Interactive Multi-Resolution Modeling with Sculptor

Stefan Birmanns
School of Health Information Sciences &
Institute of Molecular Medicine
University of Texas – Houston

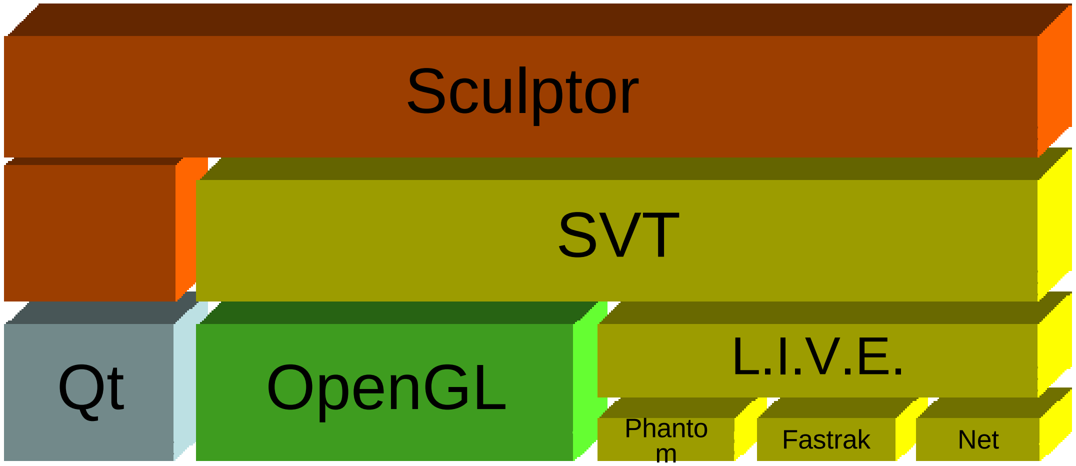
Overview

- Sculptor?
- Visualize Volumetric EM Data
- Visualize High-Resolution Structures
- Multi-Resolution Docking

Sculptor

- Interactive multi-resolution modeling
  - Application-driven development:
    - Visualization of volumetric experimental data
    - Visualization of high-resolution structures
    - Interactive and algorithmic docking techniques
- But the focus is not:
  - To develop the best volume renderer
  - To develop the best molecule renderer

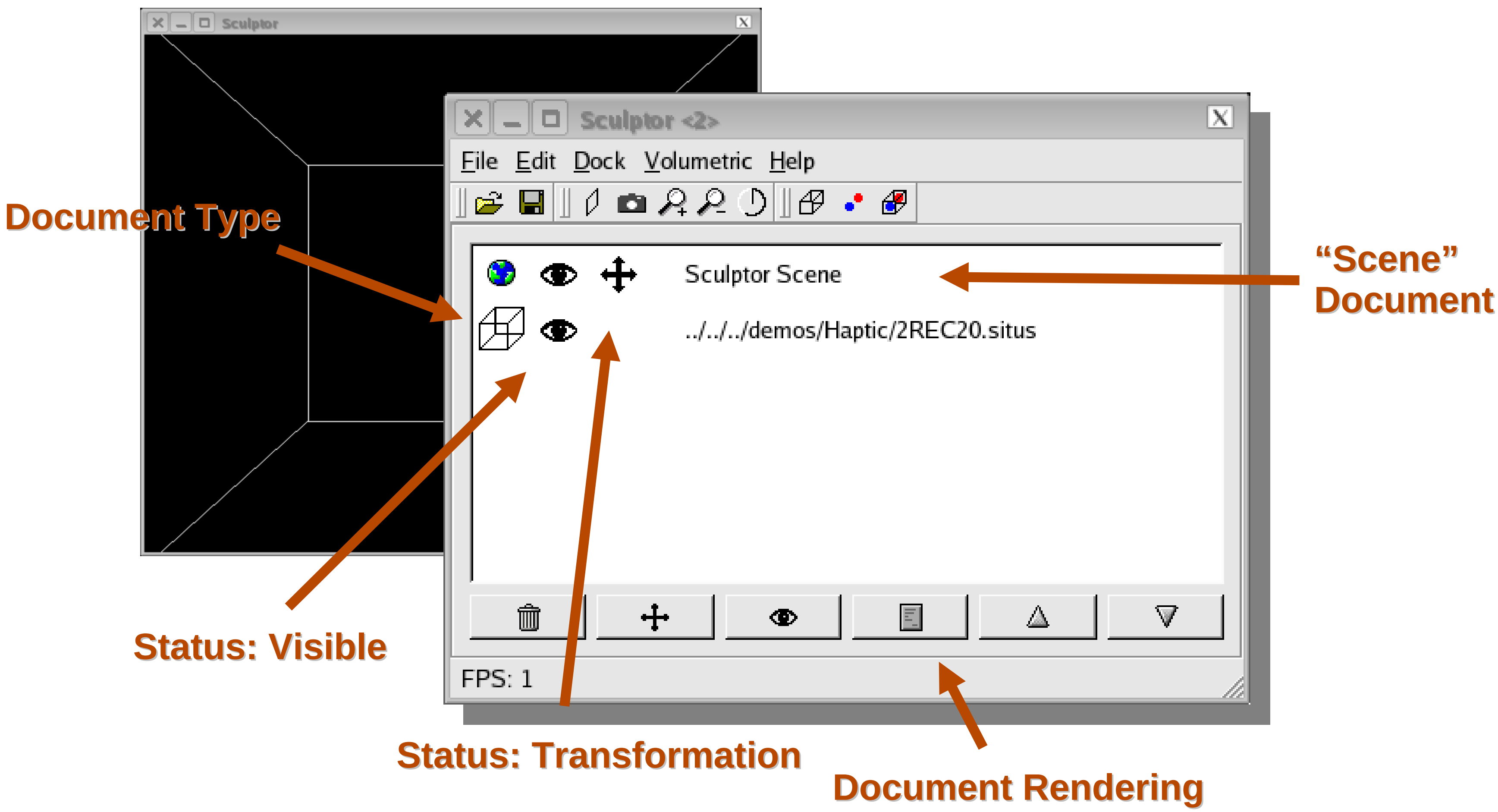
Sculptor



- Sculptor:
  - Qt[1] GUI library
  - OpenGL[2] 3D graphics library
  - SVT - VR and visualization toolkit
  - Multi-platform (Unix, Windows)

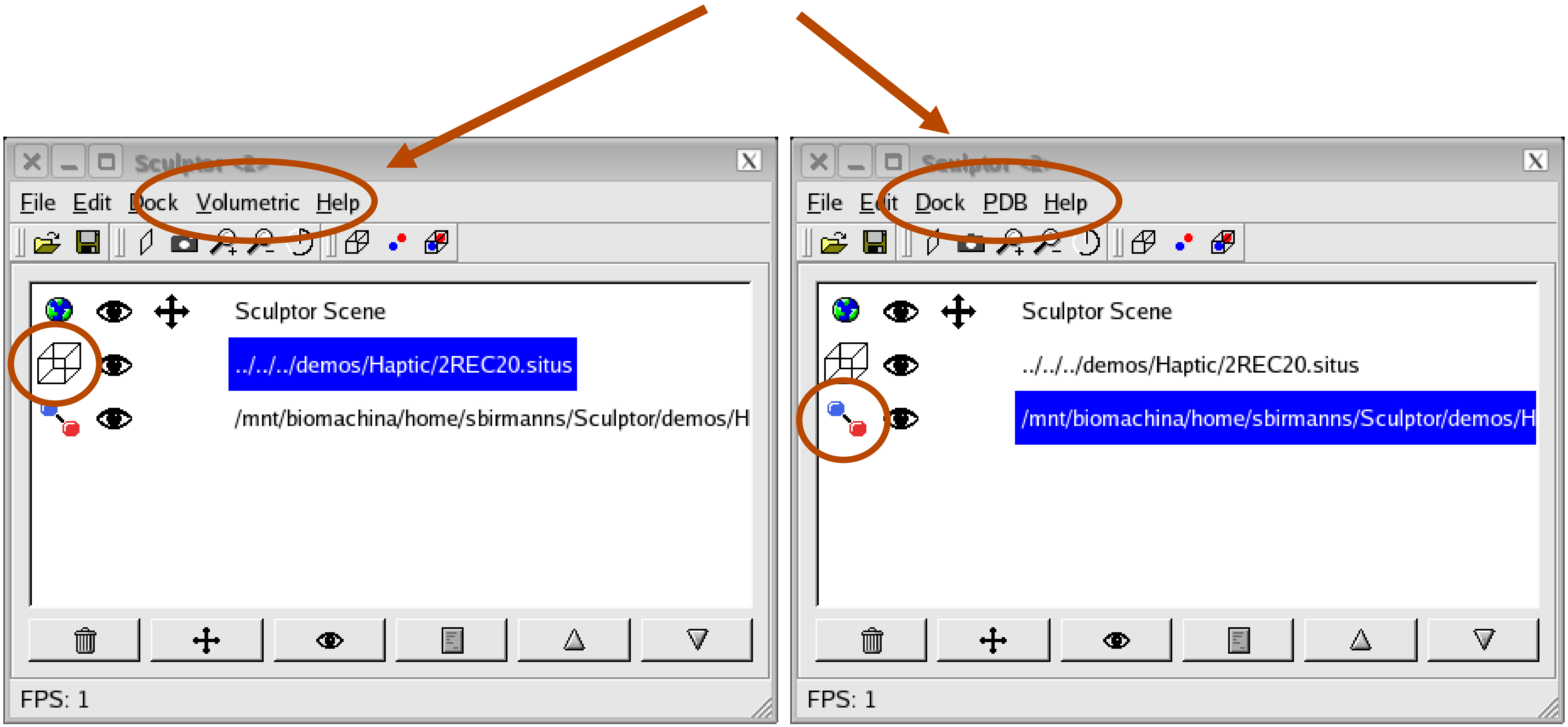
1) http://www.opengl.org
2) http://www.trolltech.com

Sculptor


Sculptor
Sculptor <2>
File Edit Dock Volumetric Help
Sculptor Scene
../../../demos/Haptic/2REC20.situs
FPS: 1
Document Type
"Scene" Document
Status: Visible
Status: Transformation
Document Rendering

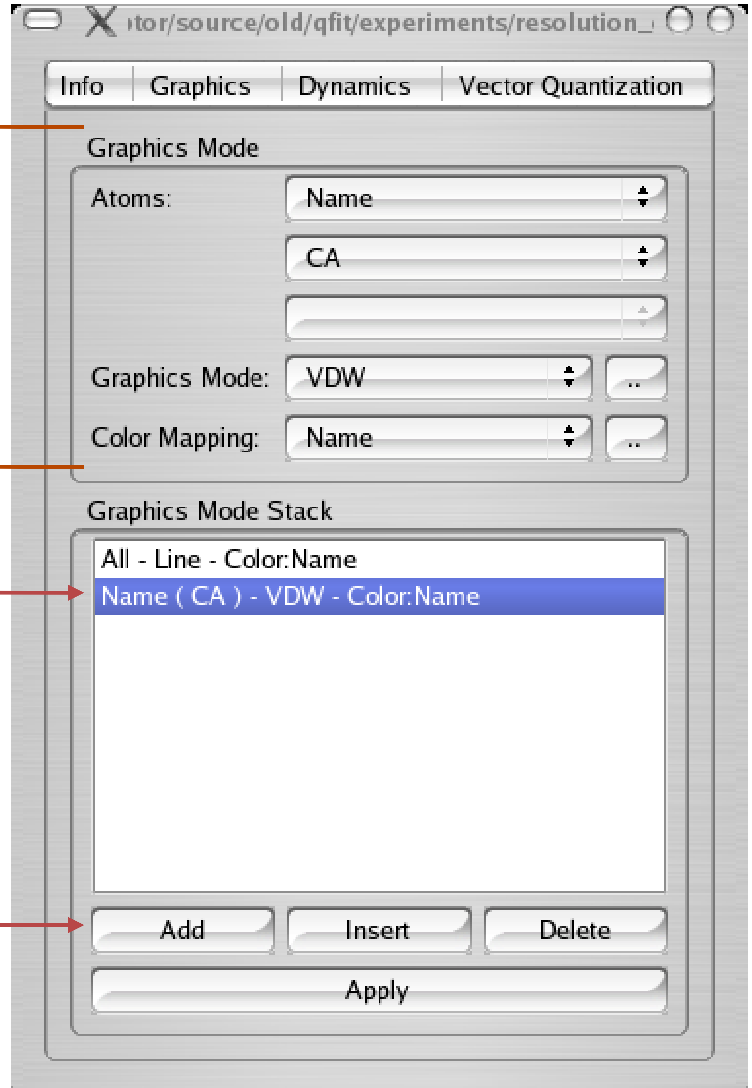
Sculptor

Context-specific menu



Structure Visualization

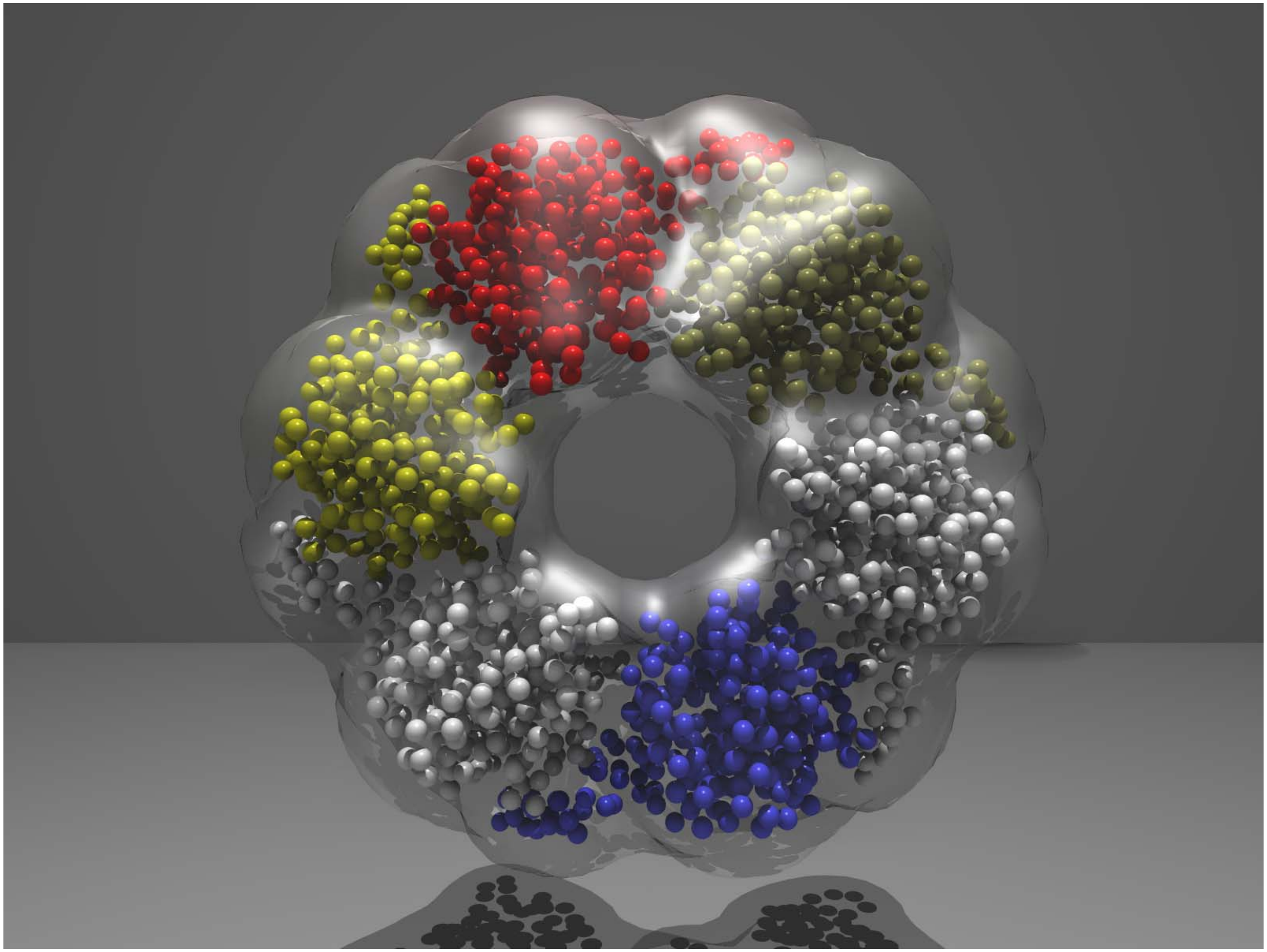
Rendering Style of a group of atoms



Graphics Mode Stack

Add and remove graphics mode changes

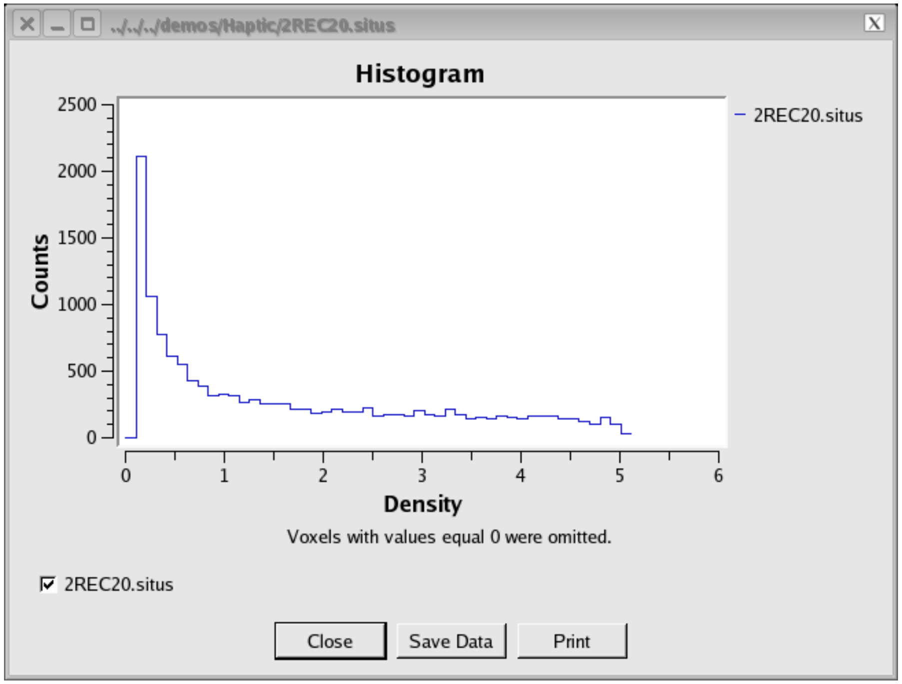
High-Quality Rendering



Volumetric Data

- Situs file-format
  - To convert from/to Situs use “map2map” (Situs)

- In context menu of volume document
  - Histogram
  - Direct inspection of density values
  - Normalization

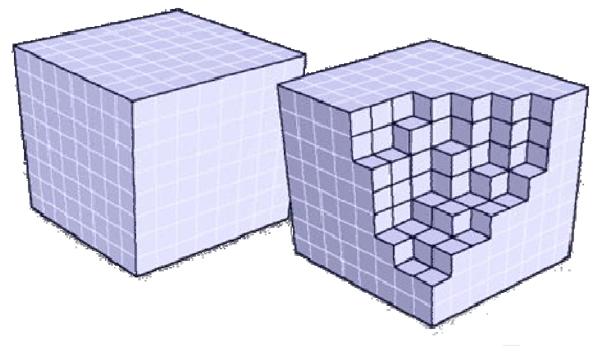


Volumetric Data

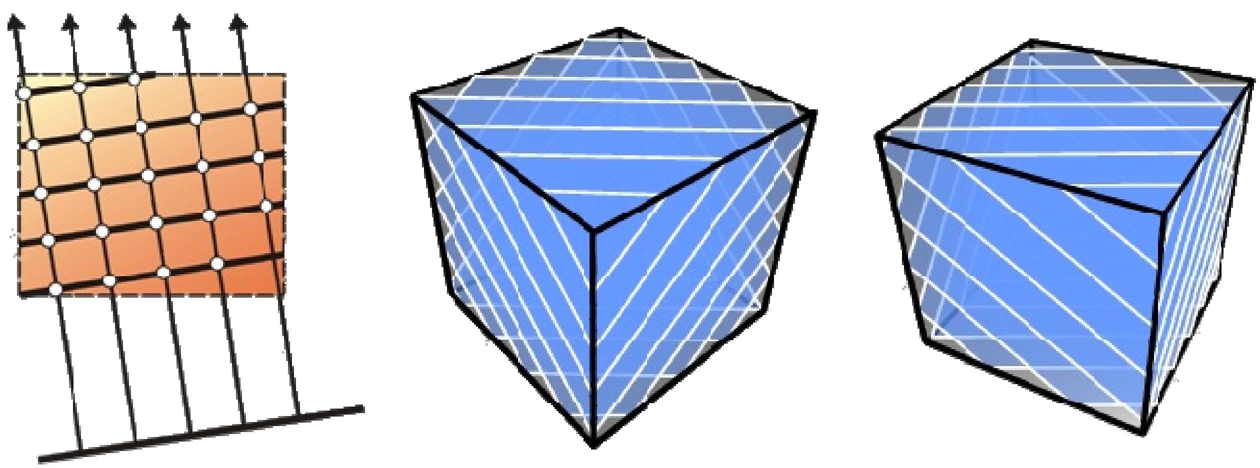
- Volume Rendering Techniques:
  - Isosurfaces
    - Conversion to triangle mesh, efficient to render
    - Single parameter, threshold value to define surface
    - EM maps often feature varying resolution and density
      - No precise, hard surface, single threshold difficult
  - Direct Volume Rendering
    - Direct rendering of voxel data
    - Soft surfaces, true transparency, intensity segmentation
    - Complex transfer-function design
    - Real-time rendering challenging

Direct Volume Rendering

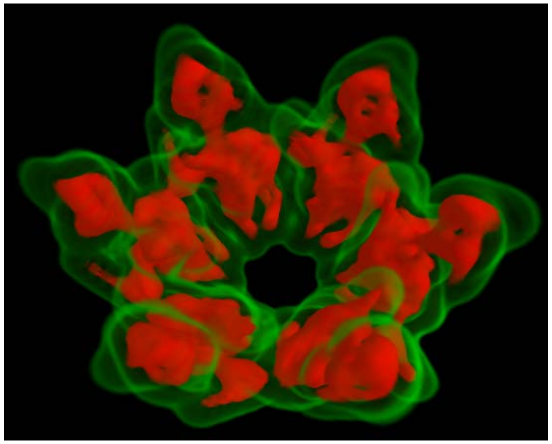
- Direct rendering of scalar field

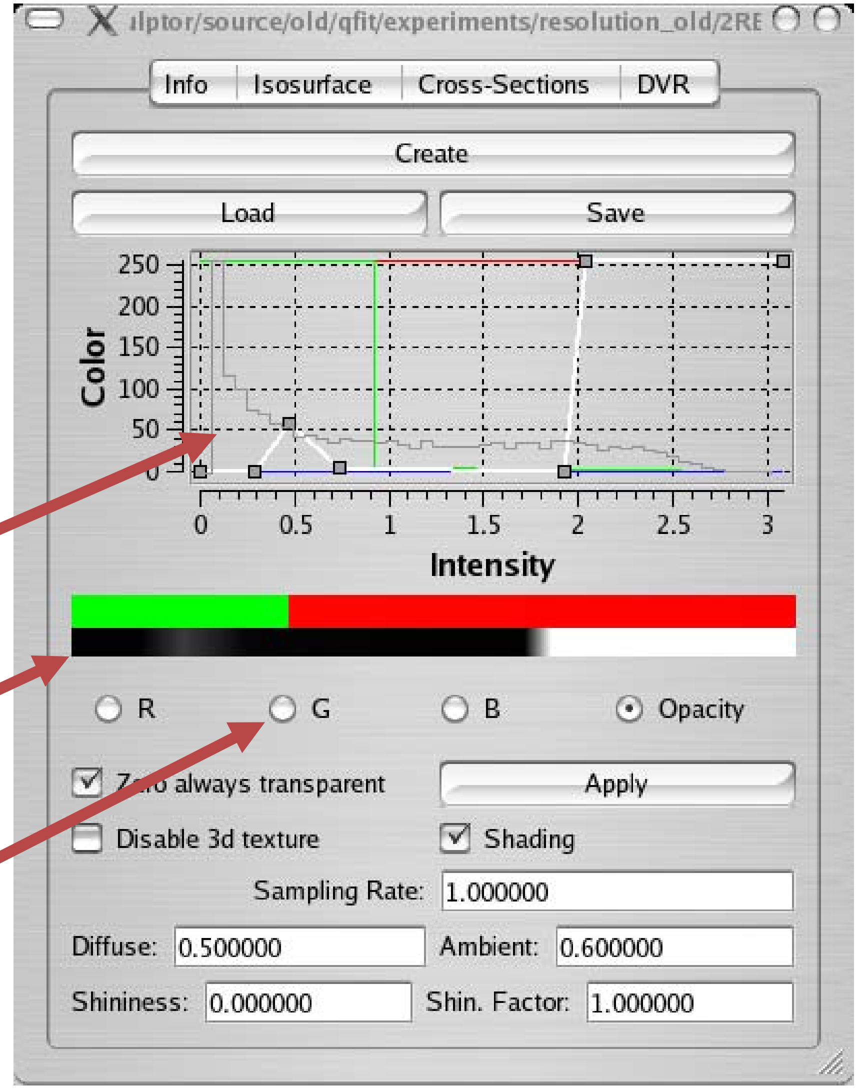


- 3D textures sampled by 2D viewer aligned slices approximate rendering integral



Direct Volume Rendering



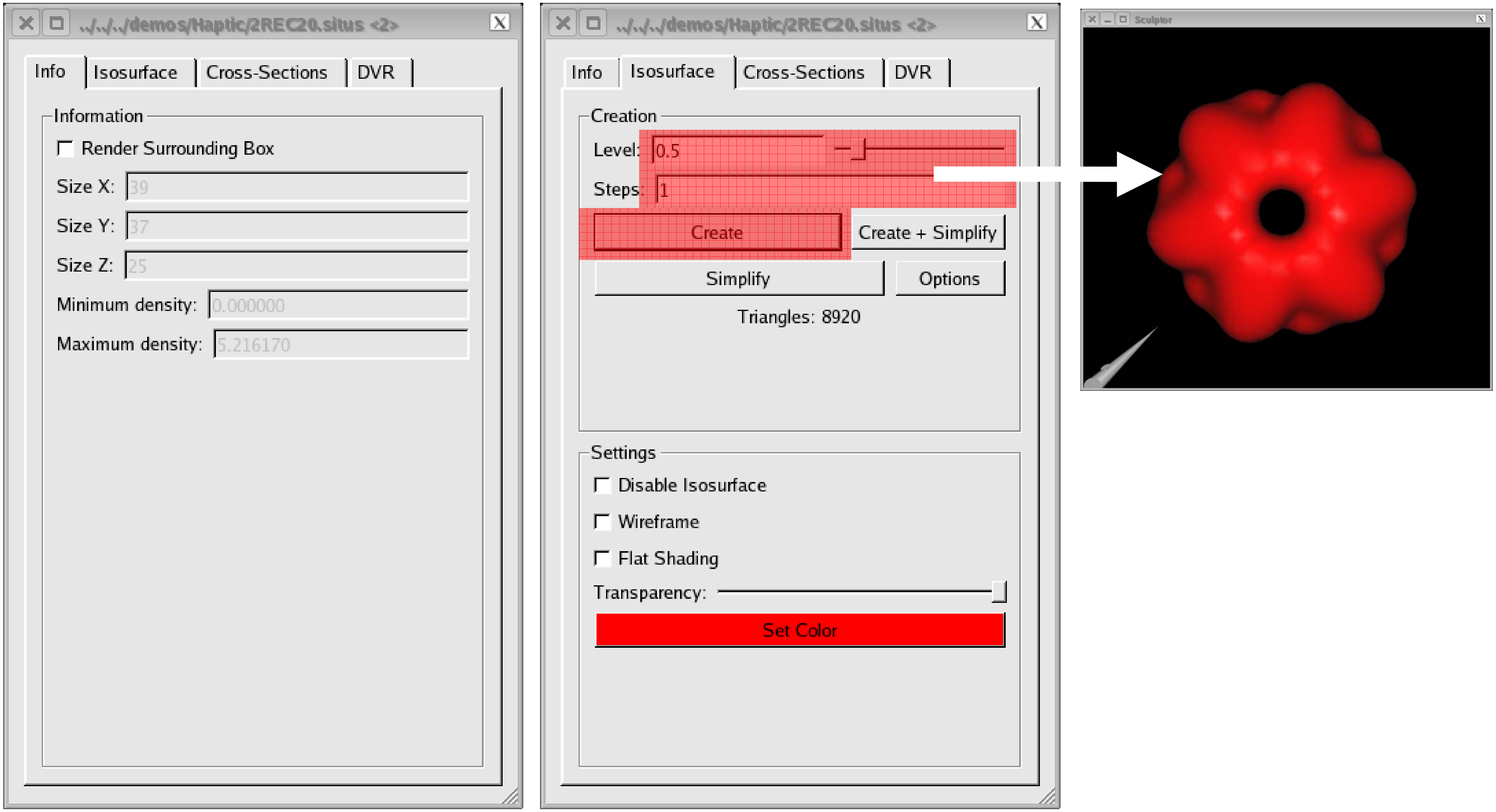


Transfer function editor

Resulting color- and opacity spectrum

Color channels are edited separately

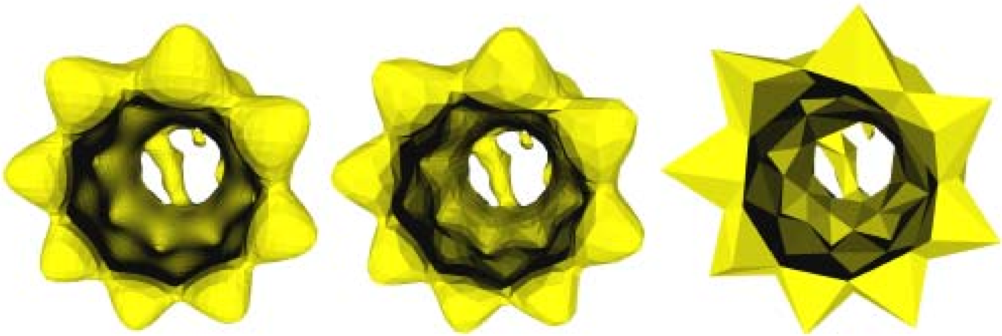
Surface Rendering



Sculptor

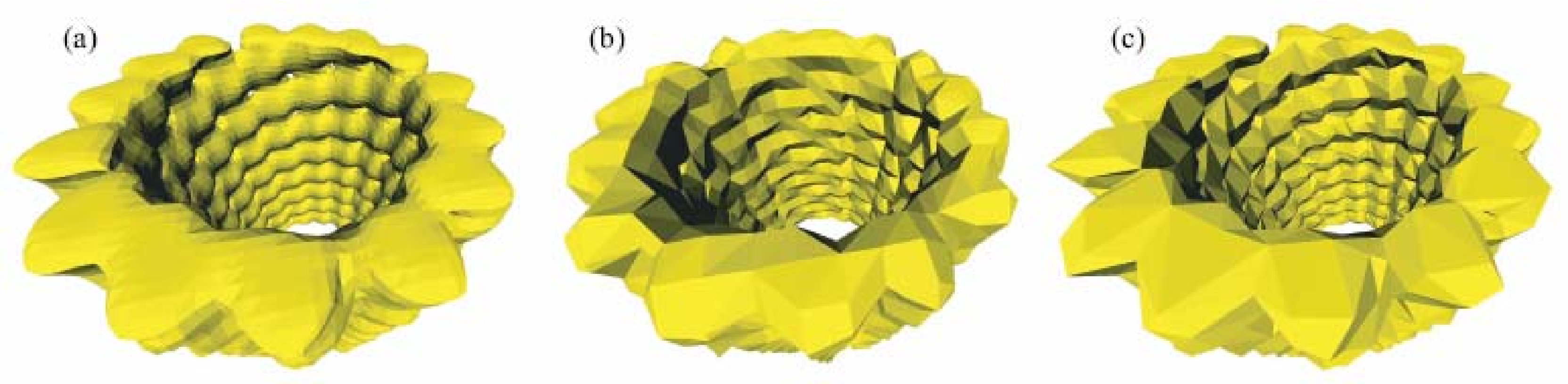
Mesh Simplification

- Large macromolecular assemblies
  - Time-consuming to render
  - Visual and haptic rendering compete for CPU time
- Haptic rendering real-time critical
  - Visual dominates haptic rendering
- Load balancing based on mesh simplification
  - Remove detail when force update rate is not sufficient



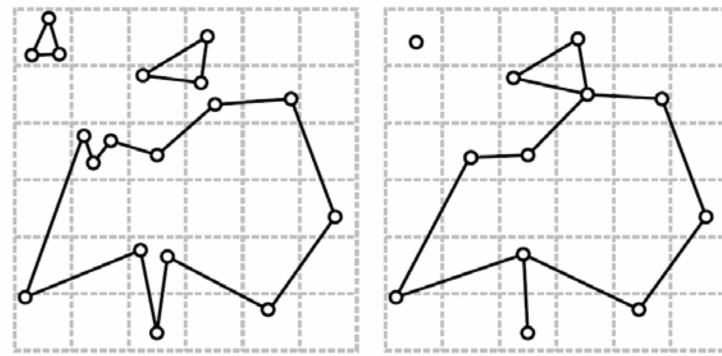
Mesh Simplification

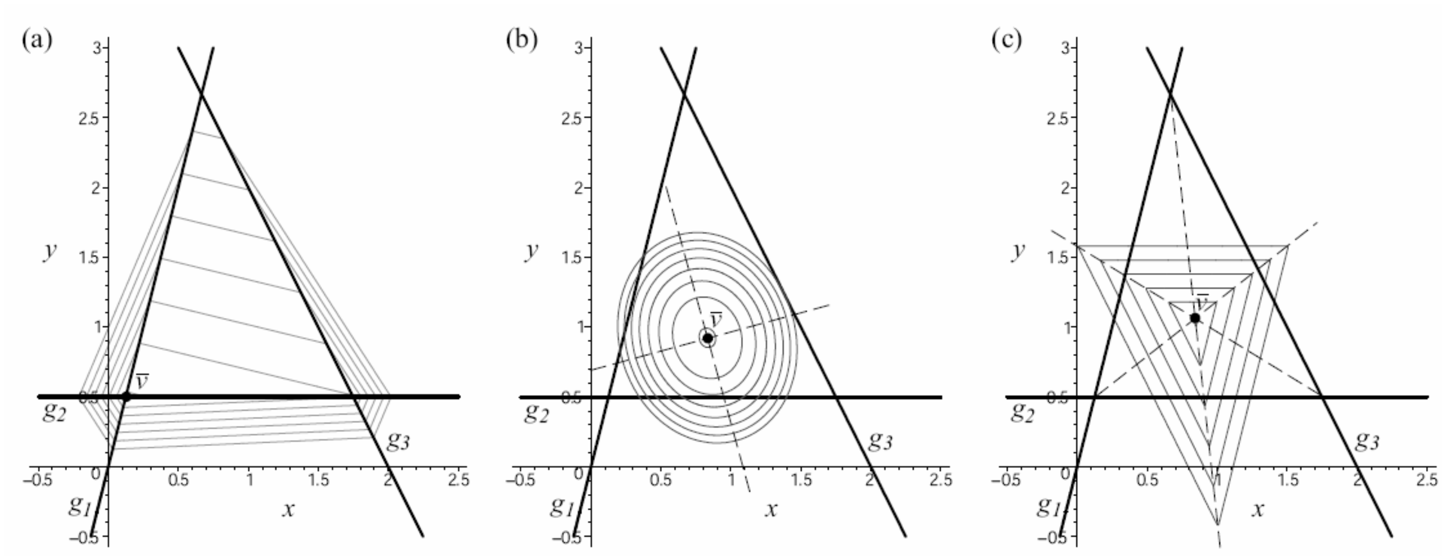
- Important properties of MS algorithms
  - Quality of the approx. meshes
  - Efficiency of the algorithm



Mesh Simplification

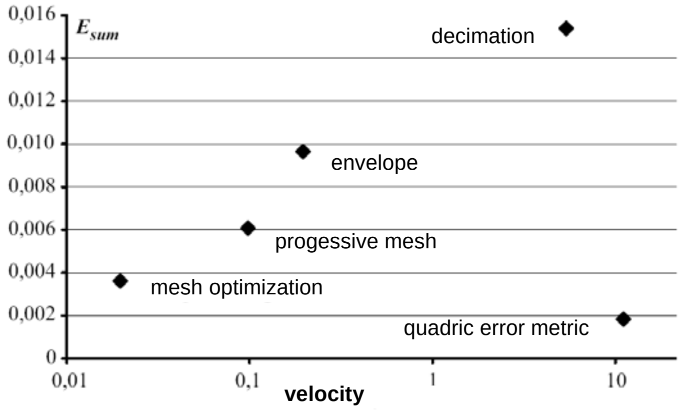
- Simplification techniques
  - Vertex Clustering
  - Vertex Decimation
  - Edge Contraction
- Error metrices



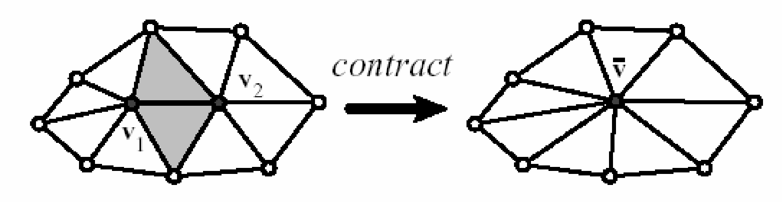


Mesh Simplification

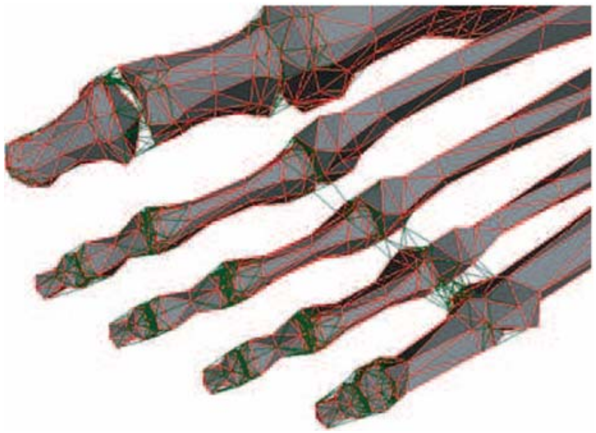
- Edge Contraction with the Quadric Error Metric
  - Fast, produces high quality meshes



  - Contract edge and replace it by a single new vertex

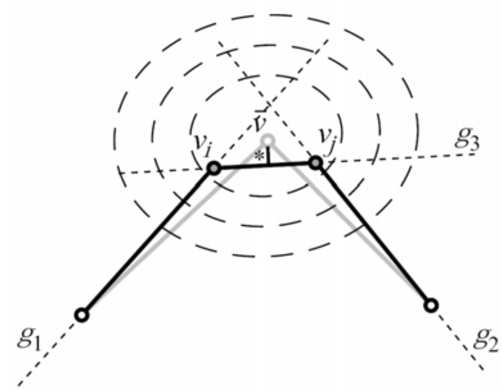


  - Imaginary edges are possible



Mesh Simplification

- Quadric Error Metric
  - Quadratic distance to all corresponding planes of the original mesh



$$Q(v) = \sum_{i=1}^{k} (n_i^t v + d_i)^2 = v^t Q v$$

  - Corresponding planes are these from which the vertex results
  - Fast because of addition theorem

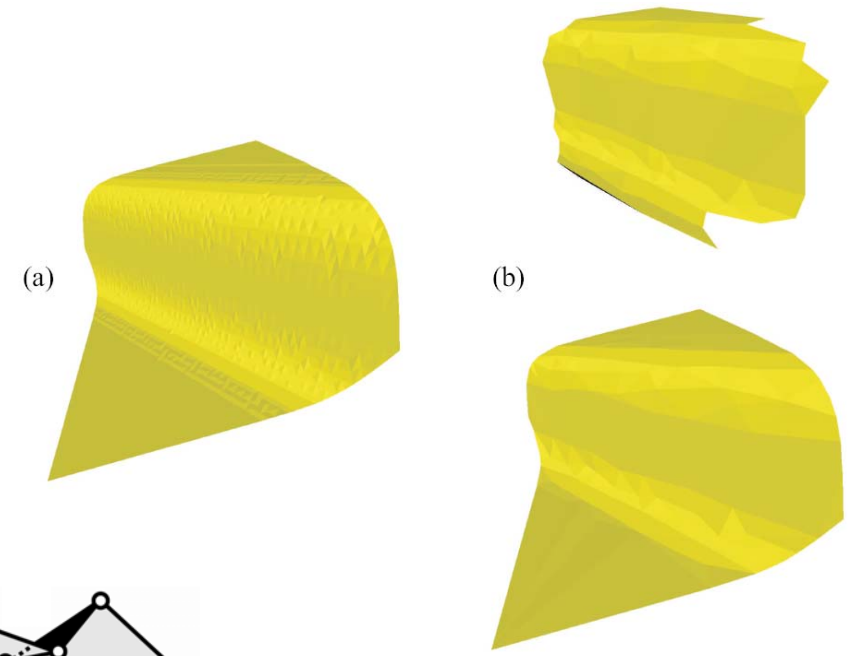
$$Q(v) = Q_i(v) + Q_j(v) = v^t (Q_i + Q_j) v$$

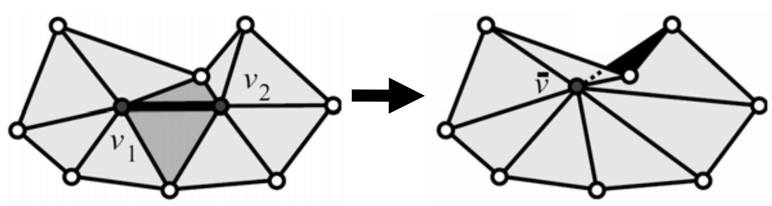
  - New vertex is the result of

$$grad(Q(v)) = 0$$

Mesh Simplification

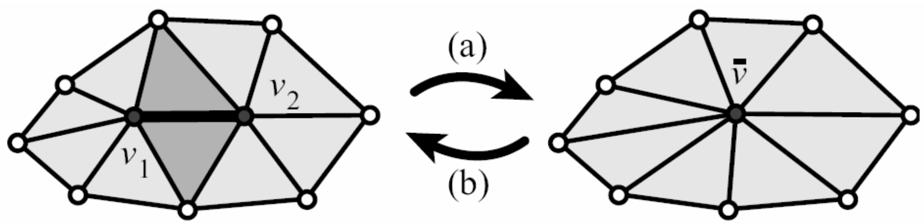
- Special cases
  - open boundaries
  - triangle identity
  - triangle twist





Mesh Simplification

- Simplification is reversible
  - Inverse operation of an edge contraction (a) is a vertex split (b)



  - Save progressive mesh simplification for later refinement

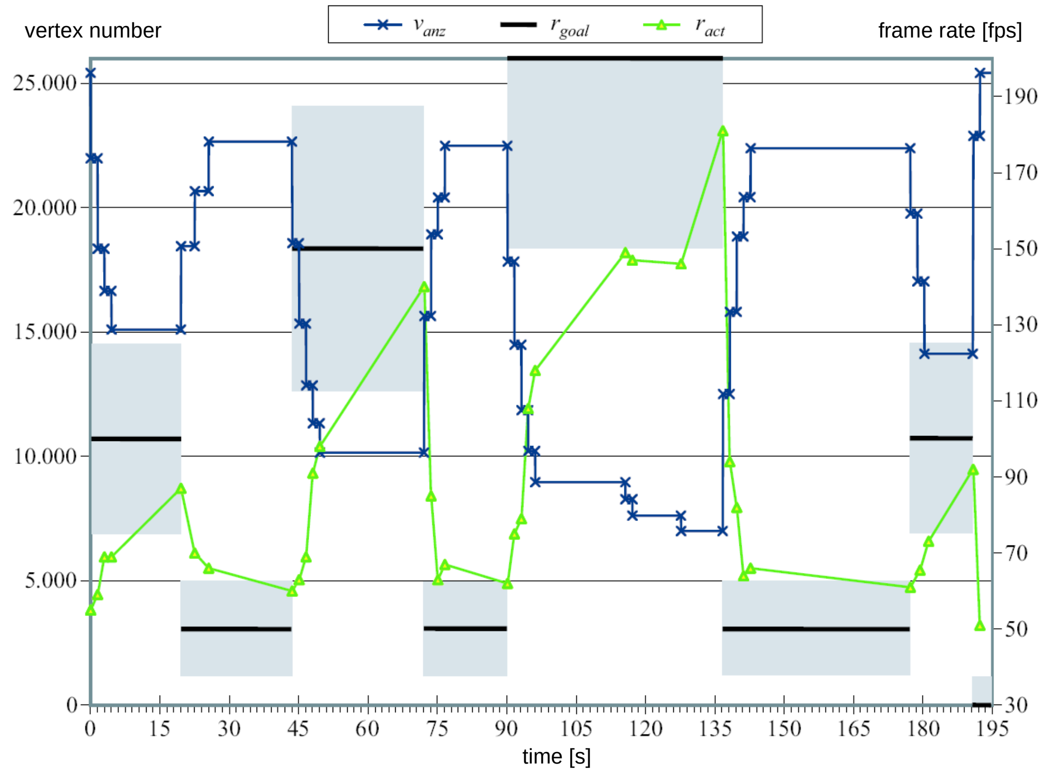
$$M_n \underset{\psi_n^{-1}}{\overset{\psi_n}{\rightleftharpoons}} M_{n-1} \underset{\psi_{n-1}^{-1}}{\overset{\psi_{n-1}}{\rightleftharpoons}} \cdots \underset{\psi_2^{-1}}{\overset{\psi_2}{\rightleftharpoons}} M_1 \underset{\psi_1^{-1}}{\overset{\psi_1}{\rightleftharpoons}} M_0$$

$$\psi_k(M_k) = M_{k-1} \qquad \psi_k^{-1}(M_{k-1}) = M_k \qquad k \in \{1, \ldots, n\}$$

$$(\psi_1 \circ \cdots \circ \psi_n)(M_n) = M_0 \qquad (\psi_n^{-1} \circ \cdots \circ \psi_1^{-1})(M_0) = M_n$$

Mesh Simplification

- Adaptable level of detail
  - Adjust the simplification level according to the desired force update rate or frame rate



Mesh Simplification

- Algorithm overview
  - Compute quadric Q_i for every vertex of the original mesh
  - Compute cost (quadric error) and optimal contraction vertex for every edge contraction by minimizing $v^t(Q_i + Q_j)v$
  - Sort possible contractions according to cost
  - Perform contraction with lowest cost and recompute cost of varied edges and resort them into the cost sorted edge contraction list
  - Save the progressive simplification
  - Adapt the level of detail corresponding to constraint (FUS, FPS)

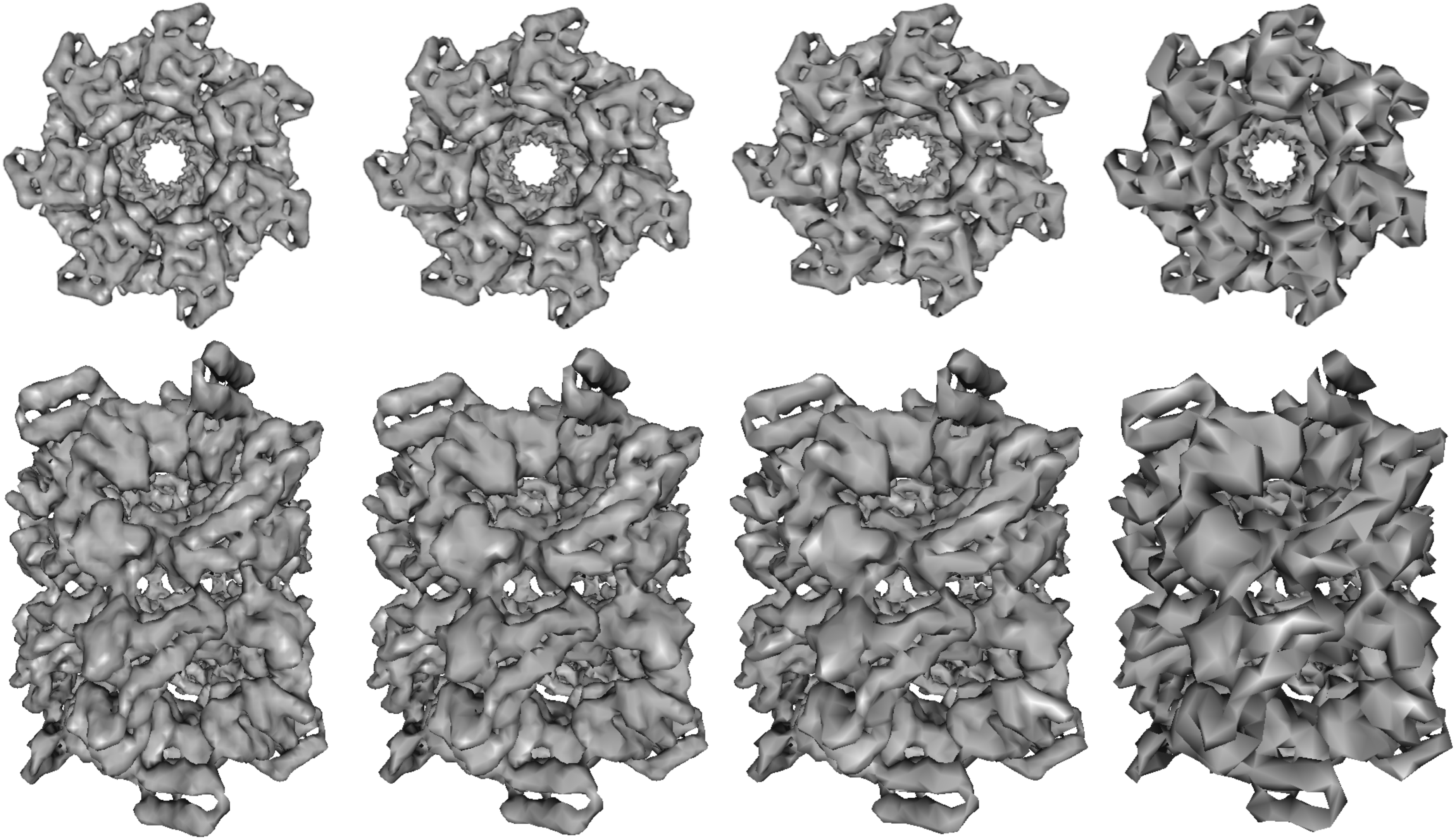
Mesh Simplification

108,989 vertices
219,718 triangles

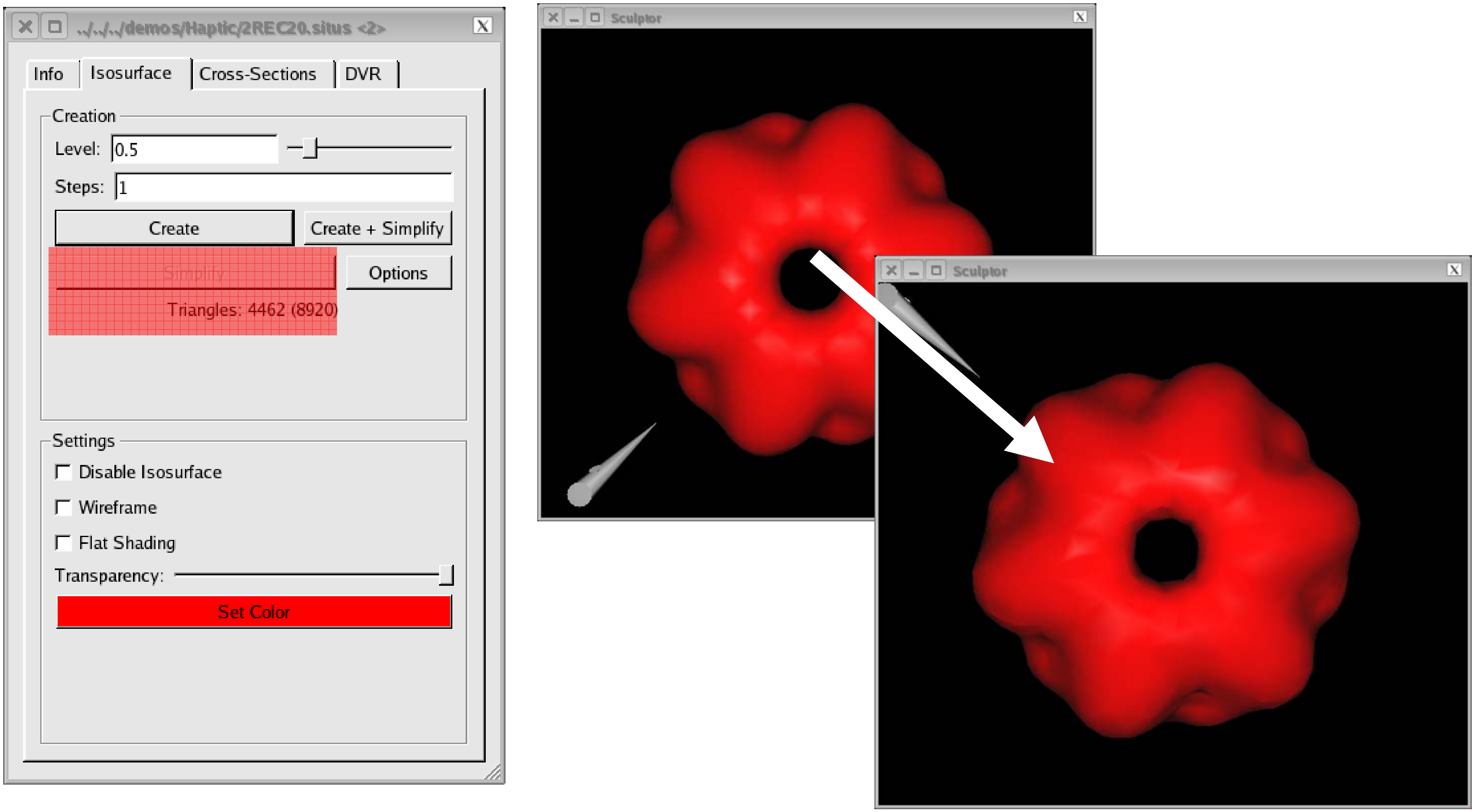
50,000 vertices
101,810 triangles

30,000 vertices
61,921 triangles

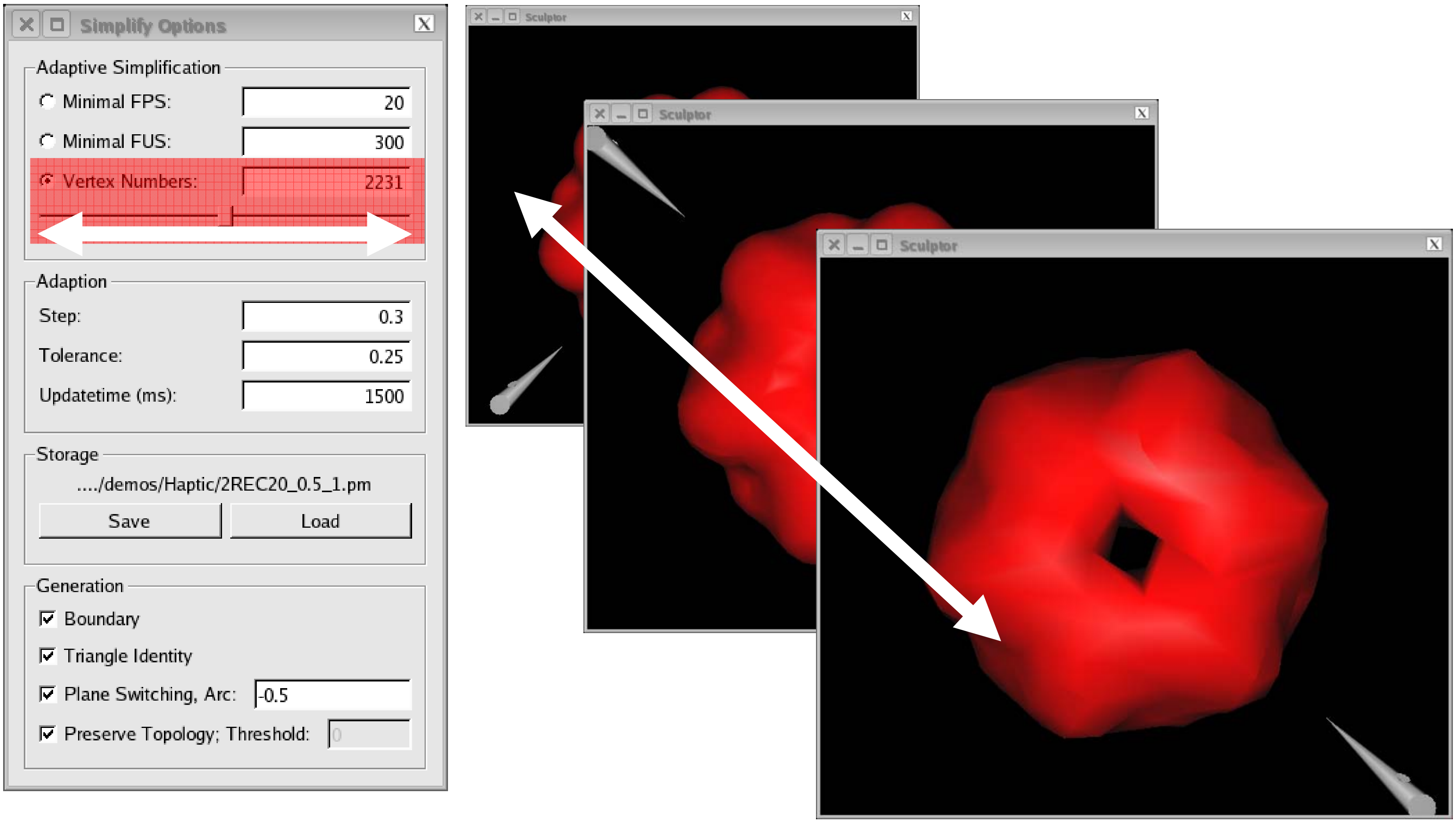
10,000 vertices
22,115 triangles



Mesh Simplification



Mesh Simplification

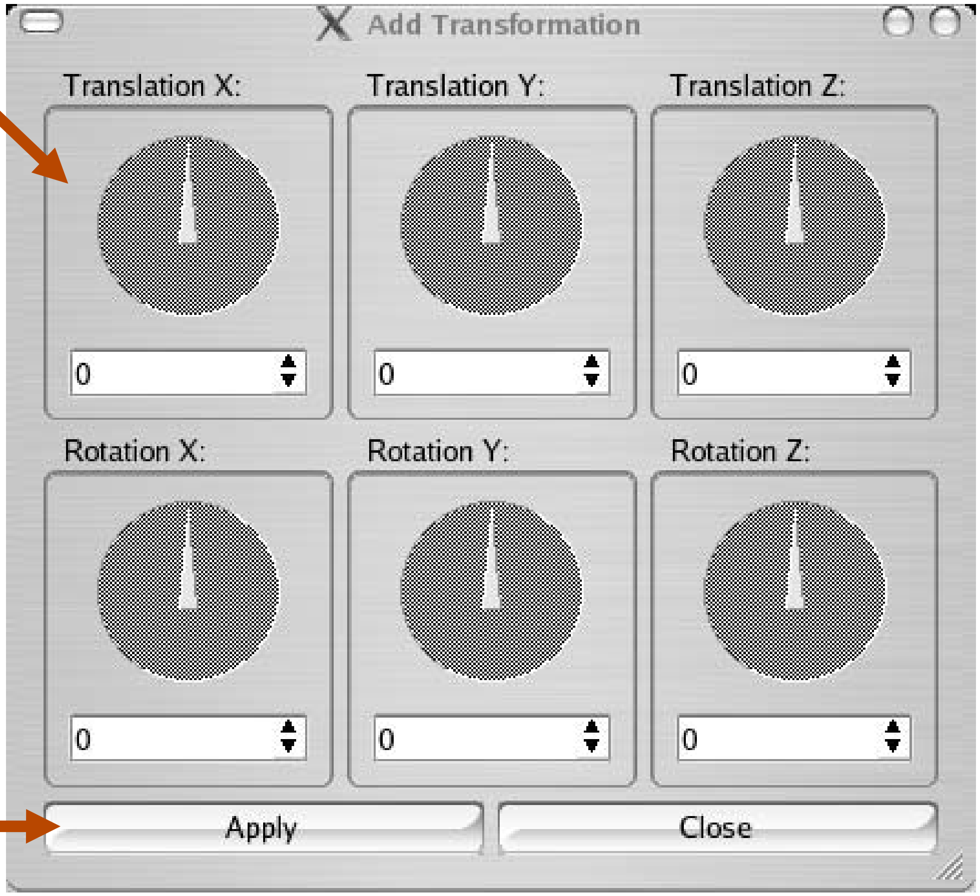


Multi-Resolution Modeling

- Interactive docking:
  - Visual docking
    - Manual docking by eye
  - Haptic Rendering
    - Manual docking augmented by force feedback
    - Reduced docking criterion
  - Algorithmic Docking
    - Pattern-recognition technique based on feature points

Transformation

Software "Dials"

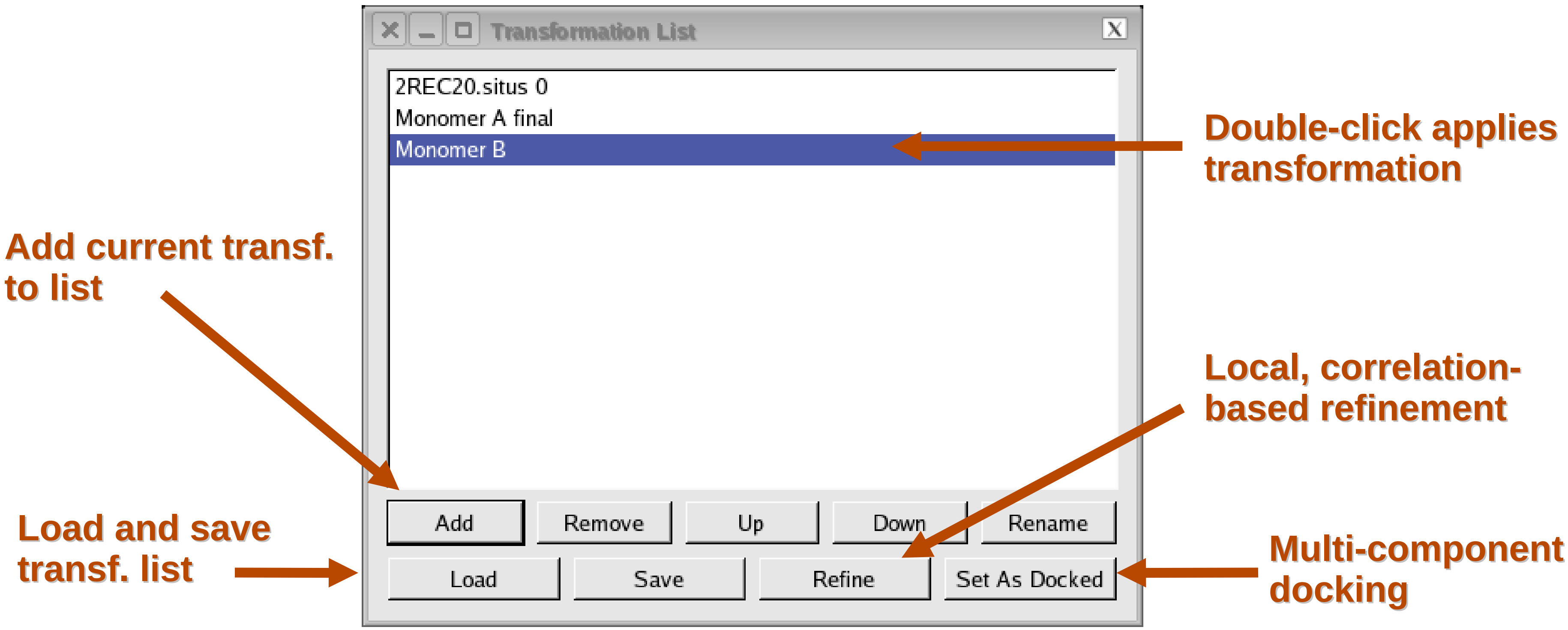


Only the activated document is manipulated

Definitely impose transf. to document and set dials back to zero

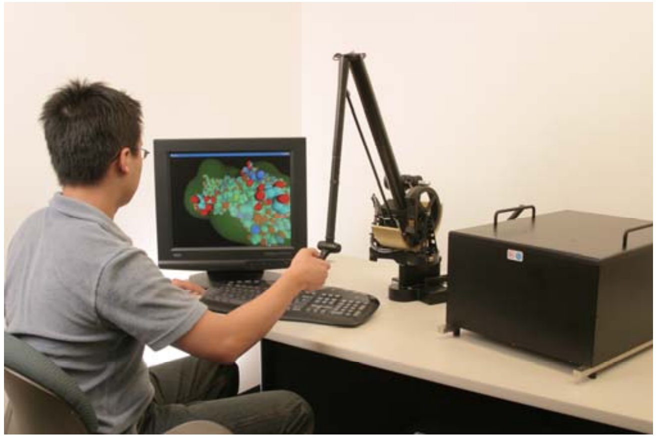
Multi-Resolution Docking

- Docking = transformation of structure into density map
- Management of transformations:



Haptic Rendering

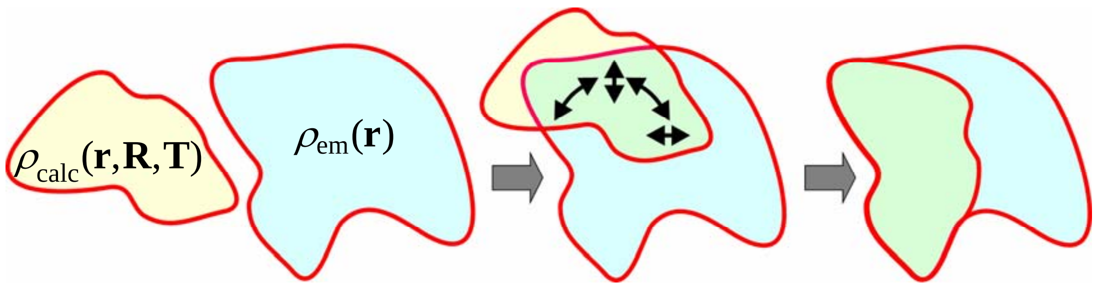
- Interactive docking augmented by haptic rendering
  - Guide the user by force-feedback through the 6D search space
  - Cross-correlation as basis for force and torque calculation
  - Combined with advanced virtual reality techniques
    - 3D stereoscopic and tracked visual rendering



Cross-Correlation

- Cross-correlation coefficient between the two objects is a popular docking criterion:

$$C(R,T) \propto \int \rho_{calc}(\mathbf{r},R,T) \cdot \rho_{em}(\mathbf{r})\, d^3\mathbf{r}$$

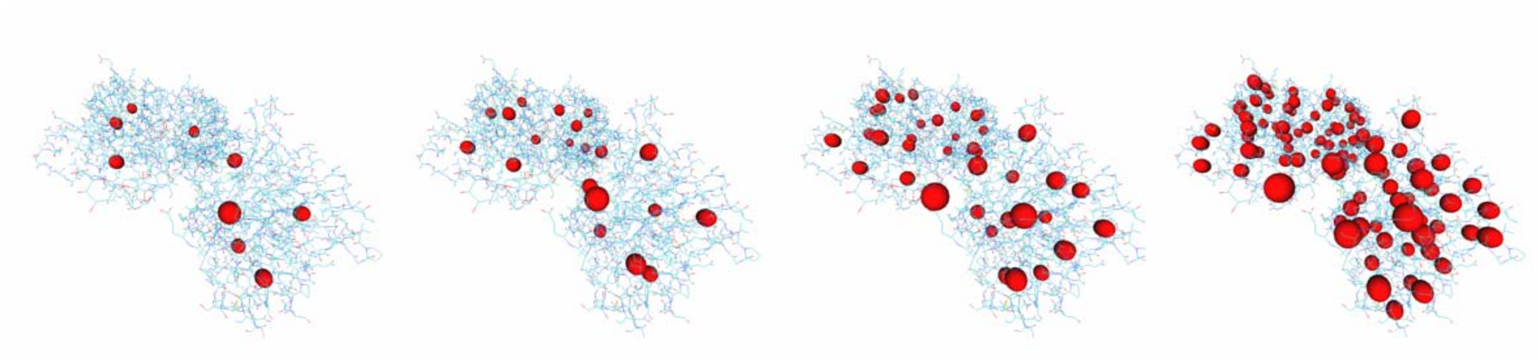


- Not time efficient enough for haptic rendering

Feature-Based Shape Description

- Vector Quantization
  - Popular method in signal processing
  - Replace complex function by compact number of feature vectors
  - Topology Representing Networks (Martinez, Schulten)

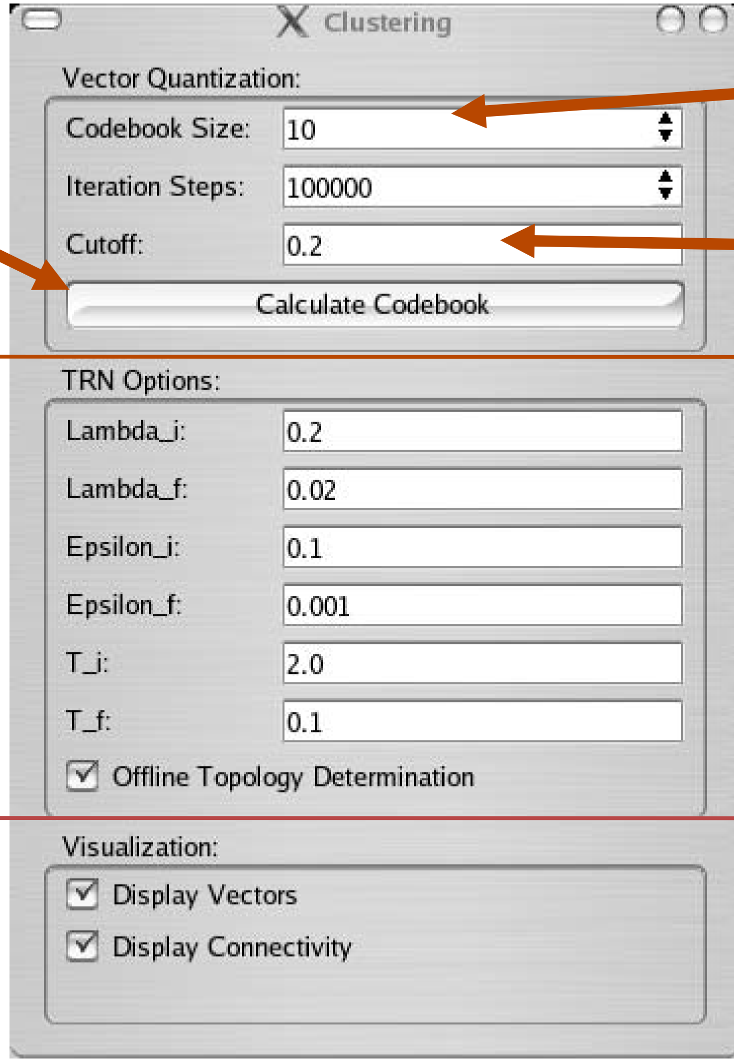
- Applied to high-resolution structure to reduce complexity of fitting problem:



10CV 20CV 40CV 100CV

Feature-Based Shape Description

Start Cluster-Analysis



Number of feature points

Only intensities above the cut-off value are considered

Parameters for the training process of the neural network

Haptic Rendering

- Correlation-based docking:

$$C(R,T) \propto \int \rho_{calc}(\mathbf{r}, R, T) \cdot \rho_{em}(\mathbf{r})\, d^3\mathbf{r}$$

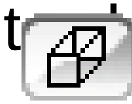
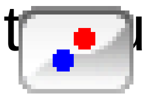
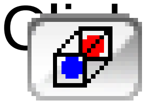
- Feature points:

$$\rho_{calc}(\mathbf{r}) \equiv \sum_{i=1}^{k} \delta(\mathbf{r} - \mathbf{w_i})$$

- Reduced docking criterion:

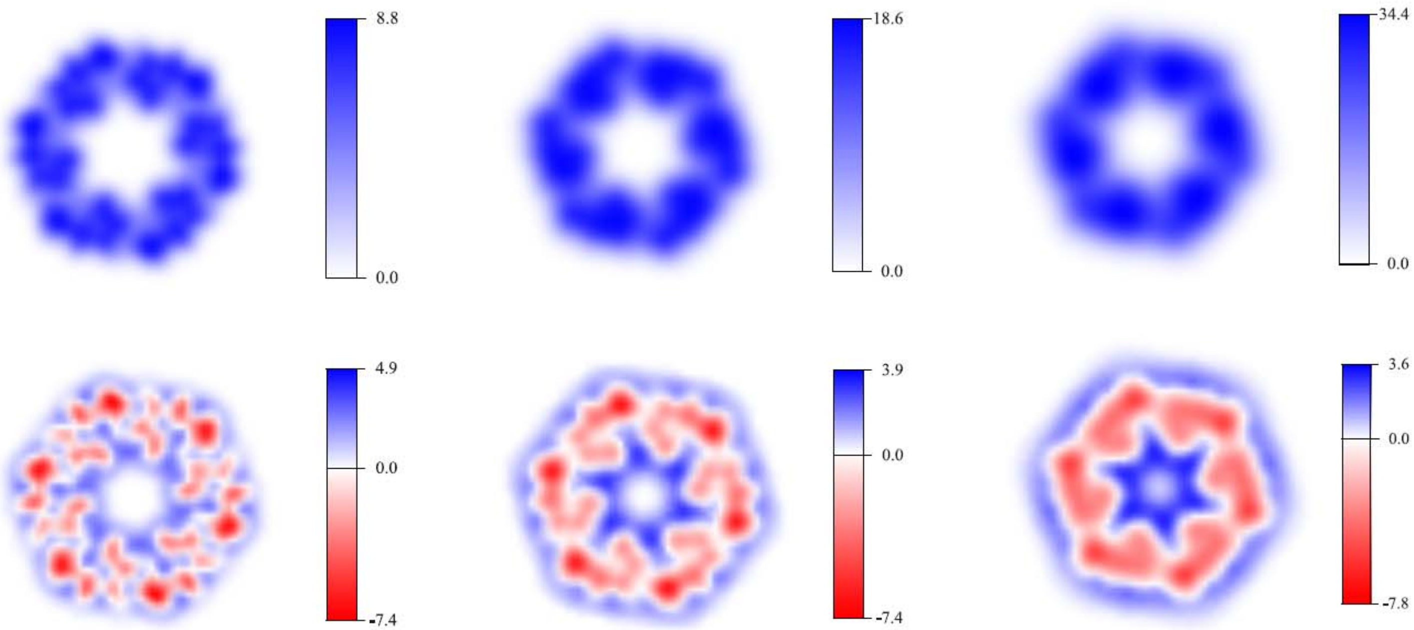
$$C(R,T) \propto \sum_{i=1}^{k} \rho_{em}(\mathbf{w_i}(R,T))$$

Haptic Rendering

- Correlation-based refinement:
  - Force used in gradient descent refinement technique
  - Highlight  umetric data in document list
  - Click on to activate as target map
  - Highlight  ucture data
  - C  on to activate as probe molecule
  - Probe will follow force vector into next local correlation maximum

Laplace Quantization

- Laplacian filter applied to low-resolution cryo-EM maps



$$L : \rho(x,y,z) \rightarrow \nabla^2 \rho(x,y,z) = \rho^L(x,y,z)$$

Laplace Quantization

- Vector Quantization demands remapping:

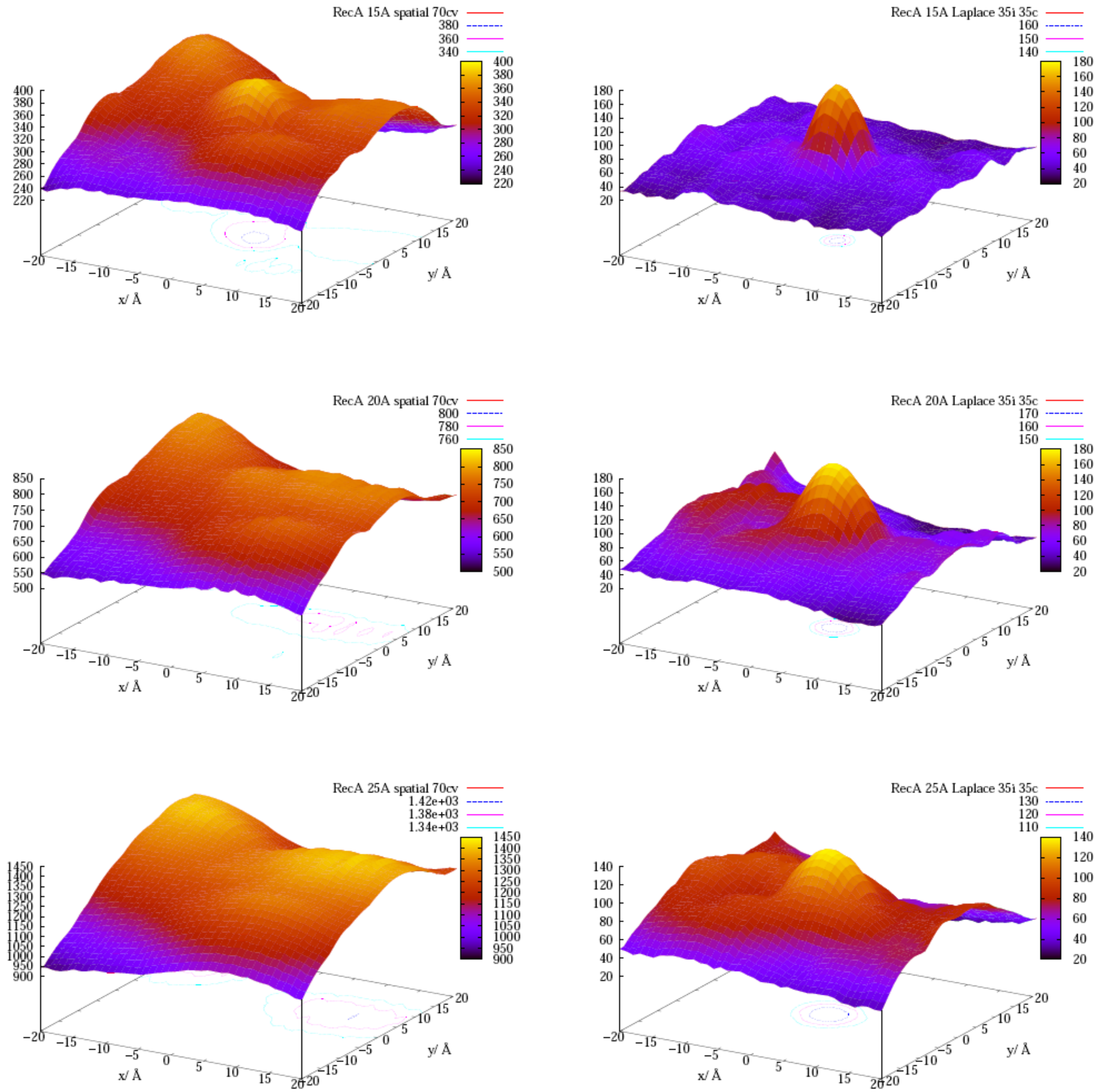
$$\rho_{\mathrm{c}}^{L}(x,y,z) \rightarrow \mathcal{M}(\rho_{\mathrm{c}}^{L}(x,y,z)) \in [0,1]$$

- Leads to separate codebooks for contour and interior

$$\begin{aligned} C^{L}(R,T) &= \int \rho_{\mathrm{c}}^{L}(R,T) \cdot \rho_{\mathrm{EM}}^{L} d^3 r \\ &= \underbrace{\sum_{i=1}^{r} \rho_{\mathrm{EM}}^{L}(w_i^{C}(R,T))}_{\text{contour-match}} - \underbrace{\sum_{i=1}^{s} \rho_{\mathrm{EM}}^{L}(w_i^{I}(R,T))}_{\text{interior-match}} \end{aligned}$$

correlation
on x/y plane
(feature-based)

Spatial
docking

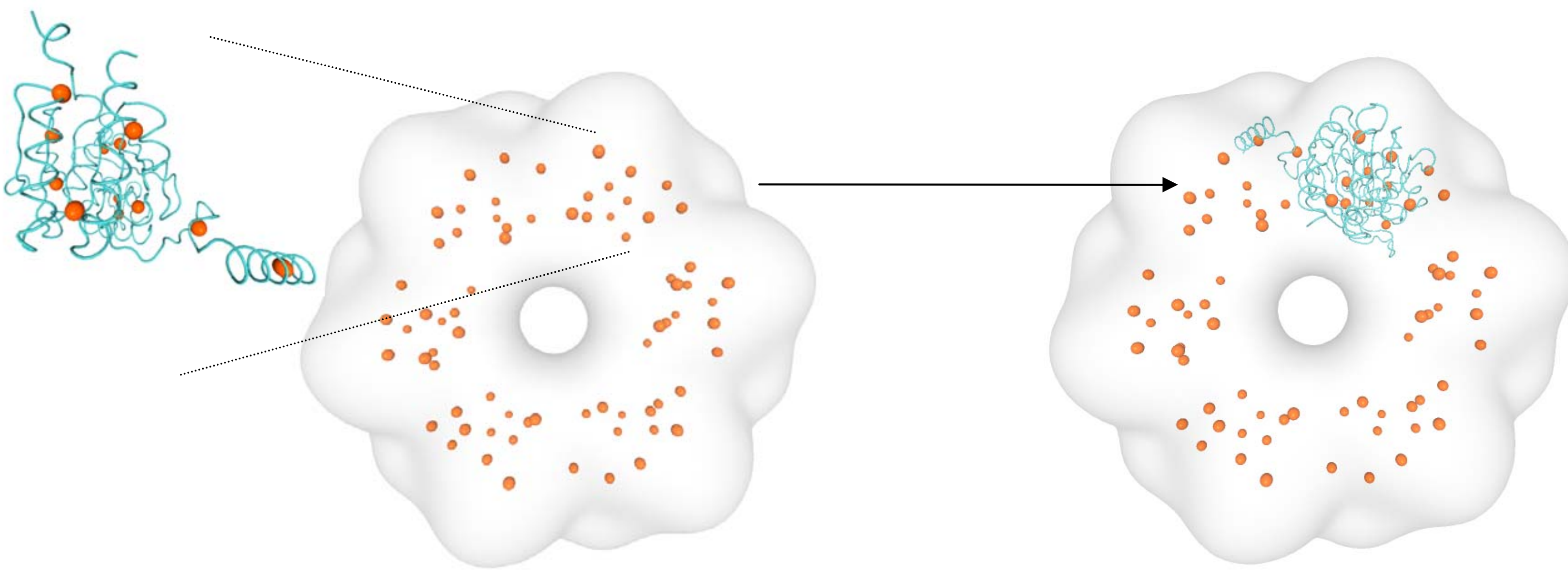


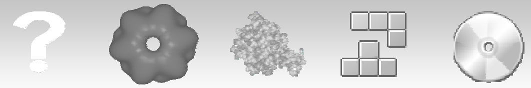
Laplacian
docking

Sculptor

Multi-Resolution Fitting

- Determine feature points in 3D structural and volumetric data
  - Point-cloud similarity alternative docking criterion





Sculptor

Point-based Shape Recognition

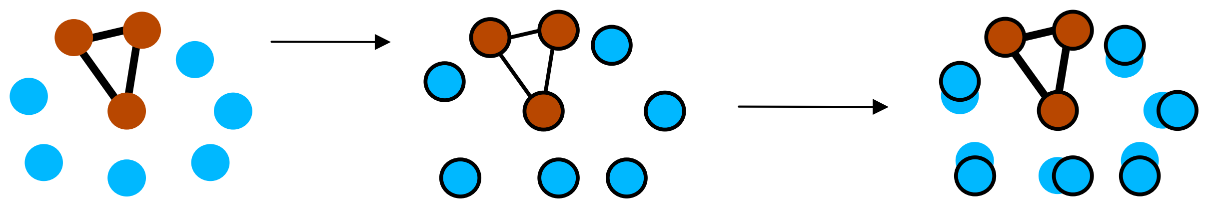
- Feature-based shape description transforms MR-docking into point-cloud matching problem:

$$rmsd(I, \mathbf{R}, \mathbf{T}) = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left\|(\mathbf{R}\mathbf{w}_j^{calc} + \mathbf{T}) - \mathbf{w}_{I(j)}^{em}\right\|^2}$$

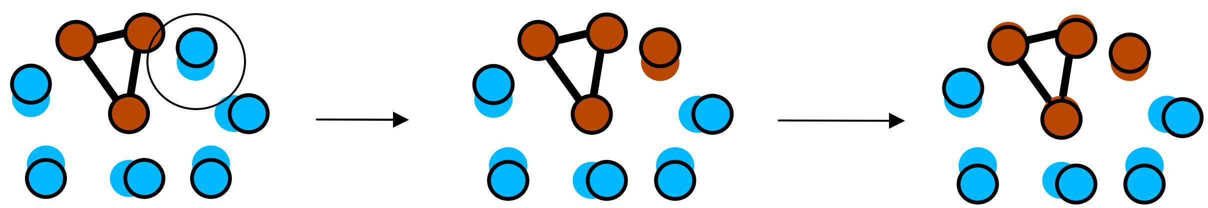
- NP-Hard
- Methods developed in other research areas
  - Structure alignment
  - Pattern matching
  - Computer vision

Anchor-Point Matching

- Anchor-point refinement matching:
  - Three pairs of anchor points give an initial (rough) transformation

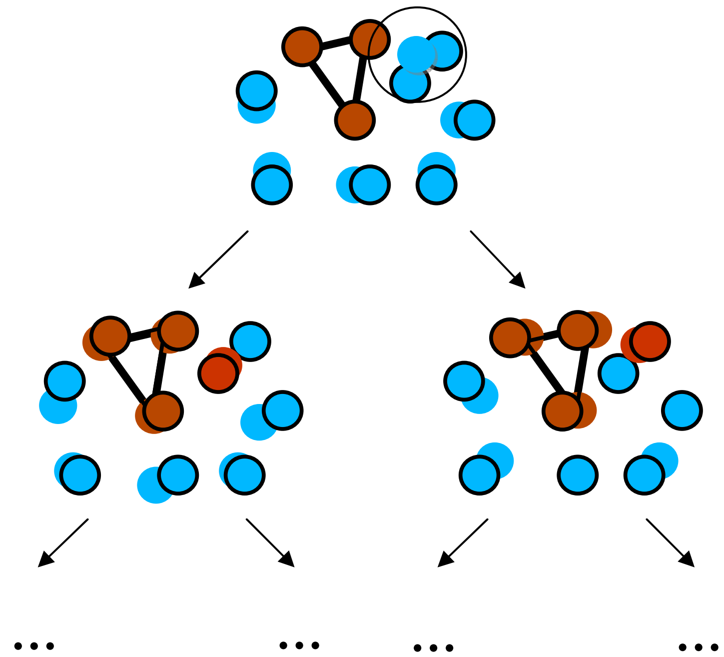


  - Iterative refinement of initial transformation



Search Tree

- Refinement leads to search tree:



- Exploit sparse distribution of feature vectors
  - Compact tree, typical runtime < 1min

Installation Instructions

- Download:
  - http://sculptor.biomachina.org

- Windows:
  - setup.exe - standard installer
- Linux:
  - RPM package for Fedora Core Linux:
    - rpm -i qwt-xxxx.rpm
    - rpm -i sculptor-xxxx.rpm
  - Compile your own package for other distributions:
    - rpmbuild -ba sculptor.spec