Repairing Inconsistent Curve Networks on Non-parallel Cross-sections

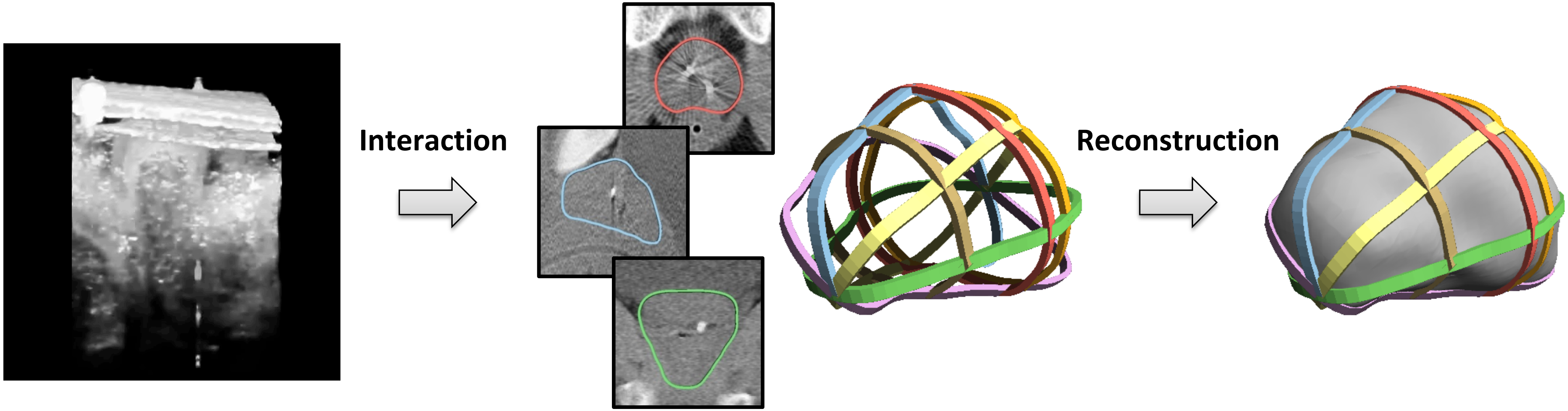
Z. Y. Huang[1], M. Holloway[1], N. Carr[2], T. Ju[1]

[1] Washington University in St. Louis, USA

[2] Adobe Inc., USA

EUROGRAPHICS 2018

Motivation: Image segmentation

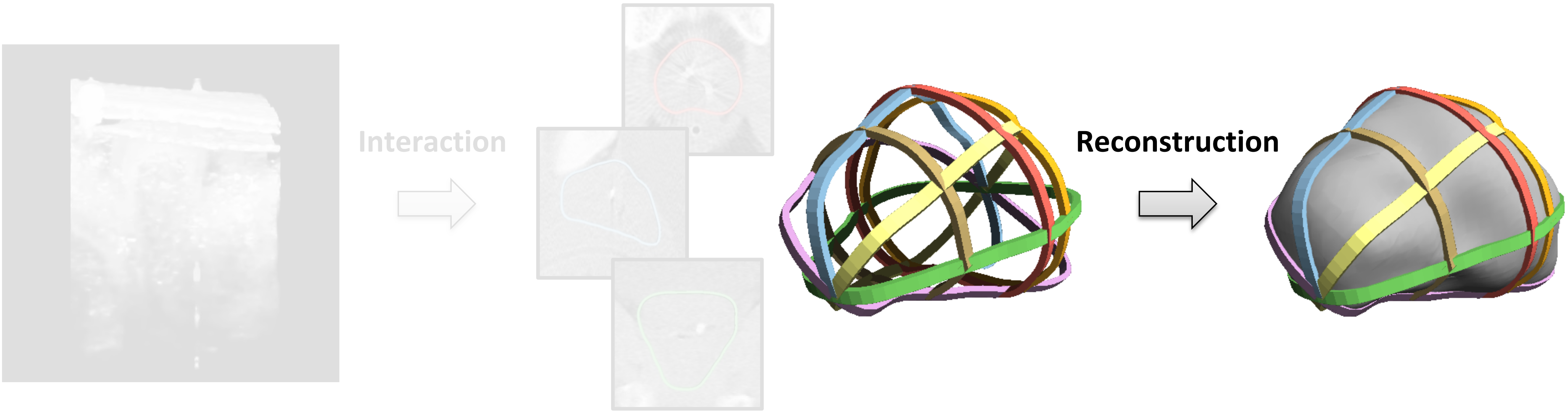


3D Image Volume

Contours on cross-sections

Segmented shape

Motivation: Image segmentation



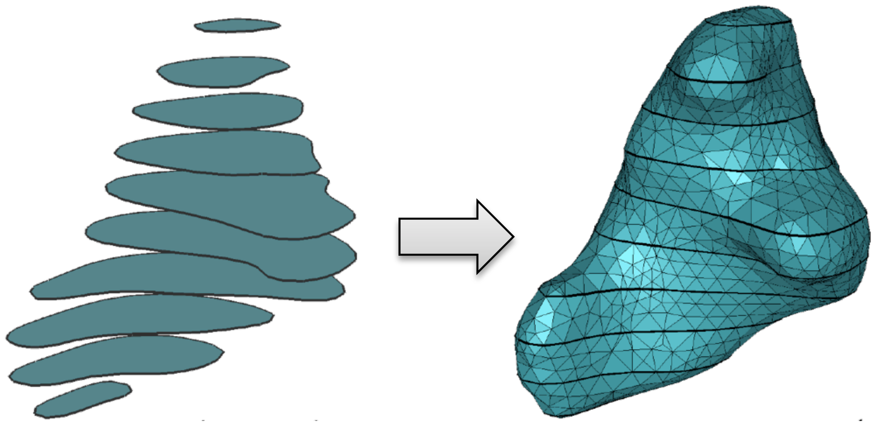
3D Image Volume

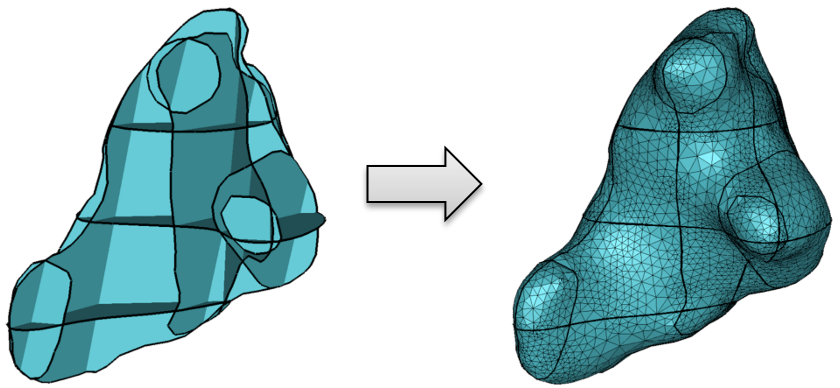
Contours on cross-sections

Segmented shape

Background: Reconstruction from cross-sections

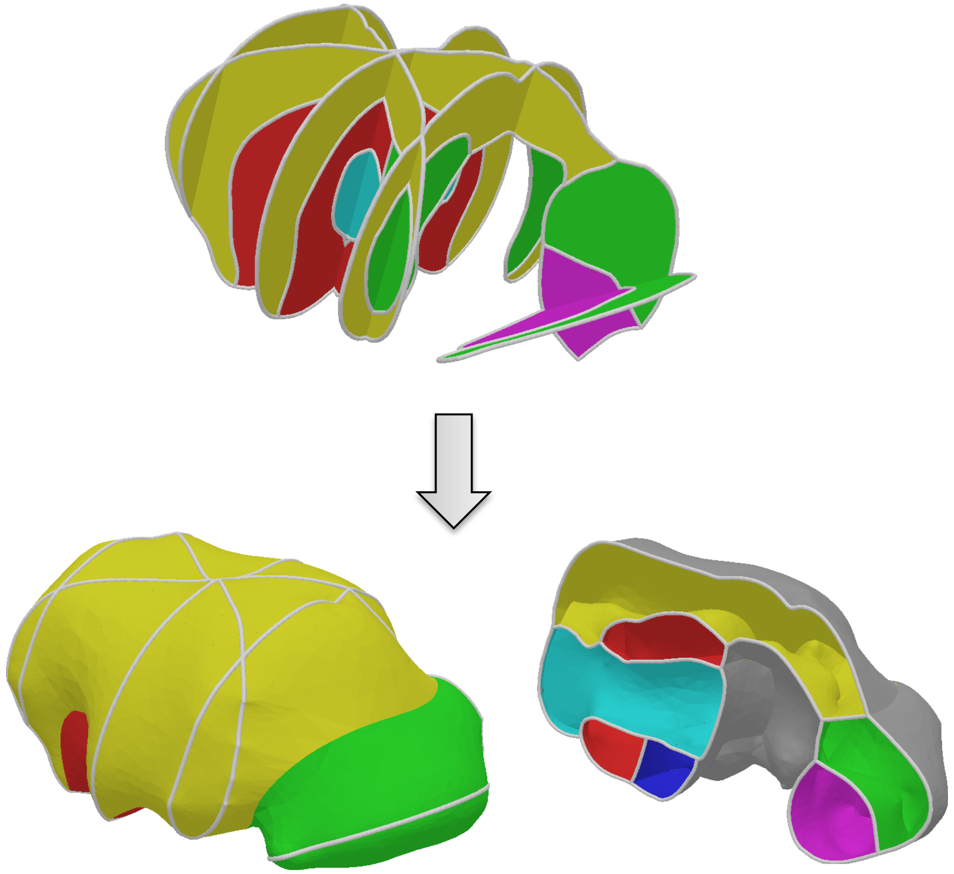
- A well-studied problem dated back to 70s
- Parallel planes
  - Natural choice for 3D images, but may require many cross-sections to describe shape
- Non-parallel planes
  - Well-chosen planes can describe shape with fewer cross-sections [Boissonnat 07, Liu 08, Barequet 09, Bermano 11, Heckel 11, Zou 15, Holloway 16, Huang 17]





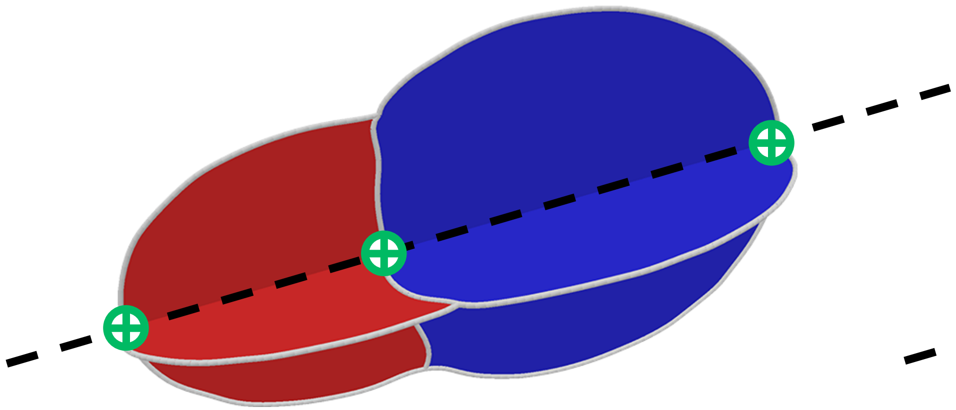
Background: Reconstruction from cross-sections

- A well-studied problem dated back to 70s
- Parallel planes
  - Natural choice for 3D images, but may require many cross-sections to describe shape
- Non-parallel planes
  - Well-chosen planes can describe shape with fewer cross-sections [Boissonnat 07, Liu 08, Barequet 09, Bermano 11, Heckel 11, Zou 15, Holloway 16, Huang 17]
  - Extension to model multi-labelled domains from multi-labelled cross-sections

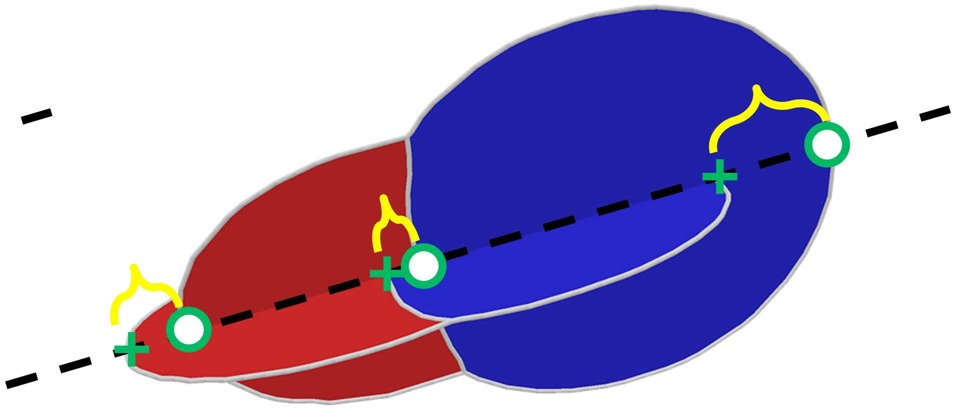


Background: Consistency

- Methods handling non-parallel cross-sections require consistent input
  - Intersecting cross-sections share the same labelling along the intersection line



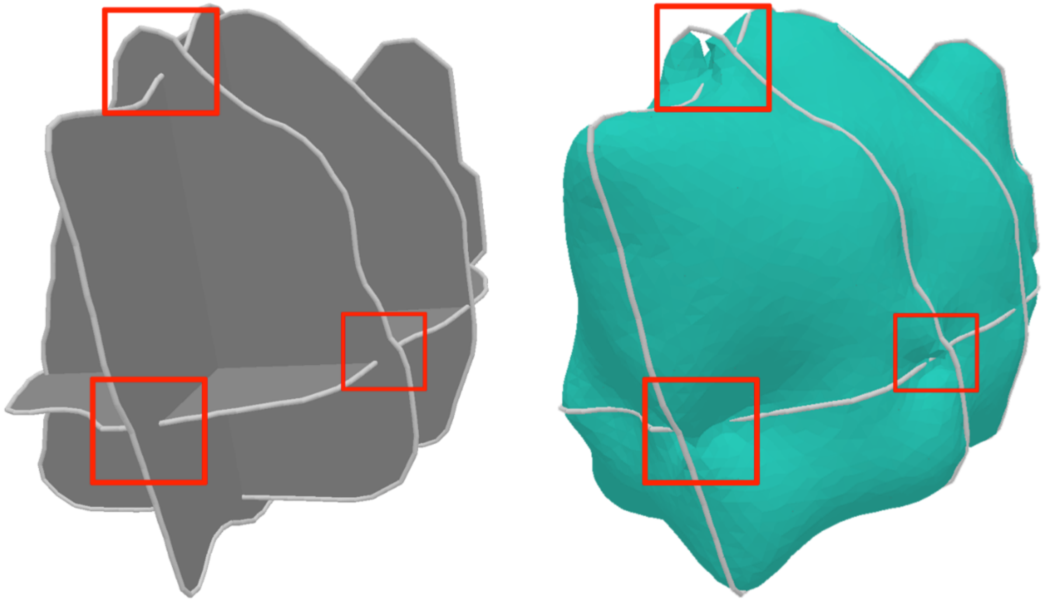
Consistent



Inconsistent

Background: Consistency

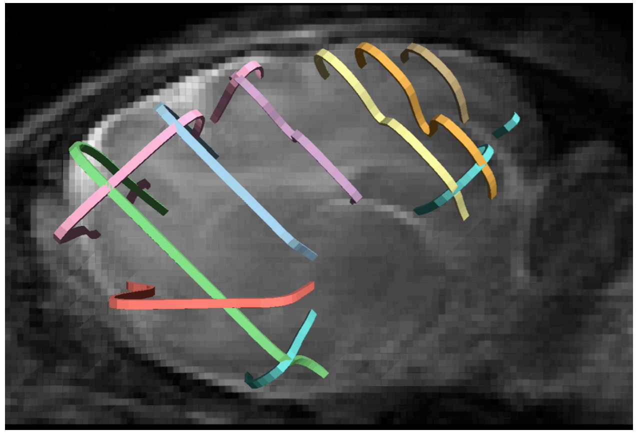
- All interpolating methods fail on inconsistent cross-sections
- Approximating methods work, but create surface artifacts

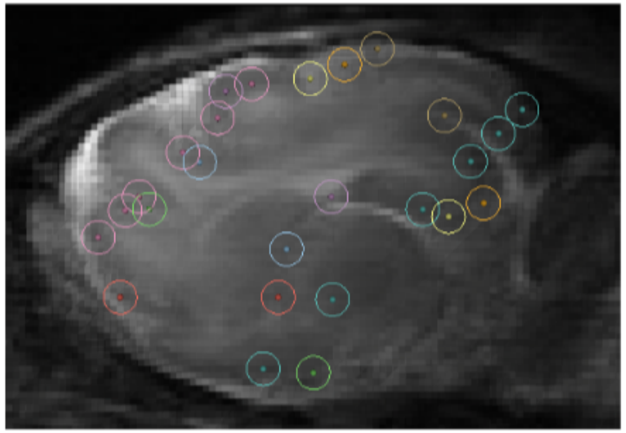


[Bermano 11]

Background: where does inconsistency come from?

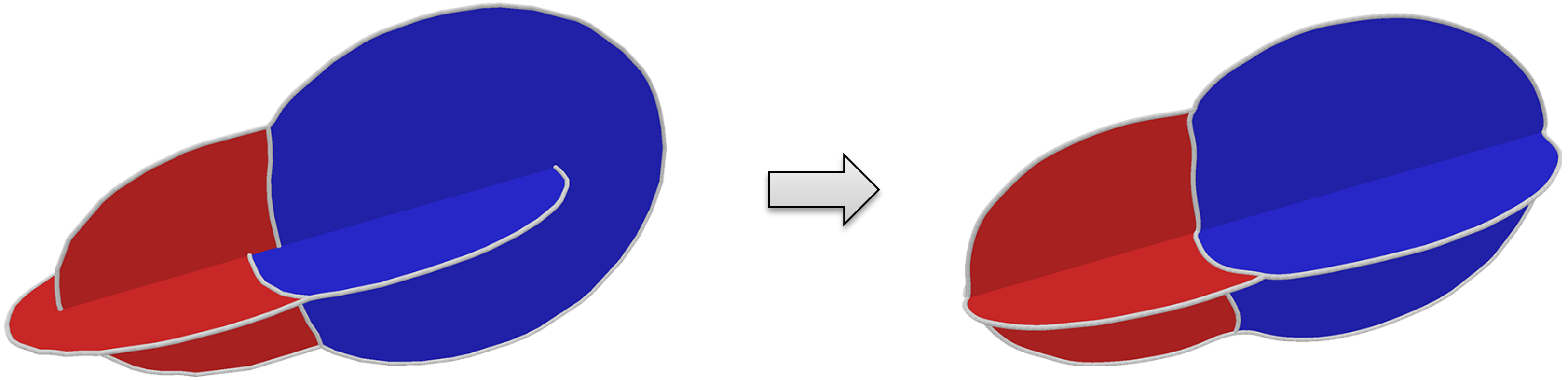
- Cross-sections are often created independently from each other
- We can ask the users/software to be more careful. But…
  - Adds labor and distraction
  - Requires changes to existing software
  - Cannot process existing data





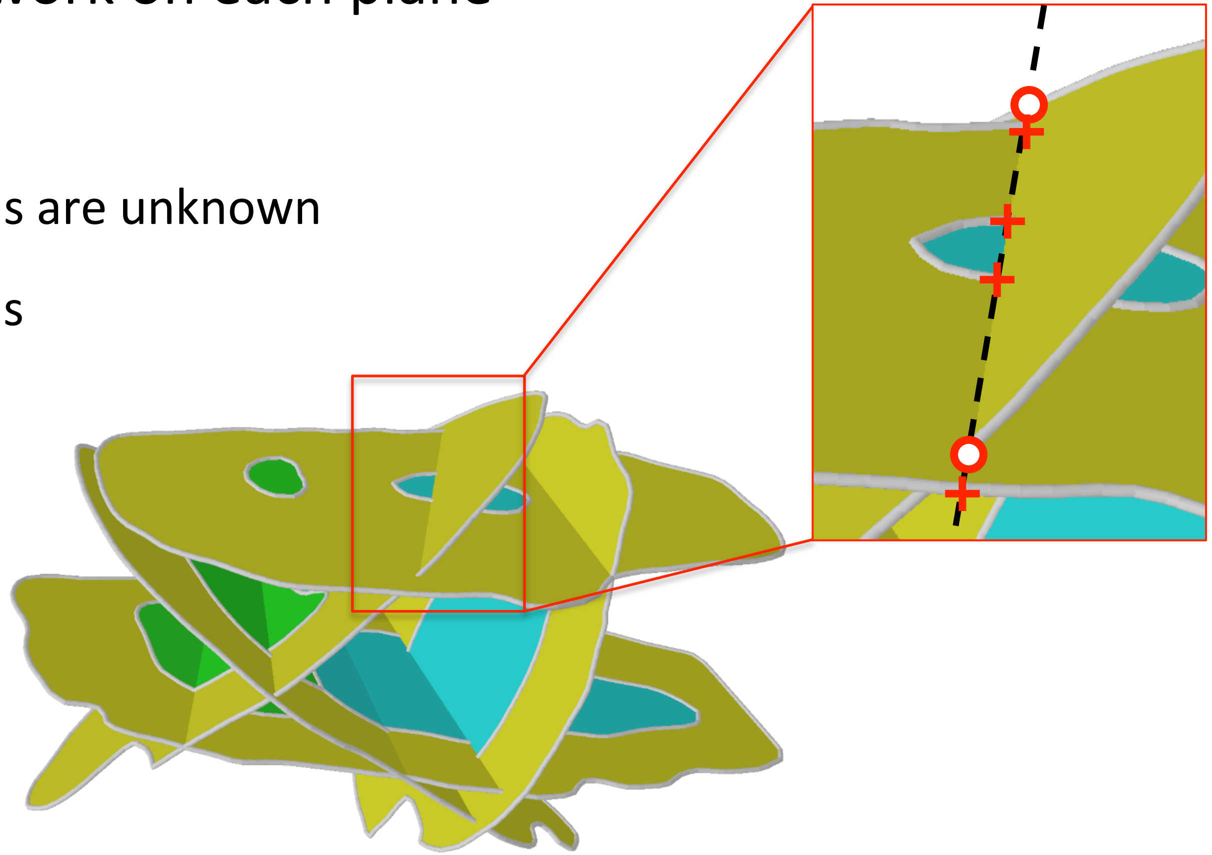
Objective

- Given a set of (possibly inconsistent) multi-labelled non-parallel cross-sections
- Modify curves on each cross-section to restore consistency



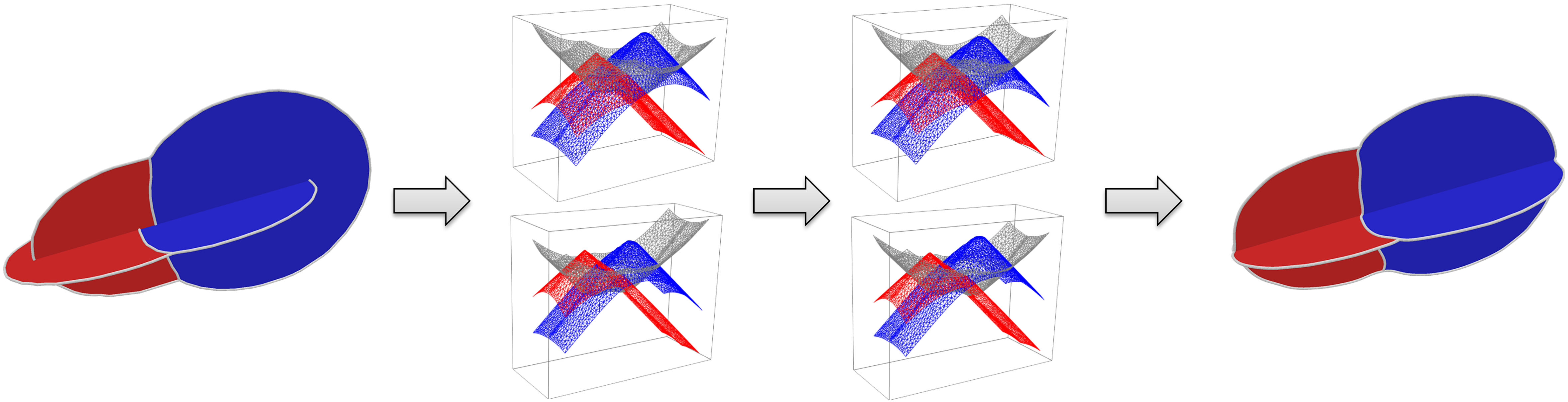
Explicit approach

- Geometric deformation of the curve network on each plane
- Difficult to enforce consistency
  - Both the number and location of intersections are unknown
  - Deformation may introduce new intersections
- Cannot change curve network topology
  - May result in large deformations



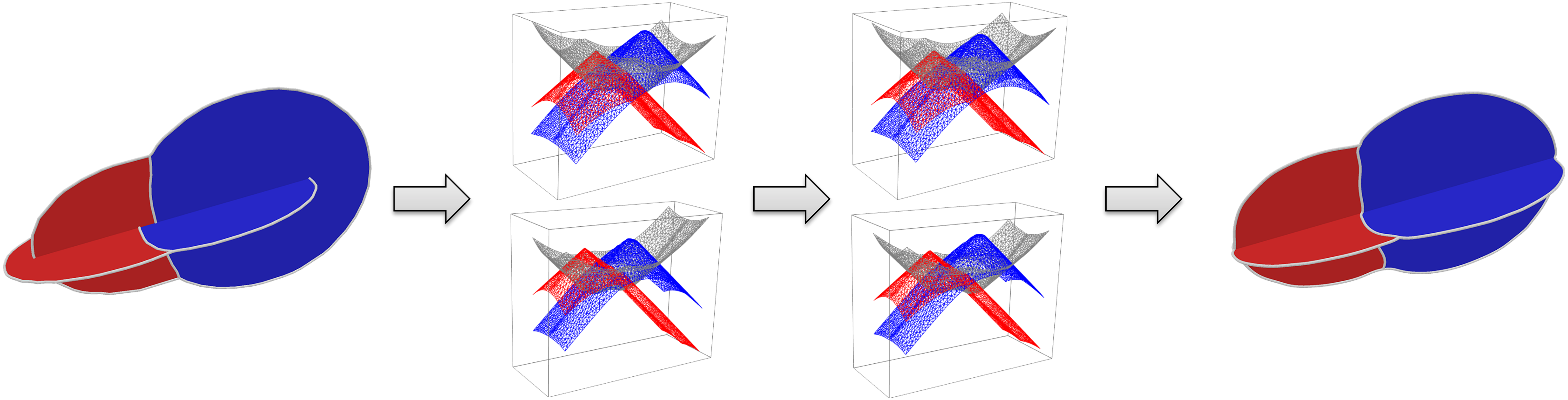
Implicit approach

- Represent the curve network on each cross-section by an implicit function
- Modify the implicit functions
- Reconstruct the curve networks from the modified functions



Implicit approach

- Easy to enforce consistency
  – As inequality constraints on the implicit functions
- Flexible in topological changes

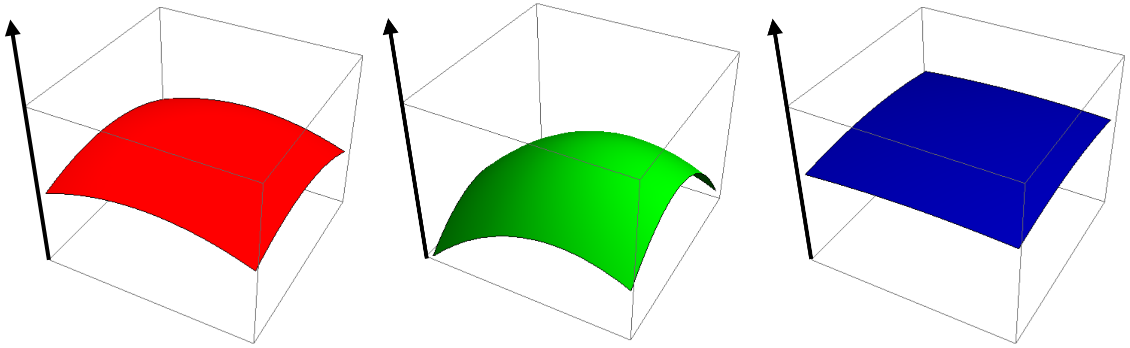


Implicit representation

- Representing a n-labelled plane:
  [Losasso 06, Feng 10, Yuan 12, Huang 17]
  - Define n scalar functions $f_1(x),\ldots,f_n(x)$
  - Label as index of the function that achieves maximum value:

$$Label(x)=\operatorname{argmax}_i f_i(x)$$

  - Labelled regions are bounded by a non-manifold curve network



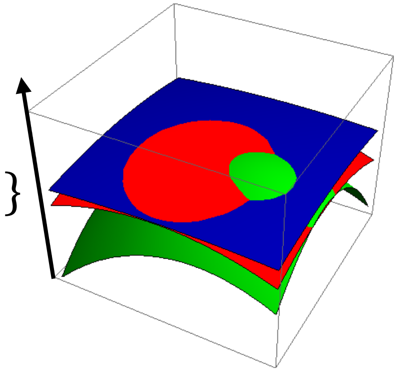
Implicit representation

- Representing a n-labelled plane: [Losasso 06, Feng 10, Yuan 12, Huang 17]
  - Define n scalar functions $f_1(x),\dots,f_n(x)$
  - Label as index of the function that achieves maximum value:

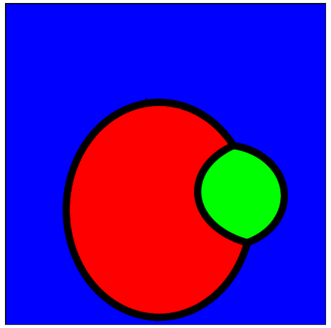
$$Label(x) = \operatorname{argmax}_i f_i(x)$$

  - Labelled regions are bounded by a non-manifold curve network

$\{f_1, f_2, f_3\}$

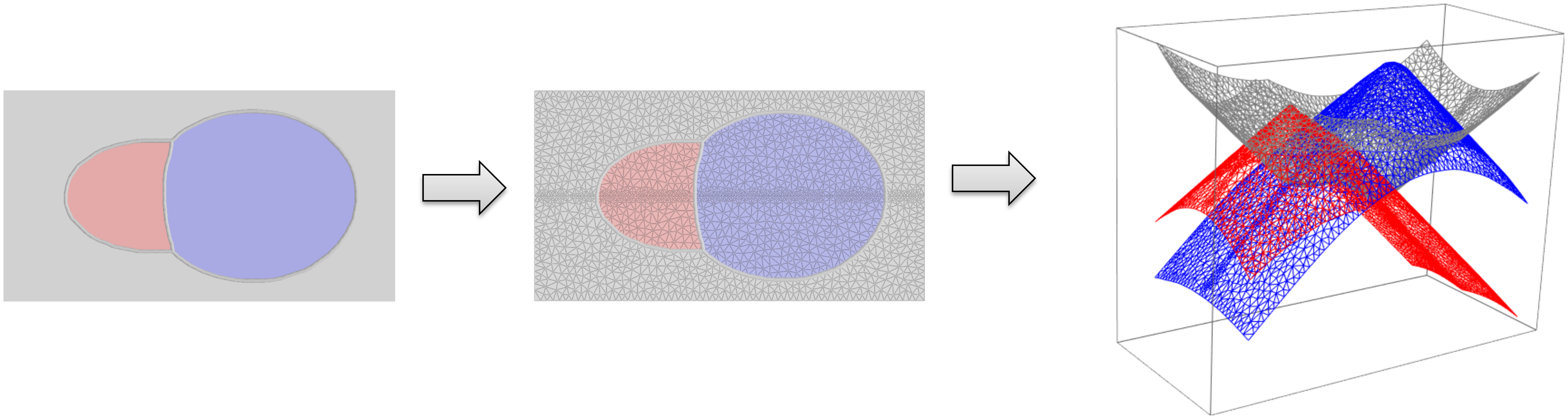


$Label$



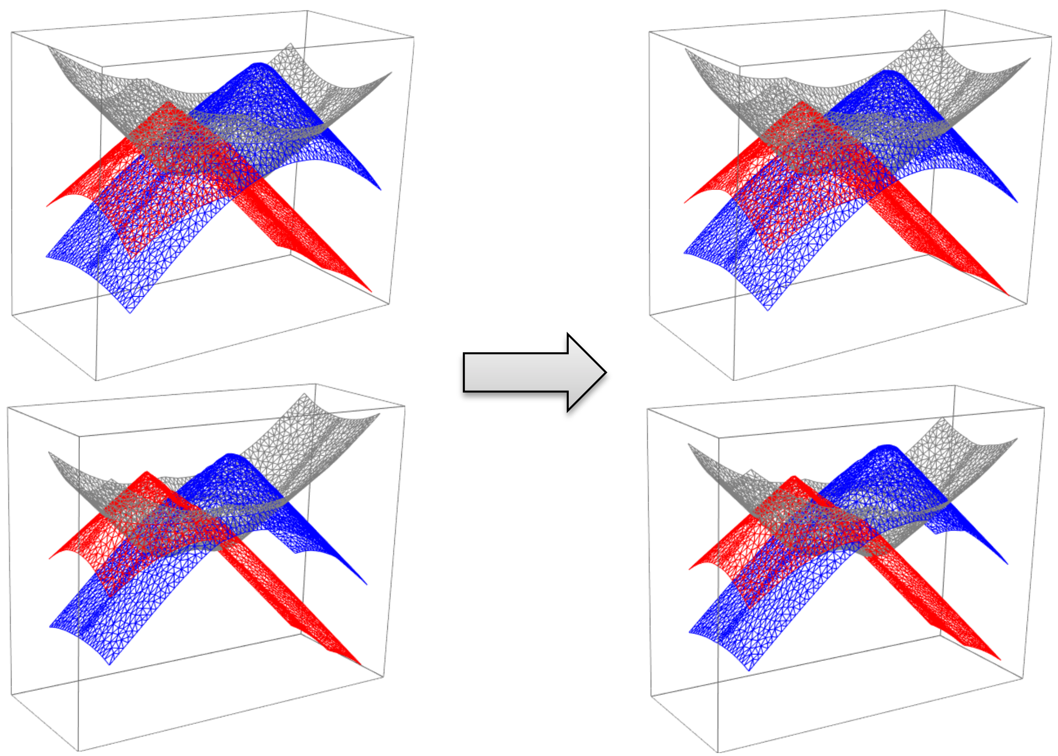
Implicit representation

- Define initial scalar functions as signed distance functions
  - Triangulate each cross-section
  - Compute $f_i^P(v)$ for label i at vertex v on plane P as signed distance to boundaries of label i



Problem formulation

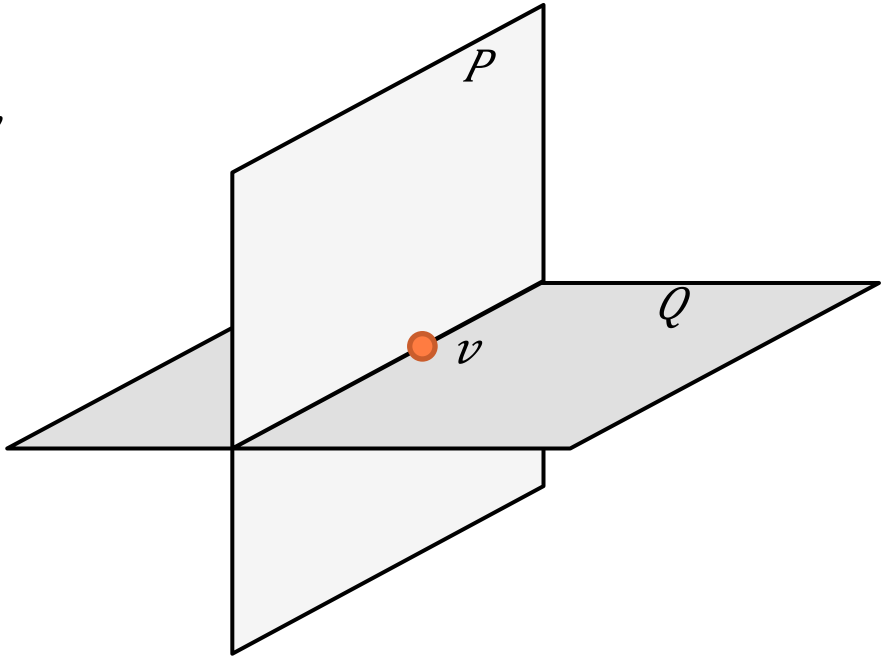
- Given implicit functions on each cross-section
- Modify the functions so that:
  - Labelling is consistent on intersection lines
  - Distortion to curve networks is minimized



Consistency constraints

- Consider vertex v on the intersection line between cross-sections P,Q
  - $f_{1,\dots,n}^{P}(v), f_{1,\dots,n}^{Q}(v)$: scalar values on plane P,Q
- Suppose the final label at v is known, $l(v)$
  - Then function value of $l(v)$ is greater than any other label at v

$$f_{l(v)}^{P}(v) \geq f_{i}^{P}(v)+\varepsilon,\ f_{l(v)}^{Q}(v) \geq f_{i}^{Q}(v)+\varepsilon,$$
$$\forall i \neq l(v)$$



- Since we don't know $l(v)$, we leave it as a variable.

Deformation energy

- Deviation from input curve networks
  - How far have the curves moved? (zero order)
  - How much have the curve tangents changed? (1st order)

$$E(f) = \lambda \sum_{P,\ v,\ i} \left(f_i^P(v) - f_i^P(v) \right)^2 \quad \text{(zero order)}$$

$$+ \sum_{P,\ v,\ i,j} \left(G(f_i^P - f_j^P)(v) - G(f_i^P - f_j^P)(v) \right)^2 \quad \text{(1st order)}$$

f : input function; L: discrete gradient

Mixed-Integer Programming (MIP)

- Continuous variables: $f_i^P(v)$ // *Implicit function values at all vertices*
- Integer variables: $l(v)$ // *Labels at vertices on intersection lines*
- Minimize: $E(f)$ // *Quadratic deformation energy*
- Subject to: $f_{l(v)}^P(v) \geq f_i^P(v) + \varepsilon$ // *Consistency constraints*

Optimization

- MIPs are computationally expensive to solve
- We propose an efficient solution strategy by iteratively solving Quadratic Programming problems

Quadratic Programming (QP)

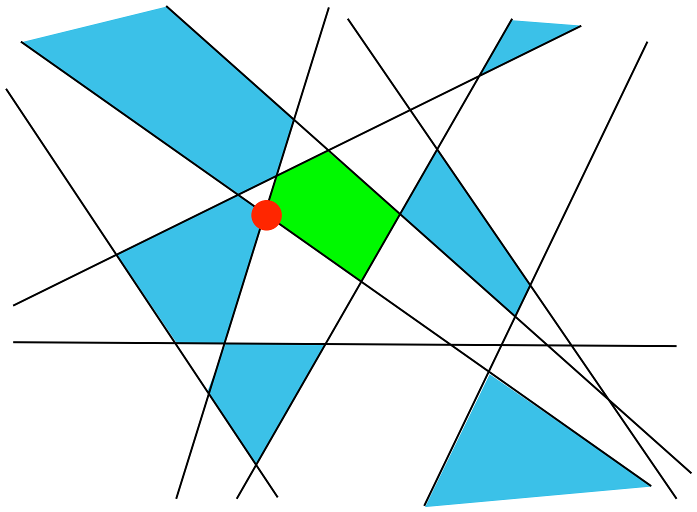
- For a given set of labels $l(v)$ at each vertex v on intersection lines:
  - Continuous variables: $f_i^P(v)$ *// Implicit function values at all vertices*
  - Minimize: $E(f)$ *// Quadratic deformation energy*
  - Subject to: $f_{l(v)}^P(v) \geq f_i^P(v) + \varepsilon$ *// Consistency constraints*

Optimization strategy

- Start with an initial set of labels on the intersection lines
  - By averaging values from multiple planes
- Solve QP
- Update labels and repeat
  - Until energy no longer decreases

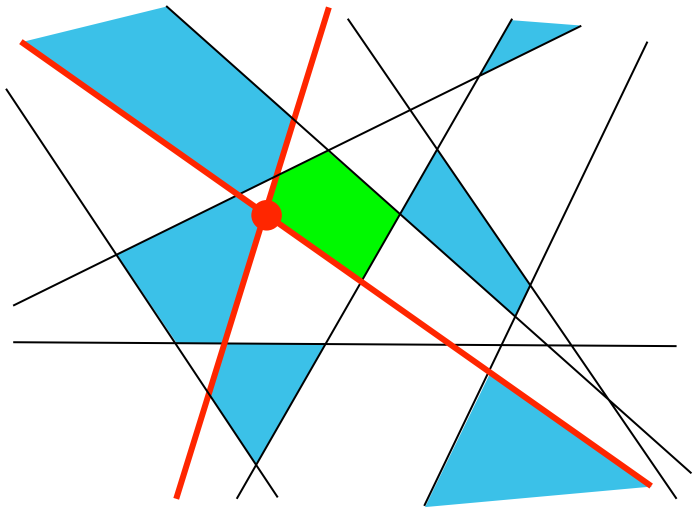
Updating labels

- A set of labels defines a set of inequality constraints
  - A convex cell in the solution space
- Minimizer of QP lies on the boundary of the cell
  - Otherwise, the input is already consistent



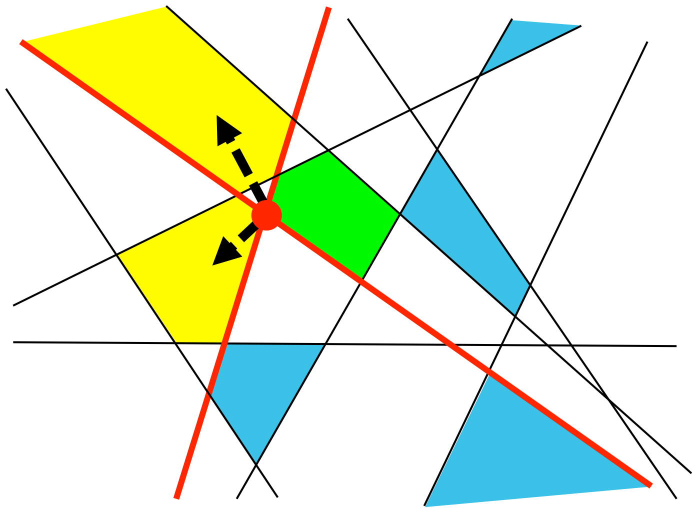
Updating labels

- A set of labels defines a set of inequality constraints
  - A convex cell in the solution space
- Minimizer of QP lies on the boundary of the cell
  - Otherwise, the input is already consistent
  - Each hyperplane containing the minimizer corresponds to a pair of labels $l(v), i$ with similar values at some vertex v
  - Setting $l(v)=i$ potentially lowers the energy



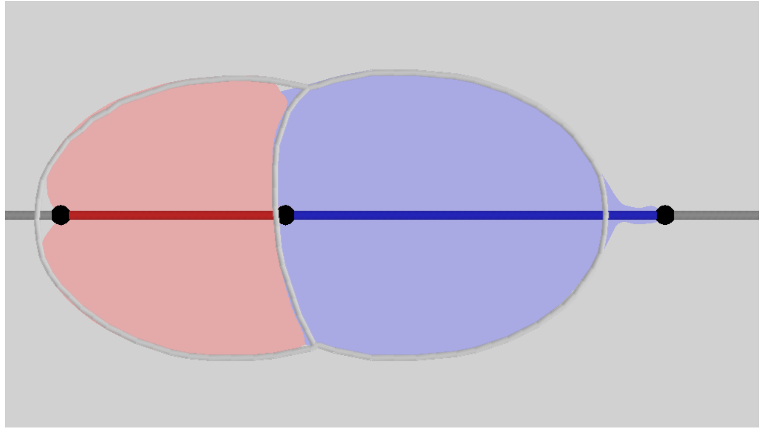
Updating labels

- Sort all hyperplanes by magnitude of energy gradient across the hyperplane
- Visit each hyperplane, flip label, and compute QP of the new label set
- Take the next label set as the first hyperplane with positive reduction in energy

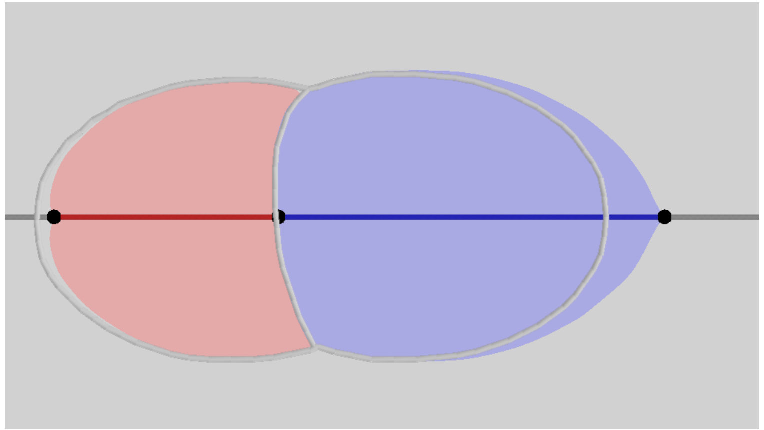


Experiments: Parameter selection

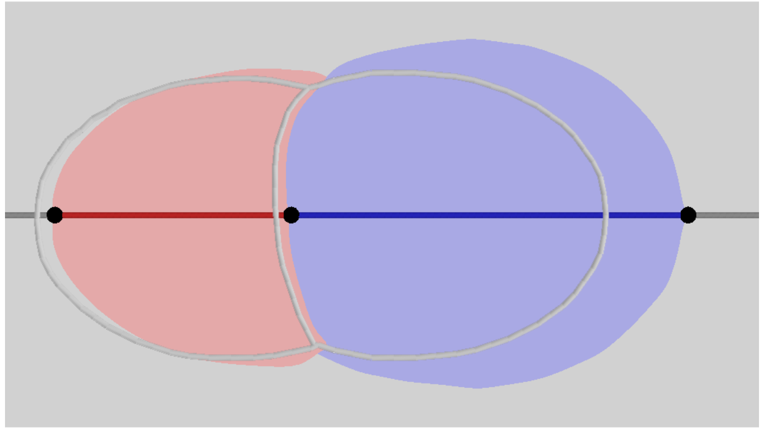
- Choosing λ: trade-off proximity with shape preservation
  - Energy = λ * 0-order difference + 1st-order difference



λ=100



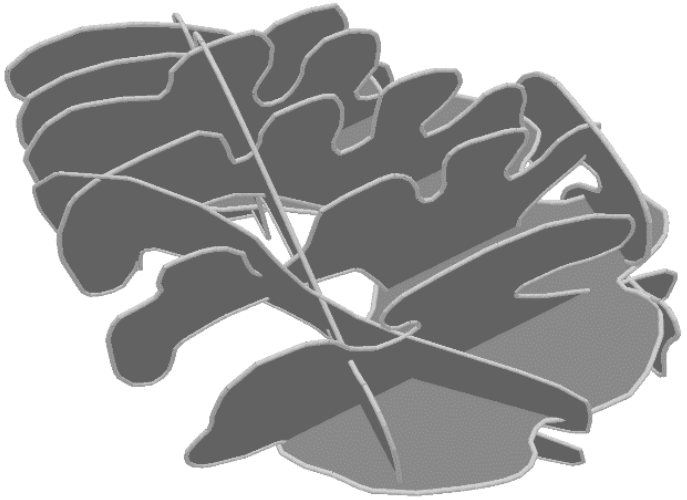
λ=1



λ=0.01

Experiments: Performance

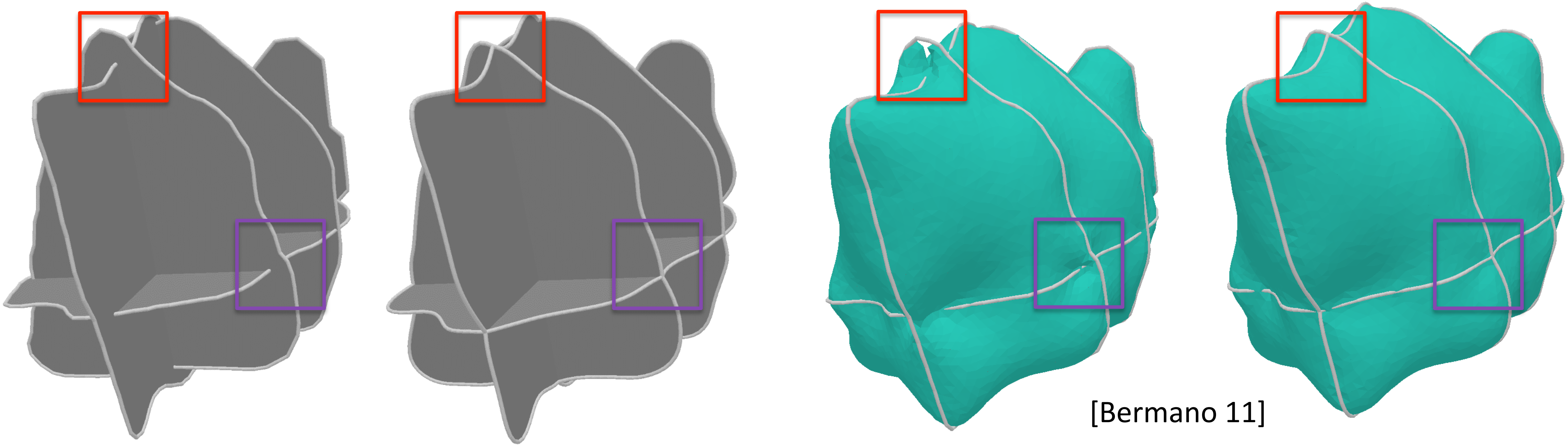
- Comparing with off-the-shelf MIP solver (Gurobi)
  - 2-labels input: increasing number of cross-section planes
  - Our method produces similar energy but using significantly less time



# planes	Our energy	Gurobi energy	Our Time (s)	Gurobi Time (s)
2	16.65	16.65	0.845	1.05
3	24.95	24.95	1.253	11.28
4	25.02	25.03	3.024	33.16
5	29.55	29.55	33.218	619.91

Experiments: More examples

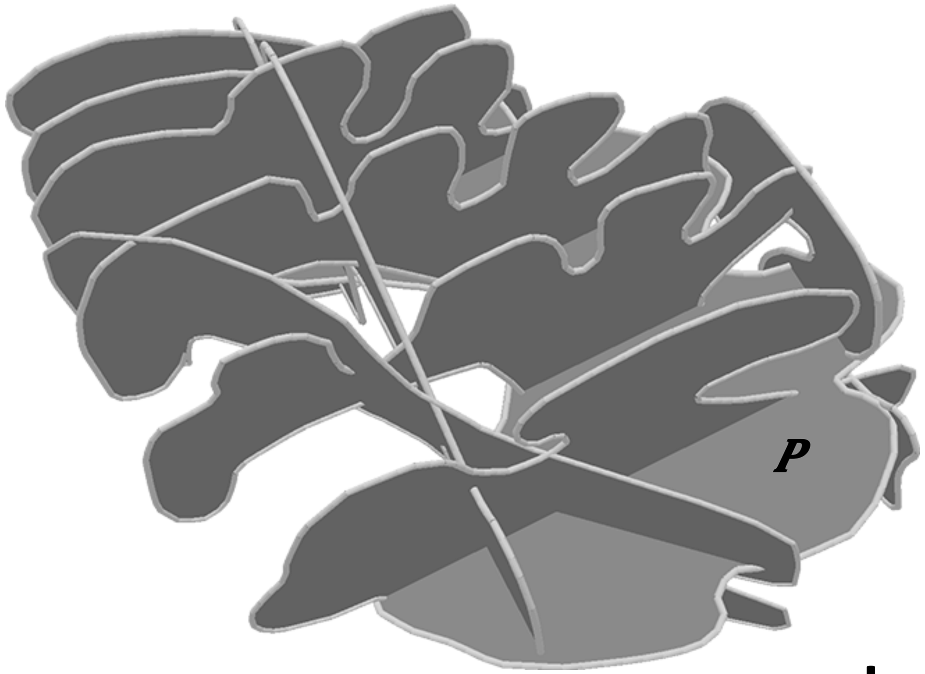
- Atrium (2 labels, 5 planes, time: 1 sec)

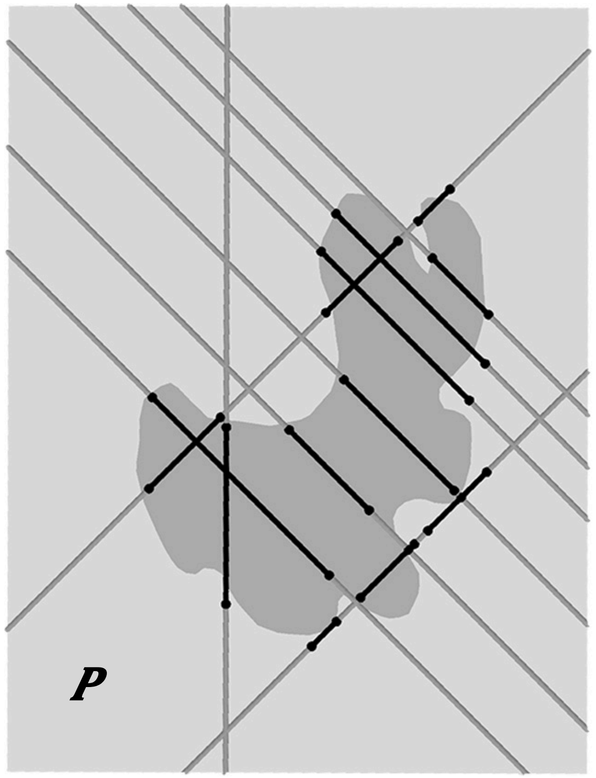


Input | Consistent output | Surface from input | Surface from output

Experiments: More examples

- Ferret brain (2 labels, 10 planes, time: 66 sec)

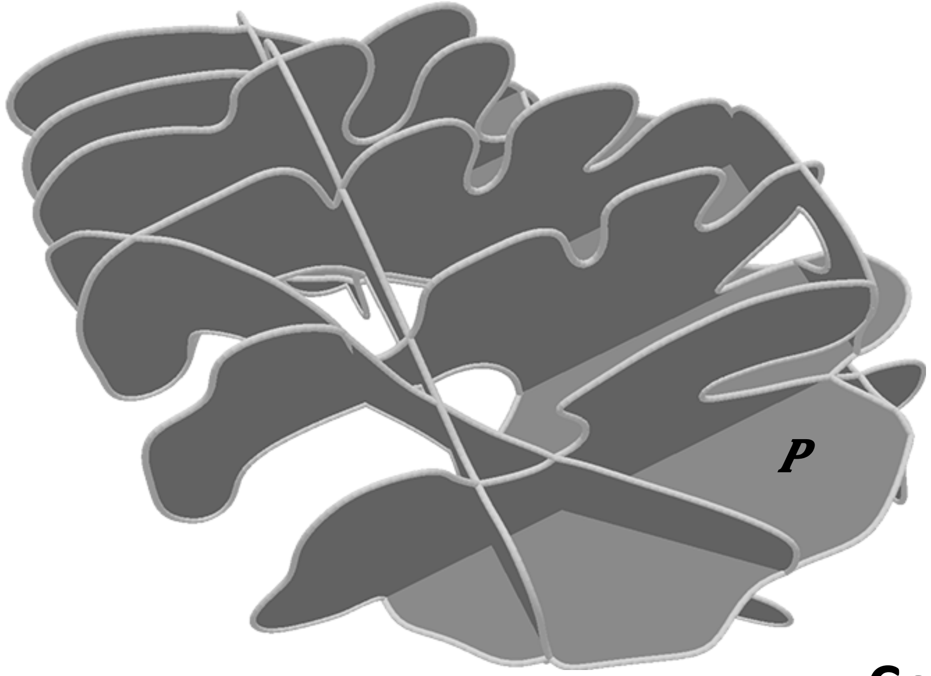


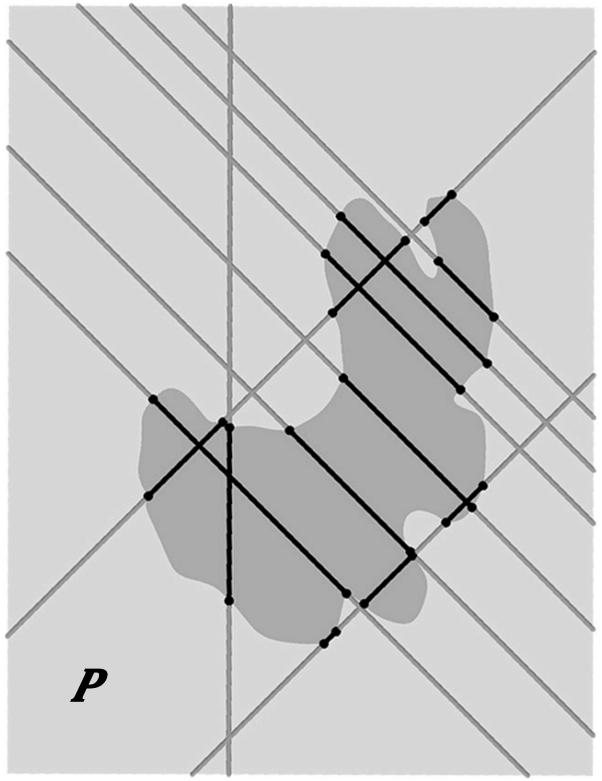


Inconsistent

Experiments: More examples

- Ferret brain (2 labels, 10 planes, time: 66 sec)

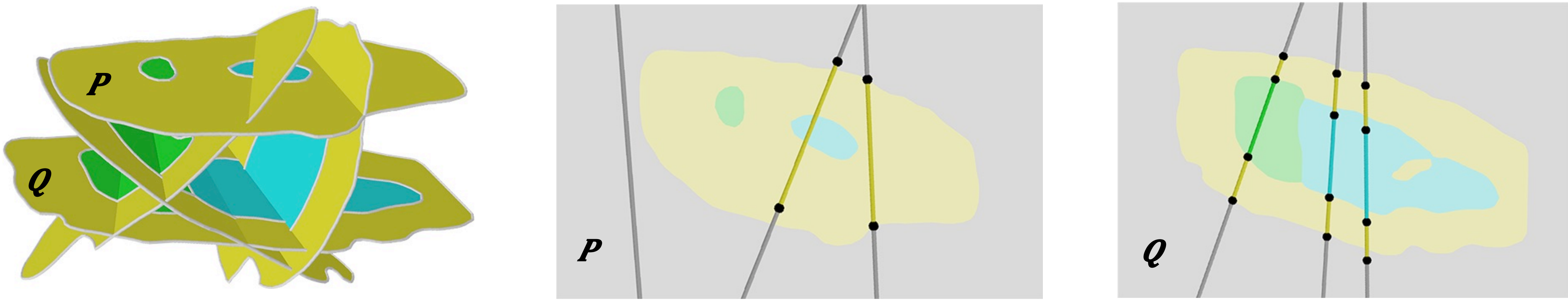




Consistent

Experiments: More examples

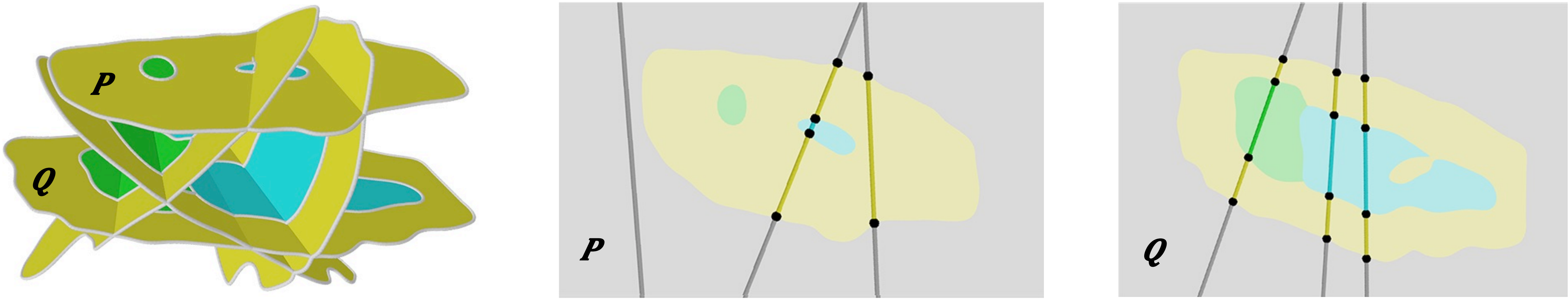
- Livers (5 planes, 4 labels , total time: 25s)



Inconsistent

Experiments: More examples

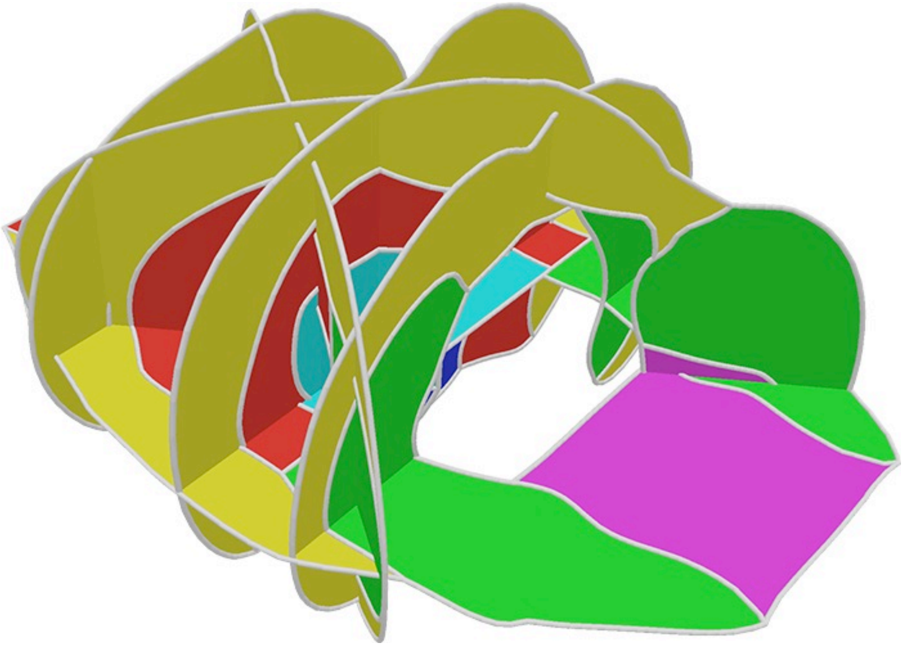
- Livers (5 planes, 4 labels , total time: 25s)



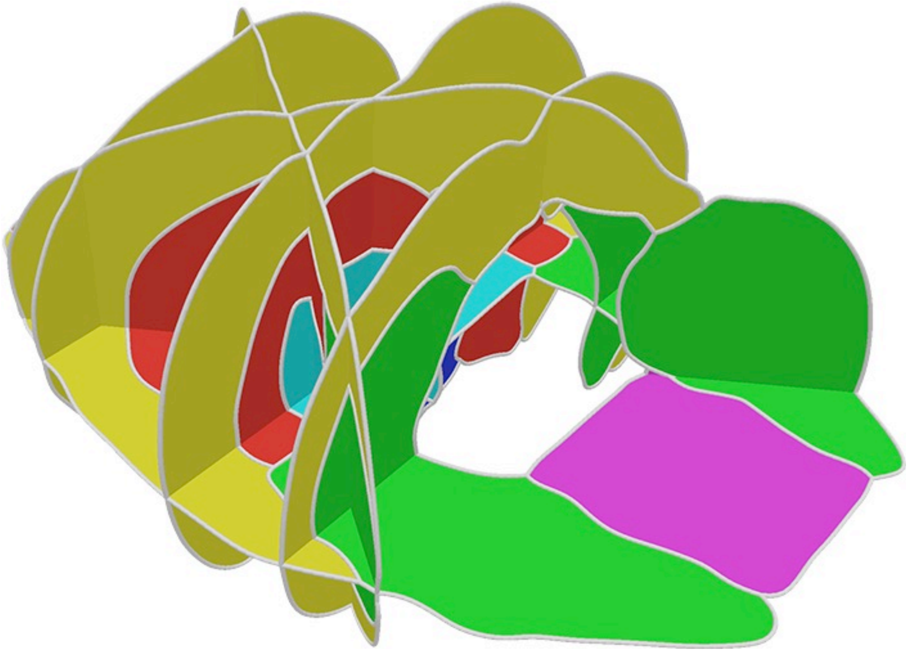
Consistent

Experiments: More examples

- Mouse brain (6 planes, 7 labels, total time: 421s)



Inconsistent



Consistent

Conclusion

- An algorithm for restoring consistency to non-parallel cross-sections
  - Formulating and solving an MIP on implicit functions
  - Allowing existing surface reconstruction methods to work on imperfect cross-section inputs

- Limitations and future work
  - Improving deformation energy to better preserve smooth/sharp features
  - Integration into interactive volume segmentation (real-time feedback to users)