IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS

Discussion on -

Hierarchical Exploration of Volumes Using Multilevel Segmentation of the Intensity-Gradient Histograms

Cheuk Yiu Ip, Amitabh Varshney, *Fellow, IEEE*,
and Joseph JaJa, *Fellow, IEEE*

Presented by-

Mohammad Imrul Jubair

University of Calgary

Research Domain:

- Volume exploration
- Visualizing meaningful information in the volumetric data

Challenge in Volume Exploration:

- Information Challenge
- Completeness Challenge
- Semantic Challenge

Challenge in Volume Exploration:

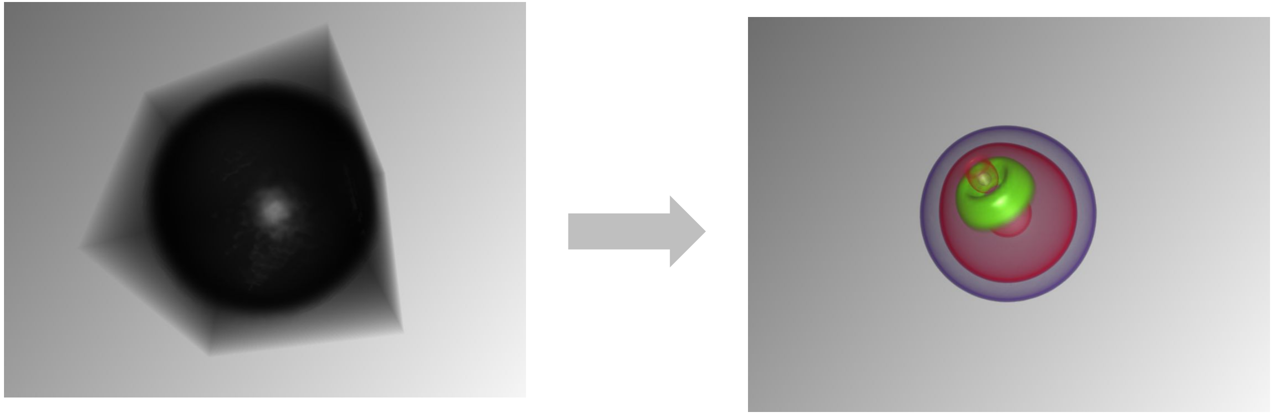
Information Challenge:

- Most research have addressed issues surrounding ***how*** to depict the data, ***what*** to depict remains an important problem

Challenge in Volume Exploration:

Information Challenge:

- Most research have addressed issues surrounding ***how*** to depict the data, ***what*** to depict remains an important problem
- Displaying meaningful feature is an important challenge



Challenge in Volume Exploration:

Completeness Challenge:

- Exhaustive data exploration is a tedious and time-intensive exercise

- But still it is important to ensure that we do not overlook any important features in the data

- Mechanisms are needed to facilitate a complete data exploration

Challenge in Volume Exploration:

Completeness Challenge:

- Exhaustive data exploration is a tedious and time-intensive exercise

- But still it is important to ensure that we do not overlook any important features in the data

- Mechanisms are needed to facilitate a complete data exploration

Semantic Challenge:

- Semantically driven navigation of the data is still a task that is designated for the user

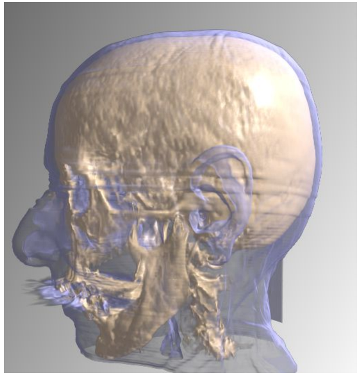
Contribution of this paper:

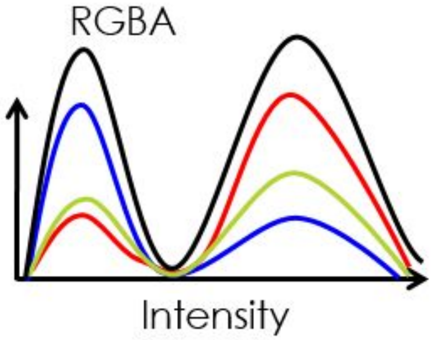
Challenges	Contributions to Address
Information	• Extracting informative regions using image segmentation on reduced statistics • Visually segmenting the 2D histograms
Completeness	• By constructing a complete exploration hierarchy • This hierarchy organizes segments of different sizes from coarse to fine • Assisted by information-theoretic measurement *(Entropy, Information gain etc.)*
Semantic	• Interactive volume exploration interface

Transfer Function Design:

- Transfer functions directly influence the visualization by assigning optical properties such as color and opacity to voxels.

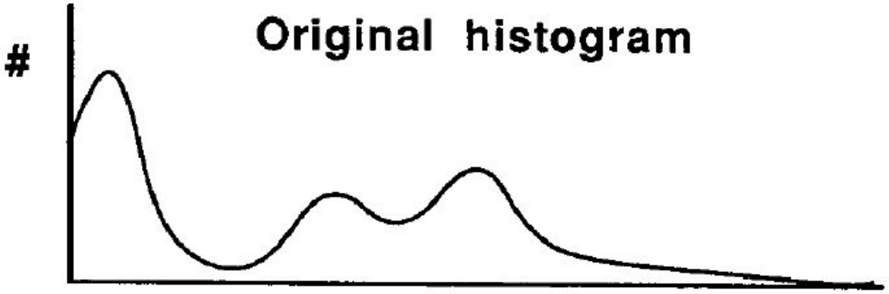
- Finding a good transfer function is critical to producing art informative rendering





Transfer Function Design:

- Histogram Helps Transfer Function Design

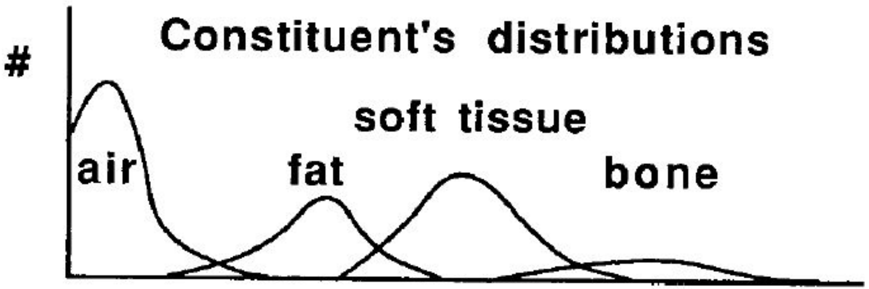


[Drebin *et al*. SIGGRAPH 88]

Transfer Function Design:

- Histogram Helps Transfer Function Design

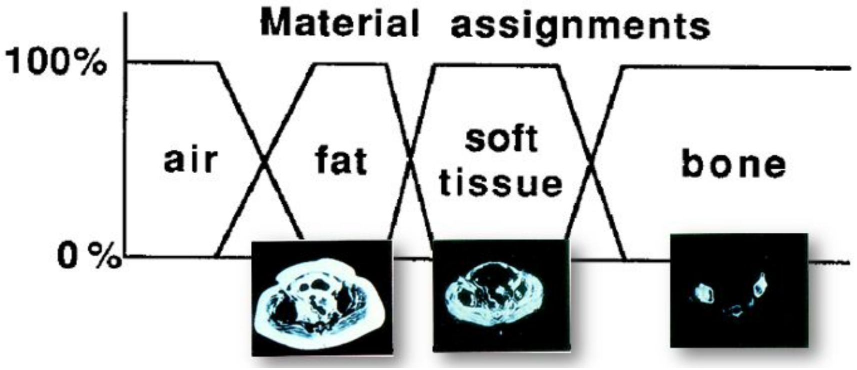
- The peaks of the histogram are composed by the distributions of different constituents



[Drebin *et al*. SIGGRAPH 88]

Transfer Function Design:

- Histogram Helps Transfer Function Design
- The peaks of the histogram are composed by the distributions of different constituents
- By assigning different colors and opacities to these components. we can visualize different materials

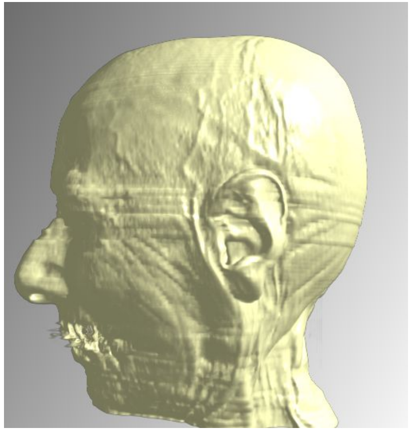


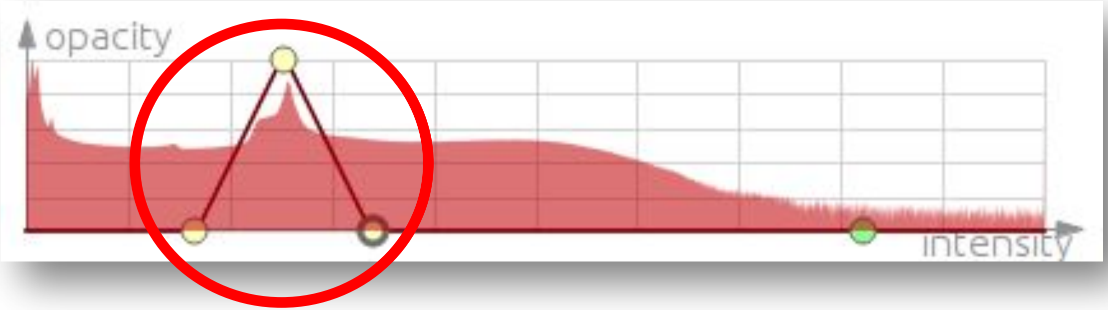
[Drebin *et al*. SIGGRAPH 88]

Transfer Function Design:

Example:

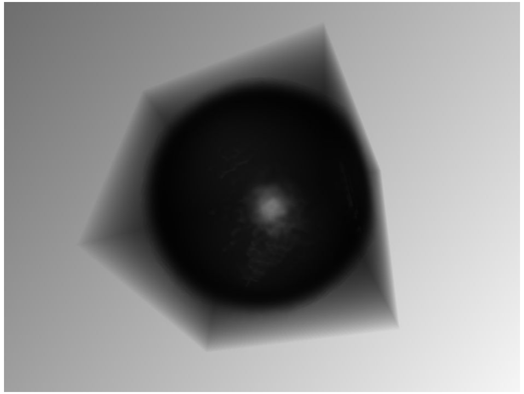
- following the peak in the histogram to obtain visualization of the cadaver head dataset

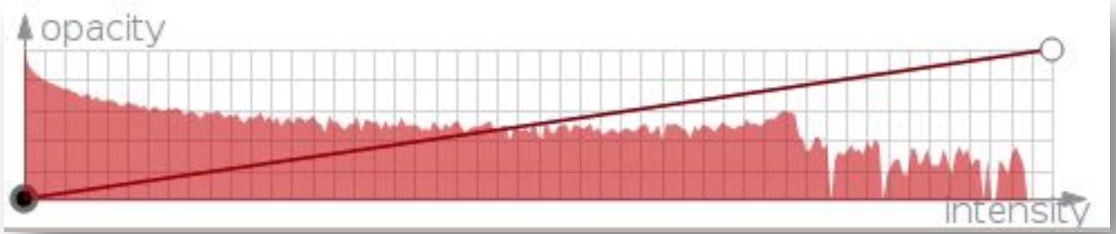




Transfer Function Design:

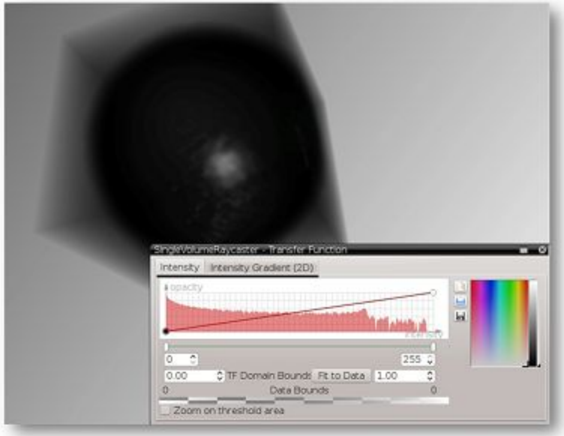
However, the separation of the distributions may not be always clear





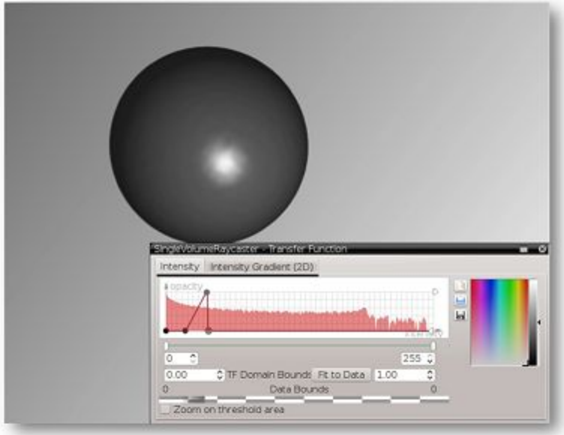
Transfer Function Design:

In this case, we have to exhaustively explore the dataset by modifying the transfer function



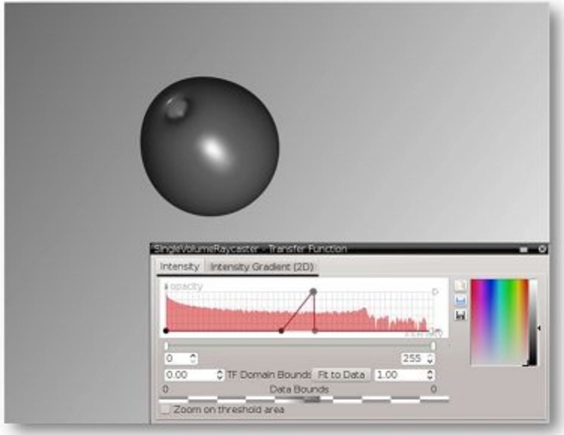
Transfer Function Design:

In this case, we have to exhaustively explore the dataset by modifying the transfer function



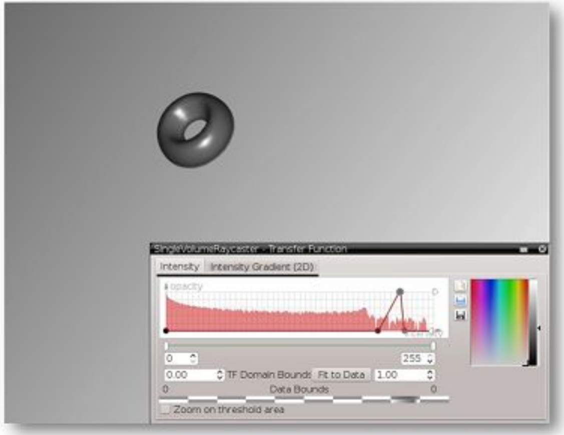
Transfer Function Design:

In this case, we have to exhaustively explore the dataset by modifying the transfer function



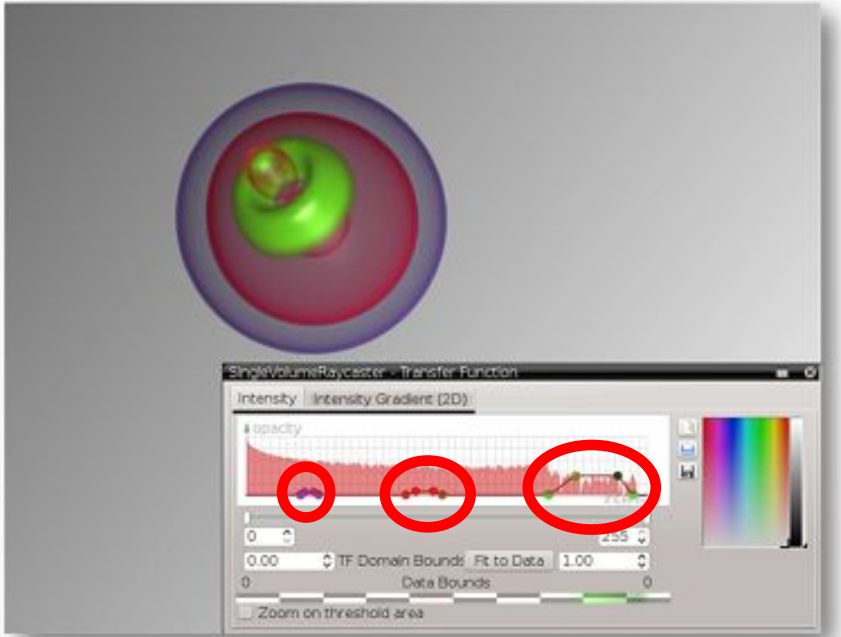
Transfer Function Design:

In this case, we have to exhaustively explore the dataset by modifying the transfer function



Transfer Function Design:

The meaningful feature can be located arbitrarily



2D Transfer Function

- To assist the search for meaningful and salient features, the transfer function has been extended to multiple dimension.

- Firstly introduced by *Levoy*

- Example: 2D-

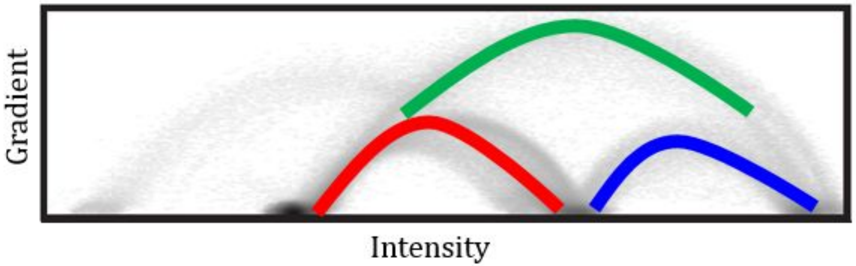
    - A histogram in 2D with the intensity on the x axis, gradient on the y axis, the darkness of the shade represents the frequency *(intensity-gradient histogram)*

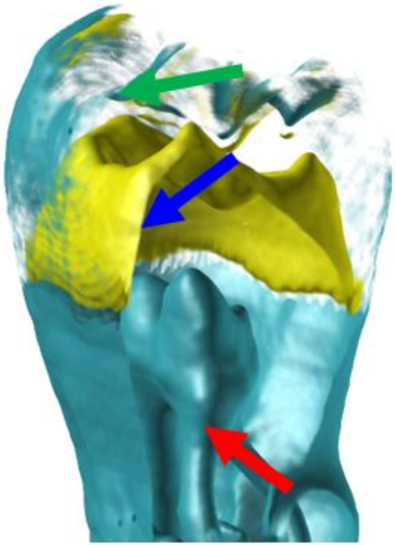
    - The gradient *f'(x)* helps capturing material boundaries.

2D Transfer Function

- Shapes on the 2D intensity-gradient histogram correspond to meaningful volumetric segments.

- This has been implemented in popular visualization packages such as *Voreen*, *ImageVis3d* and *VisIt*





Related works on TF

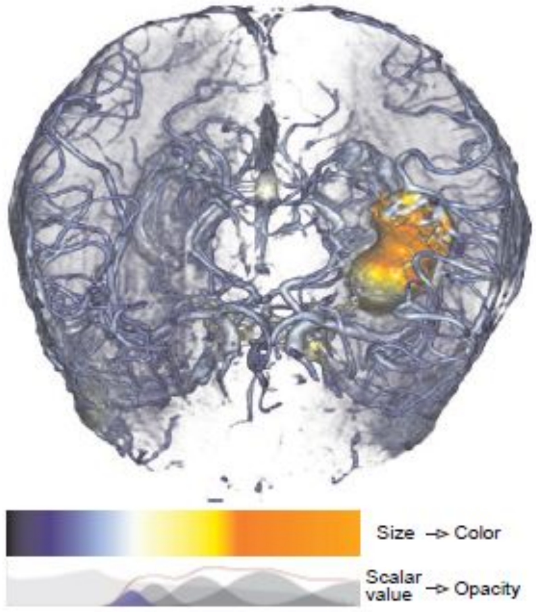
1. Kindlmann *et al.*:

- ✓ Uses **higher** derivatives: *f"(x)*
- ✓ Each bin in histogram volume represents the combination of values of three variables f, f' and f".
- ✓ Value stored in each bin signifies the number of voxels in the original volume within that same combination of ranges of these three variables.

Related works on TF

2. Correa *et al.*:

- ✓ Uses the relative **size** of features.
- ✓ Maps the relative size of local features in a volume to color and opacity



Related works on TF

3. Sereda *et al.*:

- ✓ Uses LH Histogram (Serlie *et al.*) that shows lower and higher intensities of materials that form the boundaries

4. Salama *et al.*:

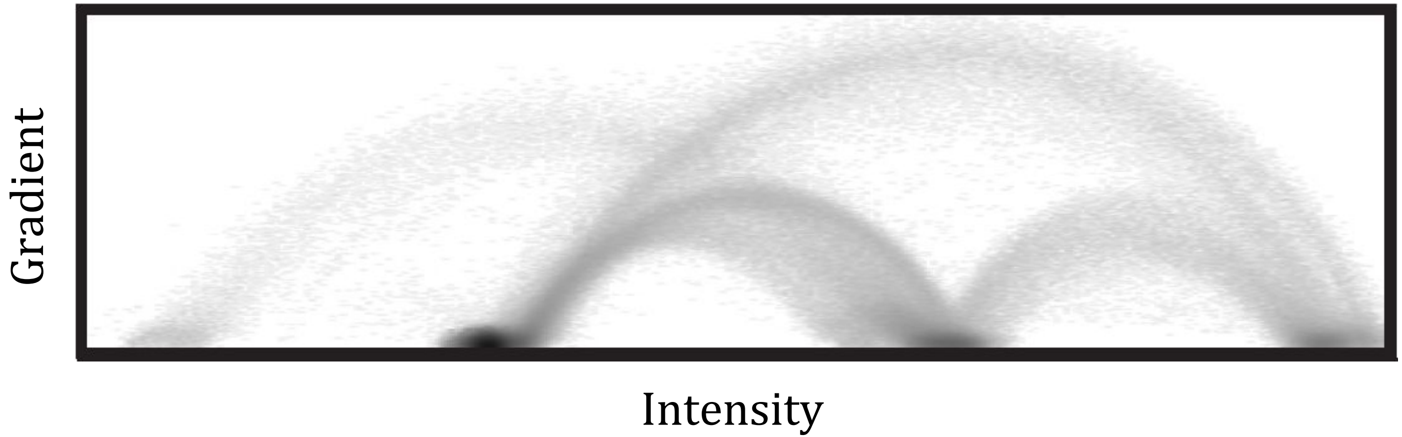
- ✓ Domain specific semantic attributes

5. Ruiz *et al.*:

- ✓ Information divergence

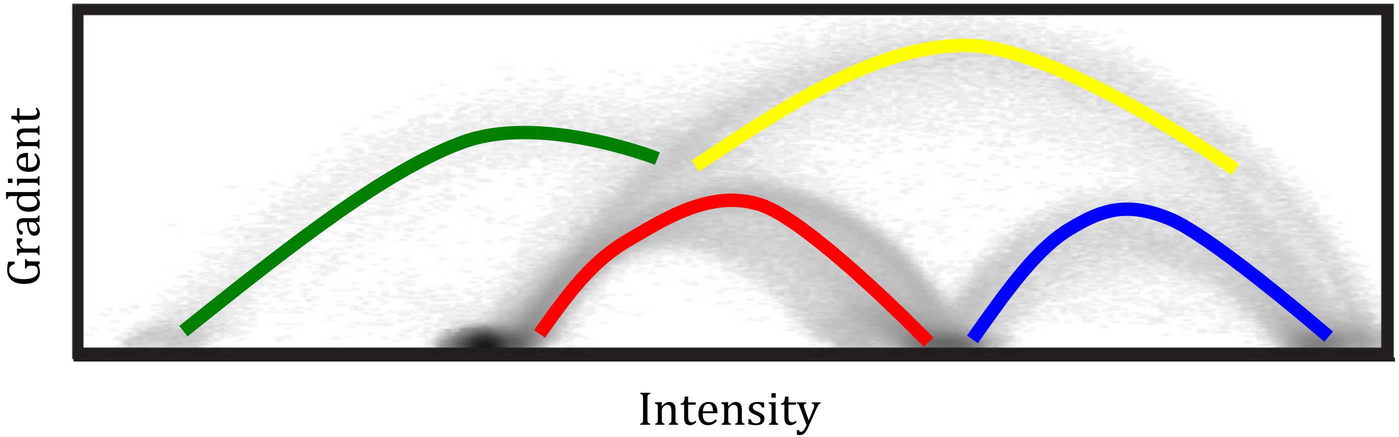
Intensity-Gradient histogram

✓ This paper focuses on Intensity-Gradient Histogram



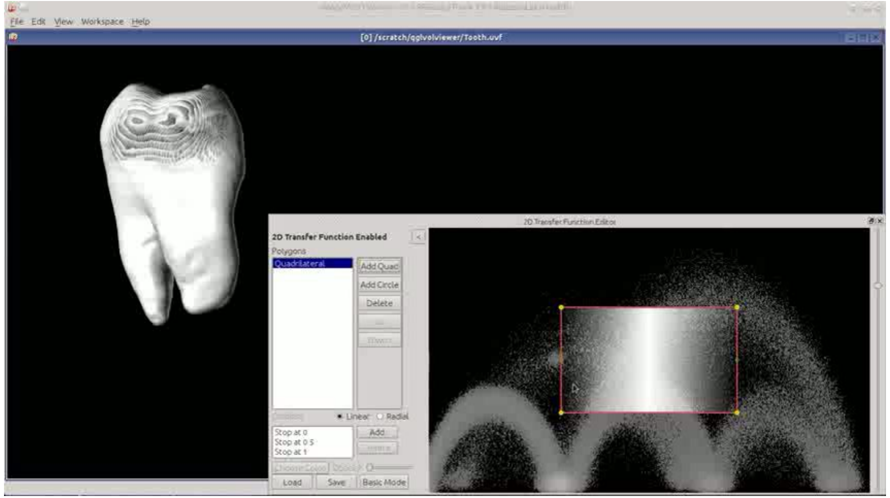
Intensity-Gradient histogram

- ✓ This paper focuses on Intensity-Gradient Histogram
- ✓ The intensity gradient histogram is very effective because human users can extract volume segments by recognizing the arcs and blobs



Search for meaningful features

✓ Users draw shapes on histogram to extract meaningful components



User trying to make polygon widget to highlight the feature that corresponds to the top arc
(ImageVis3D)

Search for meaningful features

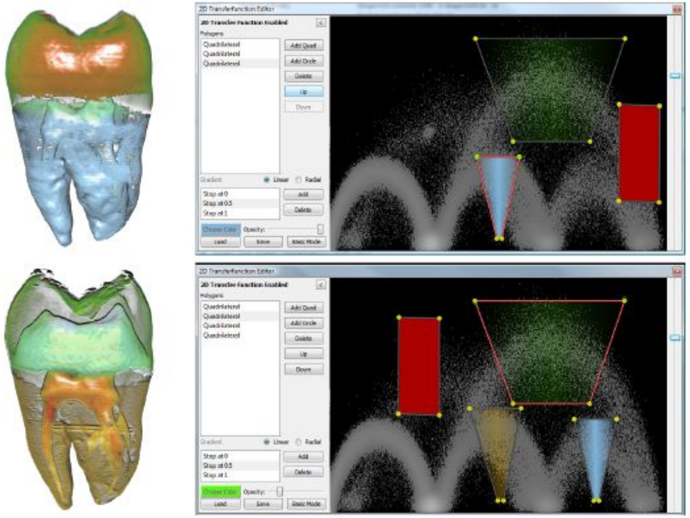
- ✓ Users draw shapes on histogram to extract meaningful components

- ✓ Manipulating the widget takes quite some effort

User trying to make polygon widget to highlight the feature that corresponds to the top arc
(ImageVis3D)

Search for meaningful features

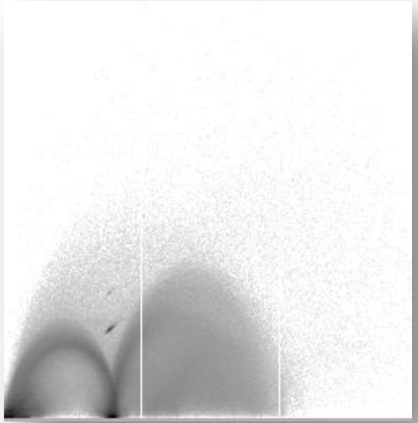
- ✓ Users draw shapes on histogram to extract meaningful components
- ✓ Manipulating the widget takes quite some effort
- ✓ The challenge here is the histogram does not show where do the good features separate



Effect of 2D Transfer Function Widget Positioning on a Dataset

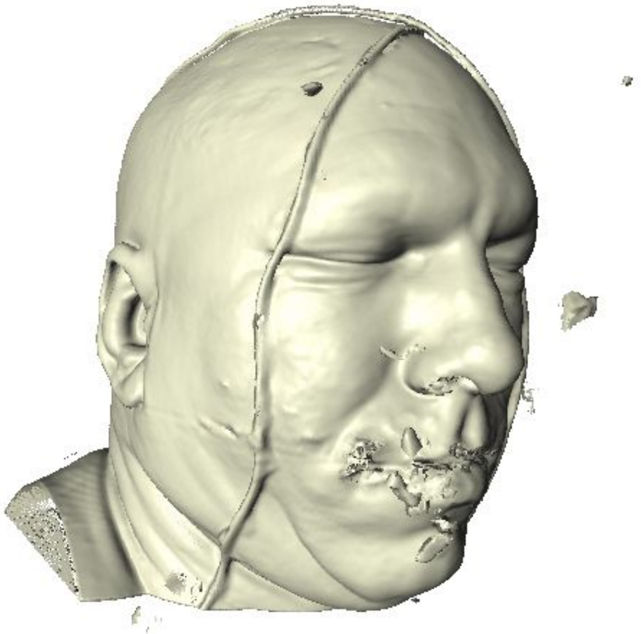
Volume features and histogram

✓ Meaningful components do correspond to histogram segments



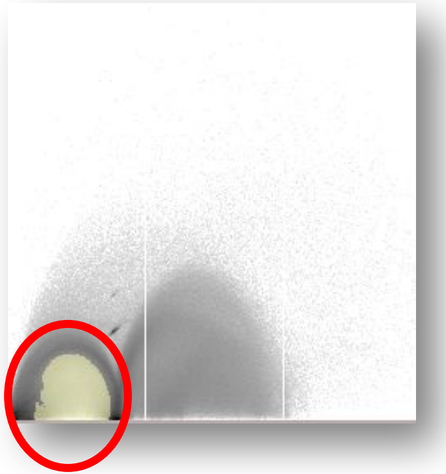
Volume features and histogram

✓ Meaningful components do correspond to histogram segments



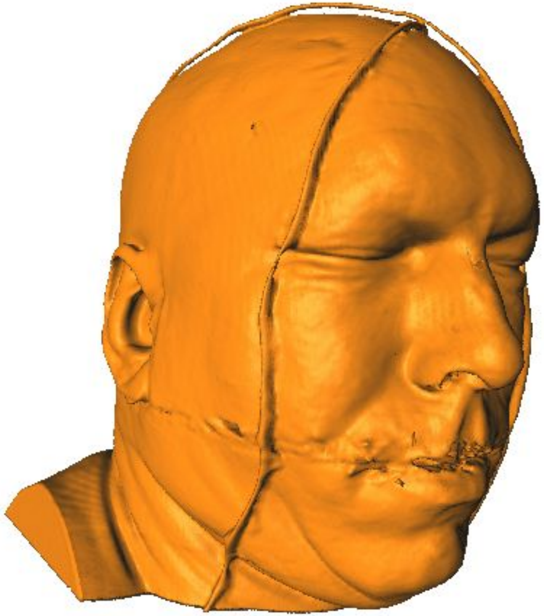
Ski

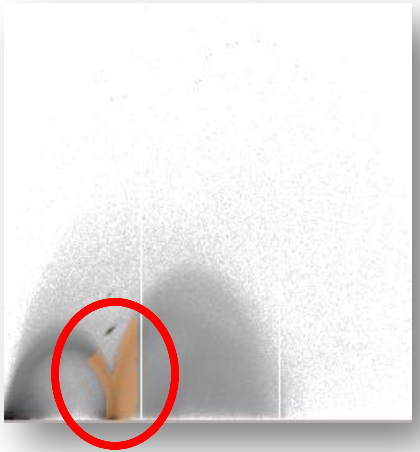
n



Volume features and histogram

✓ Meaningful components do correspond to histogram segments

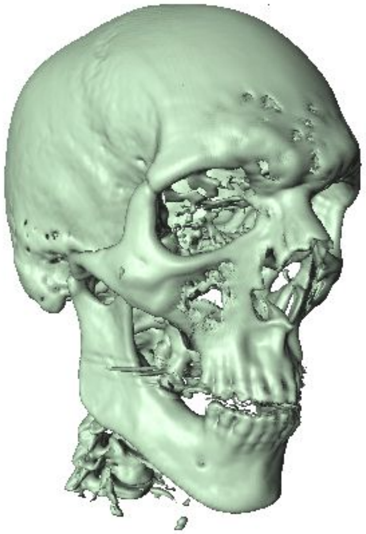




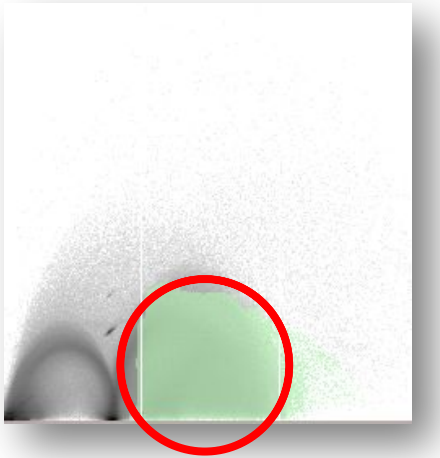
fles
h

Volume features and histogram

✓ Meaningful components do correspond to histogram segments

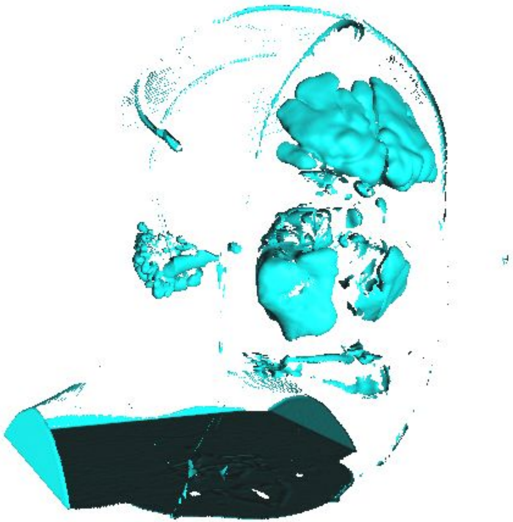


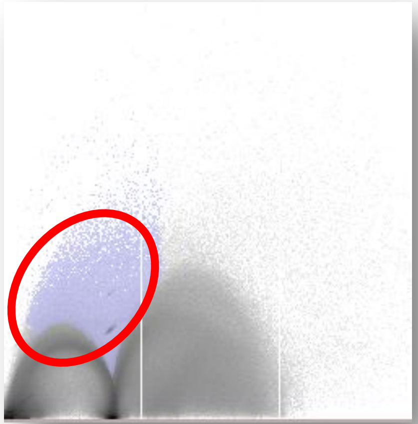
skul
l



Volume features and histogram

✓ Meaningful components do correspond to histogram segments



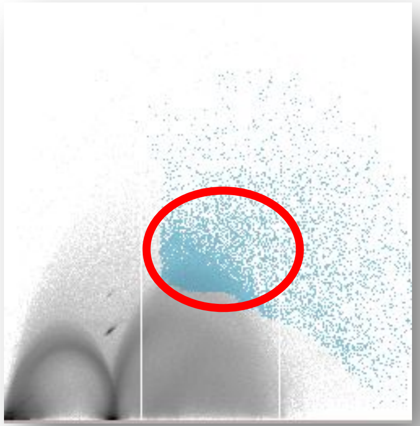


sinu
s

Volume features and histogram

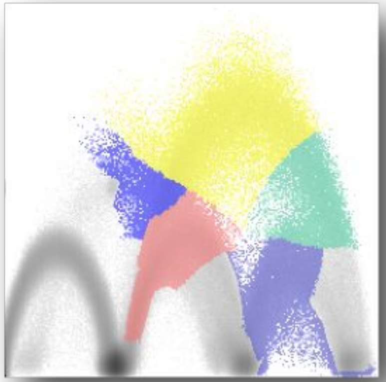
✓ Meaningful components do correspond to histogram segments

teet
h



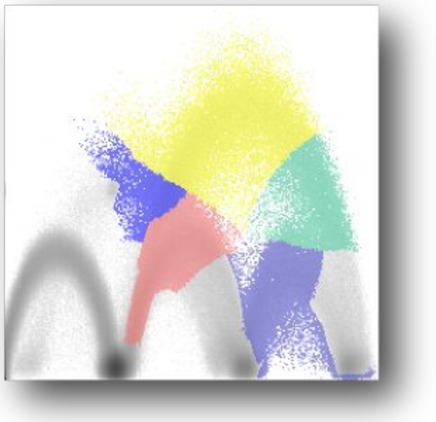
Goal

- ✓ Meaningful components do correspond to histogram segments
- ✓ Goal of this paper is to show you how to discover these regions in a systematic manner.



Reduce search to classification

- ✓ By recursively segmenting the histogram.
- ✓ A visual-inspired approach is introduced that fits the histogram with a few segments



Overview

- ✓ Segment the histogram statistics
- ✓ Build an exhaustive multilevel hierarchy
- ✓ User interactive exploration

Advantages

- ✓ Visual segmentation matches user intuition
- ✓ Augments the intensity-gradient feature space without requiring the users to learn any new features

Segmenting the histogram

- Instead of directly segmenting the volume data, the histogram is segmented
- Normalized-cut (*ncut*) *[Shi & Malik PAMI 2000]* to find shapes

Example: *normalized cut provides intuitive segments on natural images*

[Wang *et al.* PATTERN RECOGN LETT 06]

The 'Cut'

From: Shi & Malik *et al.* PAMI 2000

- A graph G = (V, E) partitioned into two disjoint sets A and B-

$$A \cup B = V,\ A \cap B = \emptyset$$

- By simply removing edges connecting the two parts

- The degree of dissimilarity between these two pieces can be computed as total weight of the edges that have been removed.

- In graph theoretic language, it is called the *cut*:

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$$

The 'Cut' *(cont….)*

From: Shi & Malik *et al.* PAMI 2000

- The optimal bi-partitioning of a graph is the one that minimizes this *cut* value.

- Although there are an exponential number of such partitions, finding the minimum cut of a graph is a well-studied problem and there exist efficient algorithms for solving it.

The 'Cut' in image segmentation

- It models an image as a graph and finds the best way to partition this graph into *k* components.

- Every pixel in the image is considered as a node on the graph.

- The edge weights, $w(u,v)$, between the nodes, u and v, are computed as *color* and *location* similarities between the pixels.

- The *closer* the pixels, the *stronger* the edge weight is.

‘Normalized Cut’ in image segmentation

- The normalized cut seeks to disconnect the graph, V, into components A, B by removing the edges with the *least normalized cost*.

'Normalized Cut' in image segmentation

- The normalized cut seeks to disconnect the graph, V, into components A, B by removing the edges with the *least normalized cost*.

- The formulation of the normalized cut is as follows:

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$$

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}$$

- *cut (A, B)* = total weight of edges connecting components A and B
- *assoc (A, V)* = total connection from nodes in A to all nodes in graph
- *assoc (B, V)* = *(similarly defined)*
- *Ncut (A, B)* normalizes *cut (A, B)*

[Shi & Malik *et al.* PAMI 2000]

‘Normalized Cut’ in image segmentation *(cont.…)*

Finding the minimum *Ncut (A, B)* is a NP-complete problem.
This is usually approximated by solving an eigenvalue problem:

$$(D - W)y = \lambda D y$$
$$d(u) = \sum_{v} w(u, v)$$

where,

W = adjacency matrix of the image graph with edge weights w (u, v)
d(u) = total connection from node *u* to all other nodes [Shi & Malik *et al.* PAMI 2000]
D = diagonal matrix with entries, *d(u)*
λ = eigenvalue

We can use the resulting eigenvectors, *y* to partition the graph.

‘Normalized Cut’ in image segmentation *(cont....)*

Finding the minimum *Ncut (A, B)* is a NP-complete problem.
This is usually approximated by solving an eigenvalue problem:

$$(D - W)y = \lambda D y$$
$$d(u) = \sum_{v} w(u, v)$$

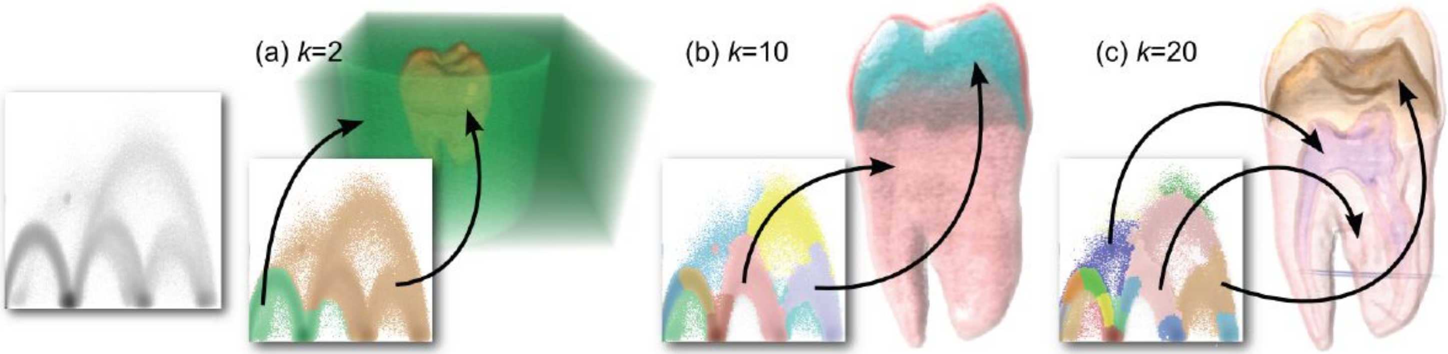
- We find the top *k* eigenvectors to approximate this minimum cut for *k* different segments

- [Yu and Shi *et al.*] shows how to find *k* partitions by finding *k* eigenvectors of the eigenvalue problem

‘Normalized Cut’ in image segmentation *(cont.…)*

Example : Normalized cut to segment 256 x 256 8 bit histogram images:-

When, k = 2 the tooth is separated from the volume box
k=10 shows segments of the tooth crown and root
k = 20 shows different material boundaries.

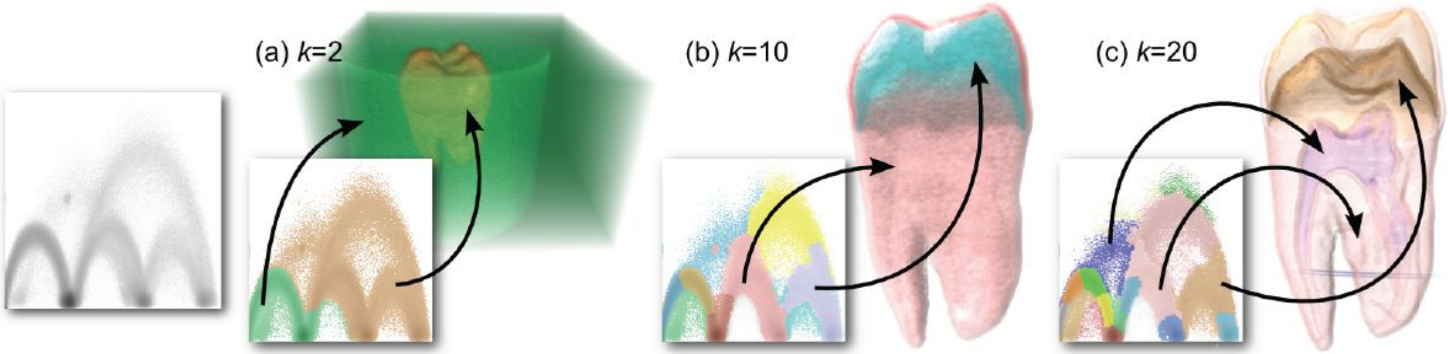


Determining the '*k*'

Example : Normalized cut to segment 256 x 256 8 bit histogram images:-

When, k = 2 the tooth is separated from the volume box
k=10 shows segments of the tooth crown and root
k = 20 shows different material boundaries.

So how many segments should we choose to compose visualization?



Determining the '*k*'

Iteratively test and pick k is time consuming and yet increasing k may not subdivide regions of interest.

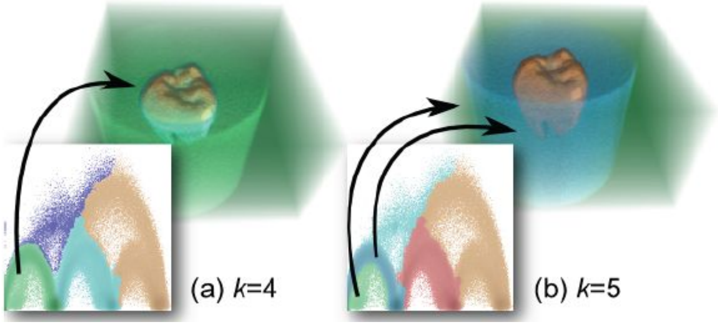
Try k = 2, 3, 4, …

Determining the '*k*'

Iteratively test and pick k is time consuming and yet increasing k may not subdivide regions of interest.

For example when k is increased from 4 to 5 , the box is subdivided. We believe users would prefer subdividing the tooth.

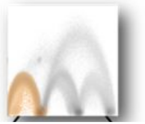
Try k = 2, 3, 4,
…

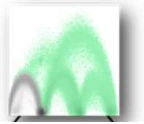


Determining the '*k*':
Replacing k with User-driven Exploration

- Allowing users to select the appropriate segments instead of choosing a specific *k*

- To accomplish this, a hierarchy is built by recursively applying the normalized cut.

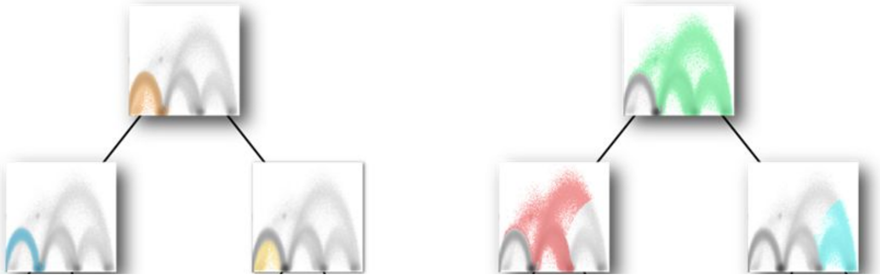




Determining the *'k'*:
Replacing k with User-driven Exploration

- Allowing users to select the appropriate segments instead of choosing a specific *k*

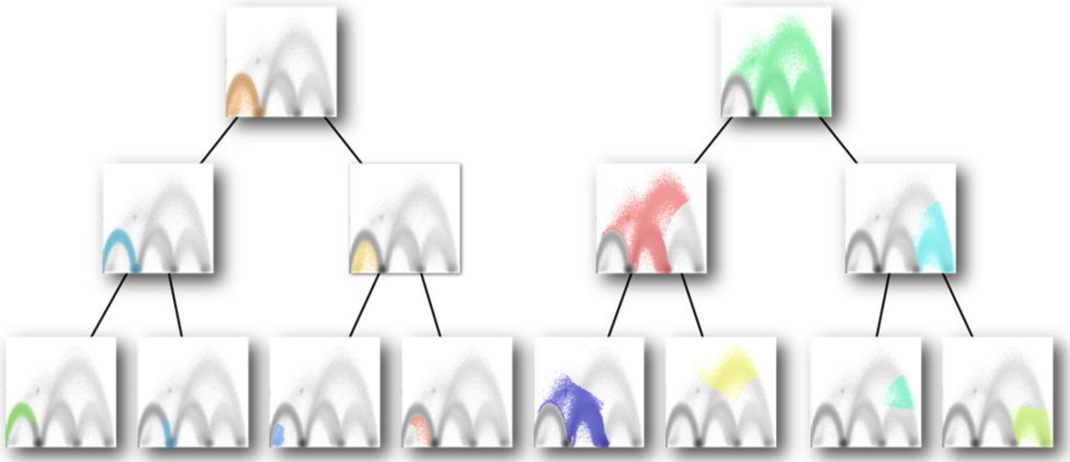
- To accomplish this, a hierarchy is built by recursively applying the normalized cut.



Determining the *'k'*:
Replacing k with User-driven Exploration

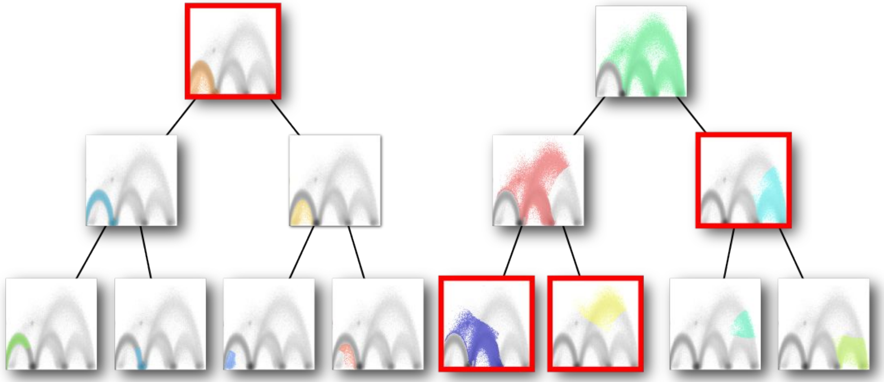
- Allowing users to select the appropriate segments instead of choosing a specific *k*

- To accomplish this, a hierarchy is built by recursively applying the normalized cut.



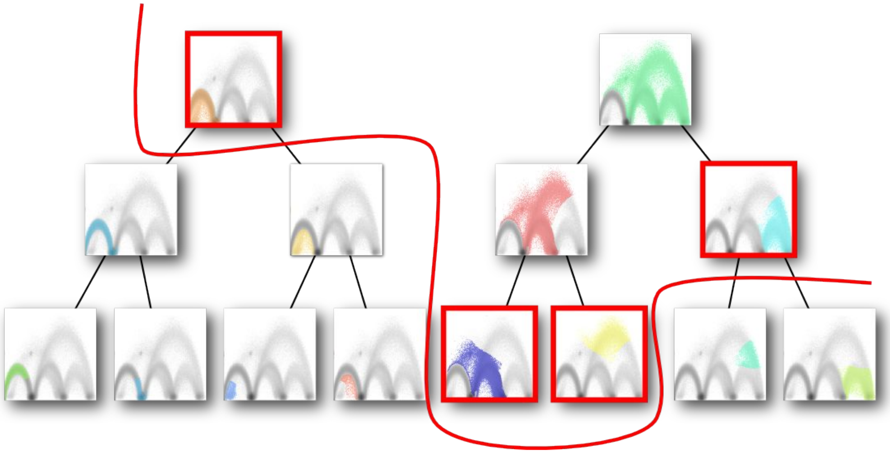
Multilevel Segmentation Hierarchy

- Hierarchy leads the users to traverse the dataset, selectively subdivide and inspect segments of their choice



Multilevel Segmentation Hierarchy

- Any cut along this hierarchy guarantees complete coverage.

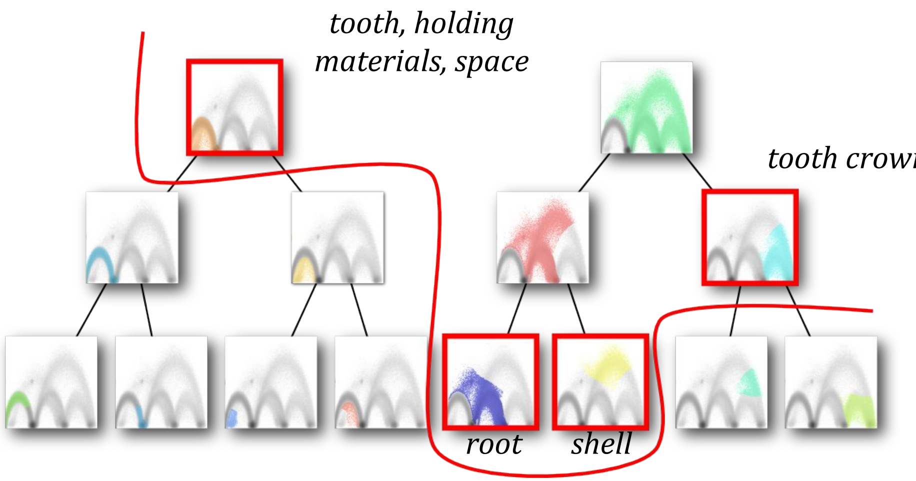


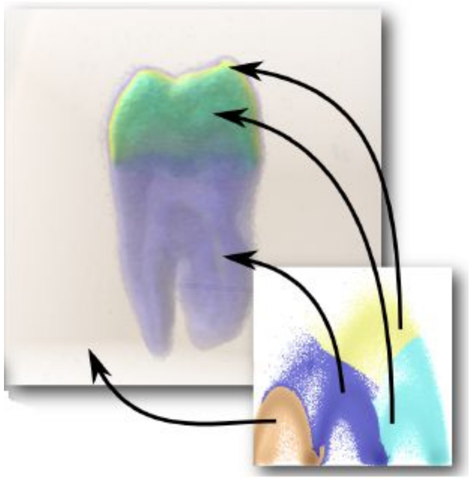
View-dependent Level of Detail hierarchies:

[Xia & Varshney Vis 96, Hoppe SIGGRAPH 97, Luebke & Erikson SIGGRAPH 97]

Multilevel Segmentation Hierarchy

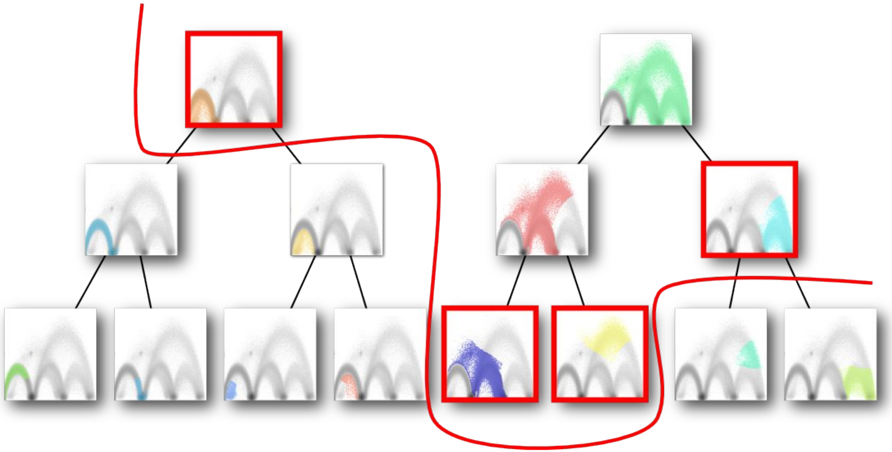
- Example: Assembling the red-framed segments produces the visualization on the right

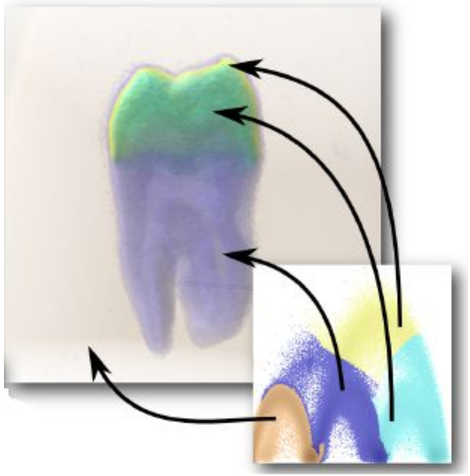




Multilevel Segmentation Hierarchy

- Example: Assembling the red-framed segments produces the visualization on the right





Next question → which segment should we subdivide?

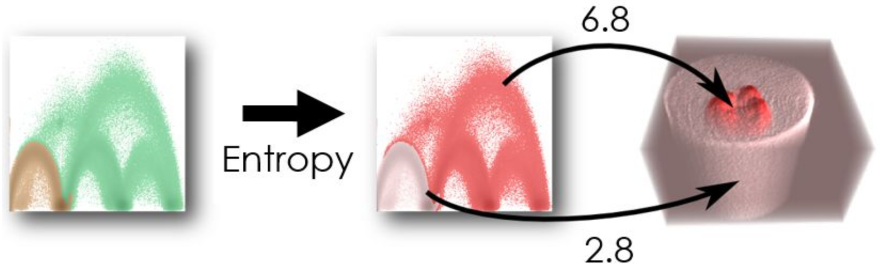
Information Guided Traversal

- Evaluating *information content* of the segments to provide some guidance
- Segment entropy: $$H(V) = -\sum_i p(v_i) \lg_2 p(v_i)$$
- where v_i is a voxel in V, $p(v_i)$ is the probability of v_i.
- $p(v_i)$ can be computed by analyzing how many voxels have the same intensity as v_i in V.

Information Guided Traversal

- Evaluating *information content* of the segments to provide some guidance
- Segment entropy: $H(V) = -\sum_i p(v_i) \lg_2 p(v_i)$
- High entropy $\rightarrow$ Complex segment

Example: the entropy of the tooth is higher than the volume box, suggesting the tooth is more complex, so tooth should be subdivided



(red is used to show which segment has a higher entropy)

Information Guided Traversal

The segment entropies can be similar

Which segment should we divide next?

Information Guided Traversal

The segment entropies can be similar

Which segment should we divide next?

- *Use Information Gain*

Information Guided Traversal

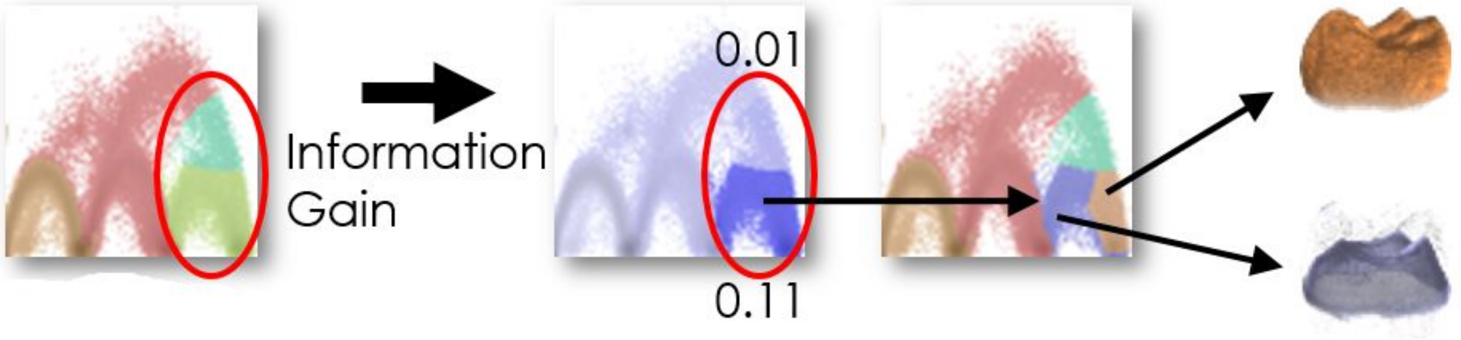
- Evaluating the *information gain* of a subdivision
- The information gain is defined as the entropy reduction of a subdivision

$$G(V) = H(V) - \sum_j \frac{|V_j|}{|V|}\left(\frac{H(V_j)}{H(V)}\right)$$

- Subdivisions with high information gain suggest separations of structures.

Information Guided Traversal

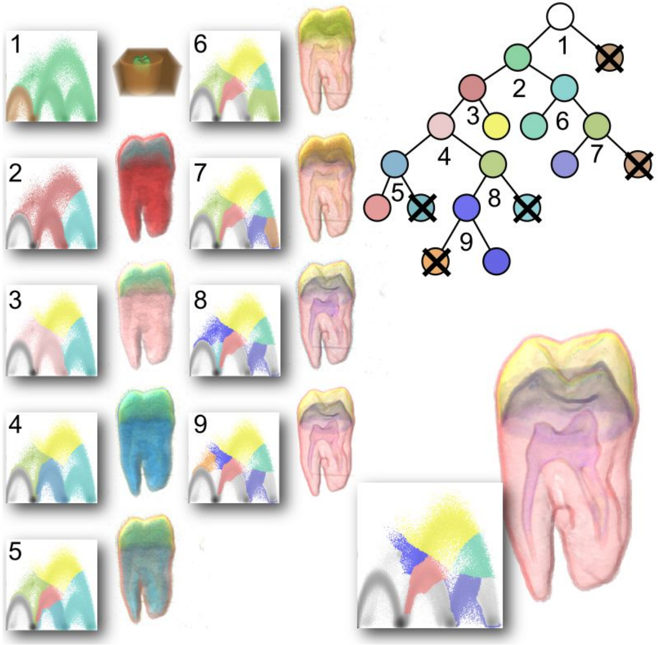
- Example: Splitting the segment with high information gain separates the surfaces of the *tooth crown* and *dentine*



(blue to represent the information gain)

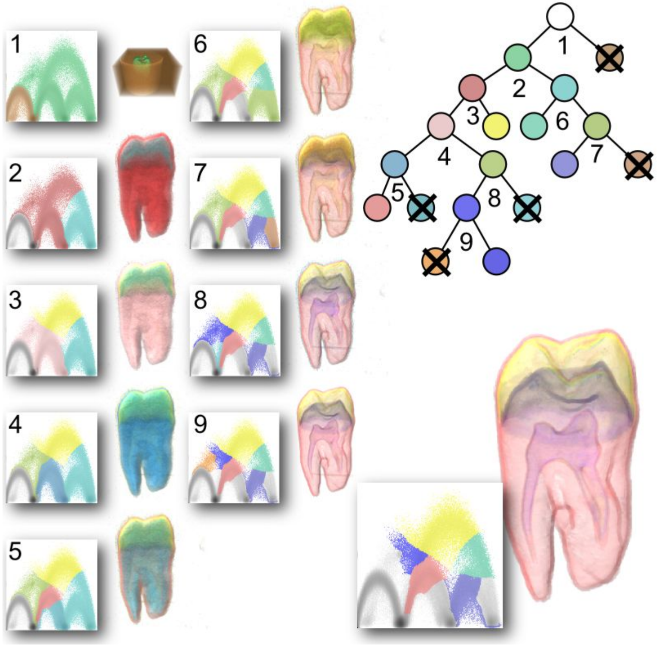
Interactive Exploration

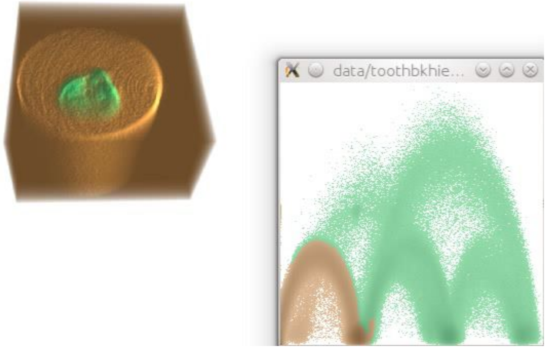
User selectively select the histogram to expand the segments and interactively compose visualizations



Interactive Exploration

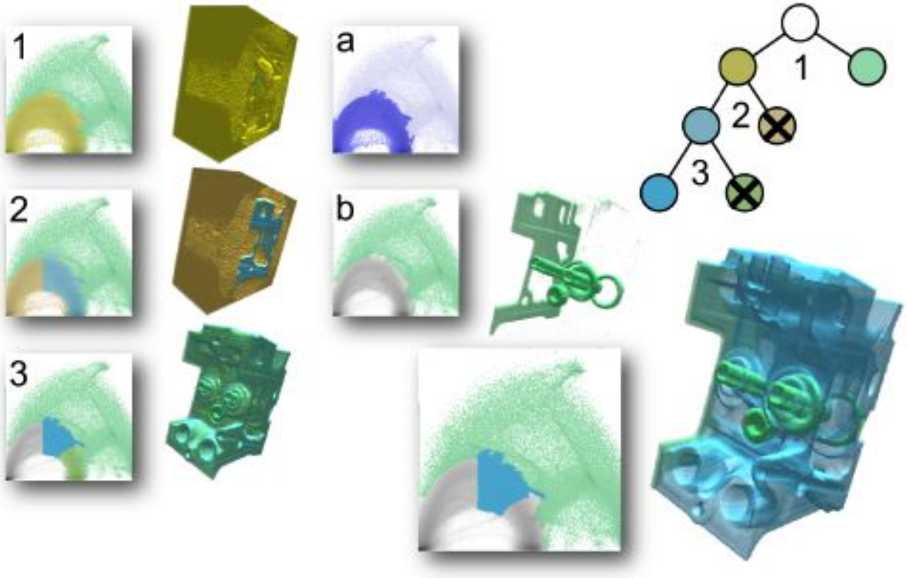
User selectively select the histogram to expand the segments and interactively compose visualizations



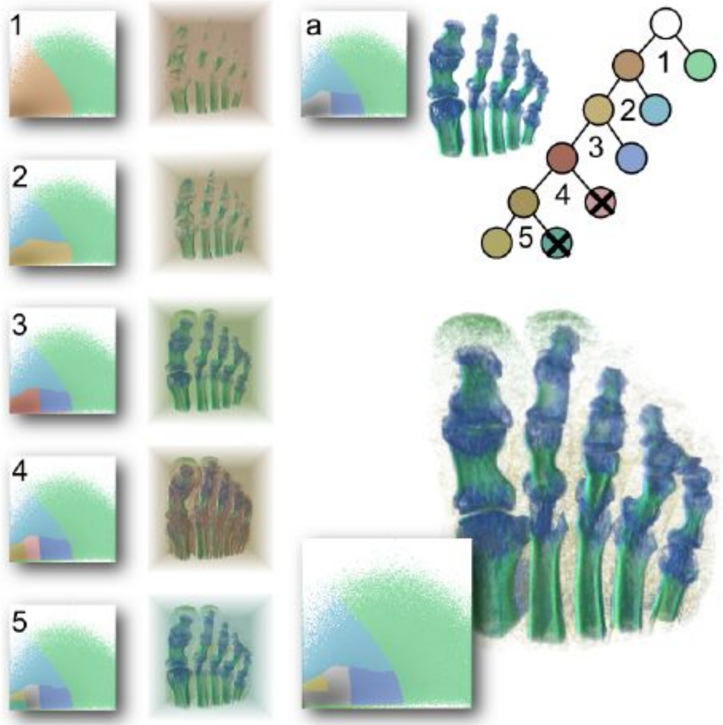


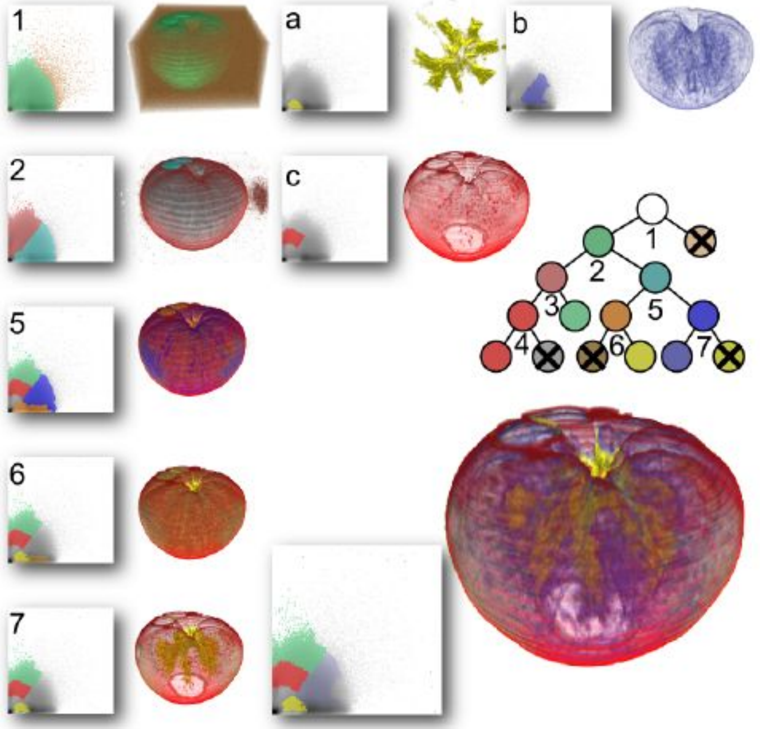
[Courtesy: Cheuk *et al.* VisWeek 2012]

Results



Results





Conclusion

- In this paper idea of using computational visual segmentation is used to effectively mimic user interaction
- It classifies intuitive volumetric regions
- An exhaustive multilevel hierarchy is built from these segments
- Information content is provided to guide the hierarchy traversal
- Users can interactively explore and visualize with these ingredients

Future Work

- Improve the information content measures
- Segment histograms with different attributes

References and resources

1. Presentation on theme: "Hierarchical Exploration of Volumes Using Multilevel Segmentation of the Intensity-Gradient Histograms Cheuk Yiu IpAmitabh VarshneyJoseph JaJa.“ VisWeel 2012 — Presentation transcript: http://slideplayer.us/slide/1704370/

2. G. L. Kindlmann and J.W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In IEEE Symposium on Volume Visualization, pages 79–86, 1998.

3. C. Correa and K.-L. Ma. Size-based transfer functions: A new volume exploration technique. IEEE Transactions on Visualization and Computer Graphics, 14(6):1380–1387, 2008.

4. P. Sereda, A. Bartroli, I. Serlie, and F. Gerritsen. Visualization of boundaries in volumetric data sets using LH histograms. IEEE Transactions on Visualization and Computer Graphics, 12(2):208–218, 2006.

5. M. Ruiz, A. Bardera, I. Boada, I. Viola, M. Feixas, and M. Sbert. Automatic transfer functions based on informational divergence. IEEE Transactions on Visualization and Computer Graphics, 17(12):1932–1941, 2011.

6. J. Shi and J. Malik. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):888–905, 2000.

Thank you all