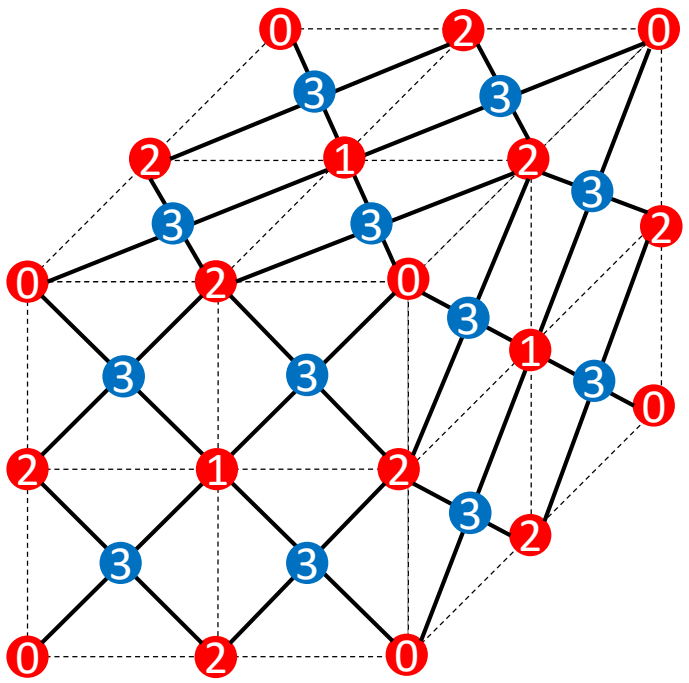
Quad Mesh Subdivision



Lowpass subband

Highpass subband

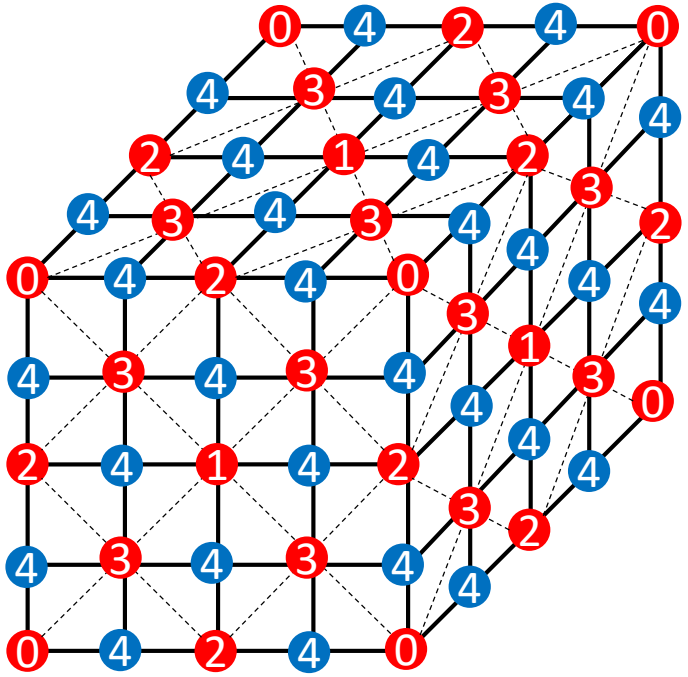
Quad Mesh Subdivision



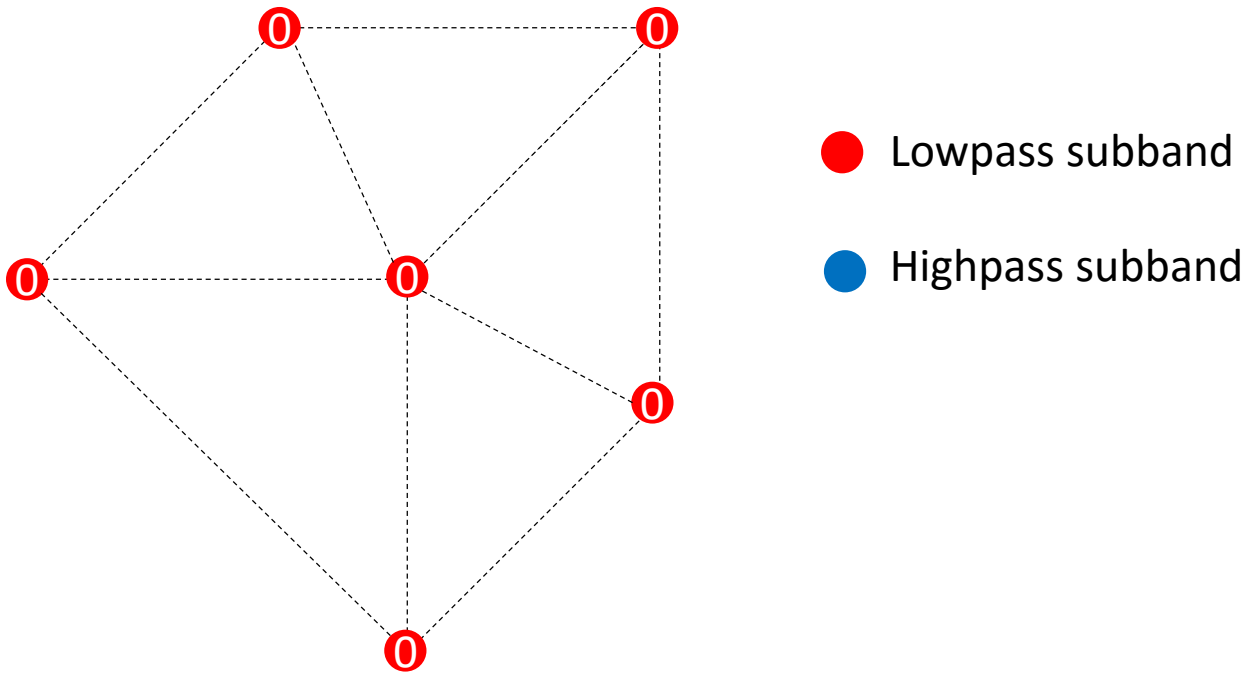
Lowpass subband

Highpass subband

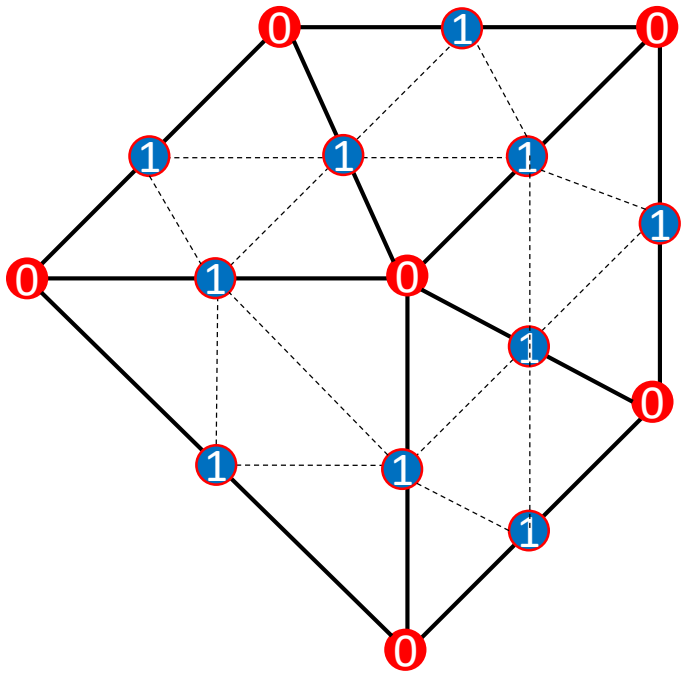
Quad Mesh Subdivision



Tri Mesh Subdivision



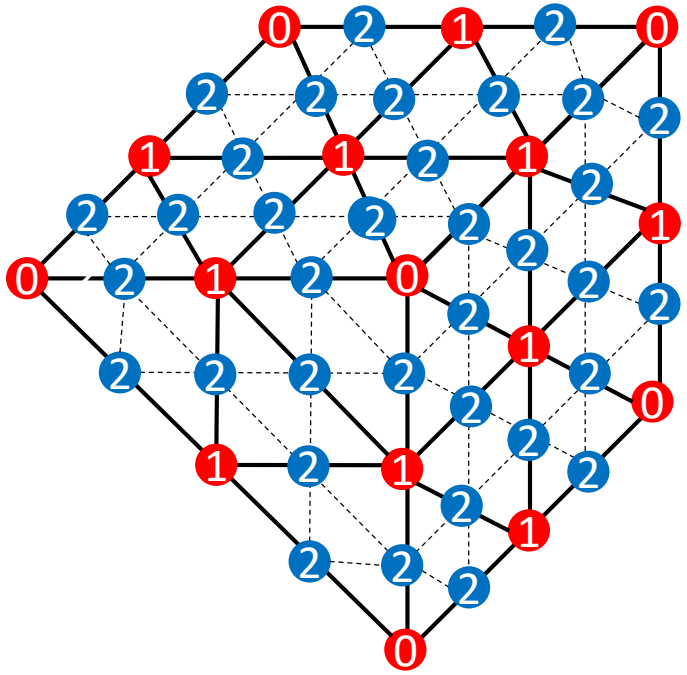
Tri Mesh Subdivision



Lowpass subband

Highpass subband

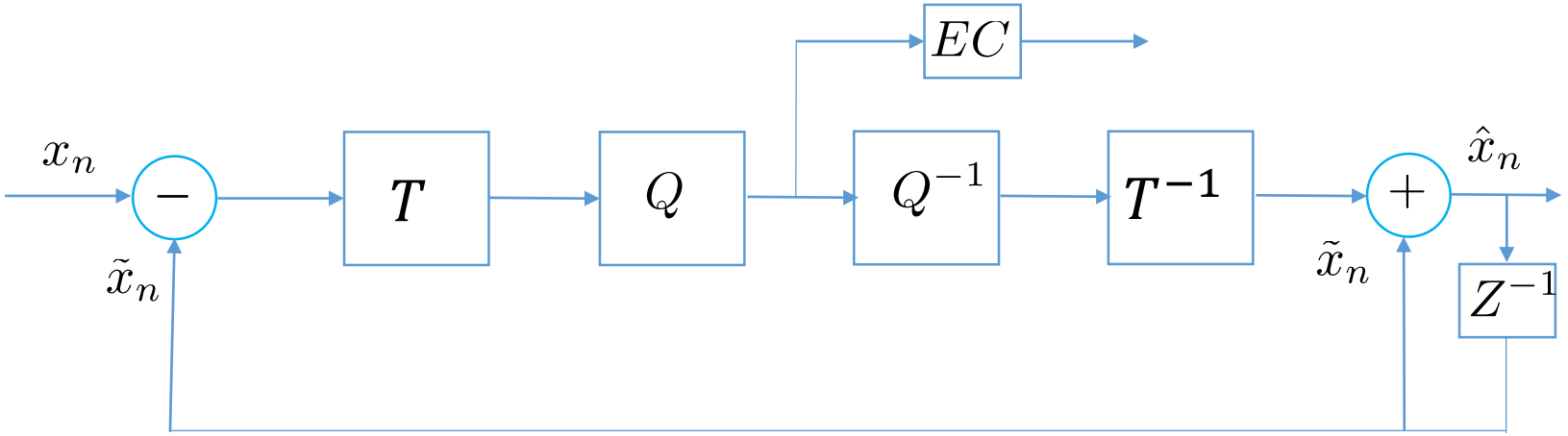
Tri Mesh Subdivision



Lowpass subband

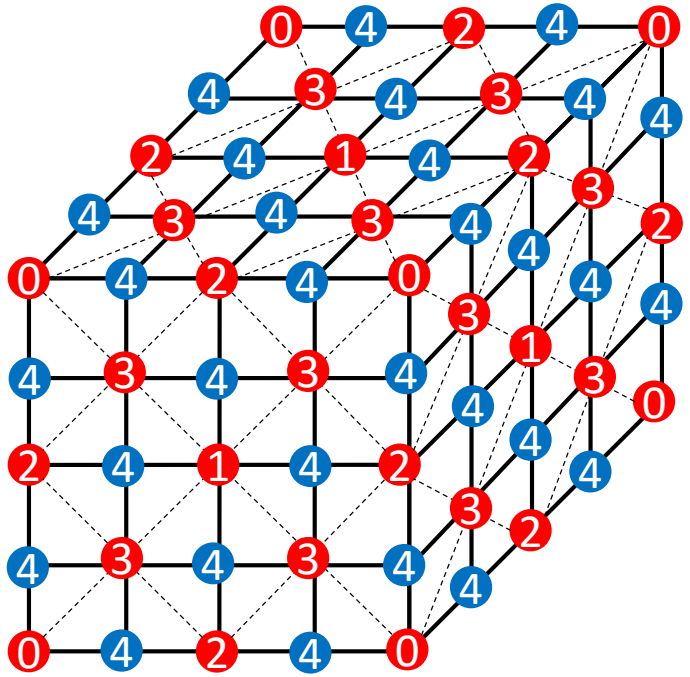
Highpass subband

Hybrid Predictive-Transform Coding



- Graph transform, uniform scalar quantization, entropy coding
- Temporal prediction removes the temporal redundancy, creates motion vectors
- Graph transform removes spatial correlation among the motion vectors

Context-Adaptive Arithmetic Coding

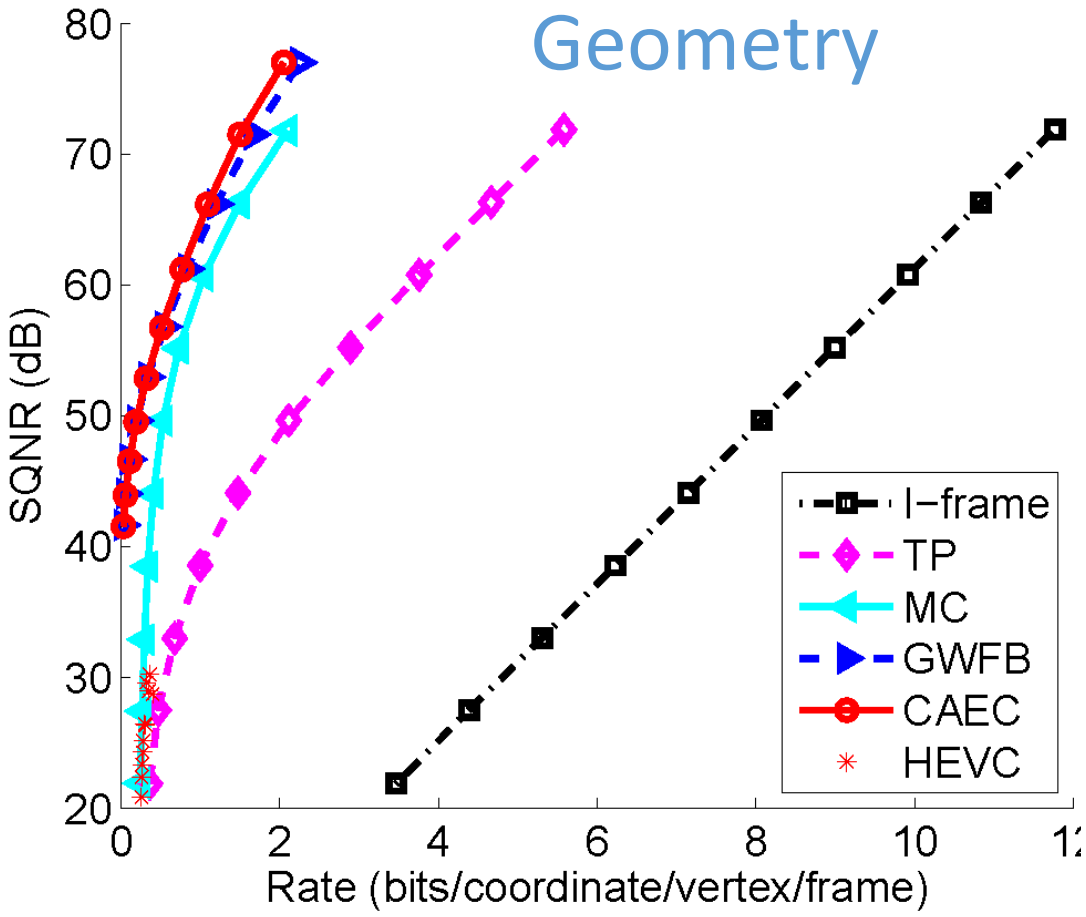


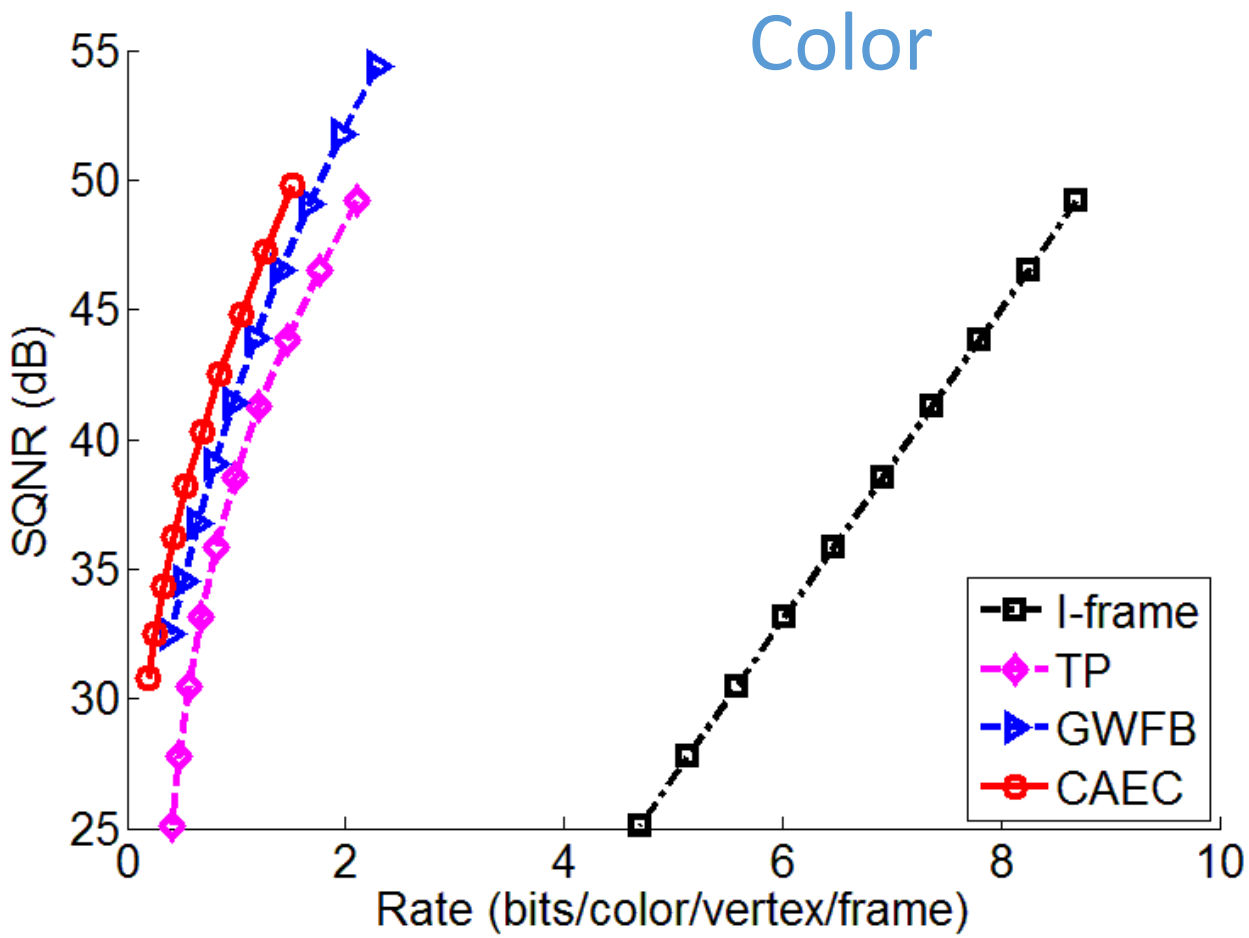
- Let $n = 0,1,2,3,4$ be the subband index
- If coefficient at node i is in subband $\mathcal{V}_n$, model it as Laplacian with parameter

$$\lambda_i = \beta_{n,0} + \sum_{j=1}^{n-1} \beta_{n,j} \left(\frac{\sum_{k \in \mathcal{N}_i \cap \mathcal{V}_{n-j}} |x_k|}{|\mathcal{N}_i \cap \mathcal{V}_{n-j}|} \right)^{-1}$$

  depending on average of magnitudes $|x_k|$ of neighbors $k \in \mathcal{N}_i \cap \mathcal{V}_{n-j}$ in previous subbands
- Fit constants $\{\beta_{n,j}\}$ to data

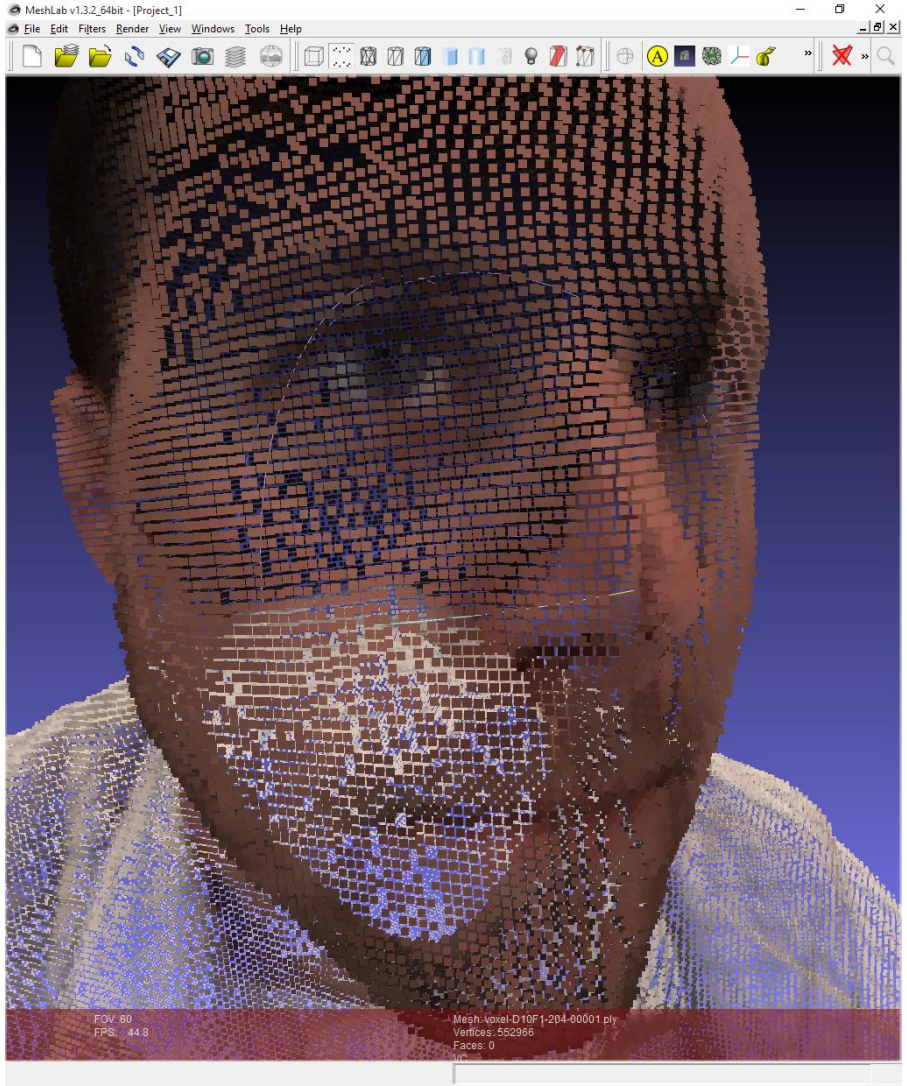
Rate-distortion Performance





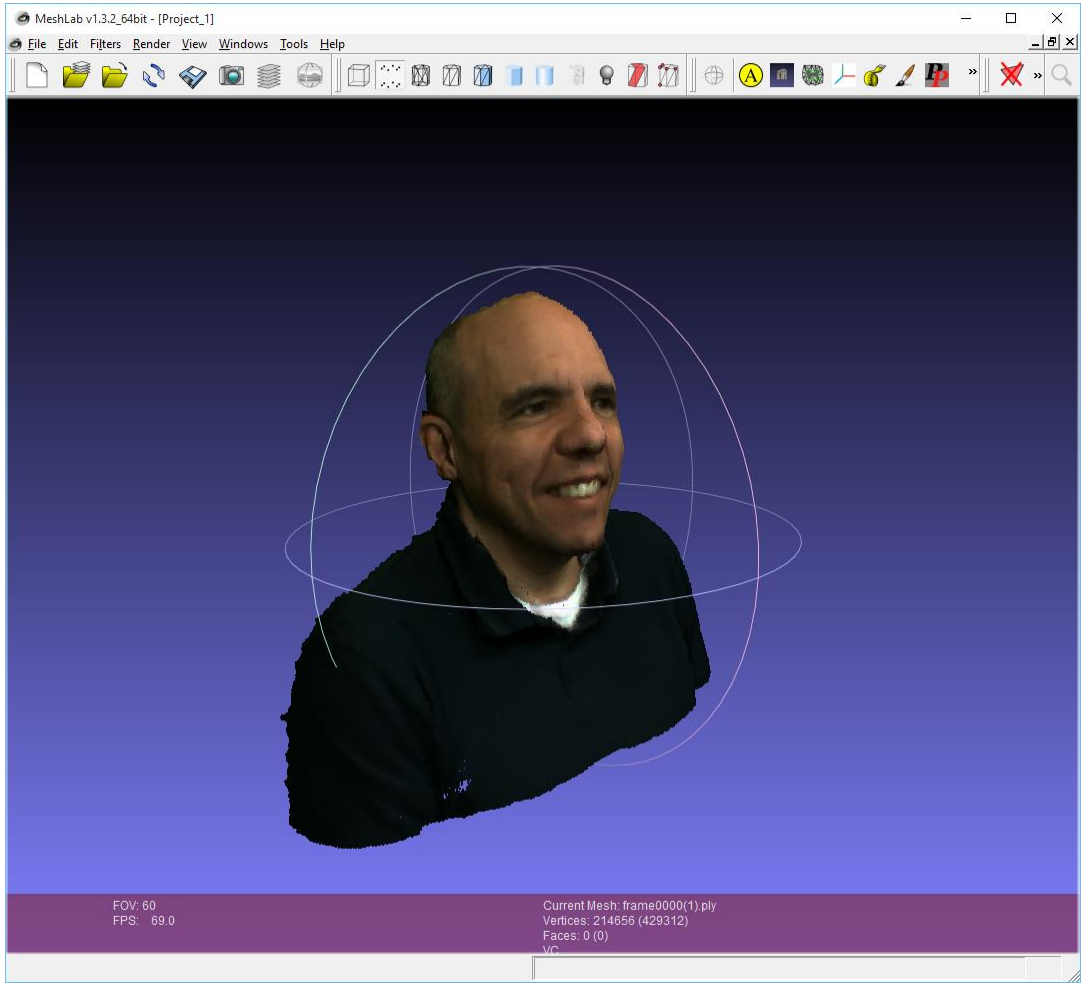
Application to Point Cloud Geometry and Color Compression

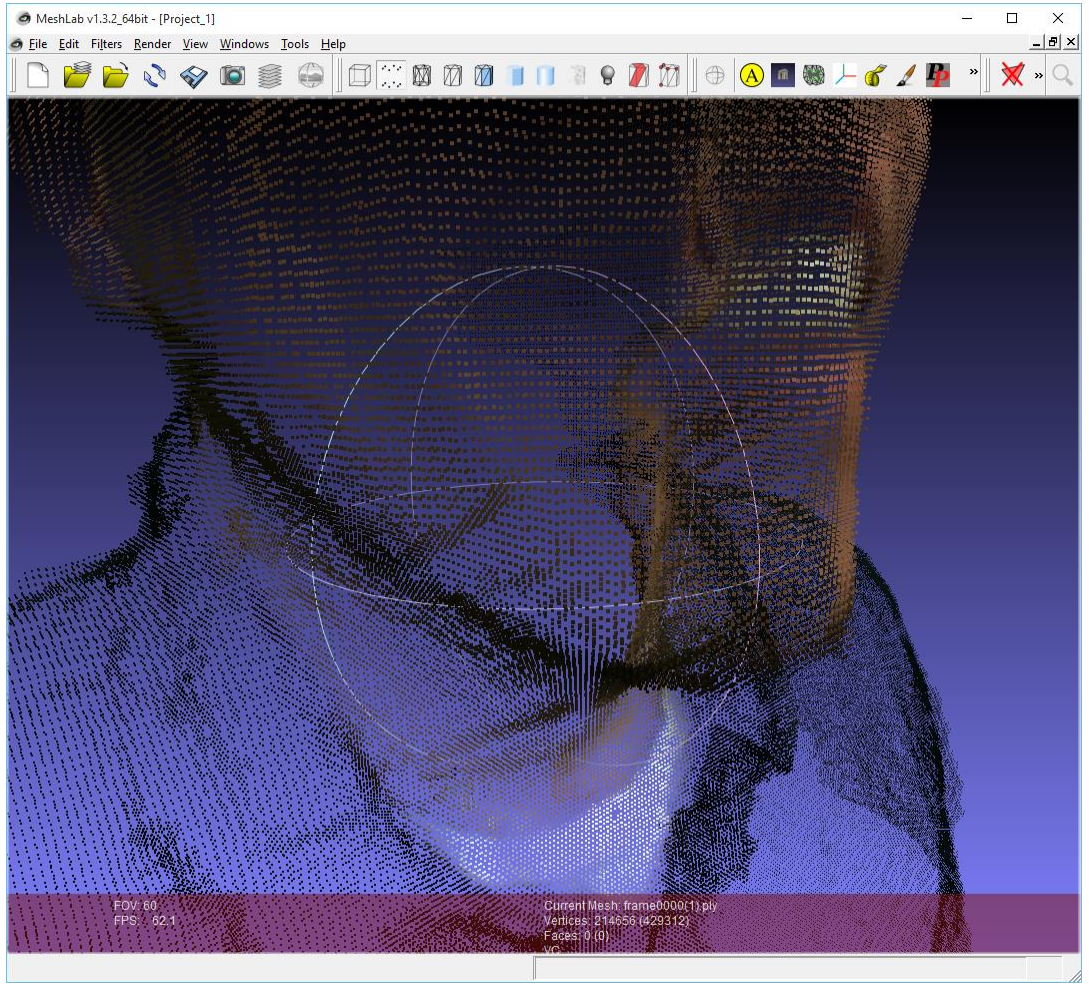
Thanou, Chou, and Frossard, “Graph-based compression of dynamic 3D point cloud sequences,” TIP 2015
Queiroz and Chou, “Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform,” TIP 2016
Queiroz and Chou, “Motion-Compensated Compression of Dynamic Voxelized Point Clouds,” TIP 2016



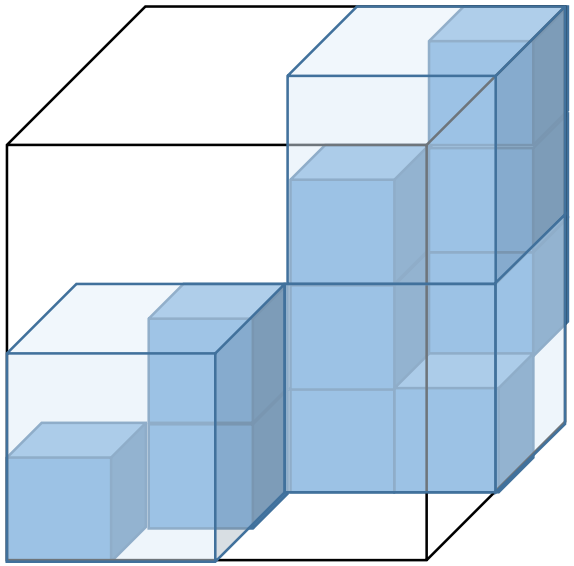
Application to Point Cloud Geometry and Color Compression

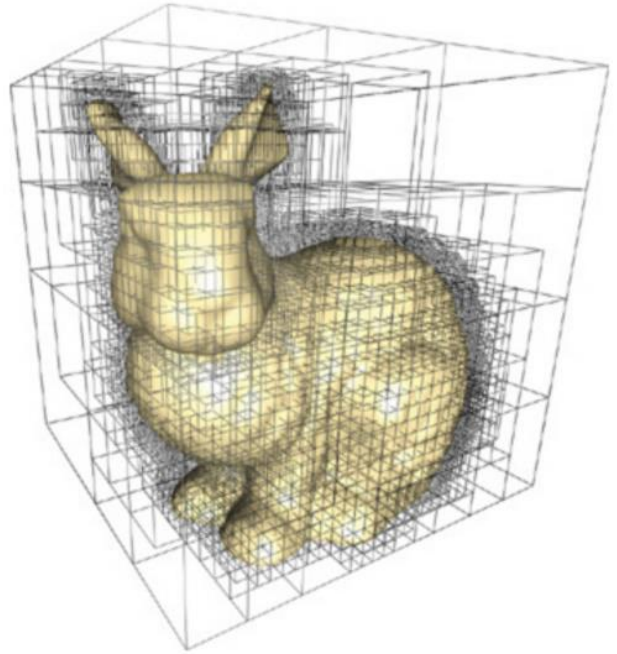
Thanou, Chou, and Frossard, “Graph-based compression of dynamic 3D point cloud sequences,” TIP 2015
Queiroz and Chou, “Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform,” TIP 2016
Queiroz and Chou, “Motion-Compensated Compression of Dynamic Voxelized Point Clouds,” TIP 2016

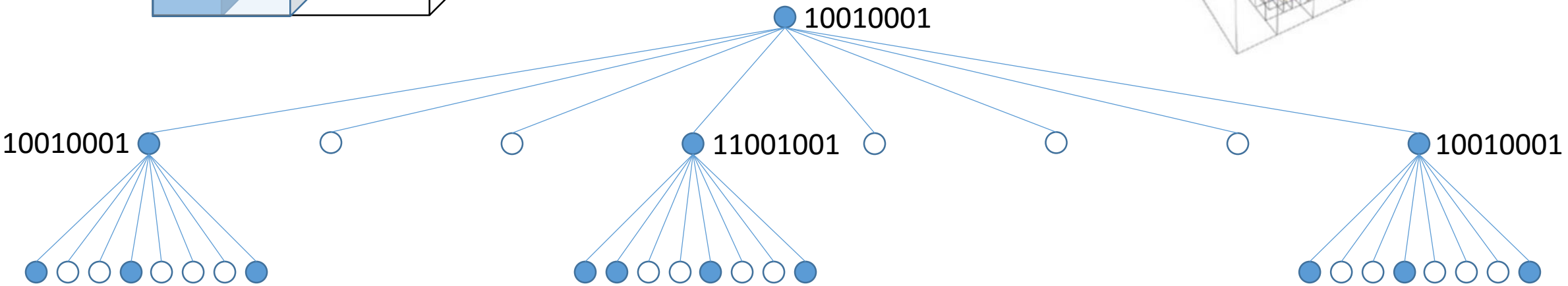




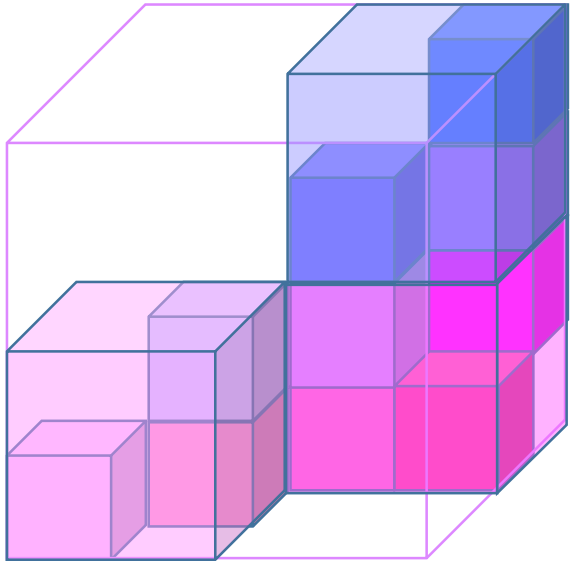
Octree Coding for (Static) Geometry

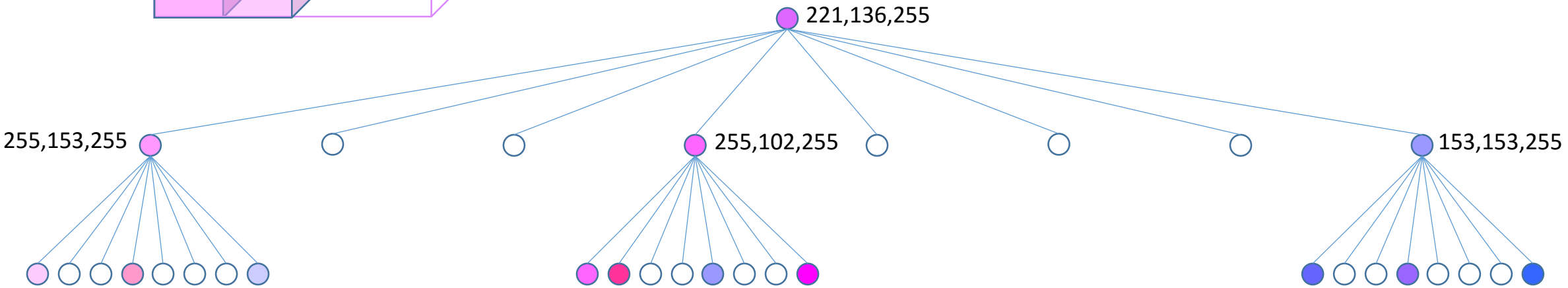




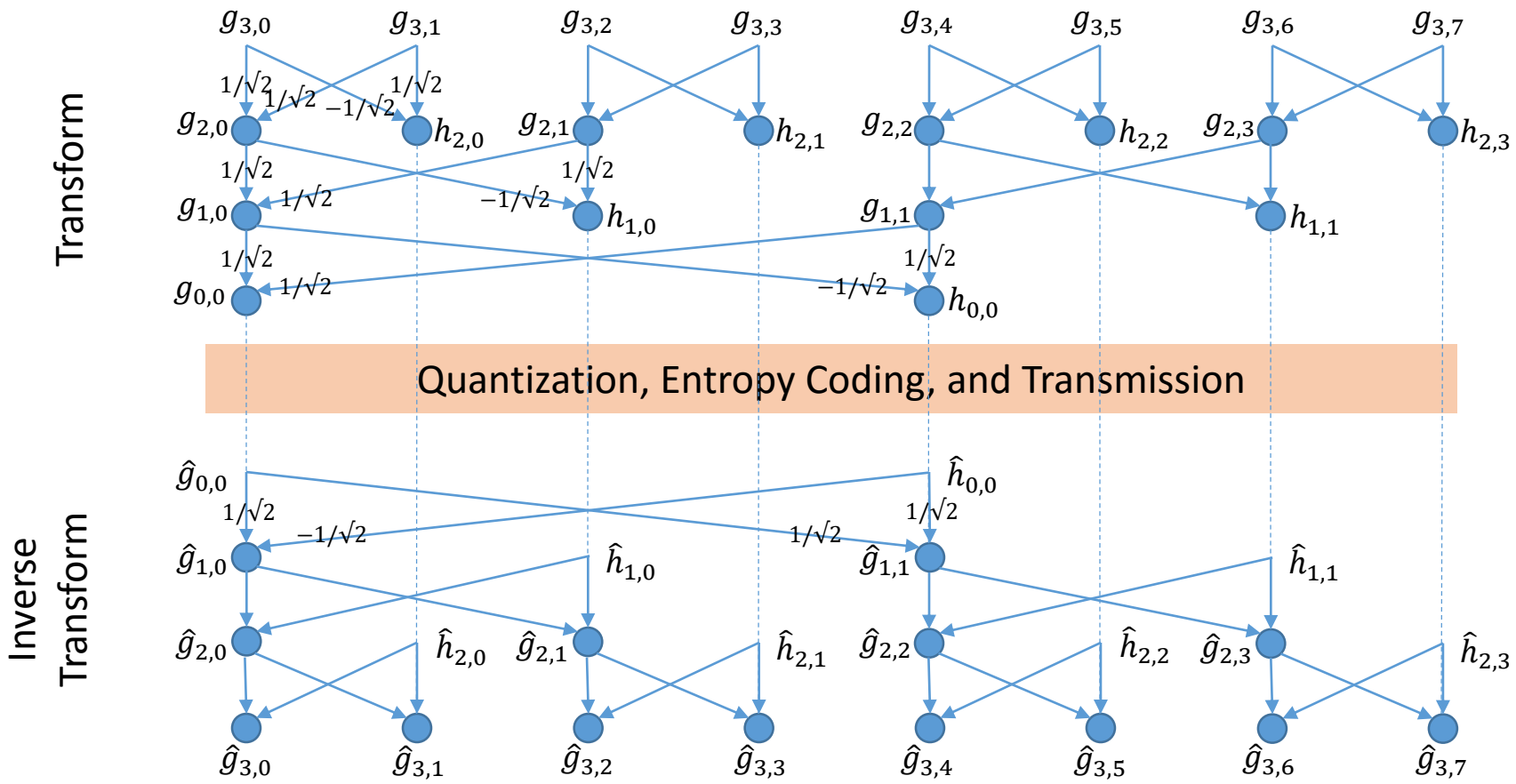


Octree Coding for Color?





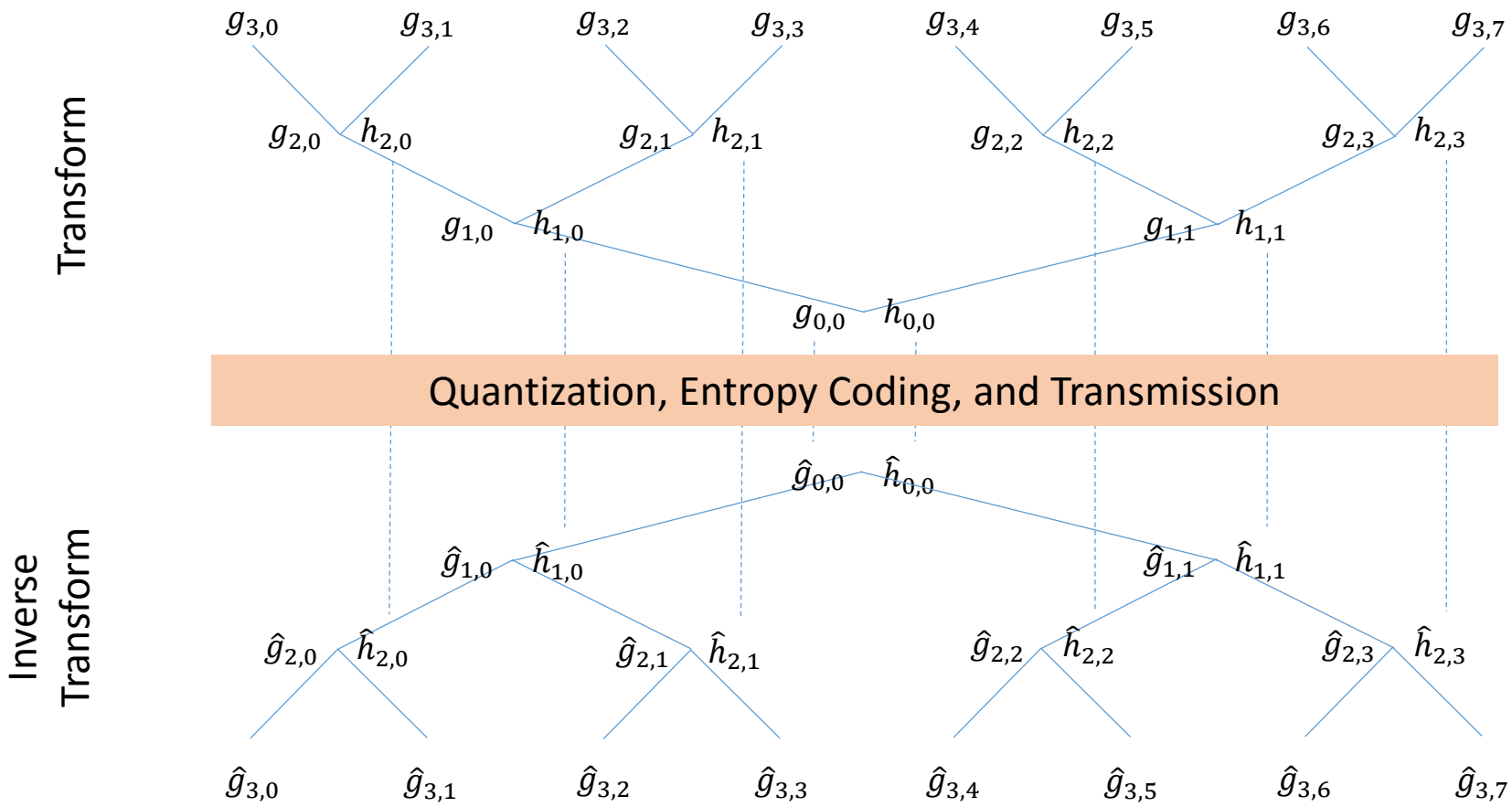
Haar Butterfly



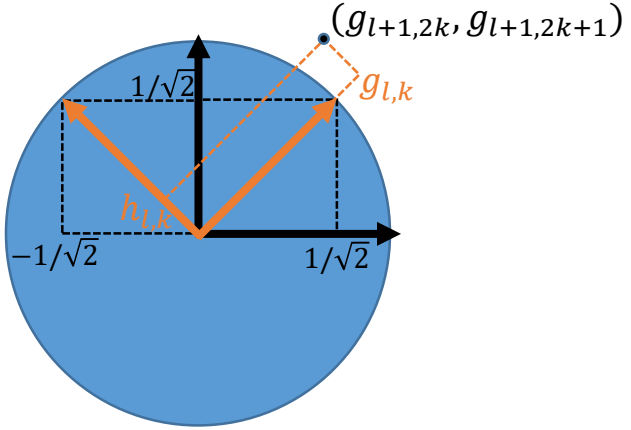
$$\begin{bmatrix} g_{l,k} \\ h_{l,k} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} g_{l+1,2k} \\ g_{l+1,2k+1} \end{bmatrix}$$

$$\begin{bmatrix} \hat{g}_{l+1,2k} \\ \hat{g}_{l+1,2k+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \hat{g}_{l,k} \\ \hat{h}_{l,k} \end{bmatrix}$$

Haar Tree

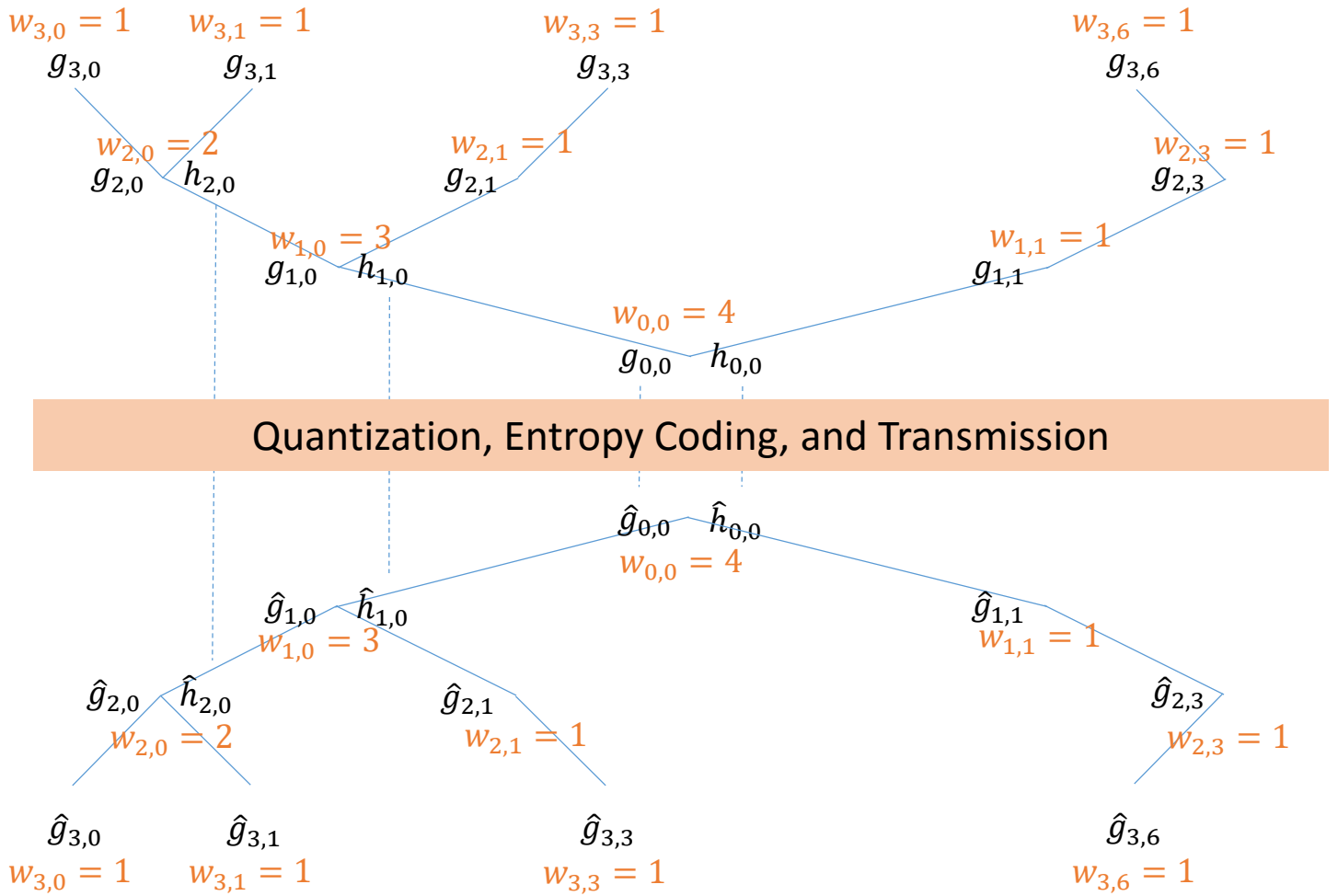


$$\begin{bmatrix} g_{l,k} \\ h_{l,k} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} g_{l+1,2k} \\ g_{l+1,2k+1} \end{bmatrix}$$



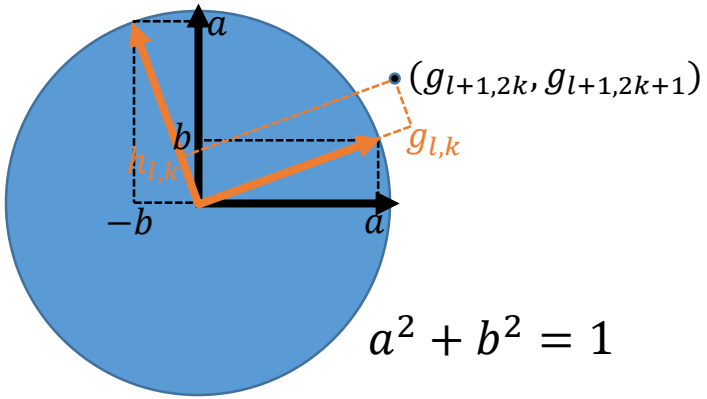
$$\begin{bmatrix} \hat{g}_{l+1,2k} \\ \hat{g}_{l+1,2k+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \hat{g}_{l,k} \\ \hat{h}_{l,k} \end{bmatrix}$$

Region Adaptive Haar Transform (RAHT)



$$\begin{bmatrix} g_{l,k} \\ h_{l,k} \end{bmatrix} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \begin{bmatrix} g_{l+1,2k} \\ g_{l+1,2k+1} \end{bmatrix}$$

$$\begin{bmatrix} \hat{g}_{l+1,2k} \\ \hat{g}_{l+1,2k+1} \end{bmatrix} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \begin{bmatrix} \hat{g}_{l,k} \\ \hat{h}_{l,k} \end{bmatrix}$$



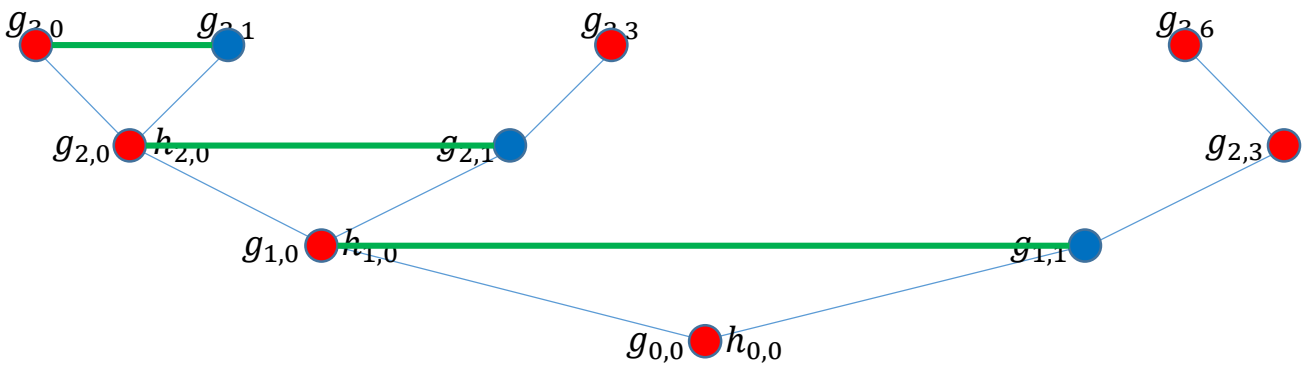
$$a^2 + b^2 = 1$$

$$a = \sqrt{\frac{w_{l+1,2k}}{w_{l+1,2k} + w_{l+1,2k+1}}}$$

$$b = \sqrt{\frac{w_{l+1,2k+1}}{w_{l+1,2k} + w_{l+1,2k+1}}}$$

Scaled DC: $\sqrt{w_{l+1,2k} + w_{l+1,2k+1}}\, g_{l,k} = \sqrt{w_{l+1,2k}}\, g_{l+1,2k} + \sqrt{w_{l+1,2k+1}}\, g_{l+1,2k+1}$

Tree specifies bi-partite graph



Lowpass subband

Highpass subband

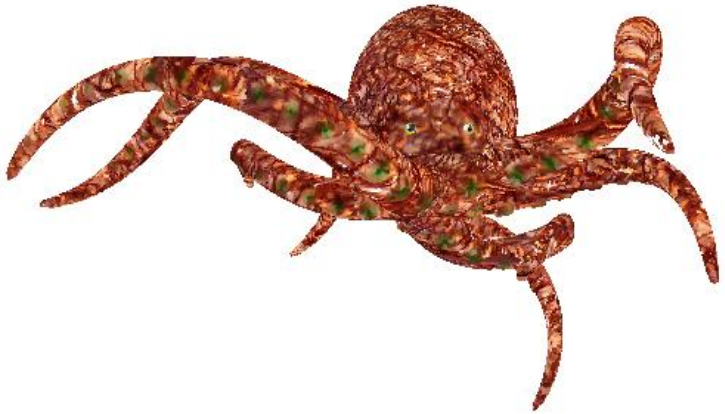
- Tree is a rule
  - At level L, how to connect and color nodes to form a bi-partite graph
  - At level $L-1$, how to reconnect low-pass nodes into a bi-partite graph
- Could be the way to generalize beyond 2-tap Haar using GSP

Feedforward Adaptive Arithmetic Coding

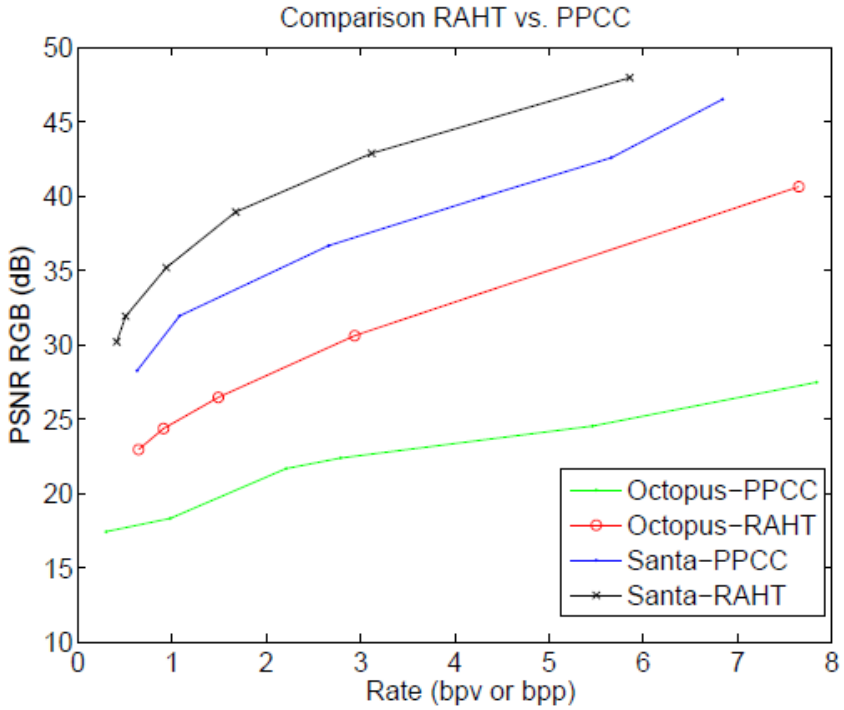
- Each final transformed data with the same weight is grouped in the same subband
- Typically we have about 1000 subbands
- Each subband is encoded using Arithmetic coding
  - Probabilities according to Laplacian distribution
  - Both encoder and decoder have to agree on the standard deviation λ
- For each subband n, we send λ_n
- λ_n are optimally quantized and encoded using Run-Length Golomb-Rice

RAHT Results (1)

Santa



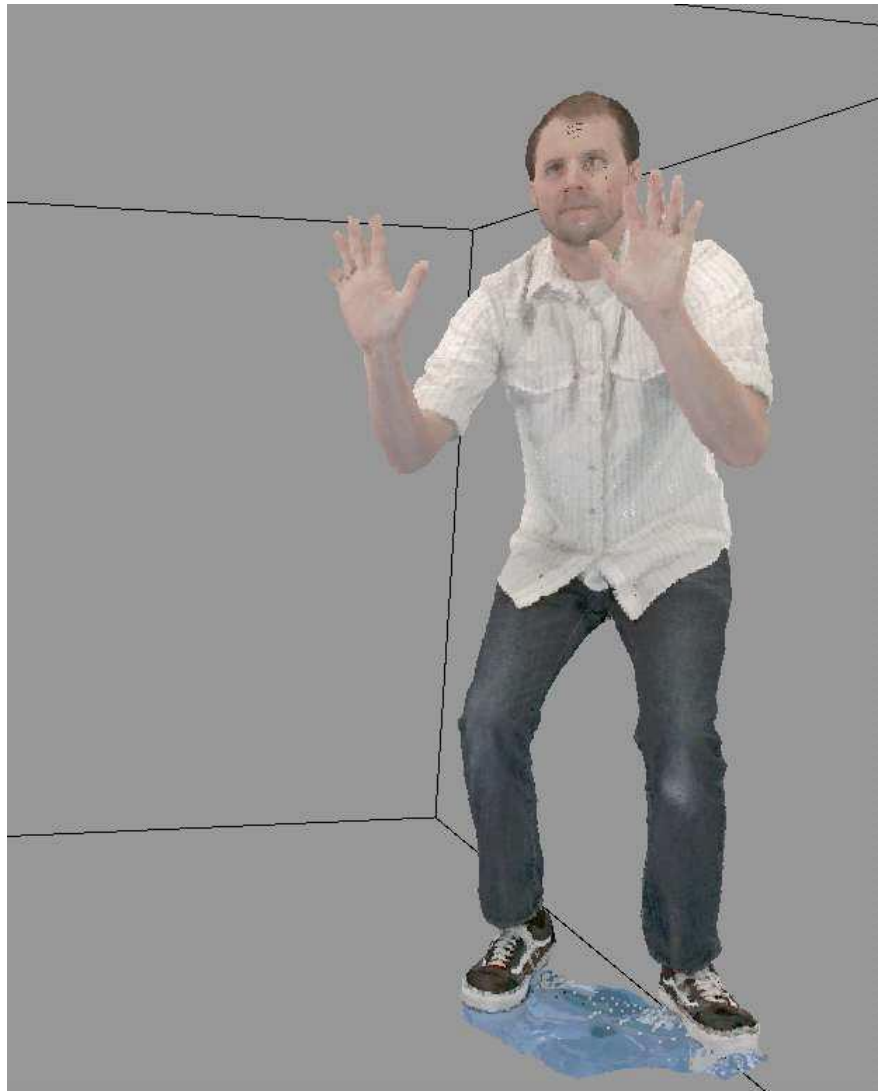
Octopus

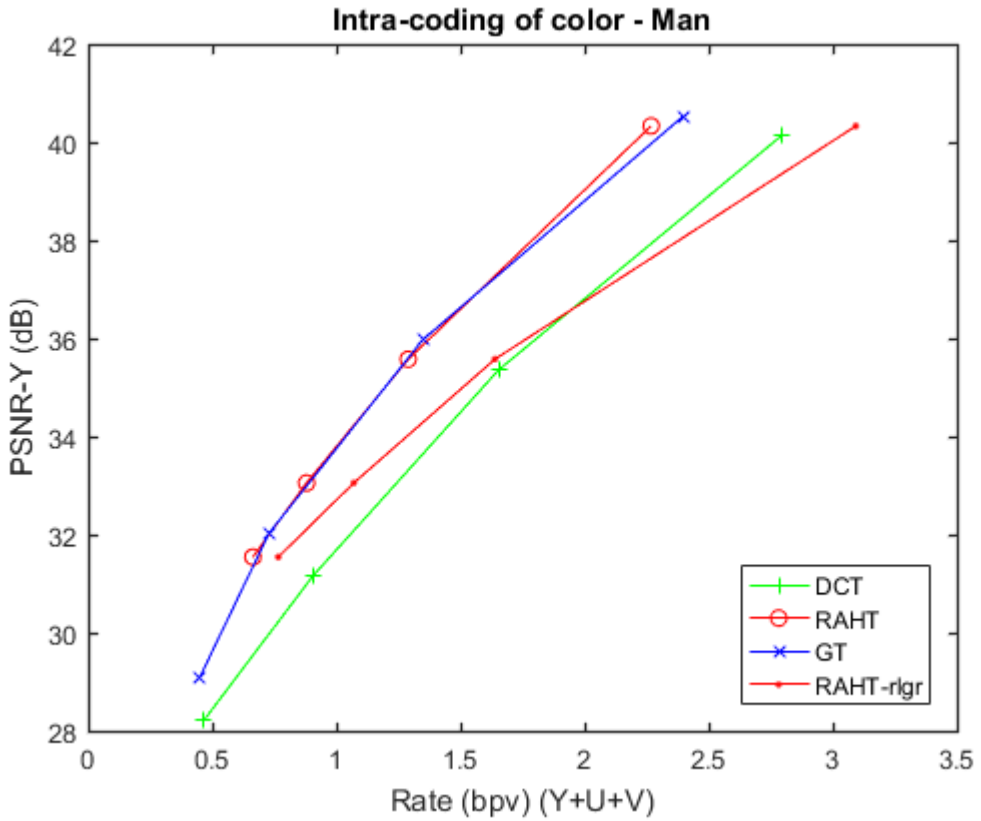


Comparison to Huang, Peng, Kuo, and Gopi, “A generic scheme for progressive point cloud coding,” IEEE Trans. Vis. Comput. Graph., 2008

Results RAHT (2)

Comparison to Zhang, Florencio, and Loop, “Point cloud attribute compression with graph transform,” ICIP 2014





Results RAHT (3)

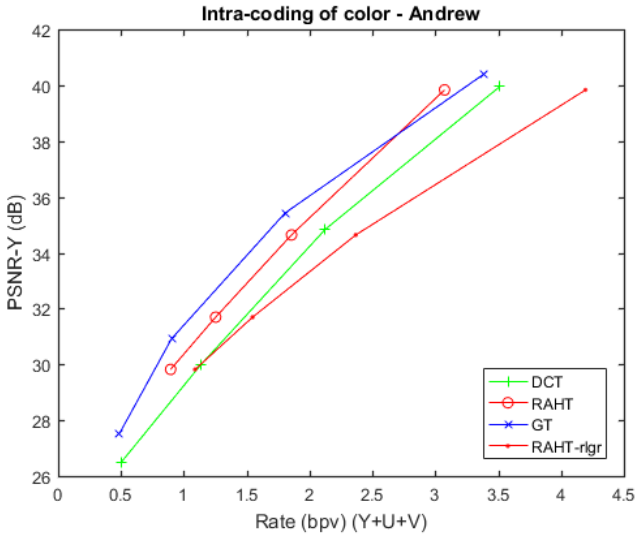
Comparison to Zhang, Florencio, and Loop, “Point cloud attribute compression with graph transform,” ICIP 2014

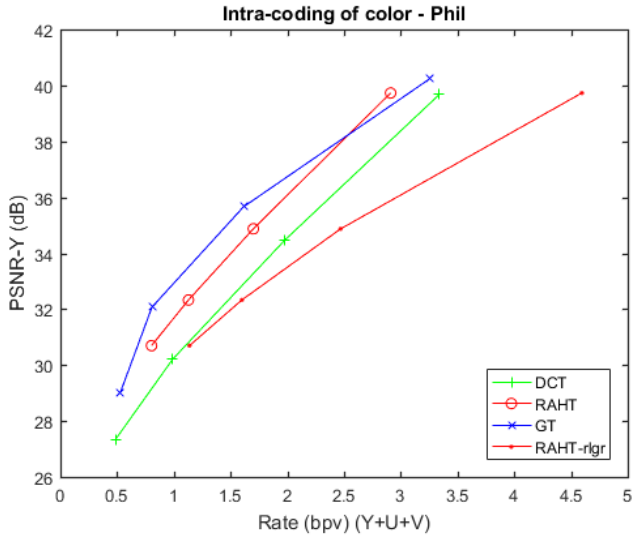
Andrew

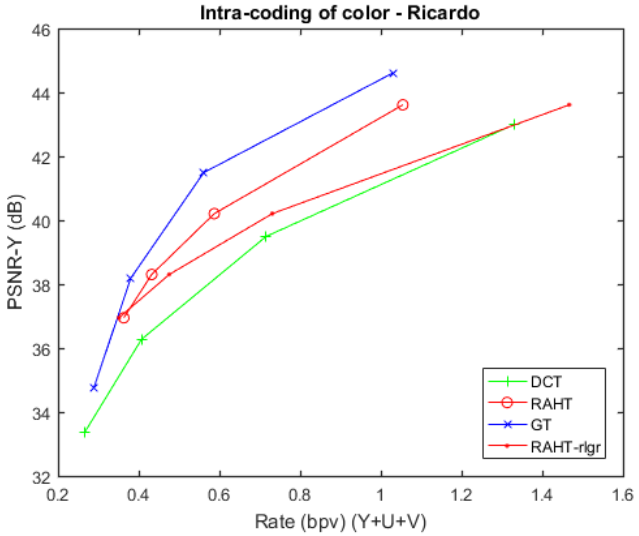
Phil

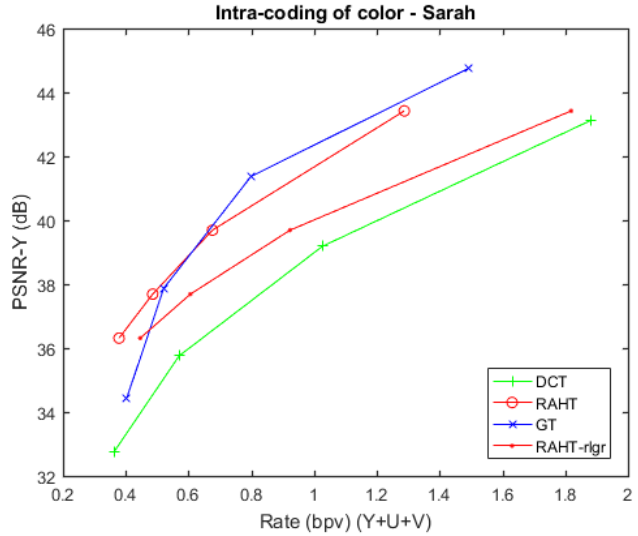
Ricardo

Sarah







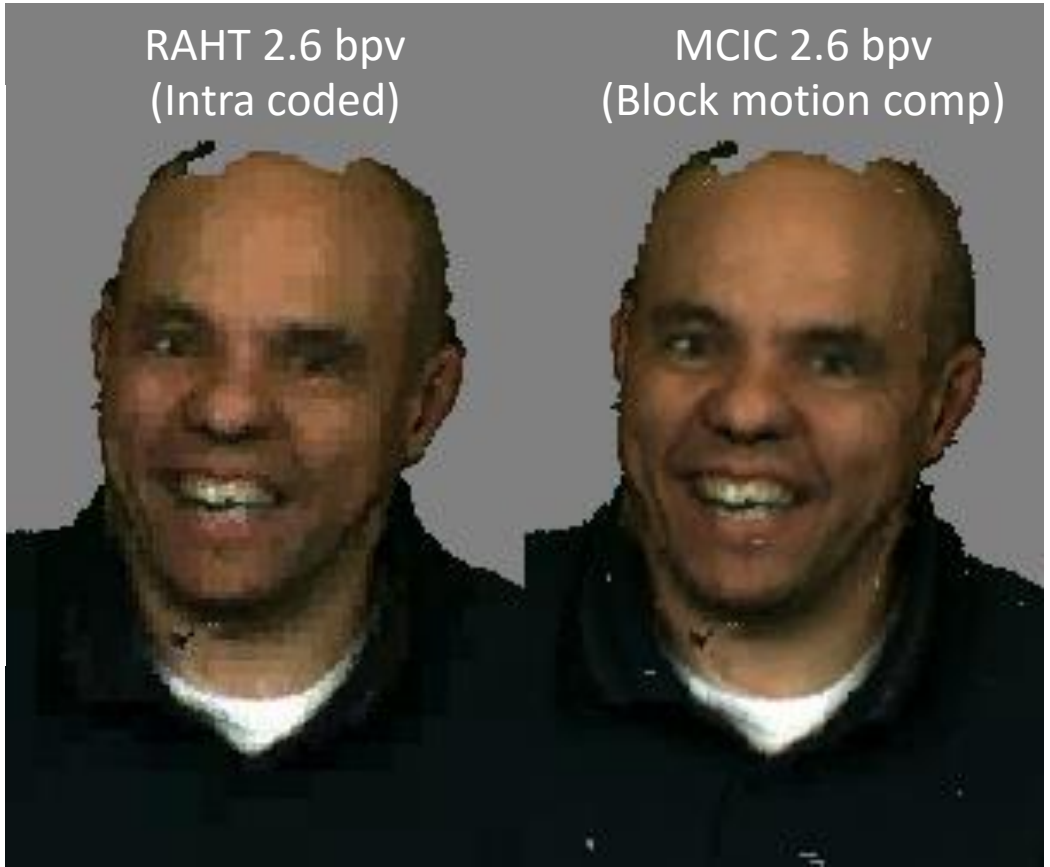


Compression of Dynamic Point Clouds

Dorina Thanou, Philip A. Chou, and Pascal Frossard, Graph-Based Motion Estimation and Compensation for Dynamic 3D Point Cloud Compression, IEEE Trans. Image Processing, Apr. 2016

Sparse motion estimation;
Dense (interpolated) motion compensation

Ricardo L. de Queiroz and Philip A. Chou, Motion-Compensated Compression of Dynamic Voxelized Point Clouds, IEEE Trans. Image Processing, submitted 2016



Rough sense of overall bitrates

- 2 Gbps: barely coded (holoportation demo) – 1 Mpixel color
- 450 Mbps: Intra-frame only, octree for geometry, uncoded 8:1:1 YUV
- 150 Mbps: Intra-frame only, octree for geometry, RAHT for YUV
- 45 Mbps: Inter-frame, hybrid (real time)
- 15 Mbps: poorly-coded mesh + H264-coded texture (siggraph paper, not real time)
- 8 Mbps: well-coded mesh + H265-coded texture
- 6 Mbps: well-coded mesh + periodic texture refresh

Future Issues

- Better motion estimation and compensation for point clouds
- Perceptually relevant distortion measures
- Hybrid mesh / point cloud coding – best of both
- Hybrid AR/VR coding – outside-in + inside-out
- Scalable coding
  - Spatial as well as temporal random access
  - Object scalability, spatial resolution scalability, signal resolution (SNR) scalability
- Rate-distortion optimization
- Error resilience
- Rate control, buffer management
- Streaming architecture
  - Low-end vs high-end devices and rendering location

Conclusion

- VR/AR are 3G immersive comm, 4G platform, to be ~$100 billion / year
- VR is closer in deployment and technology (similar to video coding); AR is later but may subsume VR (will need new coding tools)
- GSP provides an extension of classic tools necessary to process AR signals

- AR coding analogous to video coding circa 1980: coding paradigm not commonly agreed upon. History of video coding gives us a roadmap for development of AR coding.

Thanks

Special thanks go to my interns Ha Nguyen, Dorina Thanou, Amir Anis, and Eduardo Pavel Carvelli; visiting researcher Ricardo de Queiroz; colleagues Dinei Florencio, Cha Zhang, Charles Loop, Qin Cai, and the I3D group; and university collaborators Pascal Frossard, Antonio Ortega, Gene Cheung, and Minh Do.