

# Interactive Cutting Operations for Generating Anatomical Illustrations from Volumetric Data Sets

Jörg Mensmann   Timo Ropinski   Klaus Hinrichs

Visualization and Computer Graphics Research Group  
University of Münster, Germany

WSCG 2008



- 1 Anatomical Illustrations
  - Motivation
  - Occlusion Handling
- 2 Deformation Approach
  - Requirements
  - Related Work
  - Technical Realization
- 3 Interactive Cutting and Deformation
- 4 Results
- 5 Conclusions



# Motivation

Going back in time to the 16th century ...

Going back in time to the 16th century ...

Andreas Vesalius: *De humani corporis fabrica*, 1543

Going back in time to the 16th century ...

Andreas Vesalius: *De humani corporis fabrica*, 1543

Fast forward 450 years . . .

Fast forward 450 years . . .

What has changed?

# Motivation

Fast forward 450 years . . .

What has changed? *Not that much!*

Fast forward 450 years . . .

What has changed? *Not that much!*

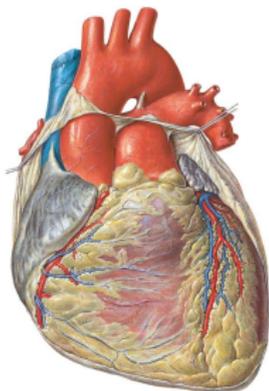
- ▶ Overall style similar
- ▶ Illustrations still mainly hand-drawn
- ▶ Not using data from imaging techniques like CT or MRI

# Motivation

Fast forward 450 years . . .

What has changed? *Not that much!*

- ▶ Overall style similar
- ▶ Illustrations still mainly hand-drawn
- ▶ Not using data from imaging techniques like CT or MRI

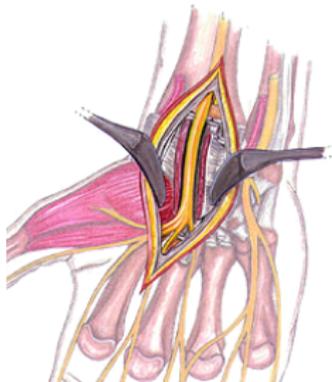
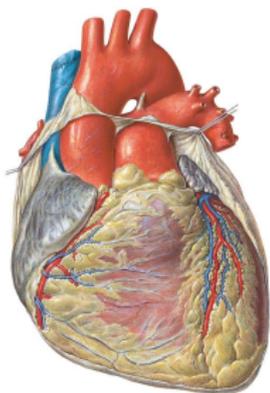


# Motivation

Fast forward 450 years . . .

What has changed? *Not that much!*

- ▶ Overall style similar
- ▶ Illustrations still mainly hand-drawn
- ▶ Not using data from imaging techniques like CT or MRI

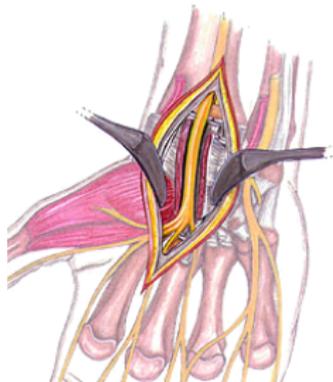
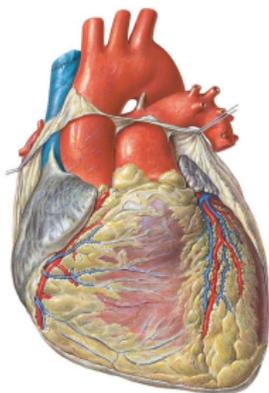


# Motivation

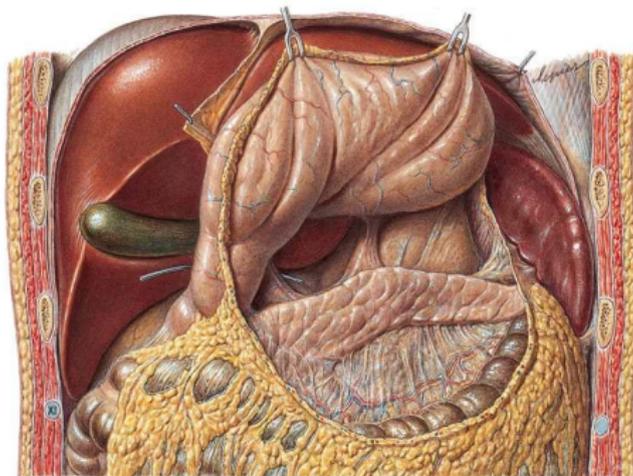
Fast forward 450 years . . .

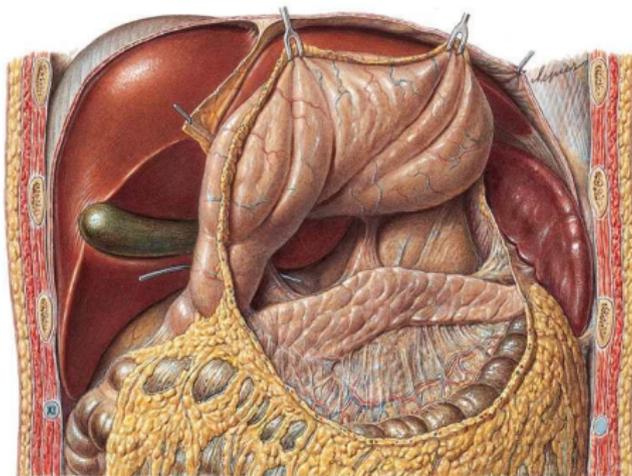
What has changed? *Not that much!*

- ▶ Overall style similar
- ▶ Illustrations still mainly hand-drawn
- ▶ Not using data from imaging techniques like CT or MRI



# Occlusion Handling





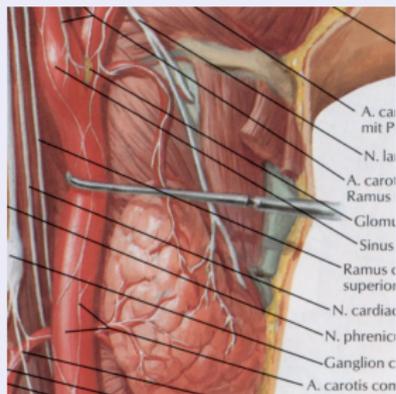
## Anatomical atlases use:

- ▶ cutaways ubiquitously
- ▶ no transparency or ghosting techniques
- ▶ deformations

# Types of Deformation and Related Cuts

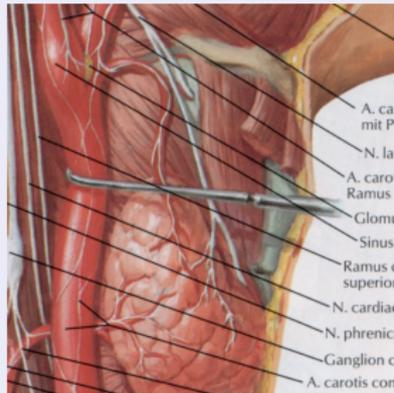
# Types of Deformation and Related Cuts

## Pulling

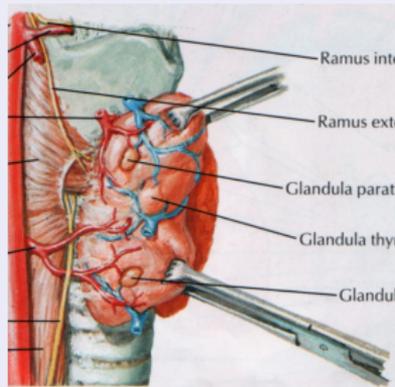


# Types of Deformation and Related Cuts

## Pulling

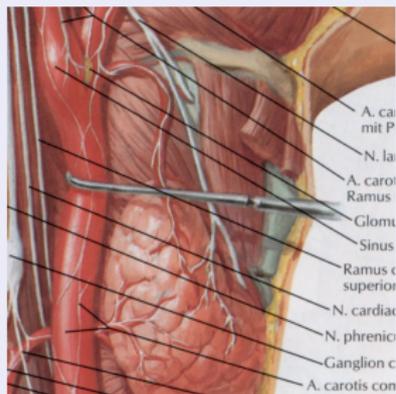


## Turning

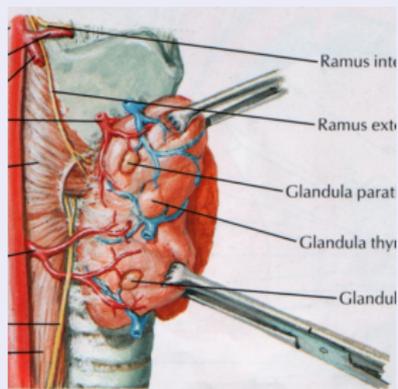


# Types of Deformation and Related Cuts

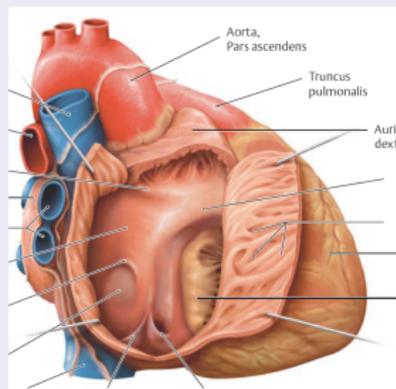
## Pulling



## Turning

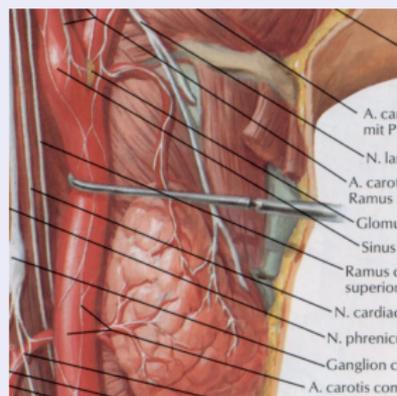


## Cut and flip open

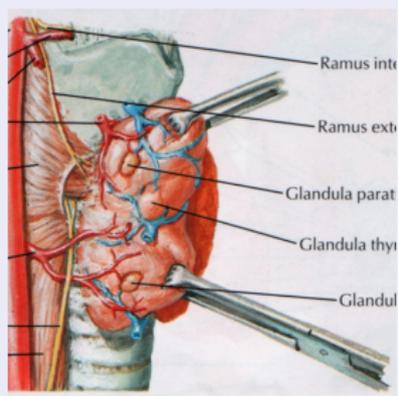


# Types of Deformation and Related Cuts

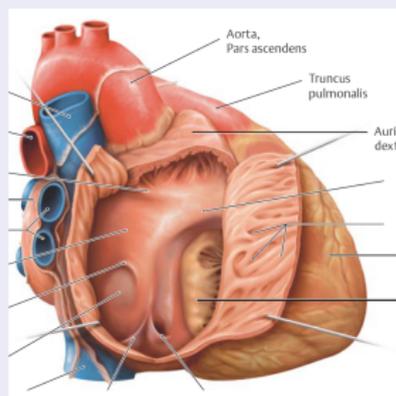
## Pulling



## Turning



## Cut and flip open



Source	Pull	Turn	Cut/Flip
<i>Netter</i>	19	6	6
<i>Sobotta</i>	33	7	23
<i>Prometheus</i>	22	16	15
<b>Total</b>	<b>74</b>	<b>29</b>	<b>44</b>



## Goals

- ▶ Create deformation illustrations directly from data generated by CT, MRI, PET, ...
- ▶ Support adaption to biological variance / non-typical cases

## Goals

- ▶ Create deformation illustrations directly from data generated by CT, MRI, PET, ...
- ▶ Support adaption to biological variance / non-typical cases

## Requirements for generating illustrative deformations

- ▶ Retain full data set resolution
- ▶ Interactive manipulation
- ▶ Physically comprehensible behaviour
- ▶ No preprocessing, no segmentation
- ▶ Optional support for transparency



## Geometry-based

- ▶ Free-Form-Deformation (Sederberg and Parry, 1986)
- ▶ Ray Deflectors (Kurzion and Yagel, 1997)
- ▶ Feature Aligned Volume Manipulation (Correa et al., 2006)
- Difficult to generate complex deformations

## Geometry-based

- ▶ Free-Form-Deformation (Sederberg and Parry, 1986)
- ▶ Ray Deflectors (Kurzion and Yagel, 1997)
- ▶ Feature Aligned Volume Manipulation (Correa et al., 2006)
- Difficult to generate complex deformations

## Physically-based

- ▶ Mass-Spring Systems
- ▶ Finite Element Method (FEM)
- Exact but slow

## Geometry-based

- ▶ Free-Form-Deformation (Sederberg and Parry, 1986)
- ▶ Ray Deflectors (Kurzion and Yagel, 1997)
- ▶ Feature Aligned Volume Manipulation (Correa et al., 2006)
- Difficult to generate complex deformations

## Physically-based

- ▶ Mass-Spring Systems
- ▶ Finite Element Method (FEM)
- Exact but slow

## Physically-inspired

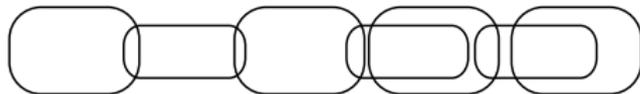
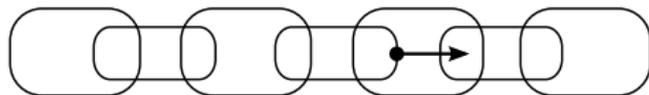
- ▶ 3D ChainMail algorithm (Gibson, 1997)
- Good compromise

# ChainMail Algorithm (Gibson, 1997)

- ▶ Linked volume representation
- ▶ Each volume element linked to its 6 neighbors
- ▶ Constraints limit relative movement
- ▶ Cutting by removing links
- ▶ Runtime proportional to number of deformed elements

# ChainMail Algorithm (Gibson, 1997)

- ▶ Linked volume representation
- ▶ Each volume element linked to its 6 neighbors
- ▶ Constraints limit relative movement
- ▶ Cutting by removing links
- ▶ Runtime proportional to number of deformed elements

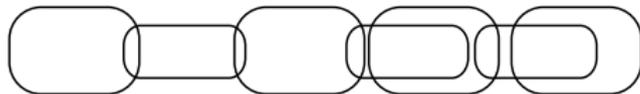


maximally  
stretched

maximally  
compressed

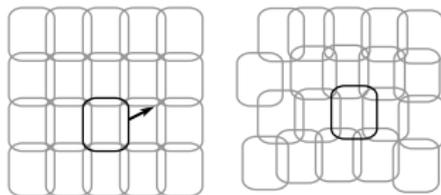
# ChainMail Algorithm (Gibson, 1997)

- ▶ Linked volume representation
- ▶ Each volume element linked to its 6 neighbors
- ▶ Constraints limit relative movement
- ▶ Cutting by removing links
- ▶ Runtime proportional to number of deformed elements



maximally stretched

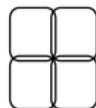
maximally compressed



relaxed



maximally compressed



maximally stretched

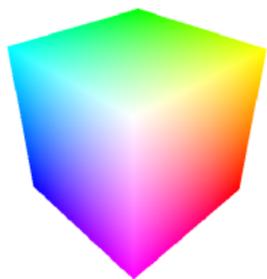


## Idea

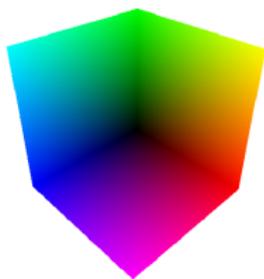
- ▶ Integrate into existing GPU ray-casting infrastructure
- ▶ Typically a box is used as proxy geometry
- ▶ Use surface extracted from ChainMail data structure instead

## Idea

- ▶ Integrate into existing GPU ray-casting infrastructure
- ▶ Typically a box is used as proxy geometry
- ▶ Use surface extracted from ChainMail data structure instead



entry

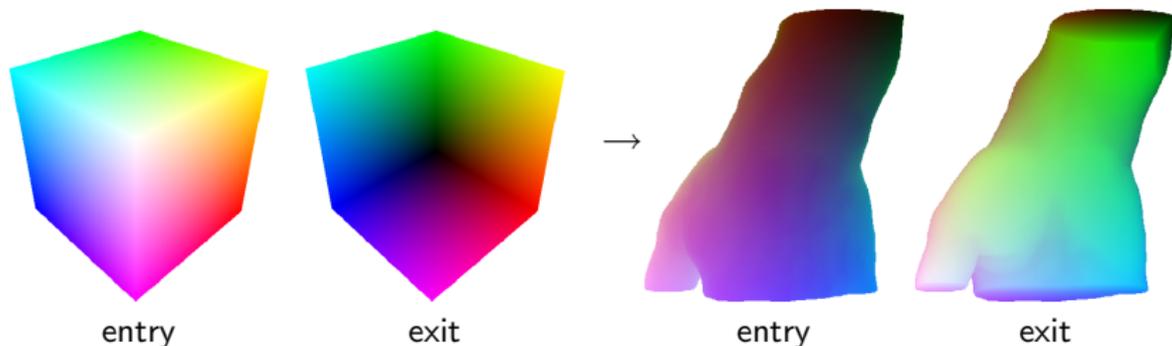


exit

Krüger/Westermann (2003)

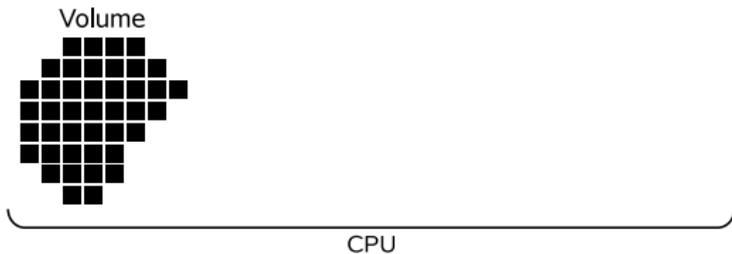
## Idea

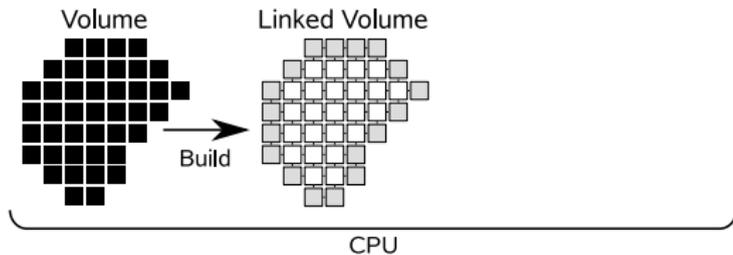
- ▶ Integrate into existing GPU ray-casting infrastructure
- ▶ Typically a box is used as proxy geometry
- ▶ Use surface extracted from ChainMail data structure instead

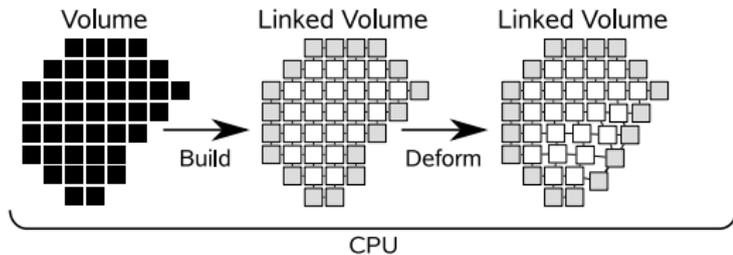


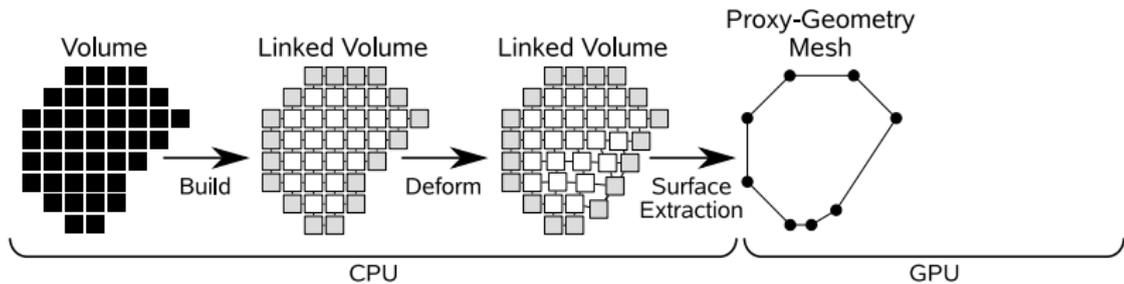
Krüger/Westermann (2003)

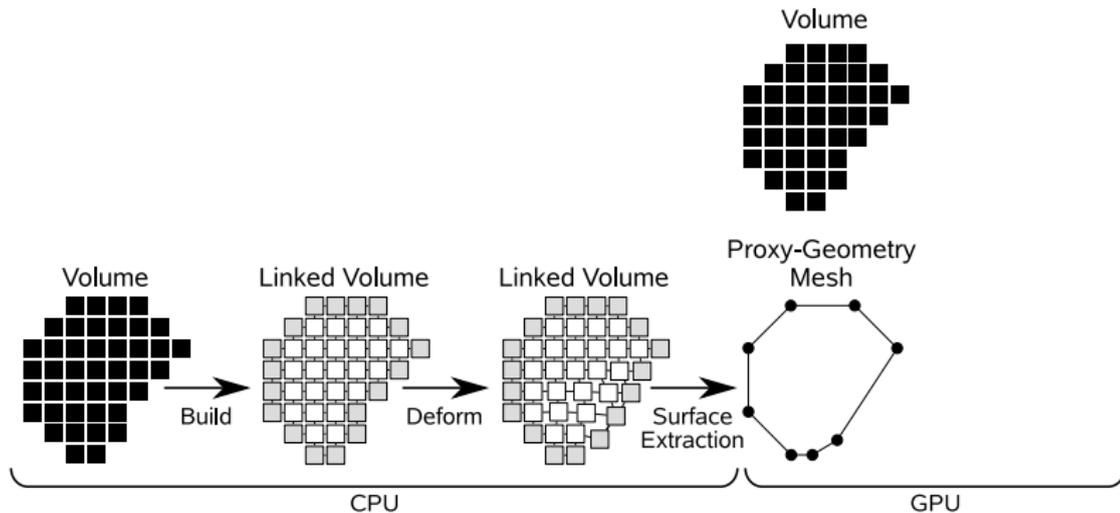




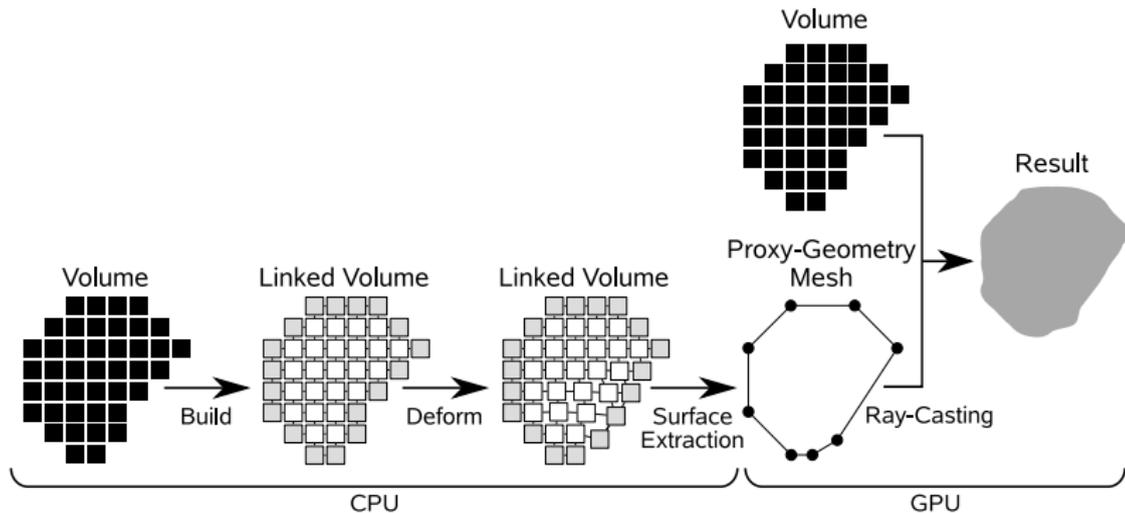








# Technical Realization



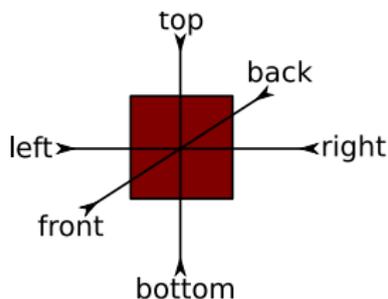


## Algorithm

- ▶ Treat ChainMail structure as rectilinear grid despite deformation
- ▶ Use surface information
- ▶ Successively look at structure from 6 principal directions
- ▶ Find cycles of 4 connected elements on surface → create quad

## Algorithm

- ▶ Treat ChainMail structure as rectilinear grid despite deformation
- ▶ Use surface information
- ▶ Successively look at structure from 6 principal directions
- ▶ Find cycles of 4 connected elements on surface → create quad



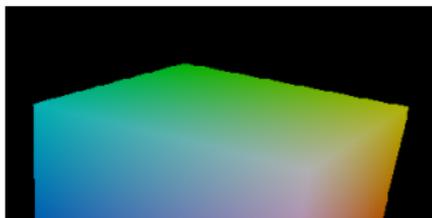
# Entry and Exit Points (EEP)

# Entry and Exit Points (EEP)

- ▶ Usually: vertex color encodes vertex position
- ▶ After deformation: vertex color encodes original voxel position
- ▶ Use depth buffer to extract min/max z-value  $\Rightarrow$  entry/exit

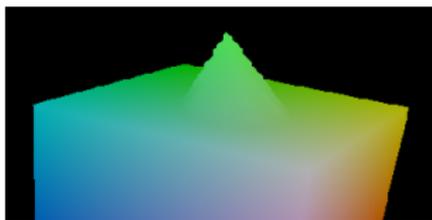
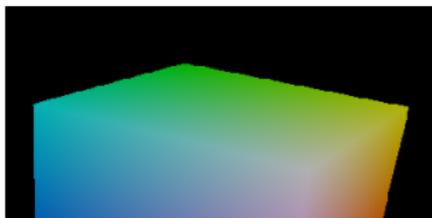
# Entry and Exit Points (EEP)

- ▶ Usually: vertex color encodes vertex position
- ▶ After deformation: vertex color encodes original voxel position
- ▶ Use depth buffer to extract min/max z-value  $\Rightarrow$  entry/exit



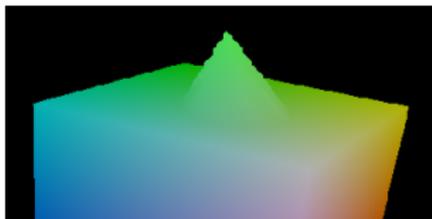
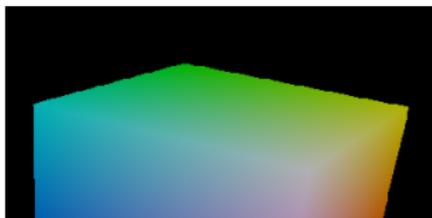
# Entry and Exit Points (EEP)

- ▶ Usually: vertex color encodes vertex position
- ▶ After deformation: vertex color encodes original voxel position
- ▶ Use depth buffer to extract min/max z-value  $\Rightarrow$  entry/exit



# Entry and Exit Points (EEP)

- ▶ Usually: vertex color encodes vertex position
- ▶ After deformation: vertex color encodes original voxel position
- ▶ Use depth buffer to extract min/max z-value  $\Rightarrow$  entry/exit





## Problem: Incorrect shading after deformation

- ▶ Normal vectors derived from voxel gradients used in Phong lighting
- ▶ Deformation not considered in gradient calculation

## Problem: Incorrect shading after deformation

- ▶ Normal vectors derived from voxel gradients used in Phong lighting
- ▶ Deformation not considered in gradient calculation

## Solution: Image-space normal estimation

- ▶ Calculate normals from first-hit image
- ▶ Use averaged forward differences to reduce noise

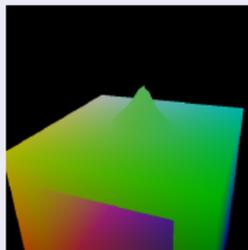
## Problem: Incorrect shading after deformation

- ▶ Normal vectors derived from voxel gradients used in Phong lighting
- ▶ Deformation not considered in gradient calculation

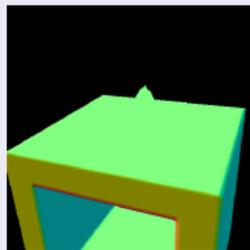
## Solution: Image-space normal estimation

- ▶ Calculate normals from first-hit image
- ▶ Use averaged forward differences to reduce noise

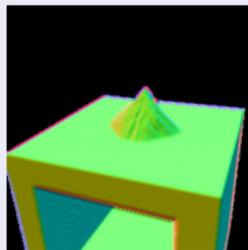
## Normal estimation example



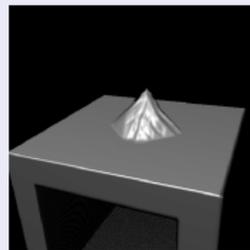
first-hit



gradient  
normals



estimated  
normals



shaded

## Two interaction concepts

- ▶ Cutting
- ▶ Deforming the cut object

## Two interaction concepts

- ▶ Cutting
- ▶ Deforming the cut object

## Common problem

- ▶ Interaction device DoF does not match scene DoF
- ▶ Requires changes of perspective to verify position

## Two interaction concepts

- ▶ Cutting
- ▶ Deforming the cut object

## Common problem

- ▶ Interaction device DoF does not match scene DoF
- ▶ Requires changes of perspective to verify position

## Cutting templates

- ▶ Only few different cuts actually used
- ▶ Provide small set of predefined cuts
- ▶ Placed freely in the data set

## Cutting templates

- ▶ Only few different cuts actually used
- ▶ Provide small set of predefined cuts
- ▶ Placed freely in the data set



single



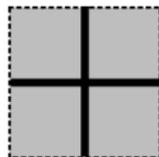
L-form



U-form



Y-form



crosshair

## Alternative: Surface placement

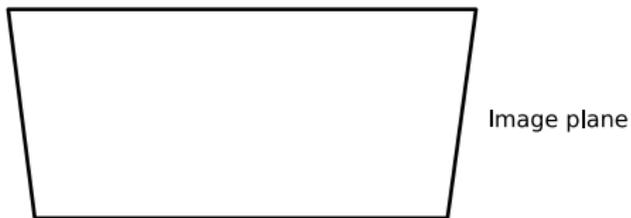
- ▶ Cutting templates fitted to surface
- ▶ Extruded: User controls cutting depth

## Alternative: Surface placement

- ▶ Cutting templates fitted to surface
- ▶ Extruded: User controls cutting depth

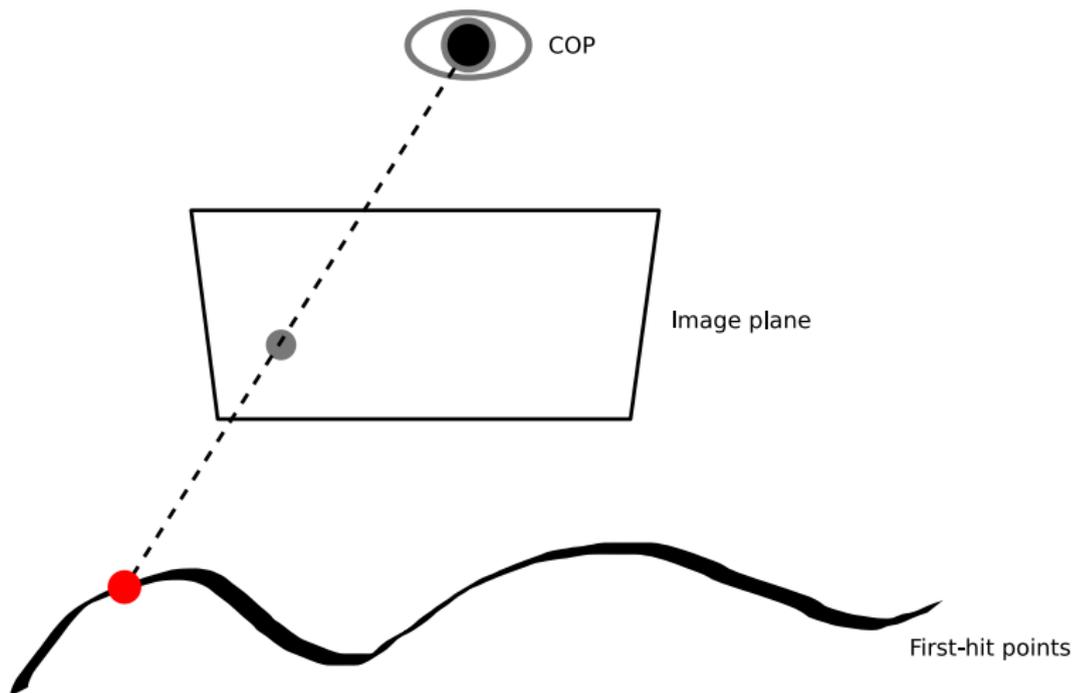
# Interactive Cutting: Surface Placement

Input: First-hit points, first-hit normals



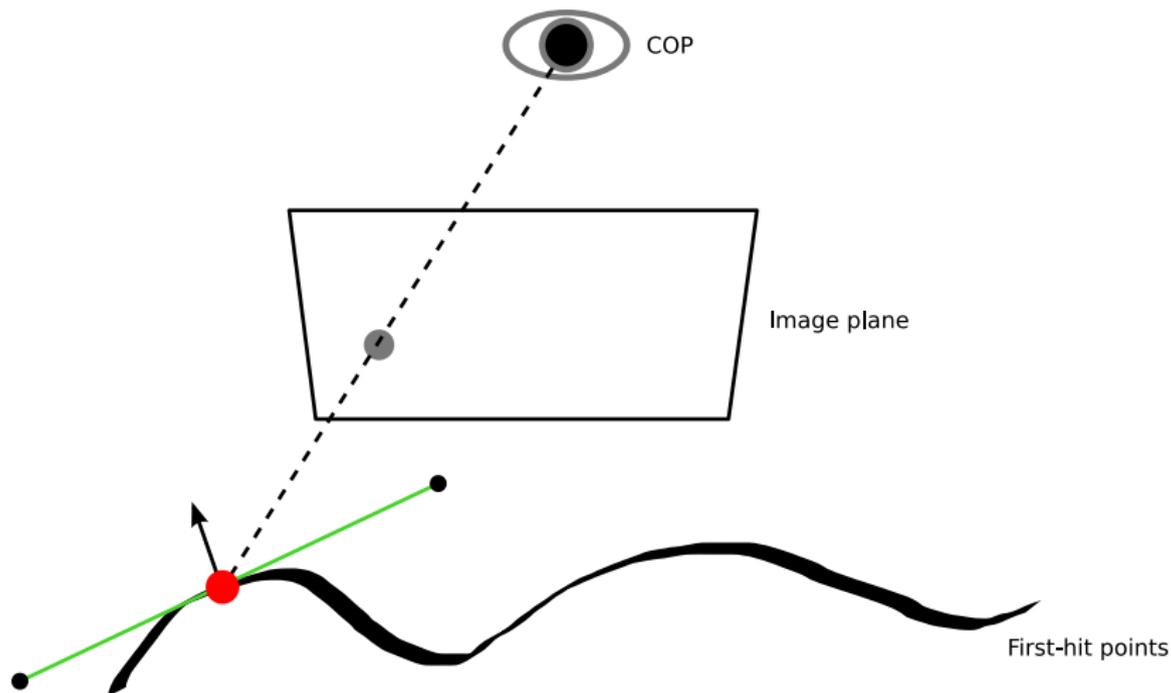
# Interactive Cutting: Surface Placement

Input: First-hit points, first-hit normals



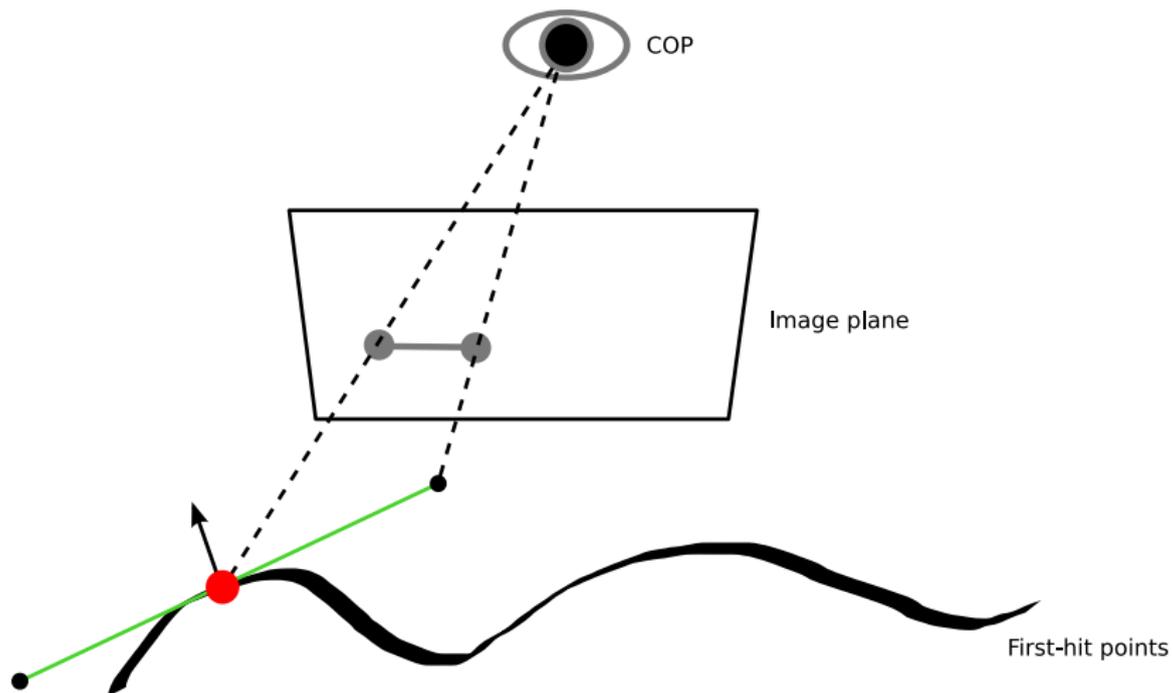
# Interactive Cutting: Surface Placement

Input: First-hit points, first-hit normals



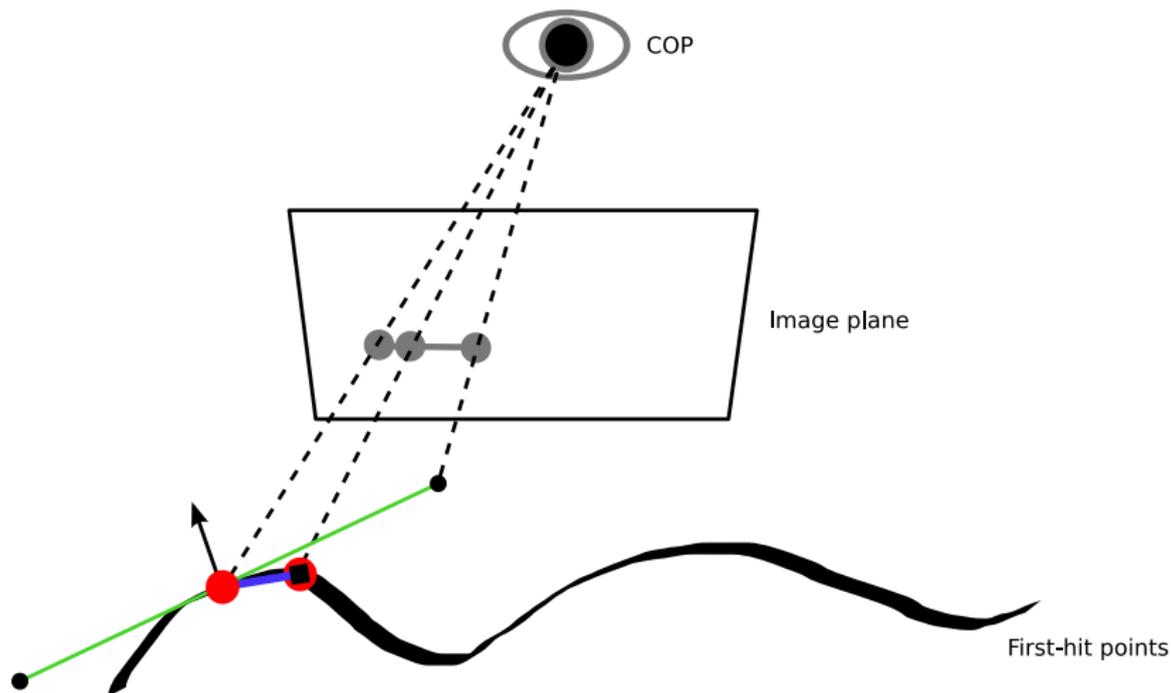
# Interactive Cutting: Surface Placement

Input: First-hit points, first-hit normals



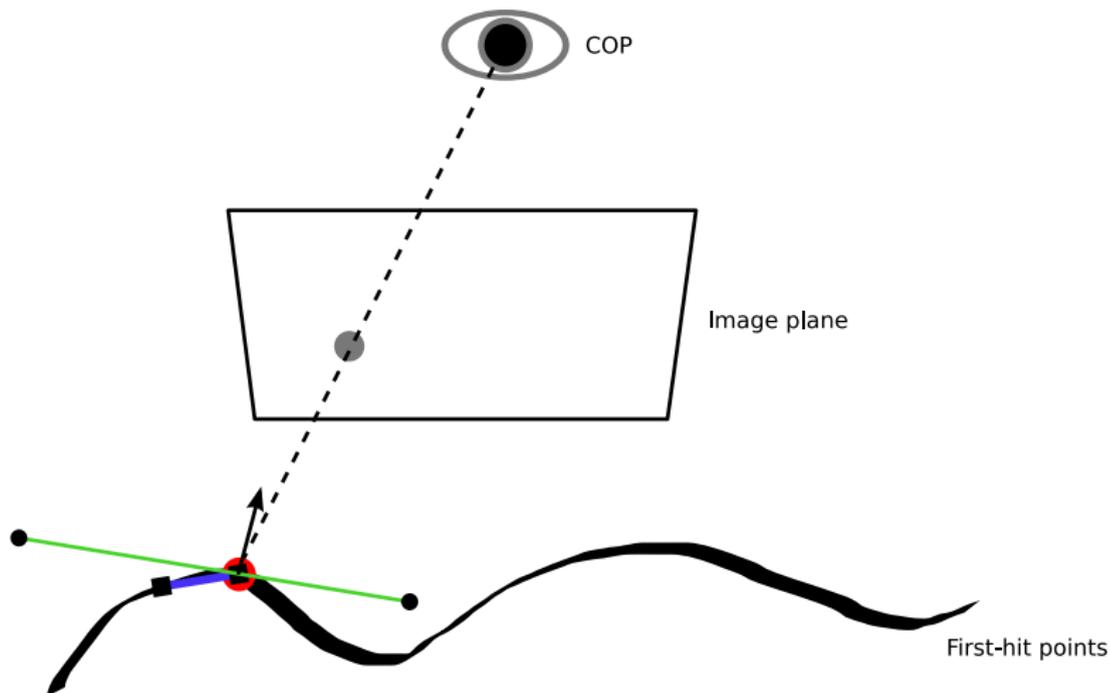
# Interactive Cutting: Surface Placement

Input: First-hit points, first-hit normals



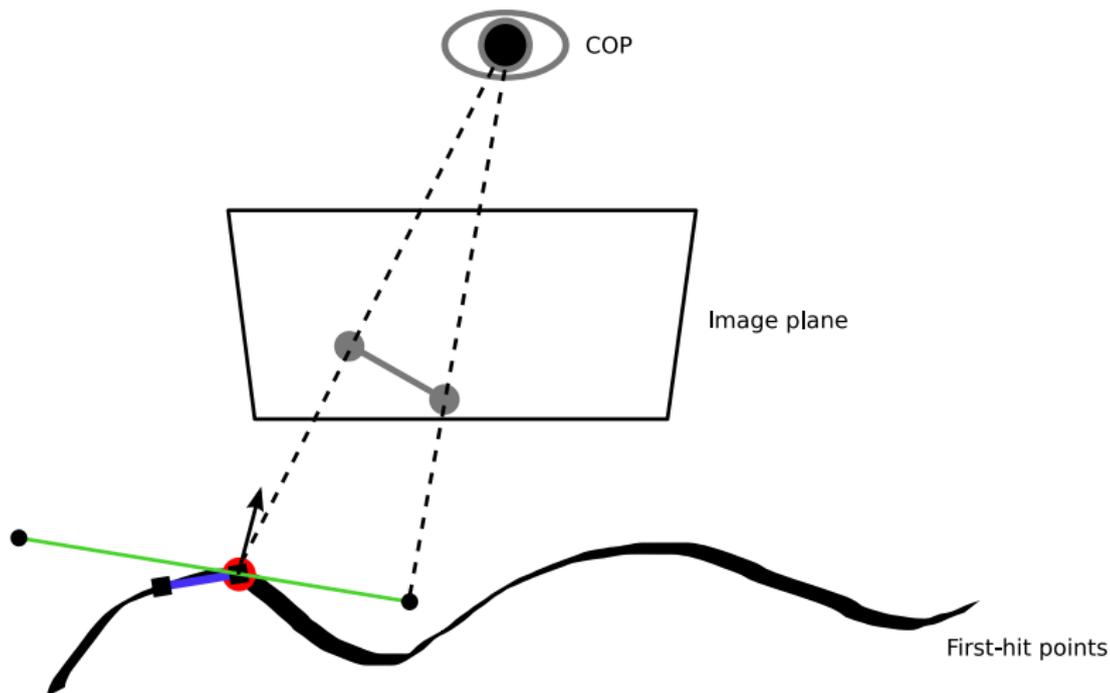
# Interactive Cutting: Surface Placement

Input: First-hit points, first-hit normals



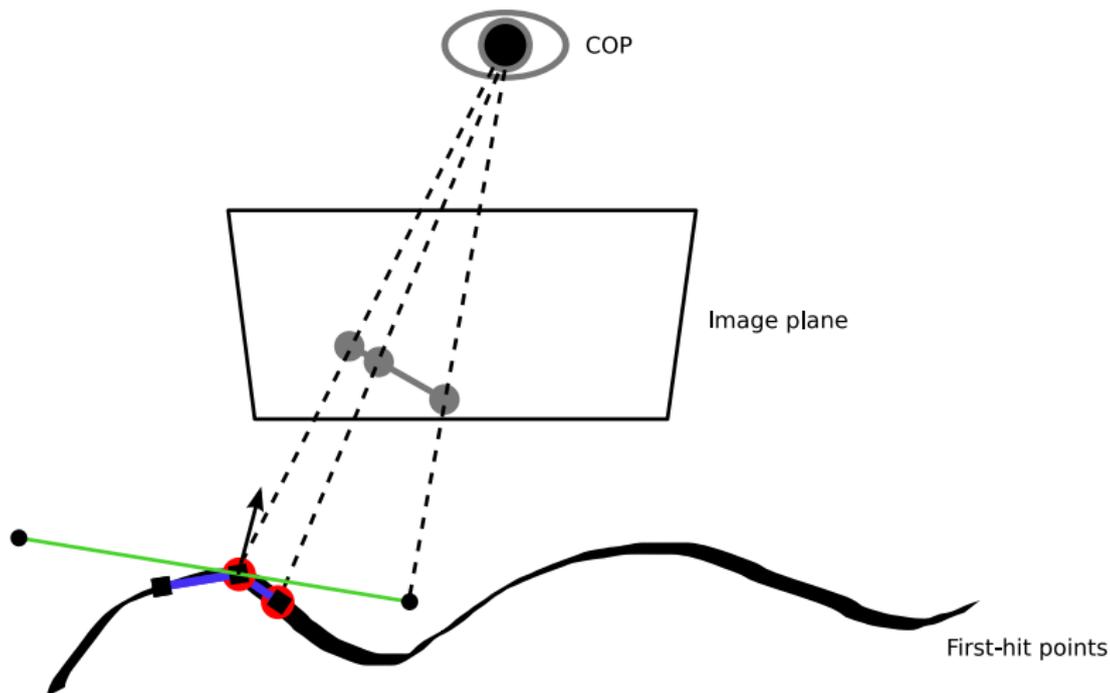
# Interactive Cutting: Surface Placement

Input: First-hit points, first-hit normals



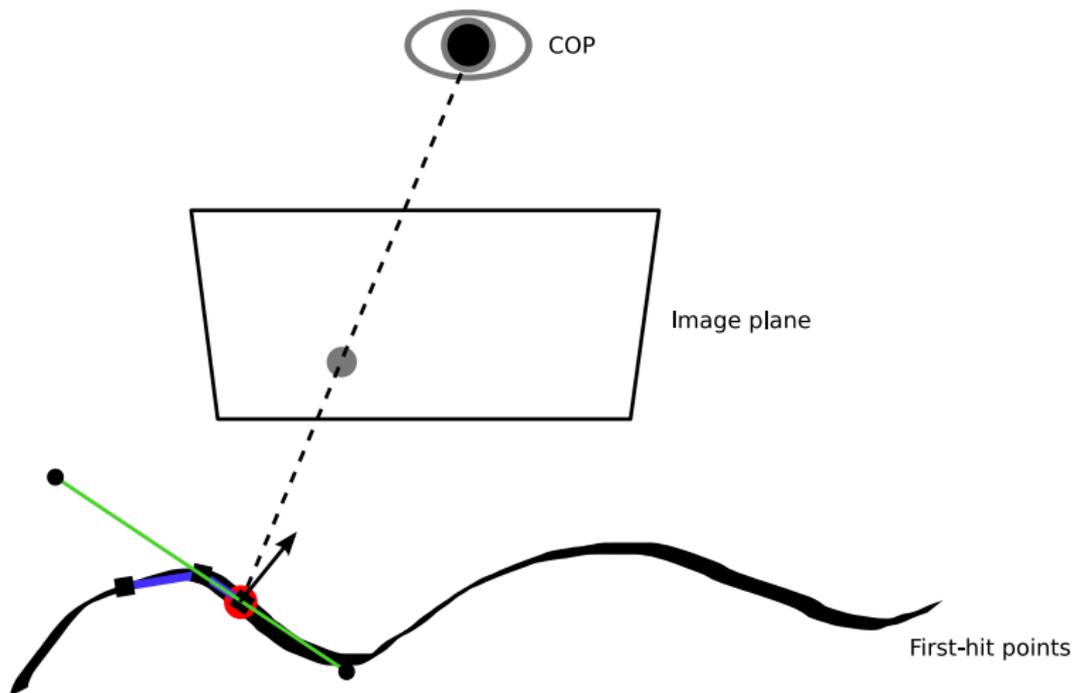
# Interactive Cutting: Surface Placement

Input: First-hit points, first-hit normals



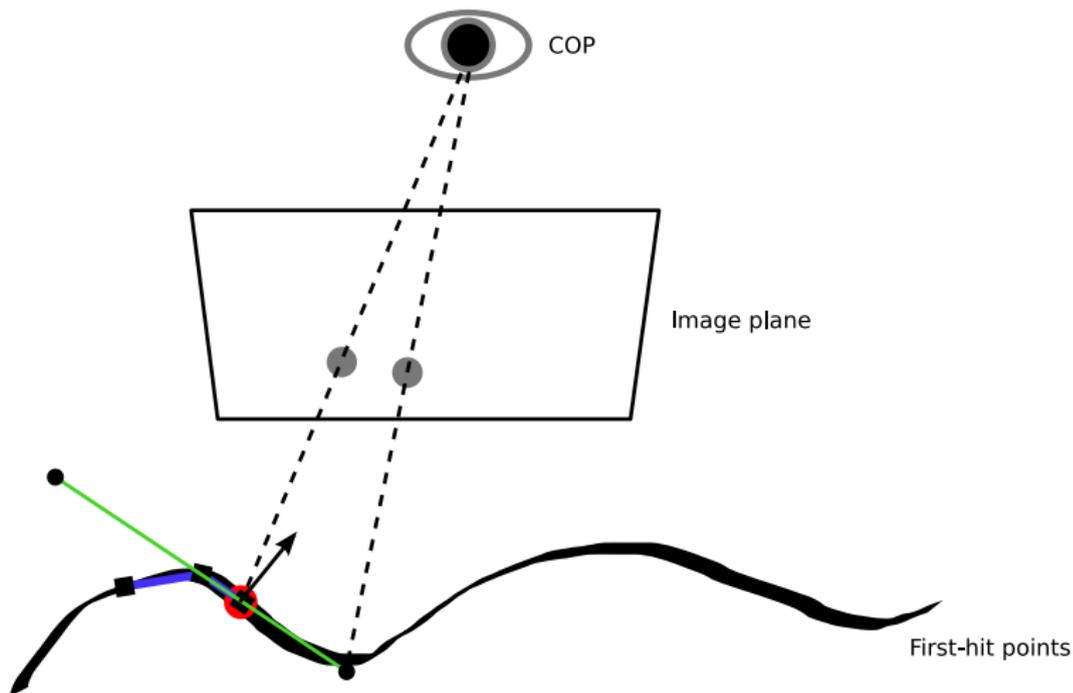
# Interactive Cutting: Surface Placement

Input: First-hit points, first-hit normals



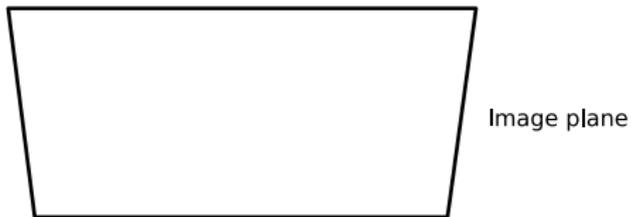
# Interactive Cutting: Surface Placement

Input: First-hit points, first-hit normals



# Interactive Cutting: Surface Placement

Input: First-hit points, first-hit normals



# Interactive Cutting: Surface Placement

Input: First-hit points, first-hit normals



COP

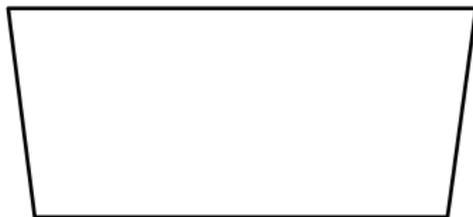
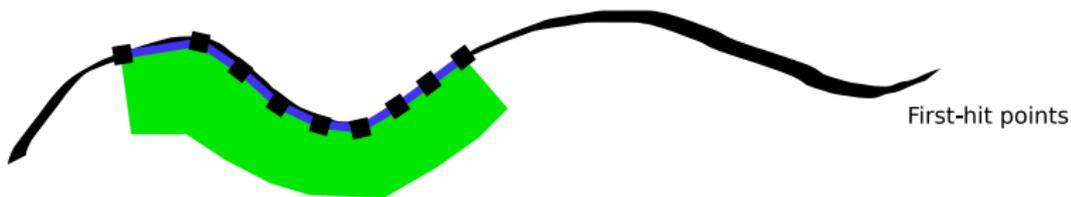


Image plane



First-hit points



## Point-and-drag interface

- ▶ Supports deformation types *pull* and *turn*
- ▶ May require several changes of perspective

## Point-and-drag interface

- ▶ Supports deformation types *pull* and *turn*
- ▶ May require several changes of perspective

## Alternative: Deformation templates

- ▶ Apply pre-defined deformation
- ▶ Supports deformation type *cut/flip*
- ▶ Corresponding to cutting templates
- ▶ Parametrizable position, size, amount of aperture

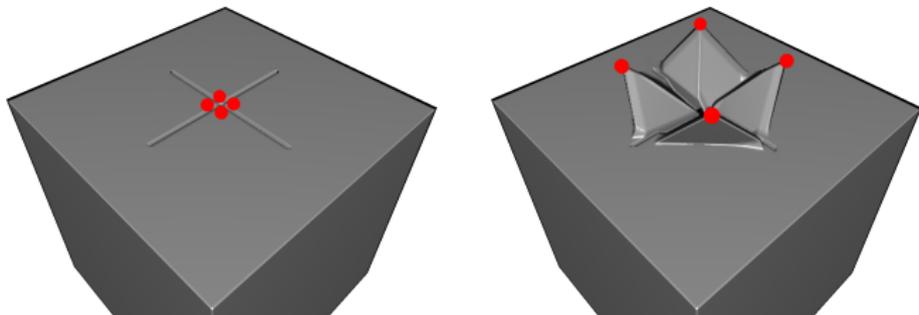
# Interactive Deformation

## Point-and-drag interface

- ▶ Supports deformation types *pull* and *turn*
- ▶ May require several changes of perspective

## Alternative: Deformation templates

- ▶ Apply pre-defined deformation
- ▶ Supports deformation type *cut/flip*
- ▶ Corresponding to cutting templates
- ▶ Parametrizable position, size, amount of aperture



# Results

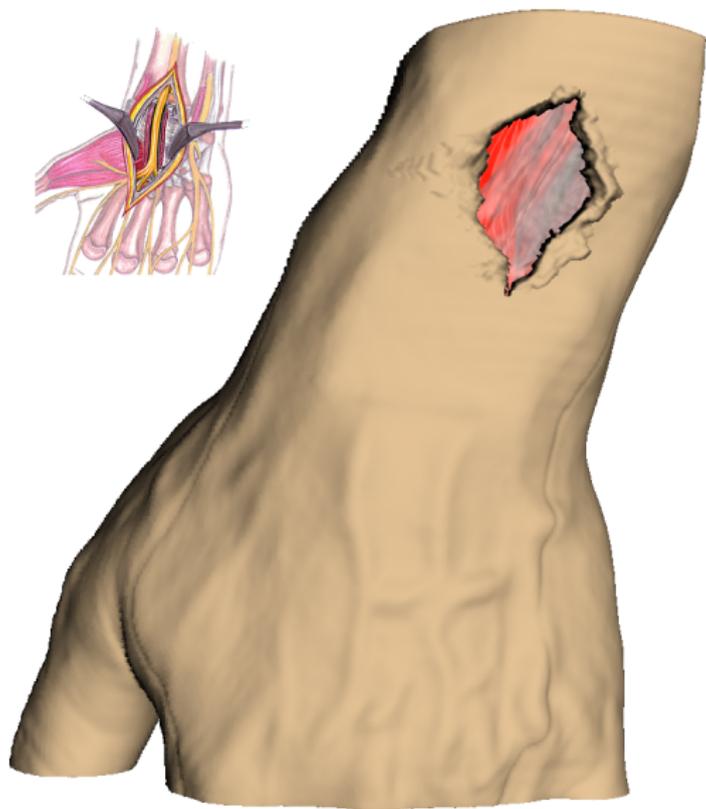
Hand data set

# Results

Hand data set

# Results

Hand data set



256 × 128 × 256 voxels

single cut

manual deformation

2,710,516 ChainMail elements

4.3% surface elements

693,200 vertices

38 fps

# Results

Visible Human head data set

256<sup>3</sup> voxels

Y-cut

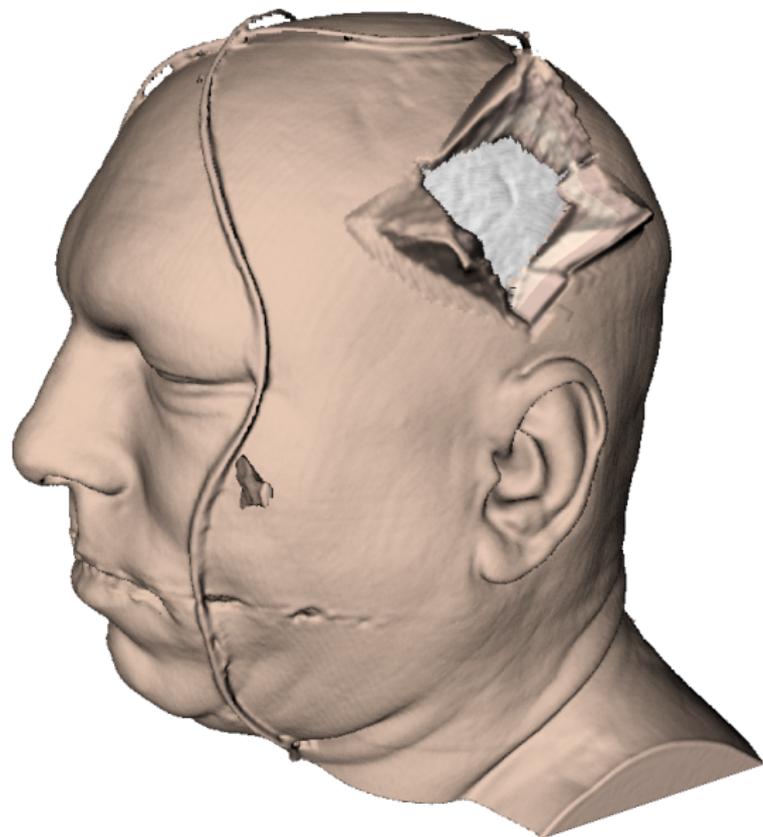
manual deformation

6,040,852 ChainMail elements

3.9% surface elements

1,529,000 vertices

23 fps





## Contributions

- ▶ Interaction techniques support easy deformation of anatomical objects
- ▶ ChainMail algorithm permits interactive deformation
- ▶ Linked volume structure allows simple cutting and surface extraction
- ▶ Concepts can be easily integrated into existing systems

## Contributions

- ▶ Interaction techniques support easy deformation of anatomical objects
- ▶ ChainMail algorithm permits interactive deformation
- ▶ Linked volume structure allows simple cutting and surface extraction
- ▶ Concepts can be easily integrated into existing systems

## Future Work

- ▶ Map material properties to deformation constraints
- ▶ Evaluate more realistic physical model (like FEM)
- ▶ Improve image quality

# Questions?



<http://www.voreen.org>