

Surface Modeling and Display from Range and Color Data

Kari	Pulli	UW
Michael	Cohen	MSR
Tom	Duchamp	UW
Hugues	Hoppe	MSR
John	McDonald	UW
Linda	Shapiro	UW
Werner	Stuetzle	UW

UW = University of Washington
Seattle, WA USA
MSR = Microsoft Research
Redmond, WA USA

Introduction

Goal

- develop robust algorithms for constructing 3D models from range & color data
- use those models to produce realistic renderings of the scanned objects



Surface Reconstruction

Step 1: Data acquisition

Obtain range data that covers the object. Filter, remove background.

Step 2: Registration

Register the range maps into a common coordinate system.

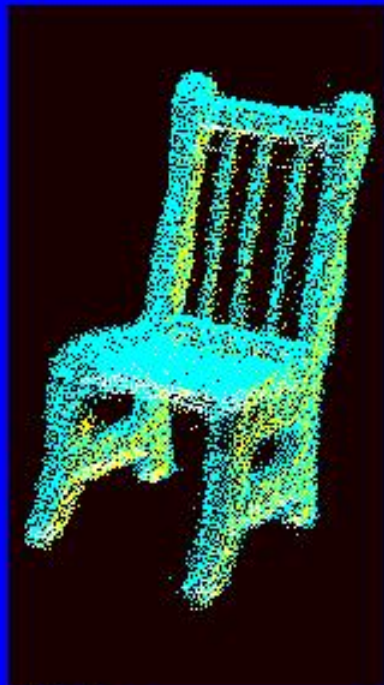
Step 3: Integration

Integrate the registered range data into a single surface representation.

Step 4: Optimization

Fit the surface more accurately to the data, simplify the representation.

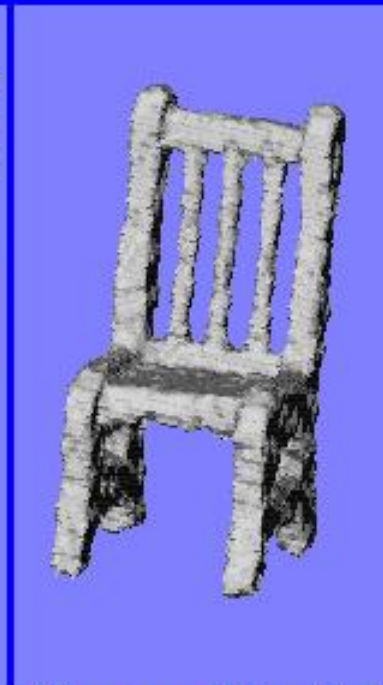
Problem



Noisy
registered
data

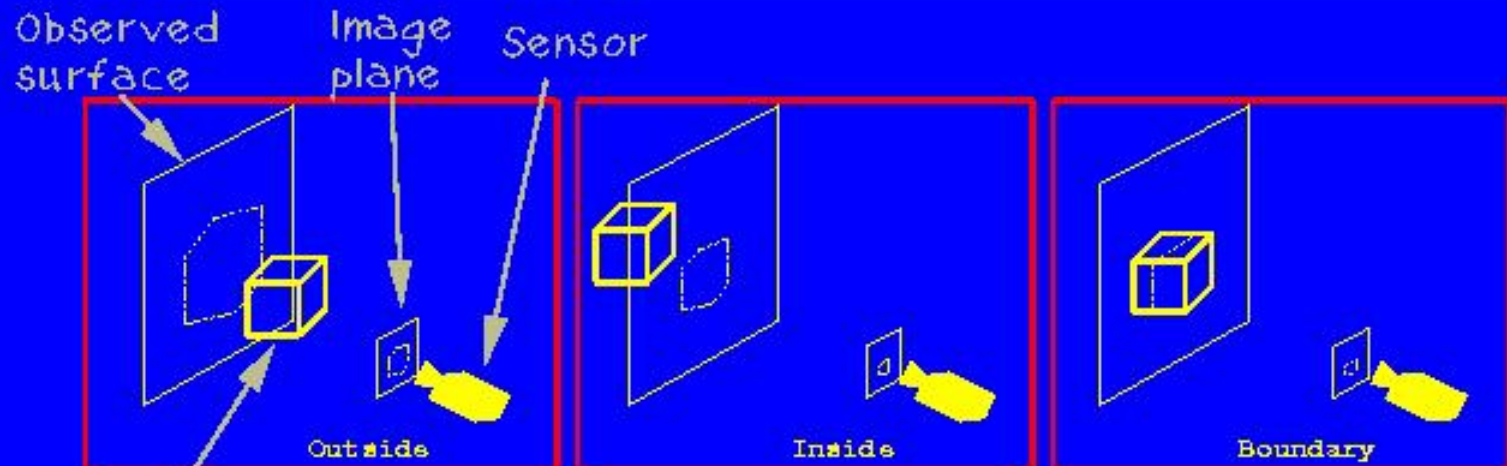


Signed
distance fn
& marching
cubes



Hierarchical &
directional
space carving

Carve space in cubes



Volume under consideration

Label cubes

- Project cube to image plane (hexagon)
- Test against data in the hexagon

Several views

Processing order:
FOR EACH cube
FOR EACH view

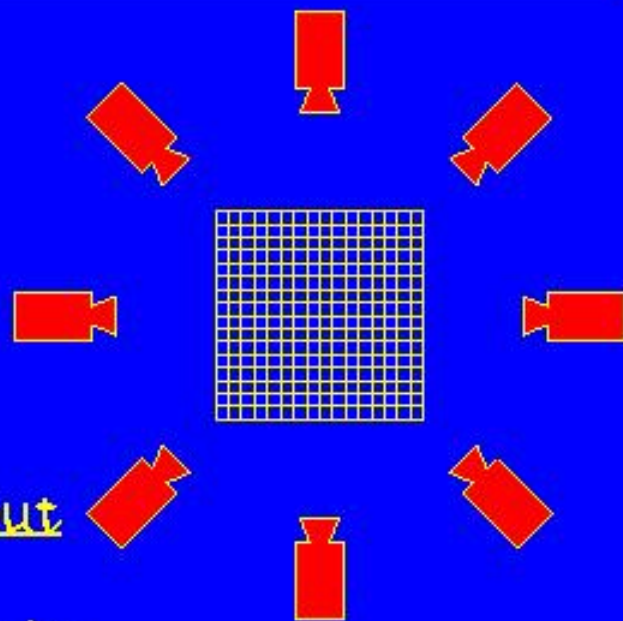
Rules:

any view thinks cube's out
=> it's out

every view thinks cube's in
=> it's in

else

=> it's at boundary

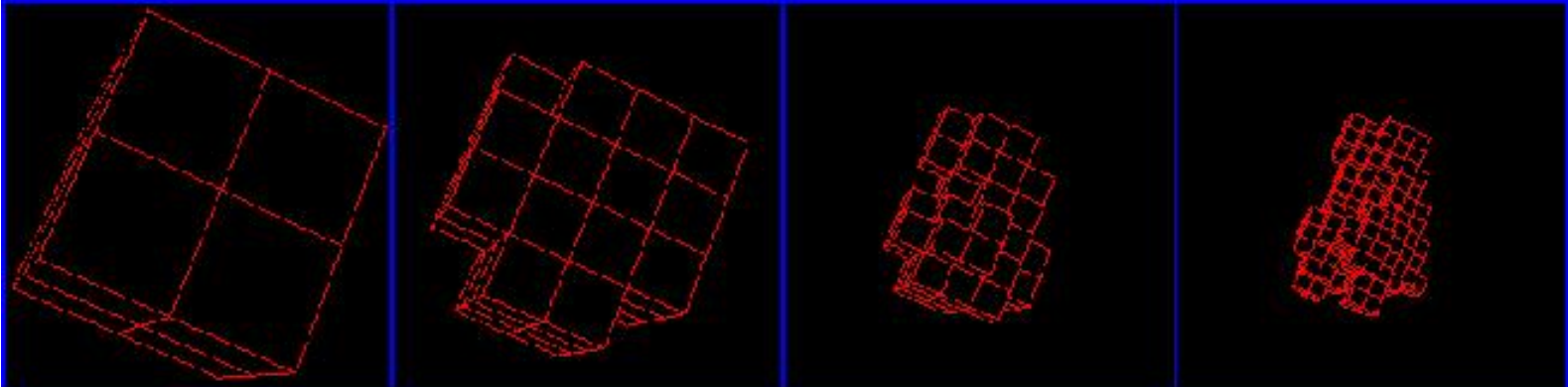


Hierarchical space carving

- Big cubes => fast, poor results
- Small cubes => slow, more accurate results
- Combination = octrees

RULES:

- cube's out => done
- cube's in => done
- else => recurse

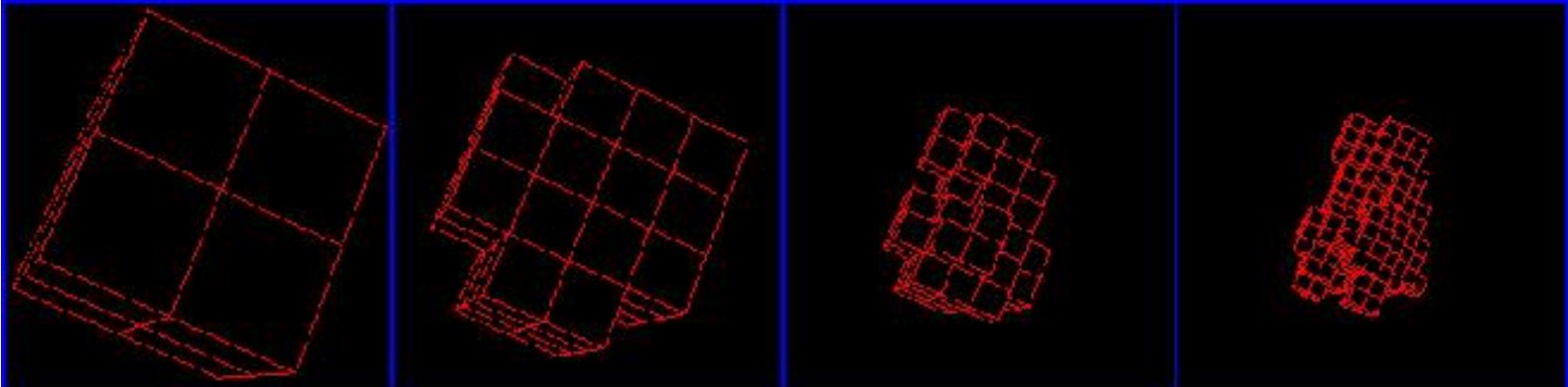


Hierarchical space carving

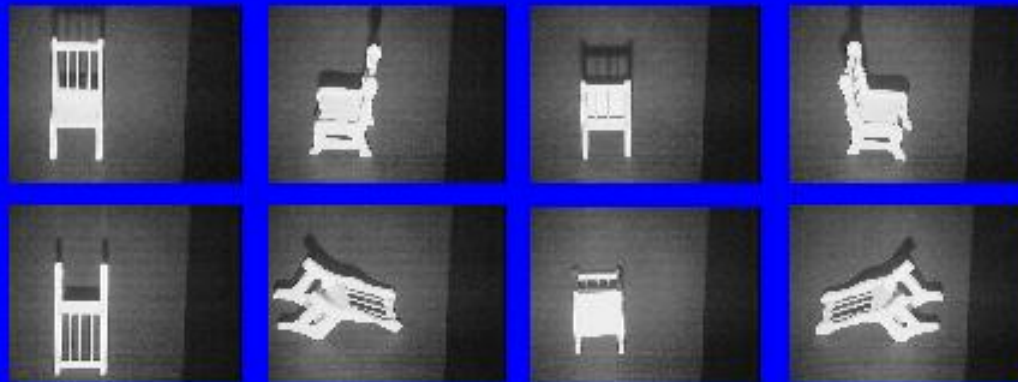
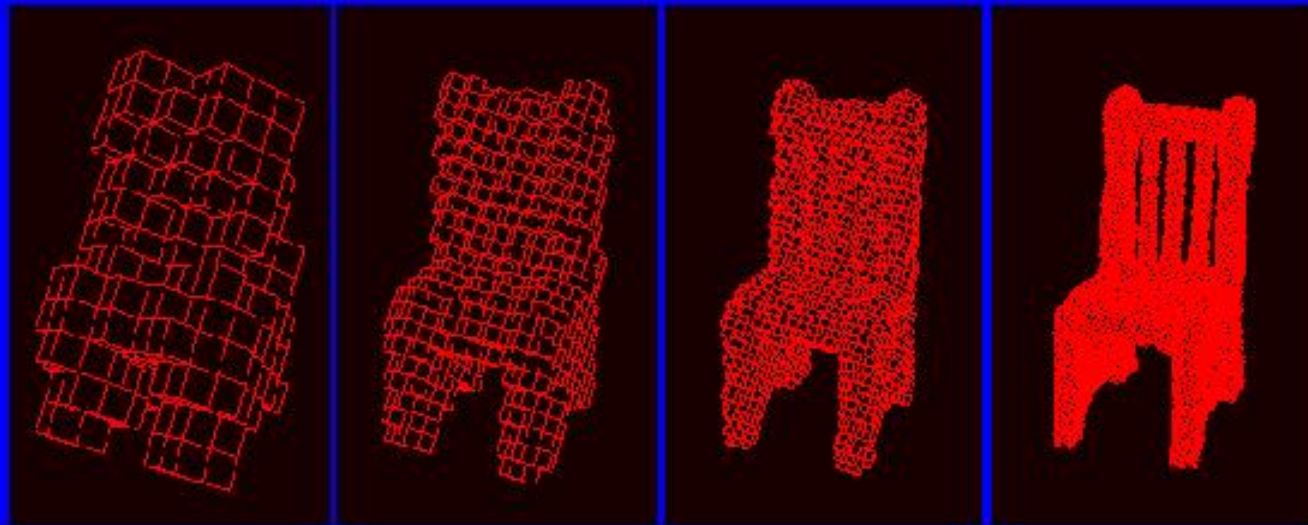
- Big cubes => fast, poor results
- Small cubes => slow, more accurate results
- Combination = octrees

RULES:

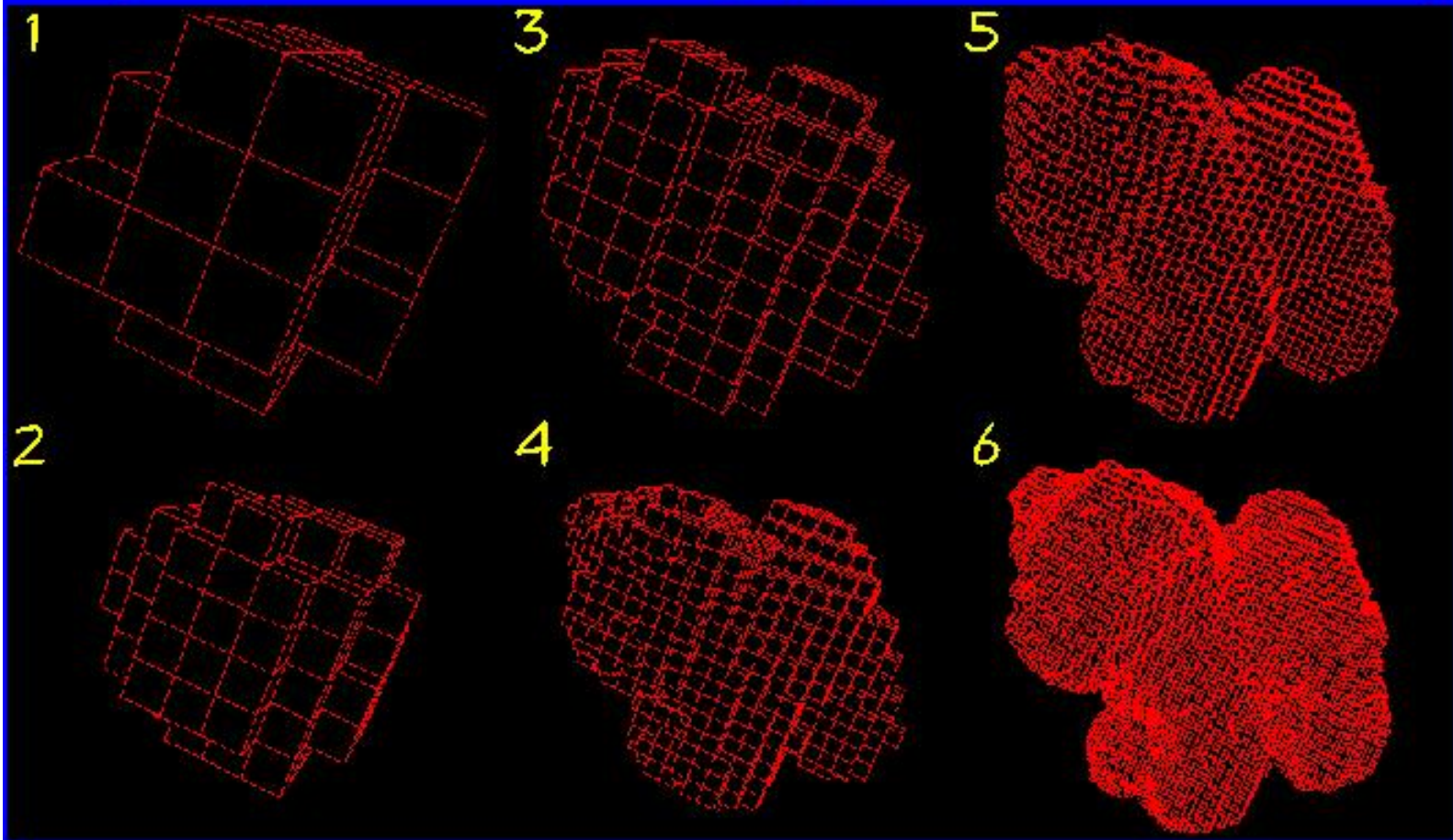
- cube's out => done
- cube's in => done
- else => recurse



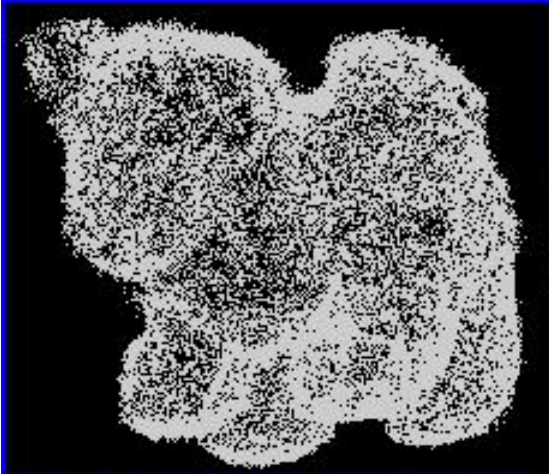
The rest of the chair



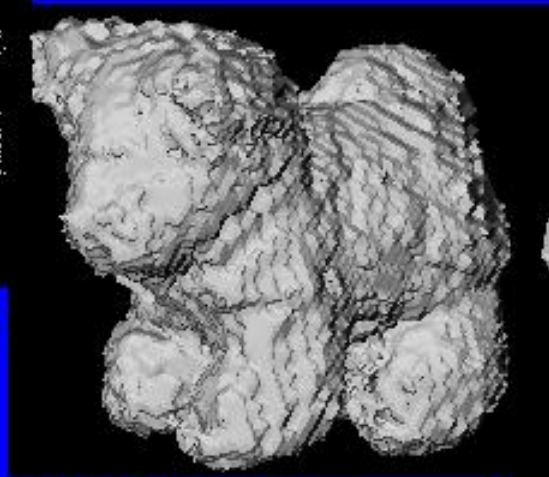
Same for a husky pup



Optimizing the dog mesh



Registered points



Initial mesh

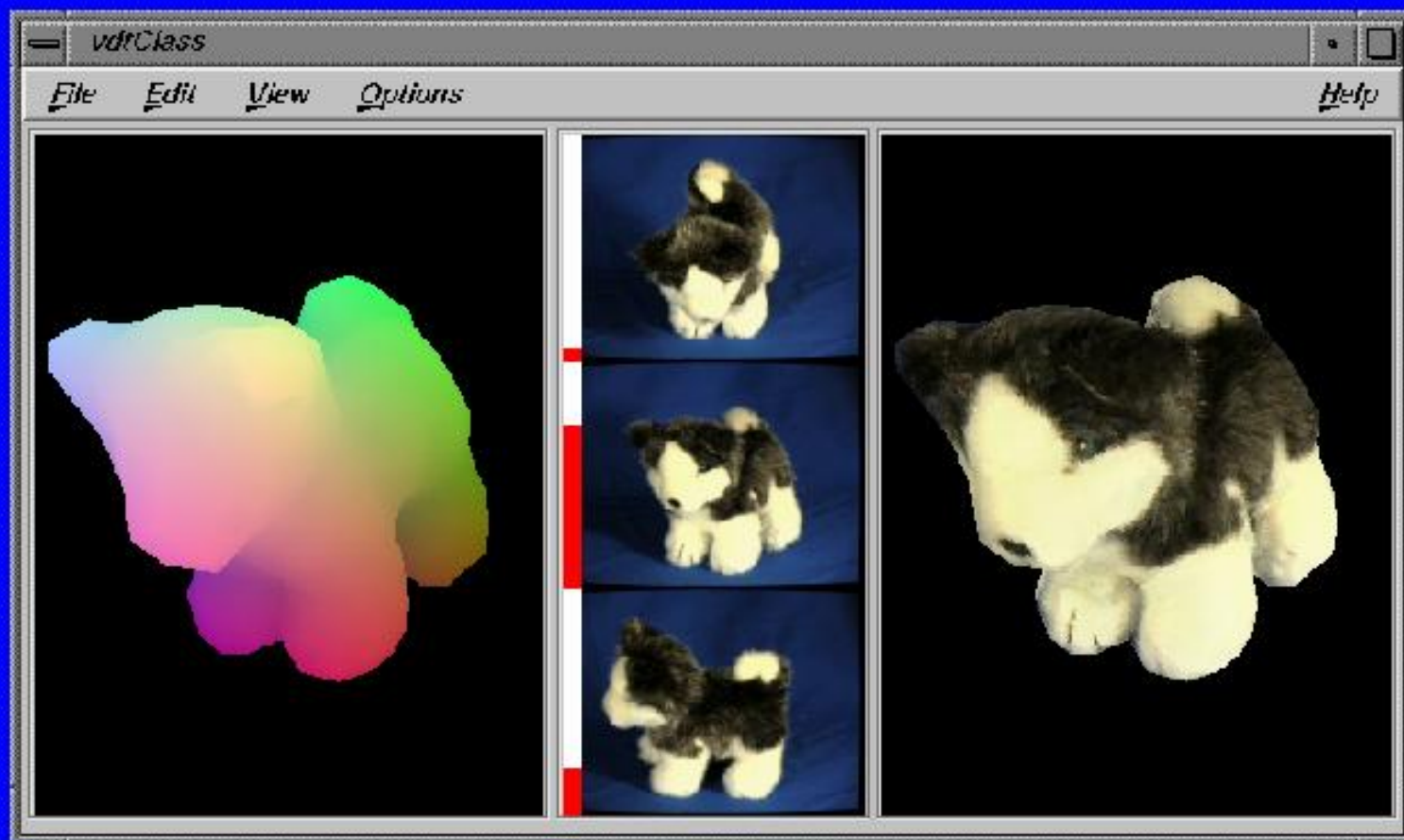


Optimized mesh

View dependent texturing



Our viewer



Overview of VBR

Choose 3 close views



Render meshes from the current viewpoint



For each pixel

- * read it from each view
- * remove occluded ones
- * calculate a weighted average
- * paint the pixel



Reconstruction of Blood Vessel Trees from Visible Human Data

Zhenrong Qian and Linda Shapiro
Computer Science & Engineering
Department
University of Washington

Introduction

- **Goal**
 - to reconstruct the blood vessels of the lungs from Visible Human Data
- **Computer vision**
 - semi-automation
 - low-level image processing
 - model construction

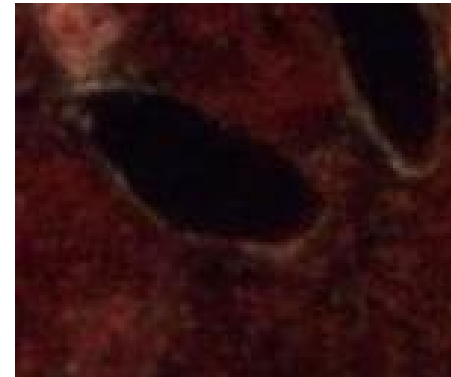
Problems Encountered

- **Data source**
 - black spots that are not blood vessels
 - variations of lighting
- **Characteristics of blood vessels**
 - similar color surrounds
 - lack of knowledge
 - close location
 - shape variety
 - continuous change not expected
 - dense data

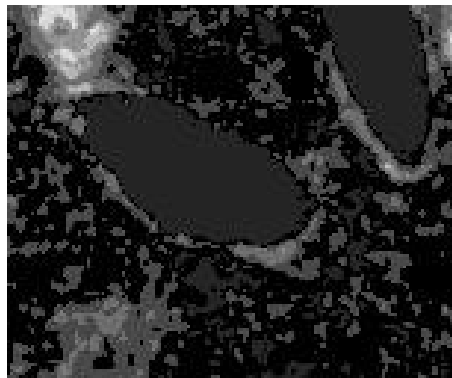
Finding the contours of a vessel being tracked (1)



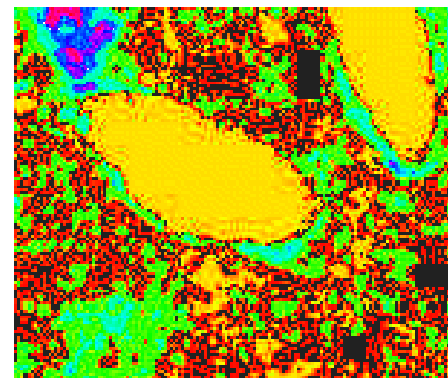
Previous contour



Current slice



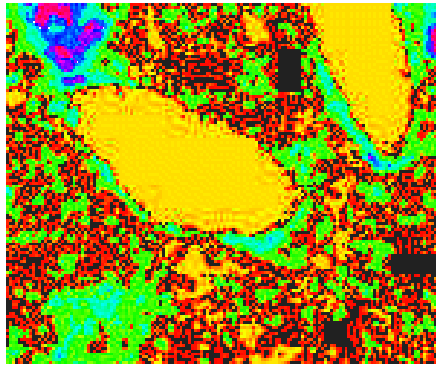
EM Segmentation



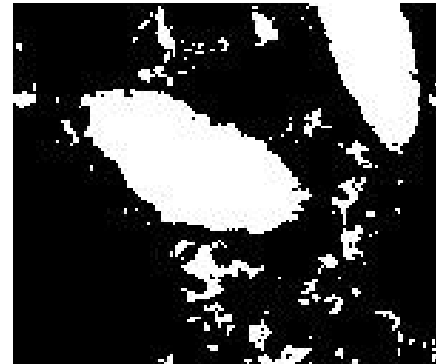
False color for the segmentation

Finding the contours of a vessel being tracked (2)

- The results after selecting regions of similar color to the tracked region



Segmentation result



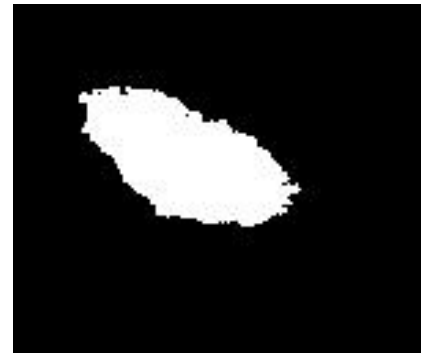
Selected regions

Finding the contours of a vessel being tracked (3)

- The results after selecting the region that overlaps most with the previous contour



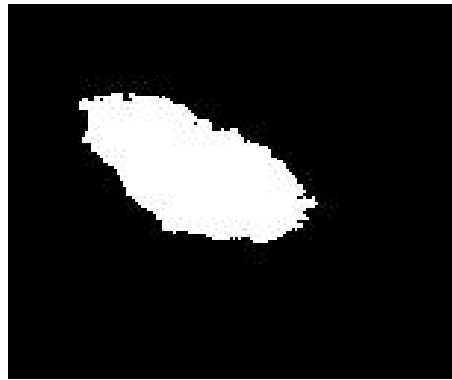
Selected regions



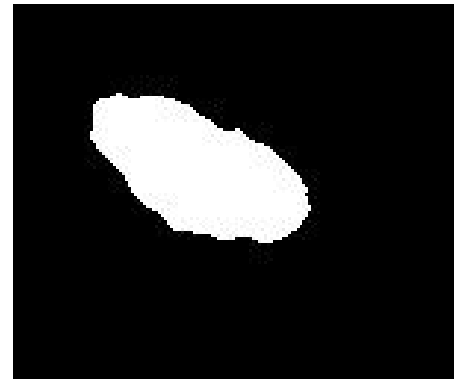
Region that overlaps
most

Find the contours of a vessel being tracked (4)

- The results after morphology to close holes and remove noise



Selected region



After noise removal

Find the contours of a vessel being tracked (5)

- The contour is determined through a fast-marching level-set approach



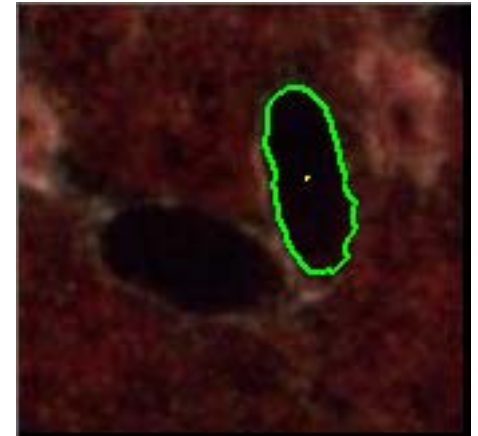
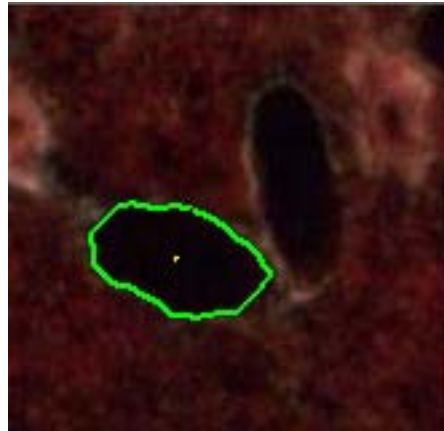
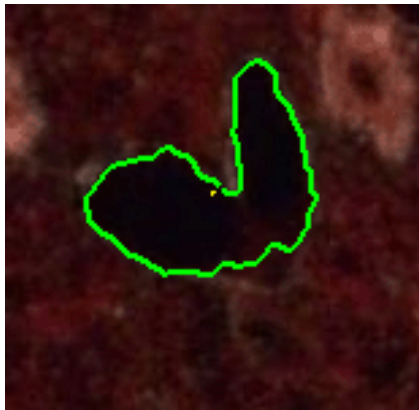
Previous contour



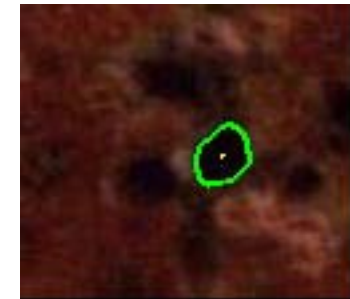
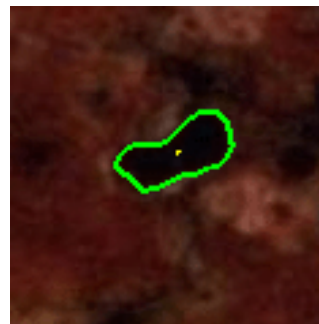
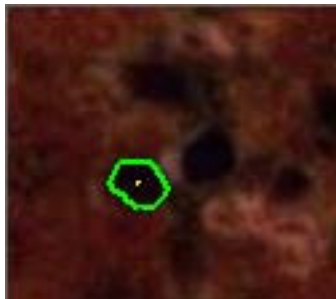
Current contour

How branching is handled

- One contour divides into two



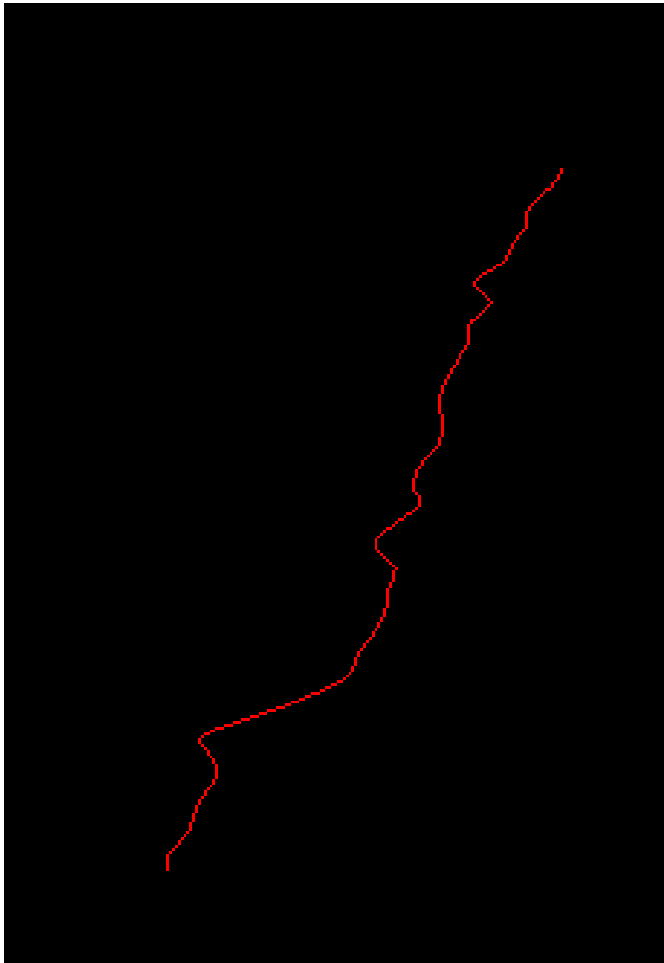
- Two contours merge into one



The use of resampling when the axis is not vertical

- **Track** the axis through the center points of found contours
- **Fit** a spline curve
- **Resample** the data perpendicular to the spline curve
- Use the resampled contours for model creation

Detect the axis

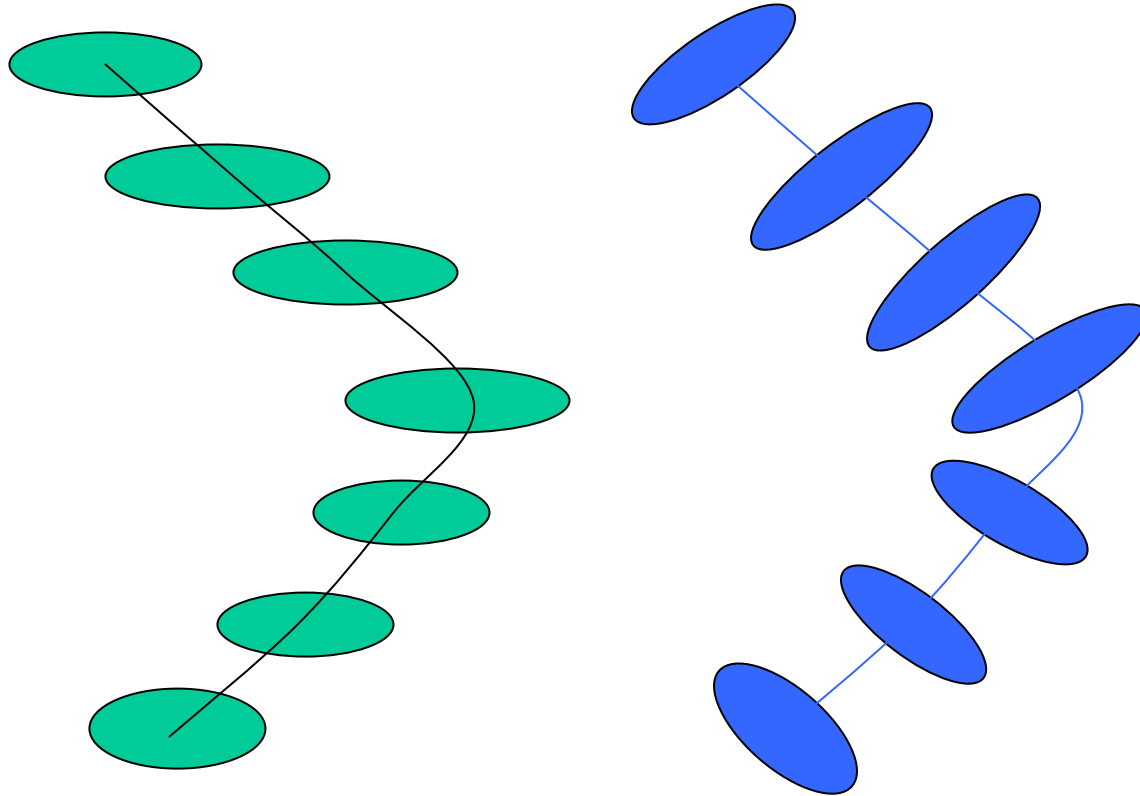


Center points of found contours



Spline-fitted axis

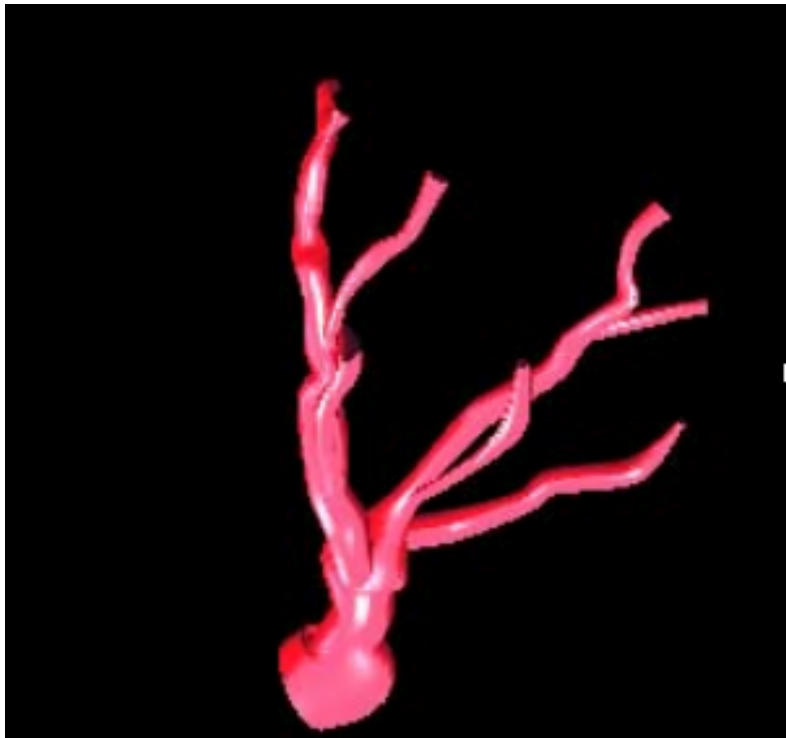
Resample the data perpendicular to the spline curve



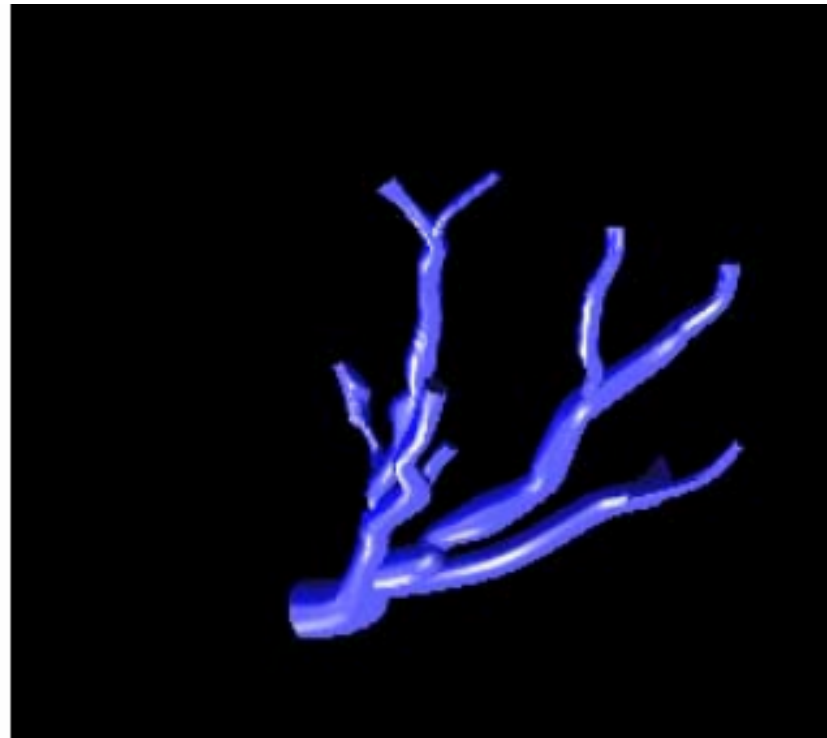
Overall Procedure for finding Vessel Trees

- The user **selects** a starting point
- The program automatically **tracks** the selected vessel and any branches it finds
- The program creates a **generalized cylinder** representation of the vessel tree
- The user may select more starting points

Some Initial Results



Artery tree from single seed



Vein tree from single seed

Results : blood vessels in right lung from previous section

