Use of 3D Graph-Theoretic Approaches in the Segmentation of Ophthalmic Structures

Mona K. Garvin, Ph.D.
Department of Electrical and Computer Engineering
The University of Iowa

VA Center for the Prevention and Treatment of Visual Loss

Collaborators involved in graph segmentation work

Faculty at Iowa:

- Michael Abramoff
- Reinhard Beichel
- Mona Garvin *(me)*
- Andreas Wahle
- Milan Sonka
- Xiaodong Wu

Faculty at Notre Dame:

- Danny Chen

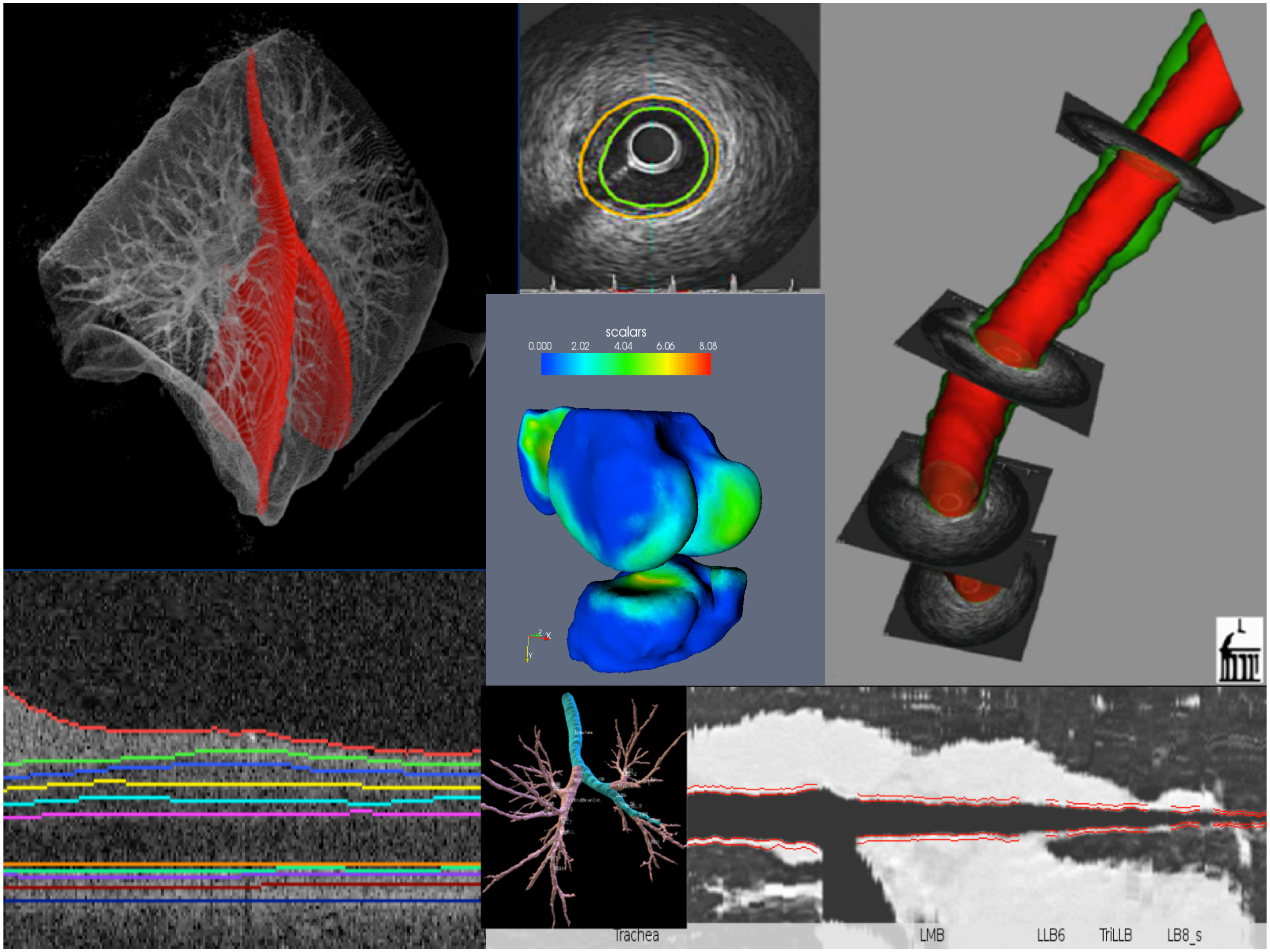
Students and post-docs (current/prior):

- Bhavna Antony
- Xin Dou
- Matt Gibson
- Dongfeng Han
- Zhihong Hu
- Kyungmoo Lee
- Qi Song
- Yin Yin
- Honghai Zhang
- Fei Zhao

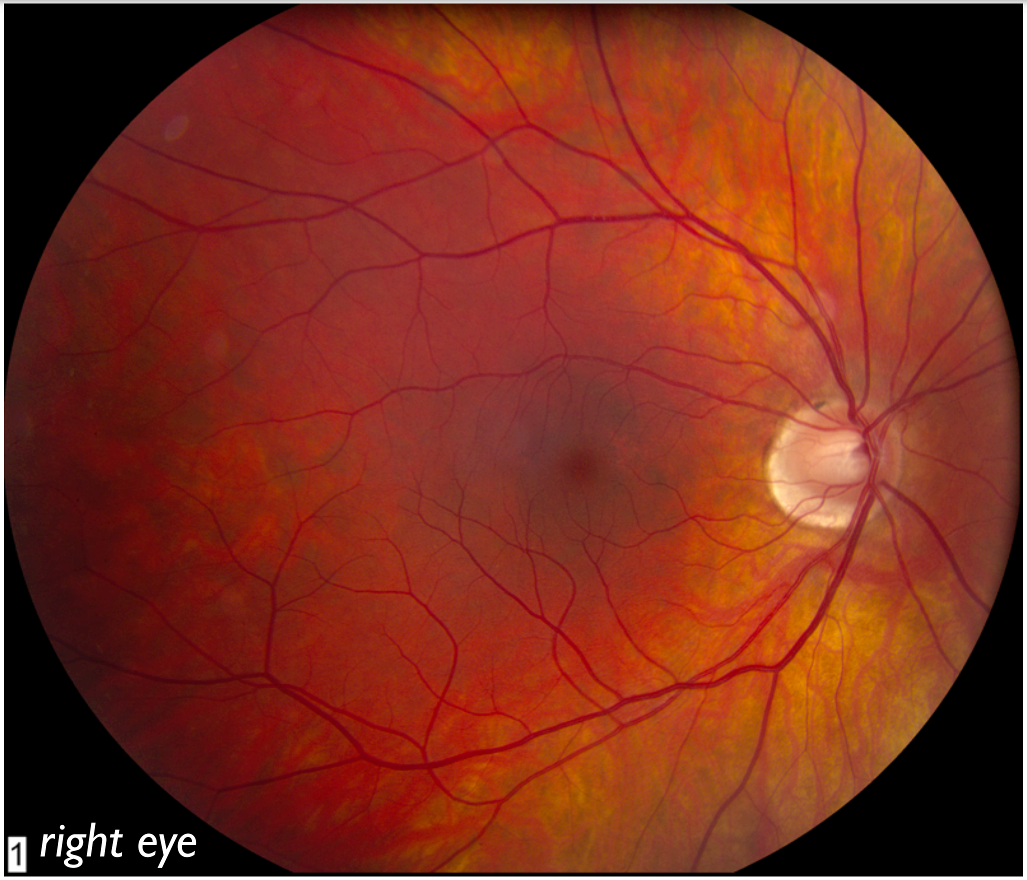
Financial support

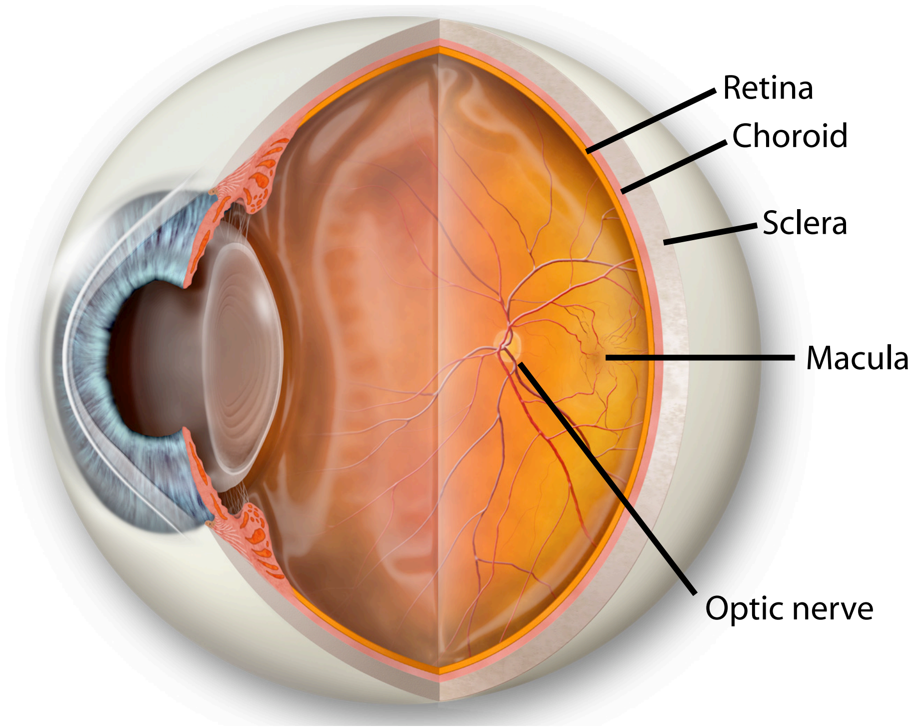
- NIH grants NIBIB R01-EB004640, NEI R01-EY017066, NEI R01-EY018853
- U.S. Department of Veteran Affairs and the VA Center to Prevent and Treat Visual Loss
- Research to Prevent Blindness
- Marlene S. and Leonard A. Hadley Glaucoma Research Fund

Motivation: need for 3D segmentation algorithms



Example: ophthalmology is experiencing a recent shift towards use of 3D imaging modalities



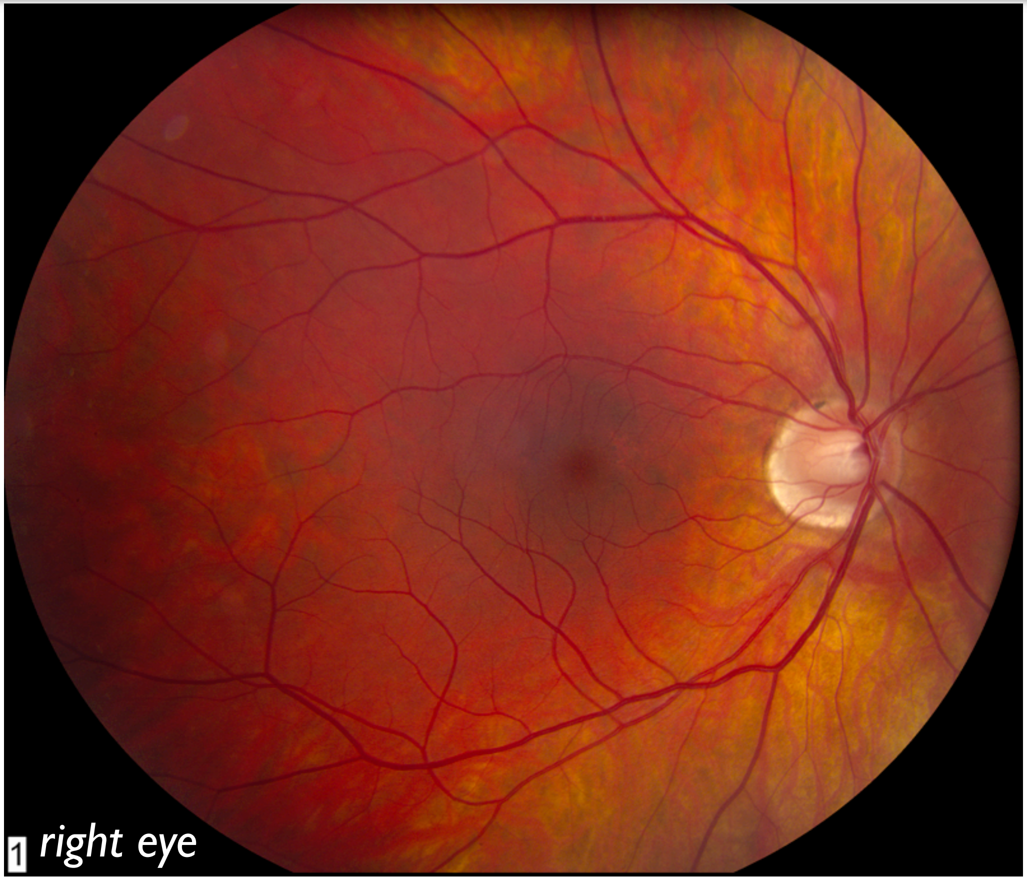


left eye

2D

fundus photography

Example: ophthalmology is experiencing a recent shift towards use of 3D imaging modalities

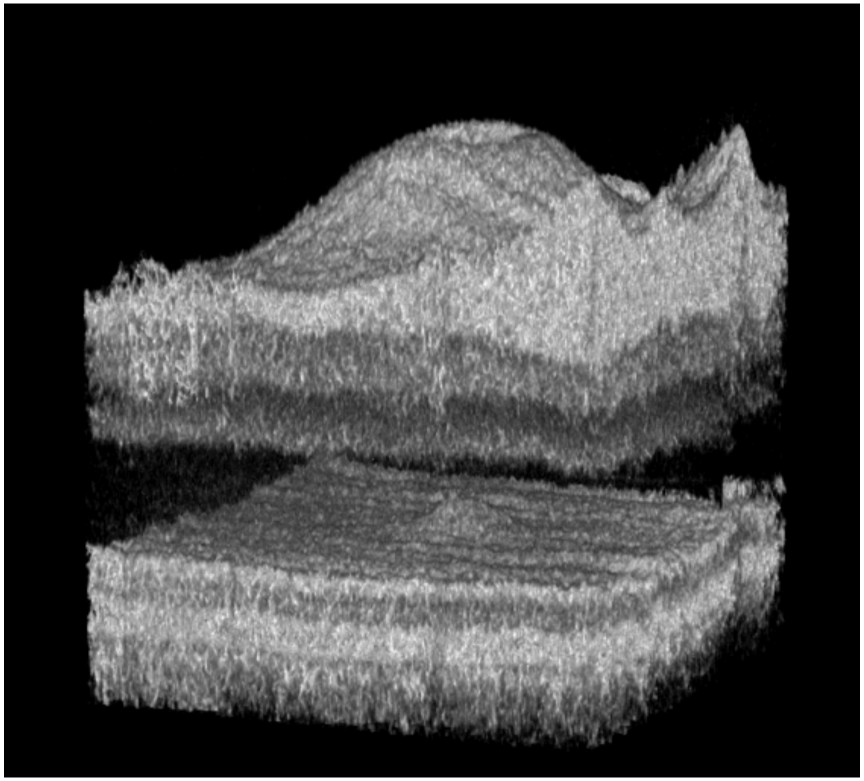


2D

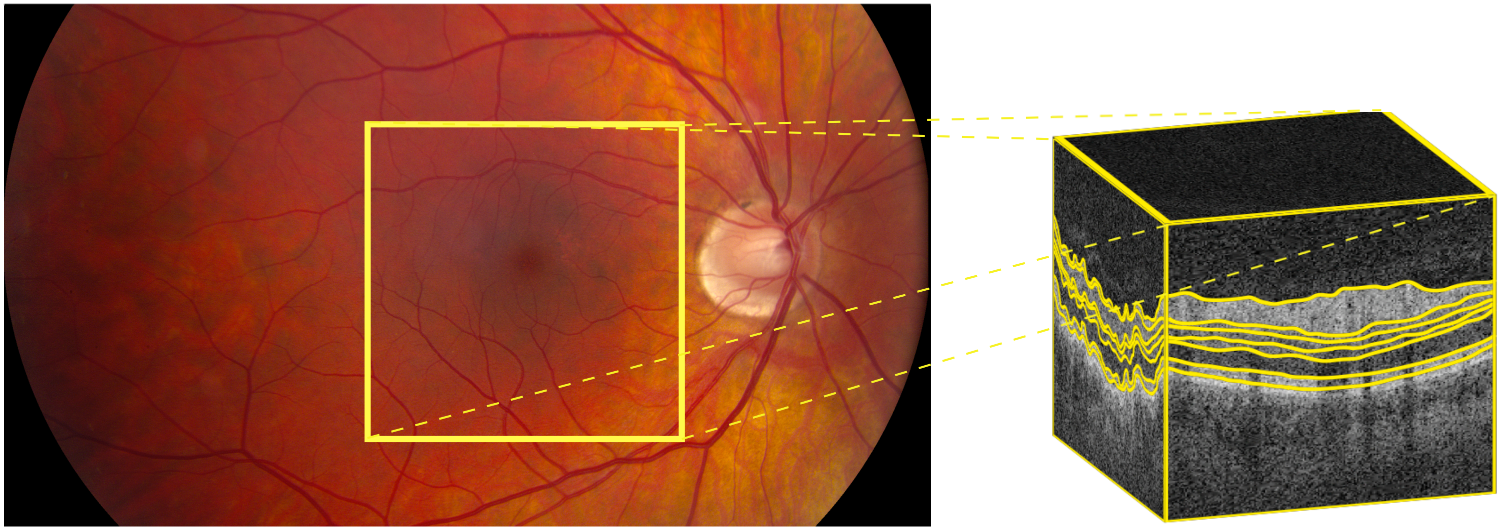
fundus photography

optical coherence tomography

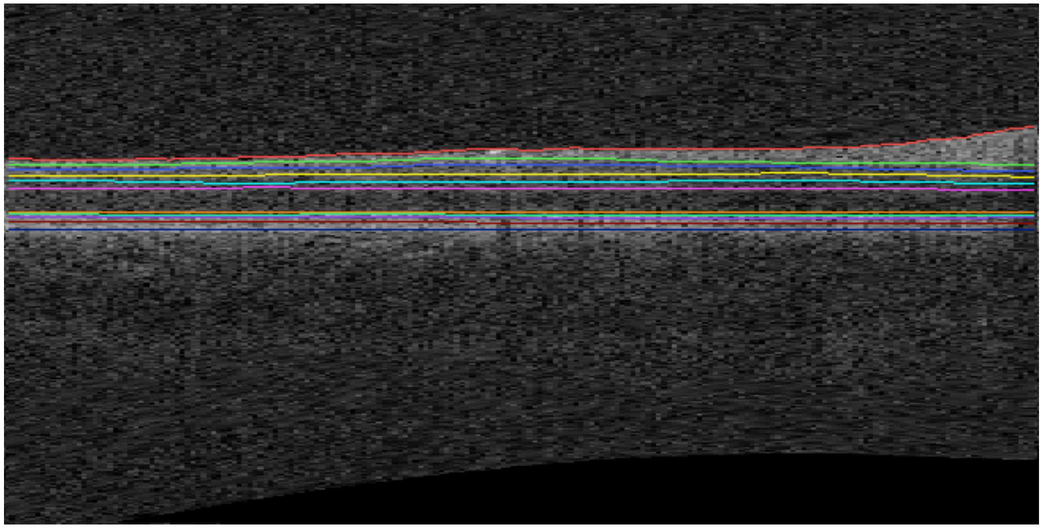
3D



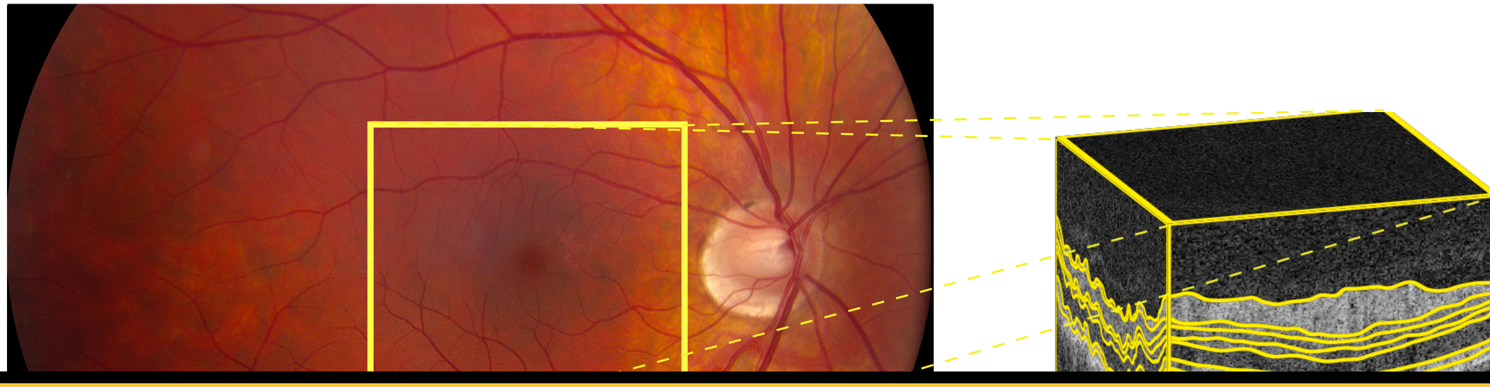
Segmenting the 3D retinal layers is important for diagnosing and managing a variety of diseases causing blindness, such as glaucoma



7-11 surfaces!

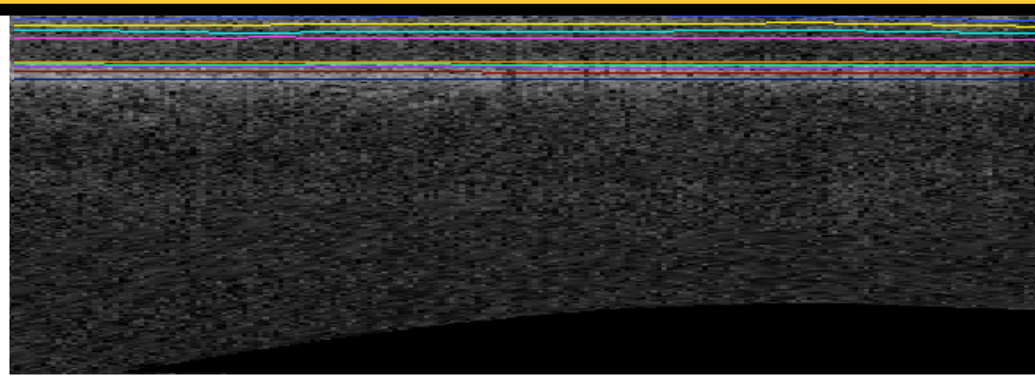


Segmenting the 3D retinal layers is important for diagnosing and managing a variety of diseases causing blindness, such as glaucoma



One challenge: developing algorithms that can **efficiently** take advantage of **3D** (or nD) contextual information and **simultaneously** find the surfaces of interacting objects

7-11 surfaces!

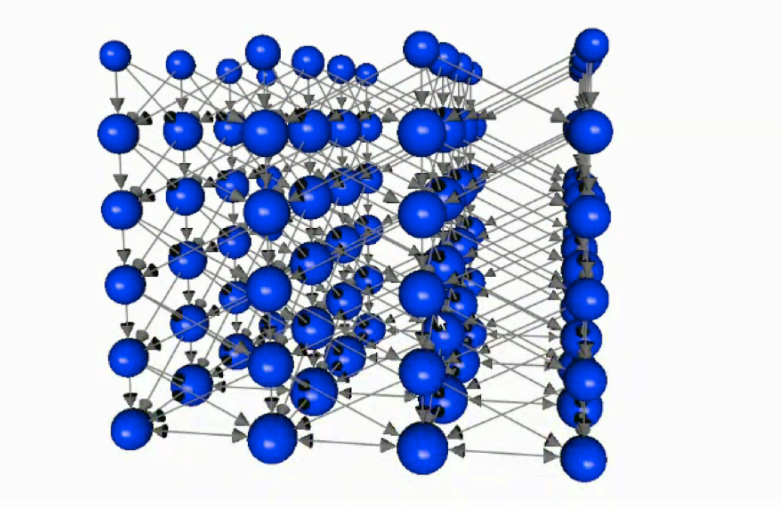


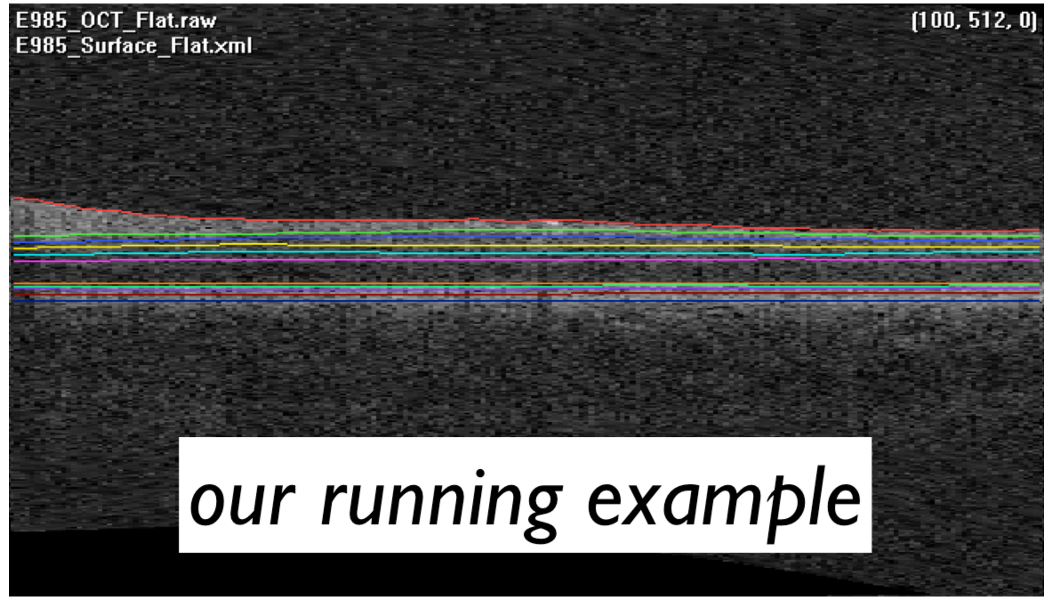
Outline

LOGISMOS approach

Intraretinal layer segmentation

LOGISMOS = Layered Optimal Graph Image Segmentation of Multiple Objects and Surfaces



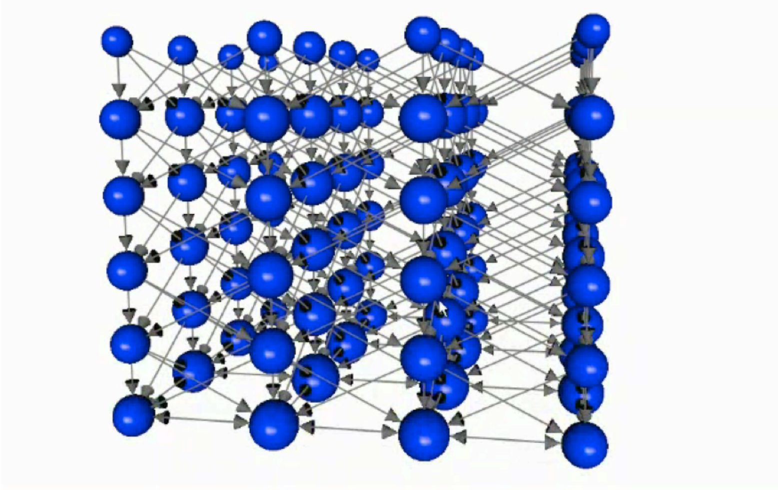


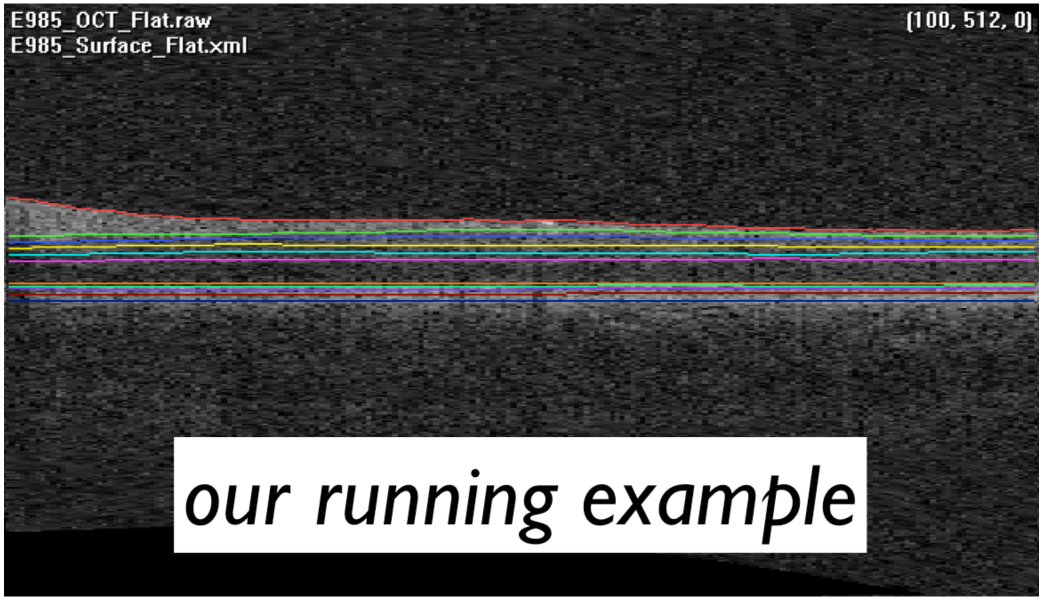
Other applications and future directions

Outline

LOGISMOS approach ⟷ **Intraretinal layer segmentation**

LOGISMOS = Layered Optimal Graph Image Segmentation of Multiple Objects and Surfaces

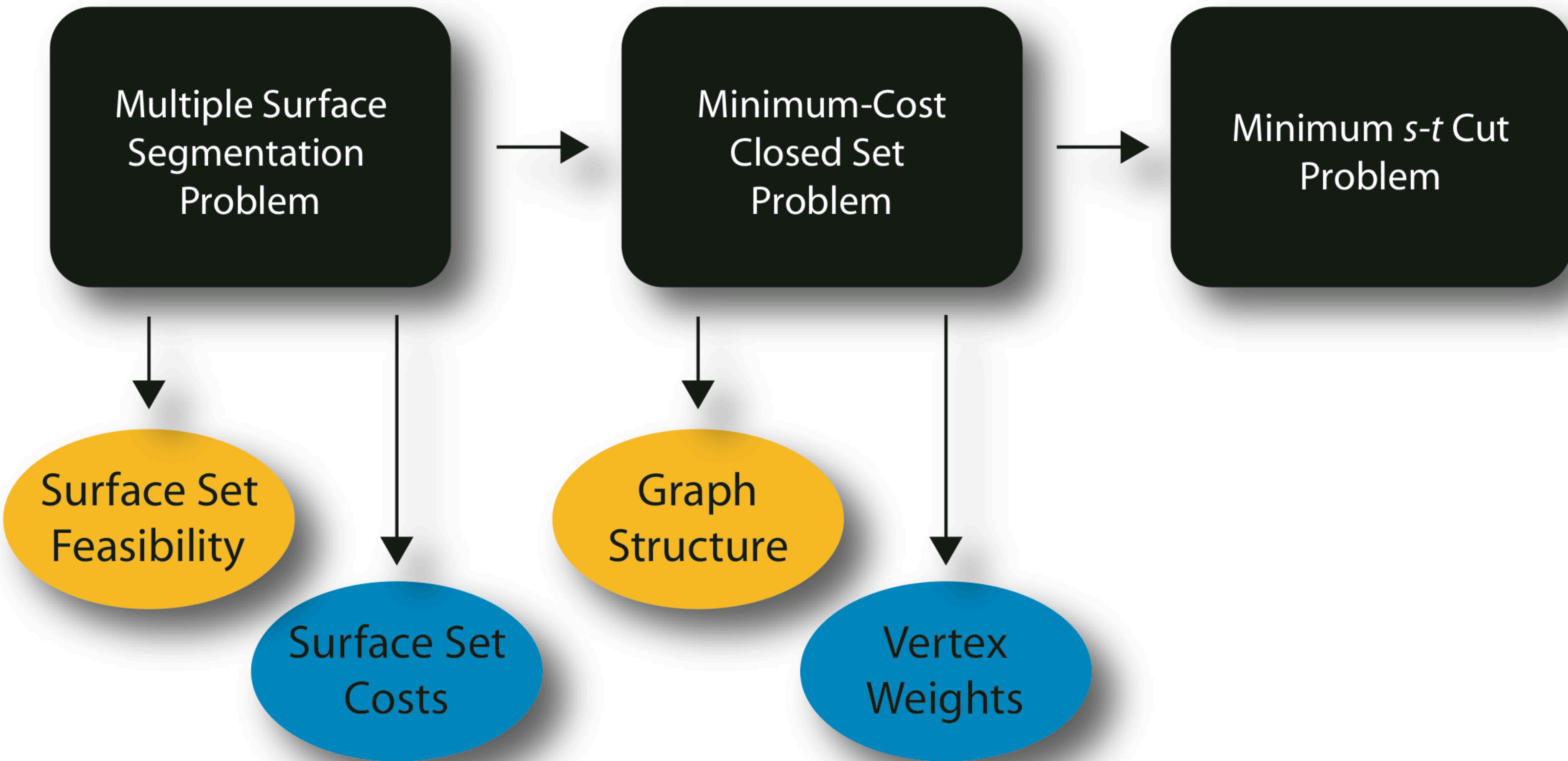




Other applications and future directions

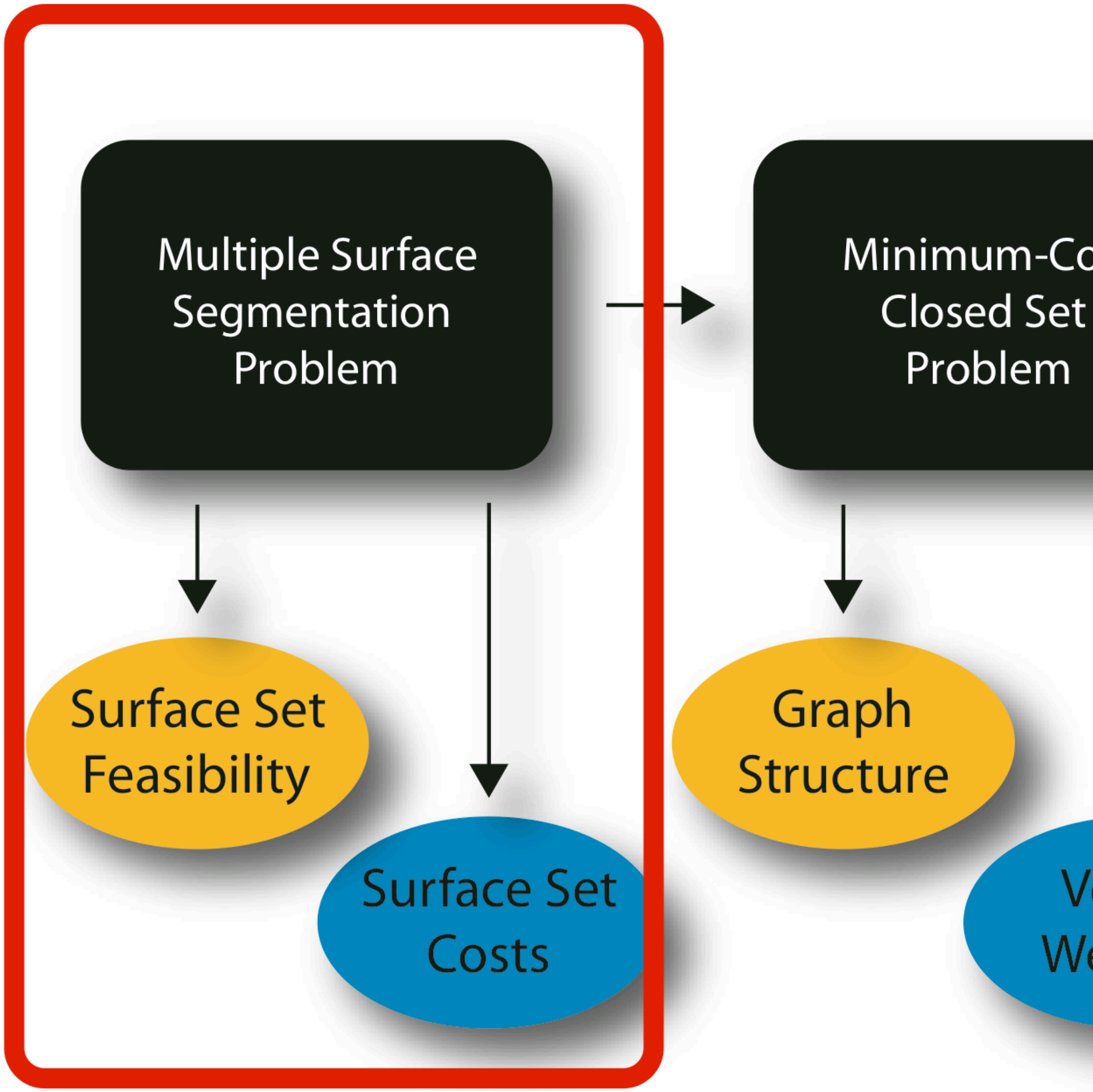
LOGISMOS basics

"Find the feasible set of surfaces that has the minimum cost."



K. Li et al., PAMI 2006, extensions: M. Garvin et al., TMI 2009

The optimality of the LOGISMOS approach allows a user to focus on cost function design (without thinking about graphs)



The graph-theoretic steps will guarantee that we will find the optimal set of surfaces subject to the feasibility constraints.

K. Li et al., PAMI 2006, extensions: M. Garvin et al., TMI 2009

The multiple surface segmentation problem

Surface set feasibility

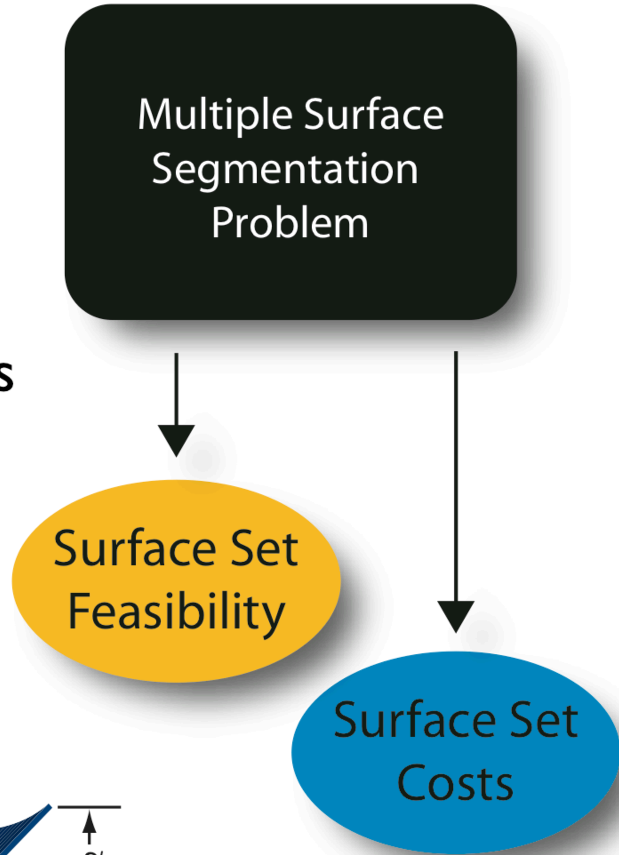
smoothness constraints

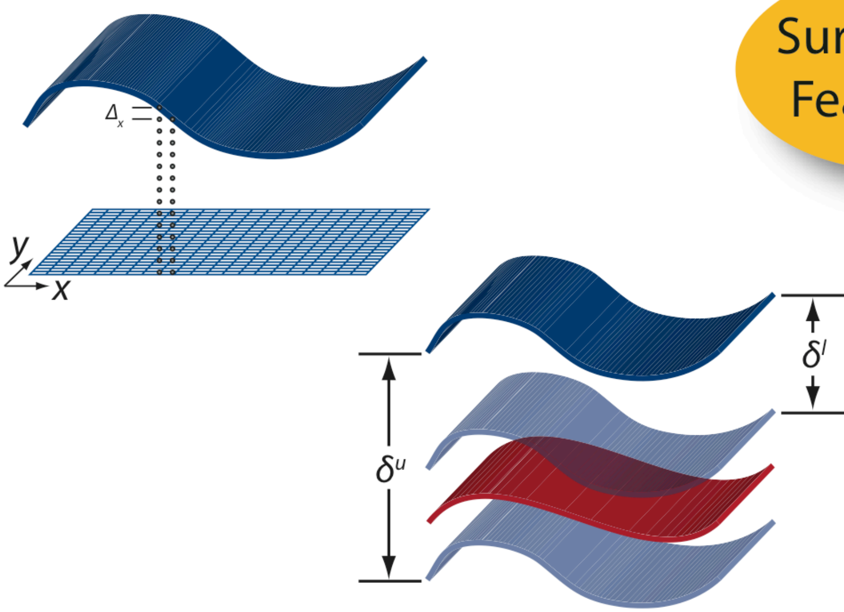
surface interaction constraints

Surface set costs

on-surface costs

in-region costs



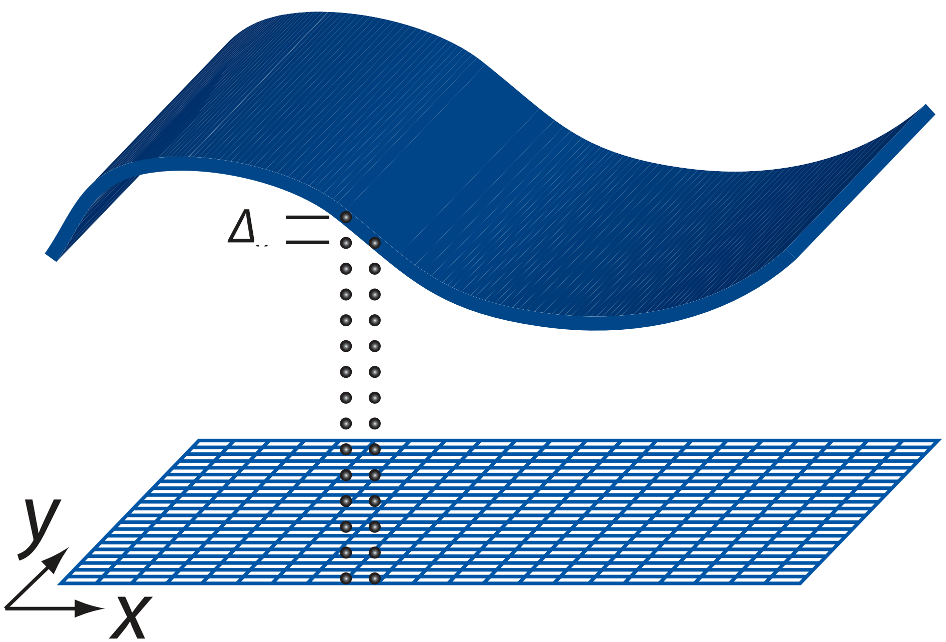


Surface set feasibility

Set of *n* surfaces: $\{f_1(x,y),\ldots,f_n(x,y)\}$

Smoothness constraints

$$-\Delta^u_{\{(x_1,y_1),(x_2,y_2)\}} \leq f(x_1,y_1) - f(x_2,y_2) \leq \Delta^l_{\{(x_1,y_1),(x_2,y_2)\}}$$

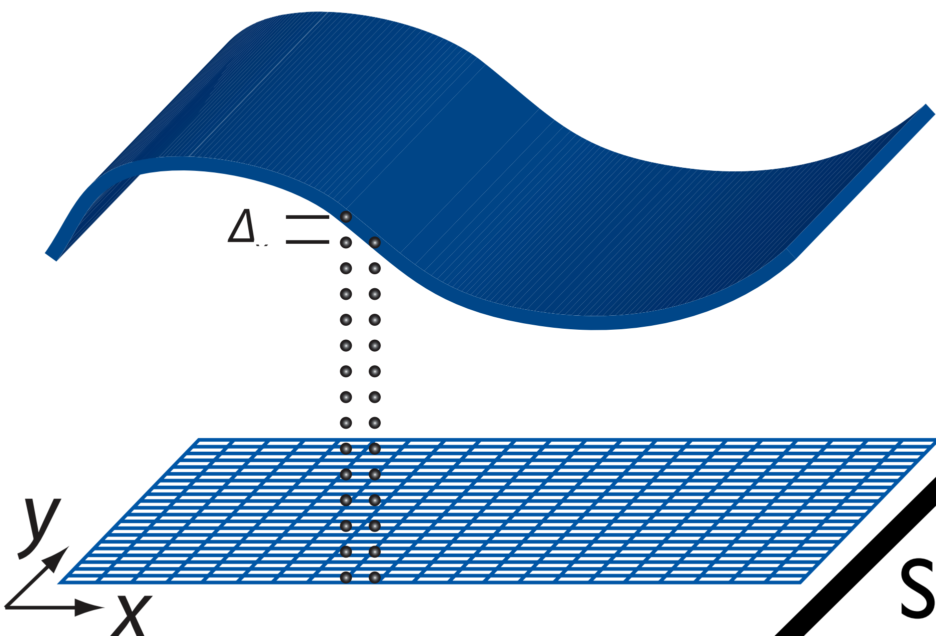


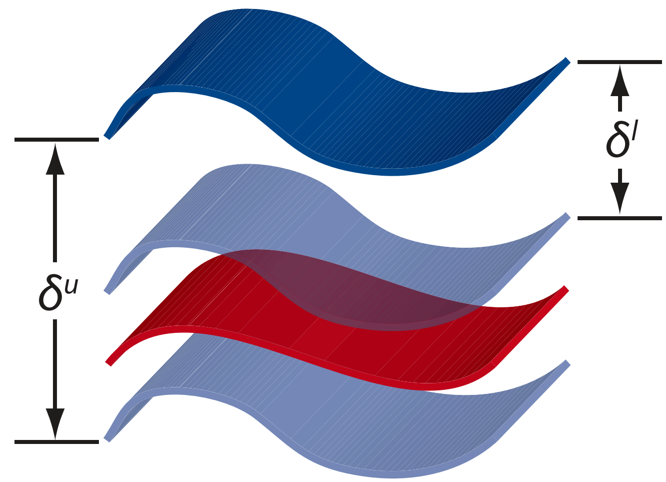
Surface set feasibility

Set of *n* surfaces: $\{f_1(x,y),\ldots,f_n(x,y)\}$

Smoothness constraints

$$-\Delta^u_{\{(x_1,y_1),(x_2,y_2)\}} \leq f(x_1,y_1) - f(x_2,y_2) \leq \Delta^l_{\{(x_1,y_1),(x_2,y_2)\}}$$





Surface interaction constraints

$$\delta^l(x,y) \leq f_i(x,y) - f_j(x,y) \leq \delta^u(x,y)$$

The multiple surface segmentation problem

Surface set feasibility

smoothness constraints

surface interaction constraints

Multiple Surface Segmentation Problem

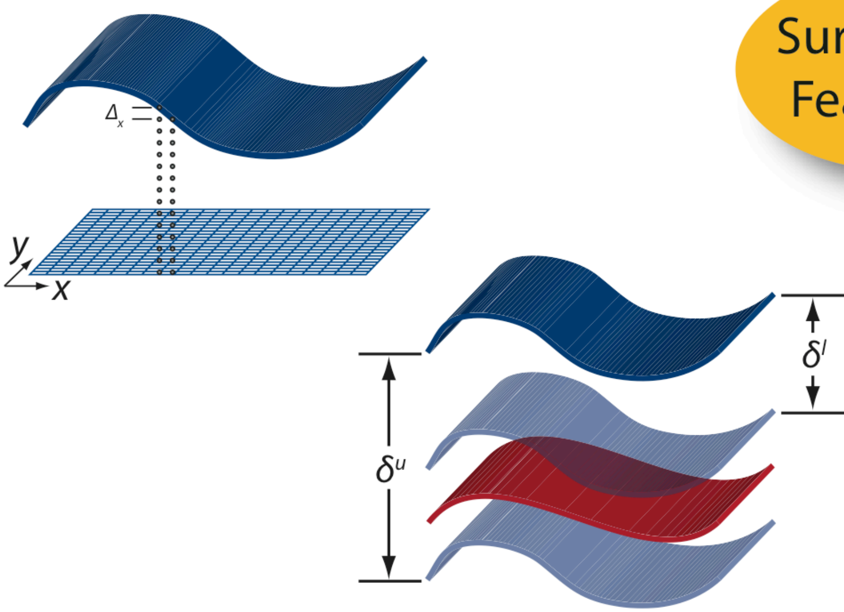
Surface Set Feasibility

Surface Set Costs

Surface set costs

on-surface costs

in-region costs



More on surface set costs...

Two categories of cost functions

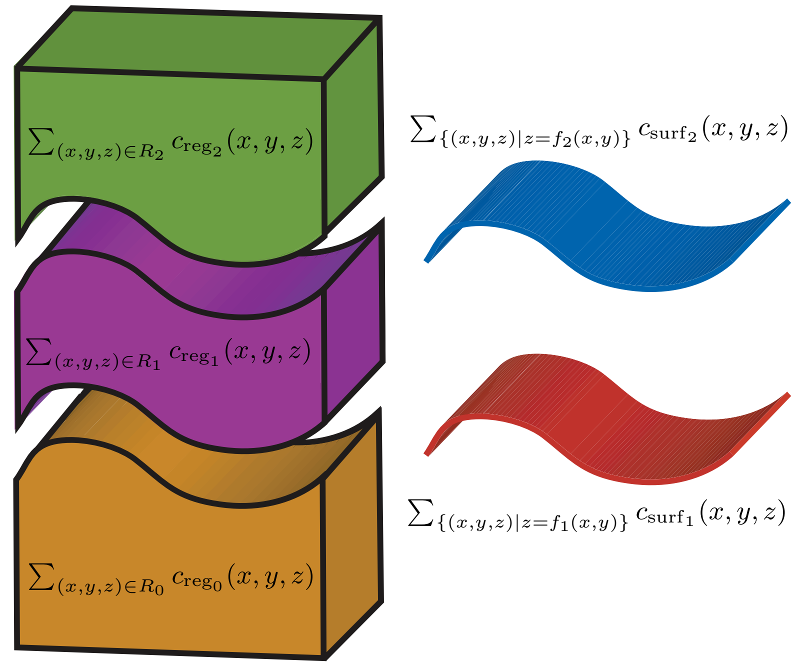
n surfaces, n+1 regions

- **On-surface costs:** Each voxel has *n* on-surface costs corresponding to the unlikeliness of belonging to each surface.
- **In-region costs:** Each voxel has *n+1* in-region costs corresponding to the unlikeliness of belonging to each region.

Cost function using both on-surface and in-region costs

Surface set cost function:

$$C_{\{f_1(x,y),f_2(x,y),\ldots,f_n(x,y)\}} = \sum_{i=1}^{n} C_{f_i(x,y)} + \sum_{i=0}^{n} C_{R_i}$$



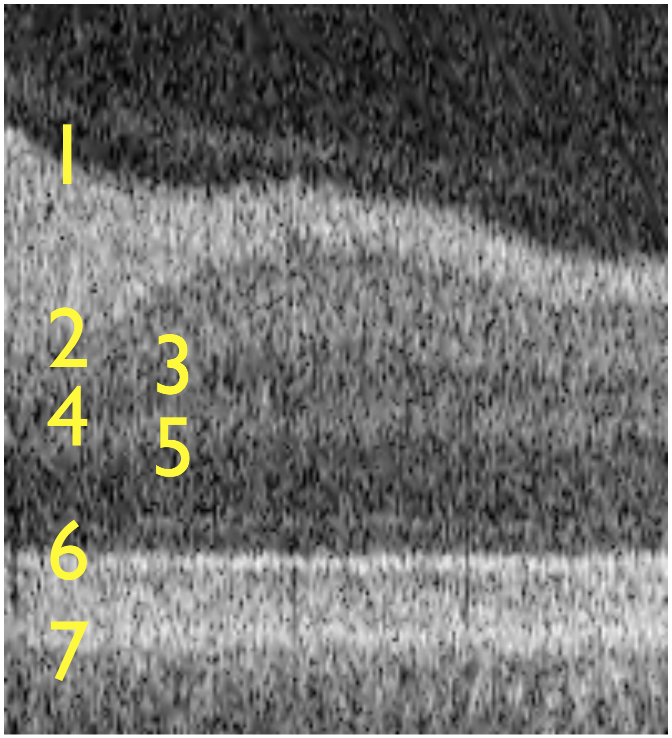
On-surface costs:

$$C_{f_i(x,y)} = \sum_{\{(x,y,z)|z=f_i(x,y)\}} c_{\text{surf}_i}(x,y,z)$$

In-region costs:

$$C_{R_i} = \sum_{(x,y,z)\in R_i} c_{\text{reg}_i}(x,y,z)$$

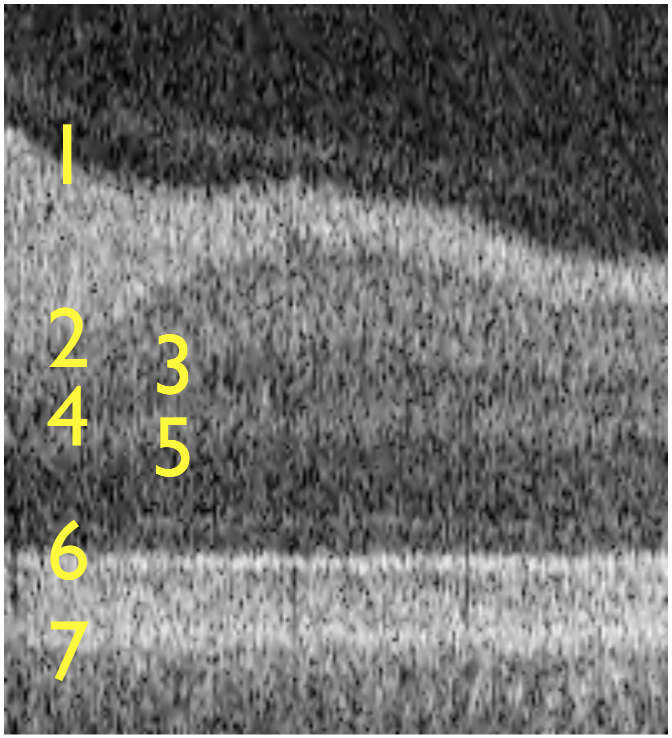
Example using only on-surface costs



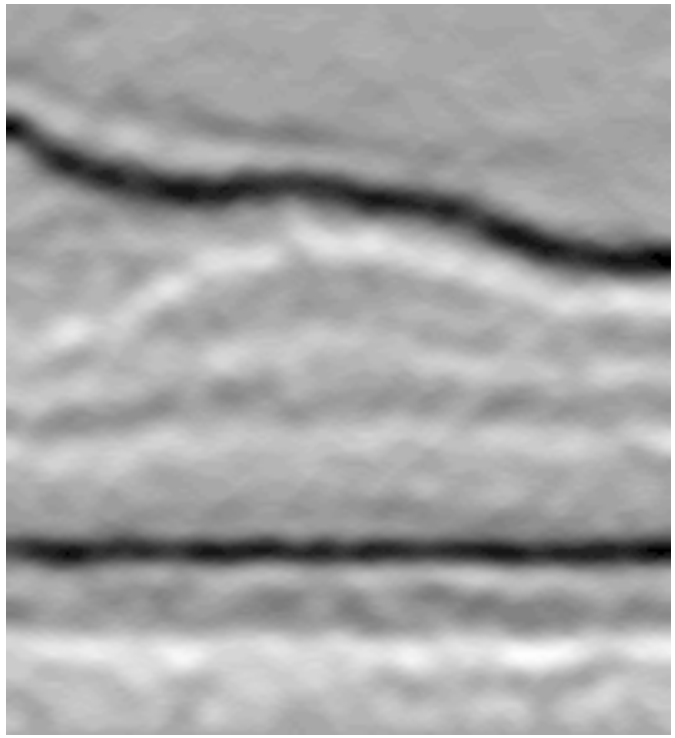
OCT Image

Find indicated 7 surfaces in OCT image using only on-surface costs.

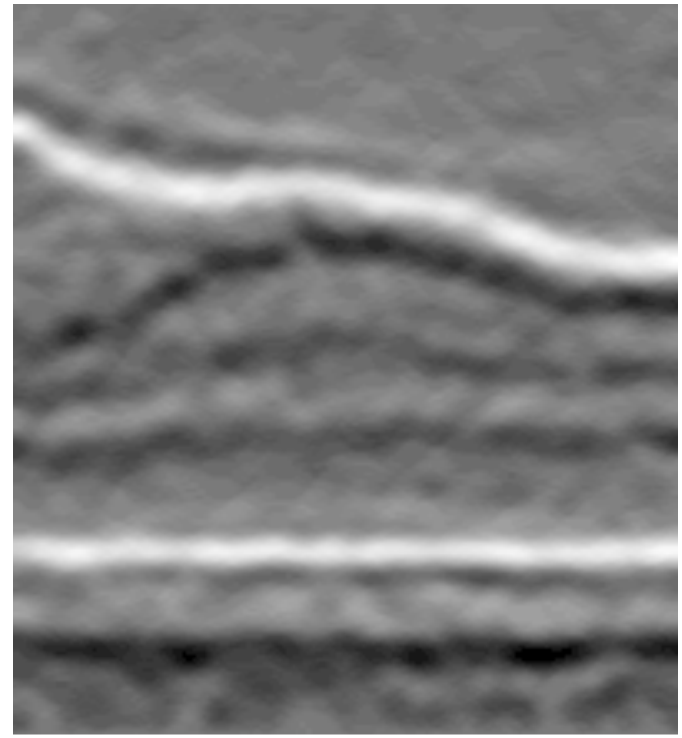
Example using only on-surface costs



OCT Image



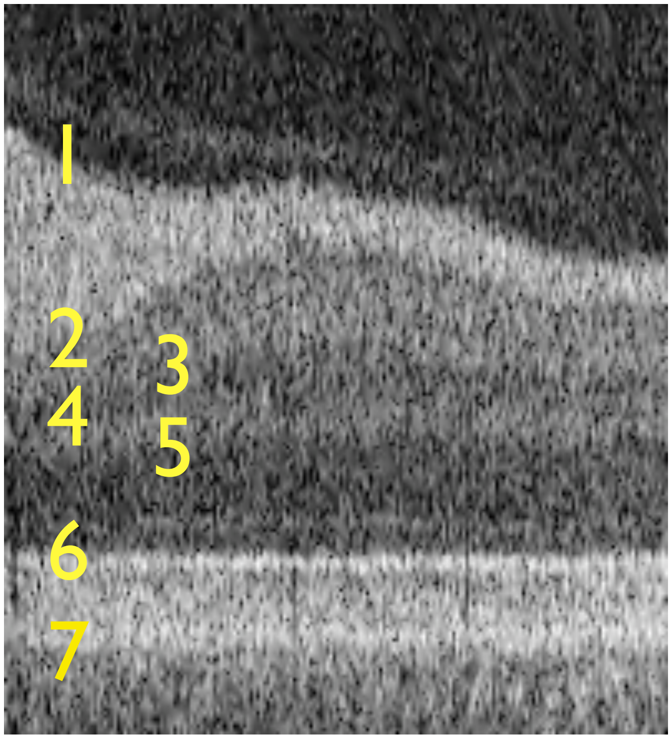
Encourage dark-to-bright transitions



Encourage bright-to-dark transitions

Example using only on-surface costs

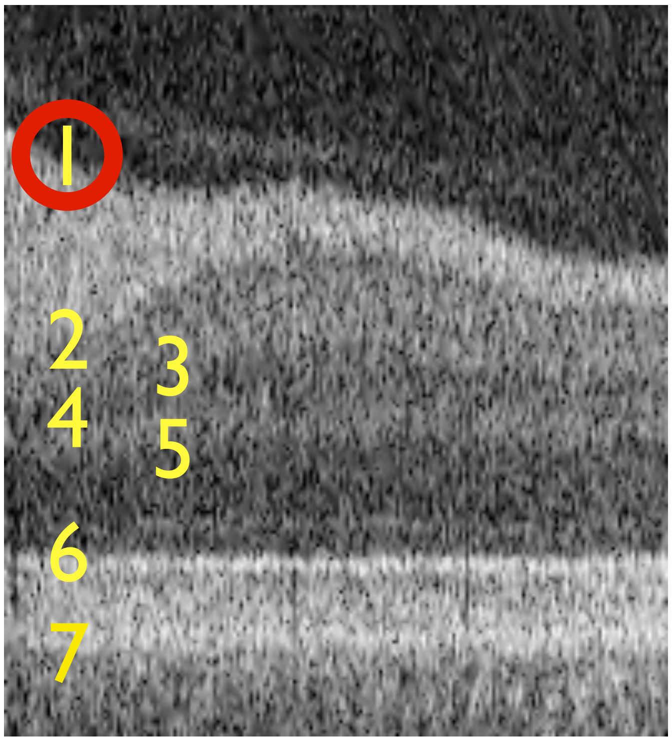
Seven cost images:



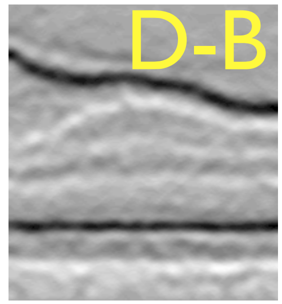
OCT Image

Example using only on-surface costs

Seven cost images:



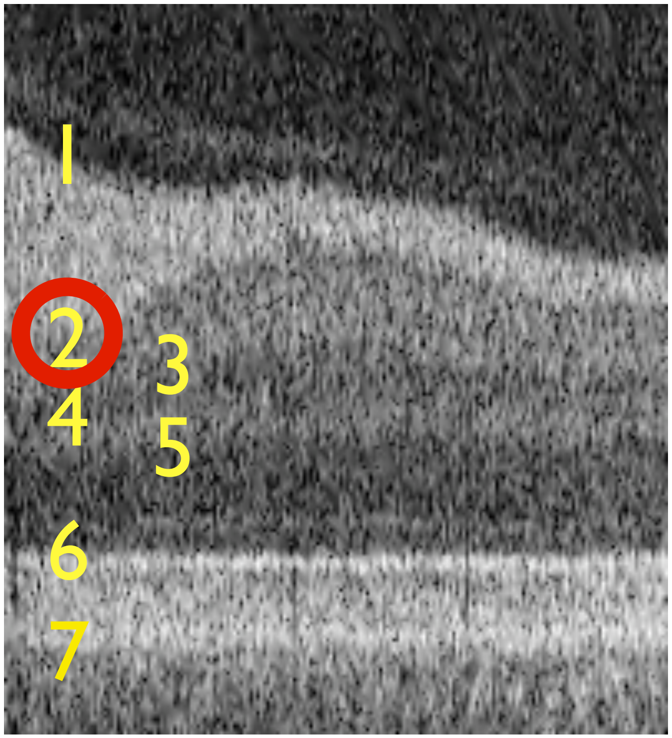
OCT Image



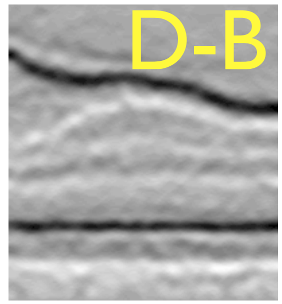
1

Example using only on-surface costs

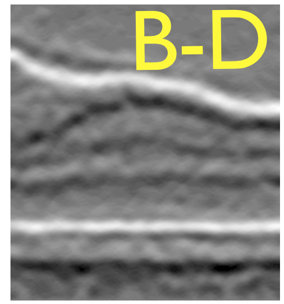
Seven cost images:



OCT Image



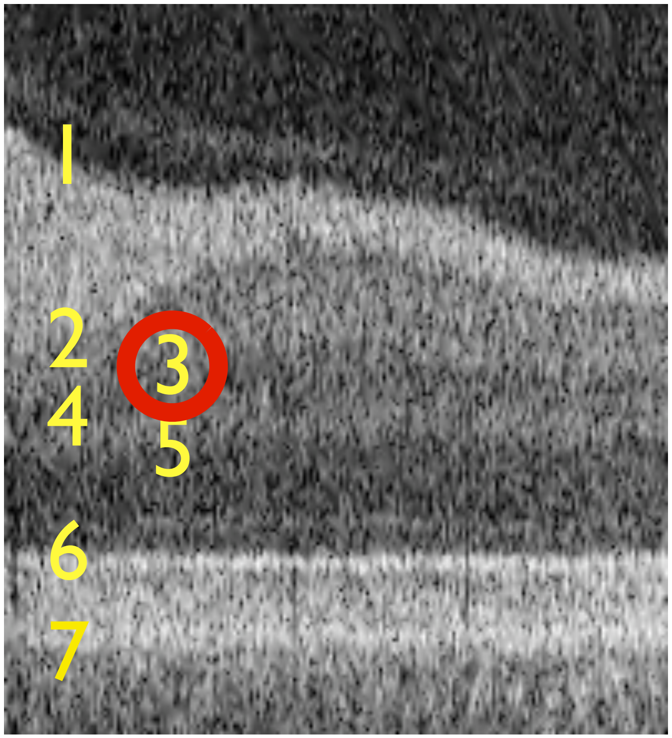
1



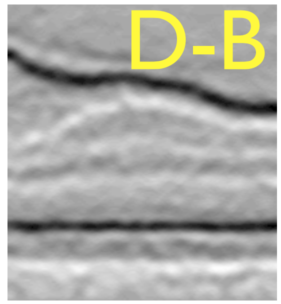
2

Example using only on-surface costs

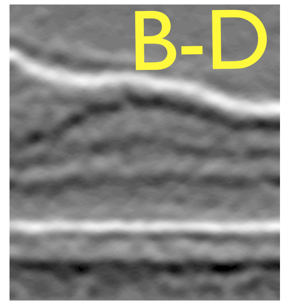
Seven cost images:



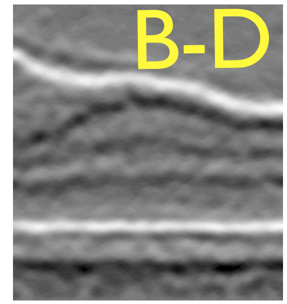
OCT Image



1

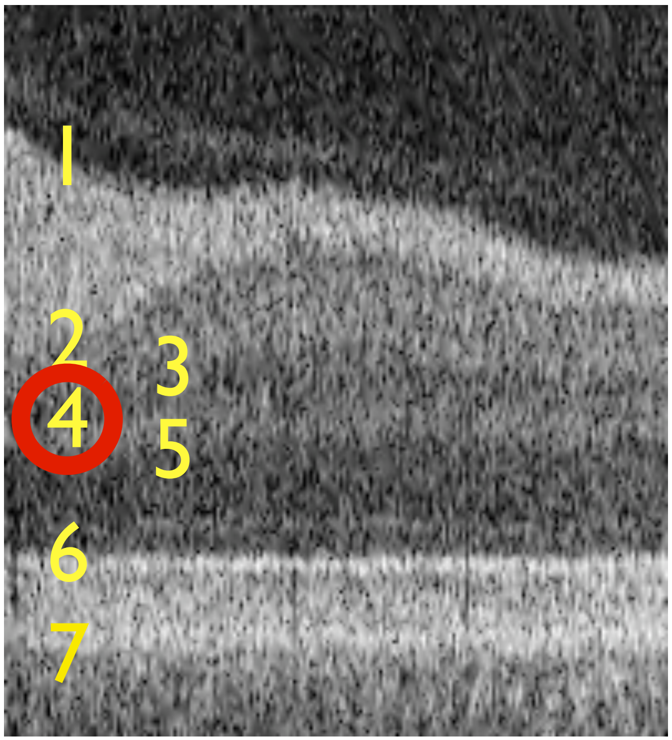


2



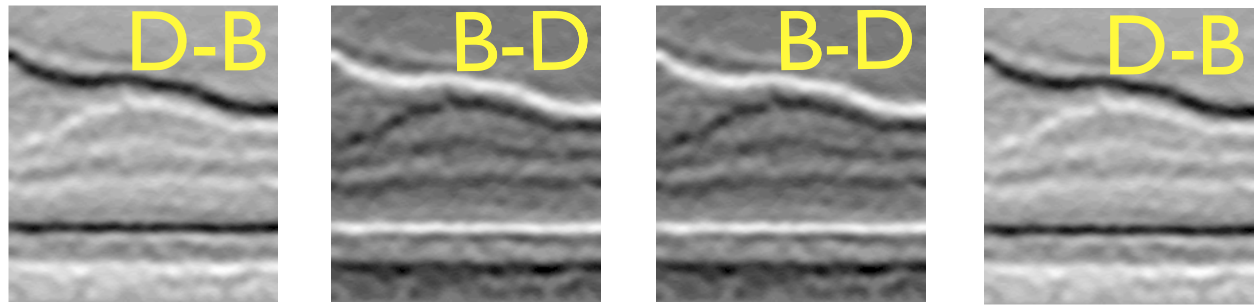
3

Example using only on-surface costs



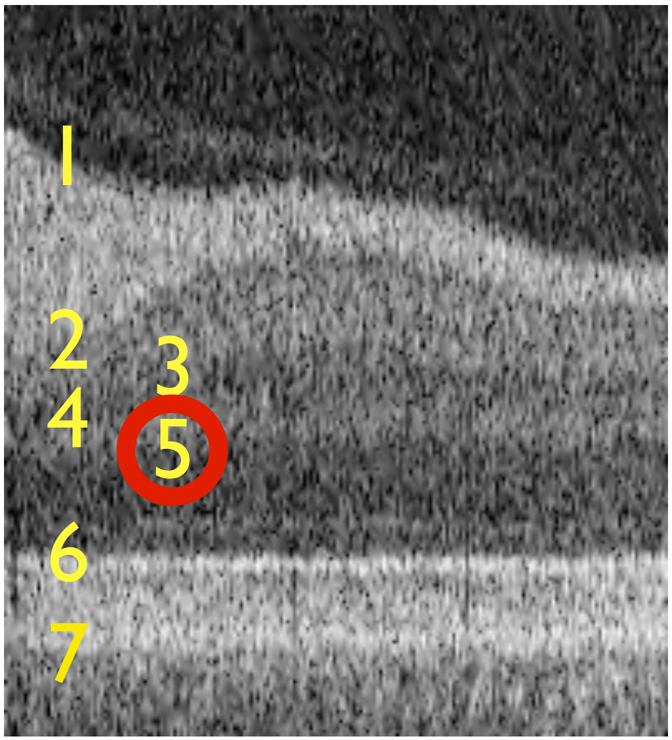
OCT Image

Seven cost images:



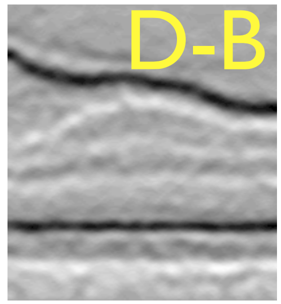
1 2 3 4

Example using only on-surface costs

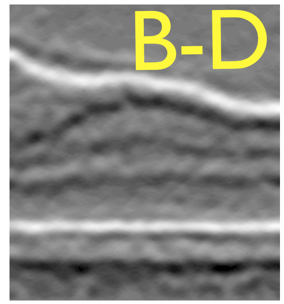


OCT Image

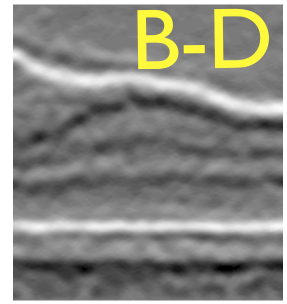
Seven cost images:



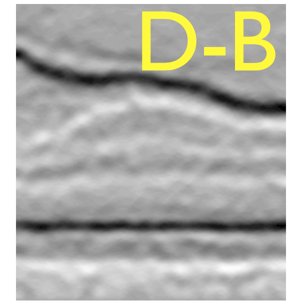
1



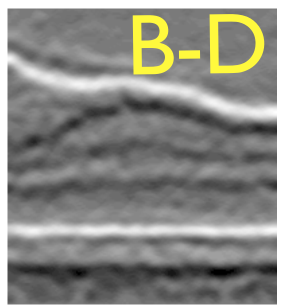
2



3



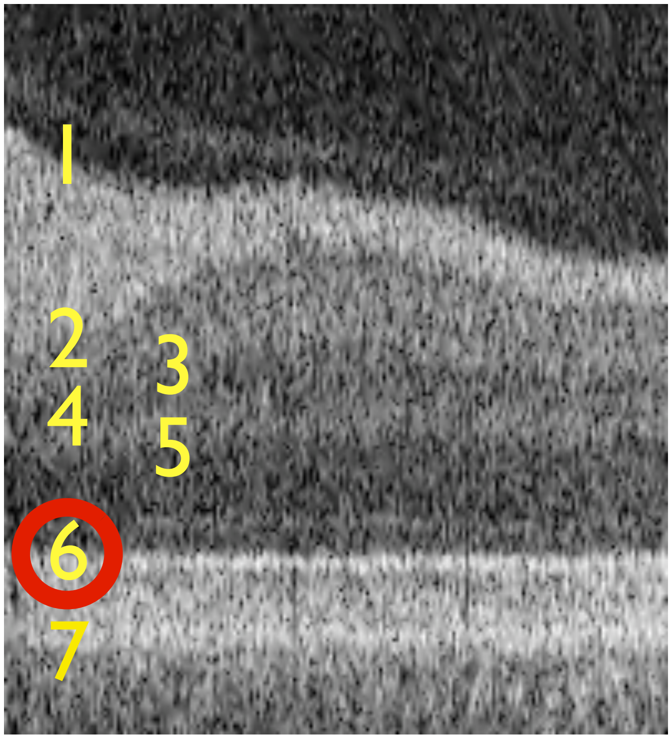
4



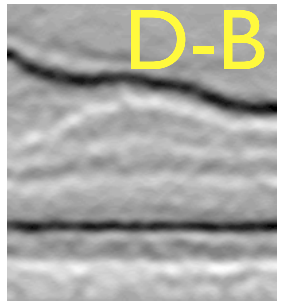
5

Example using only on-surface costs

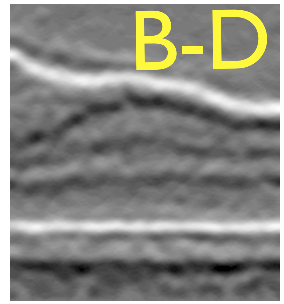
Seven cost images:



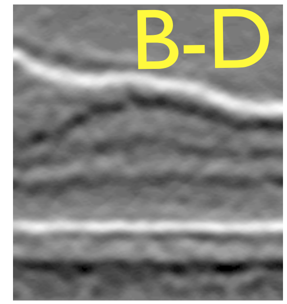
OCT Image



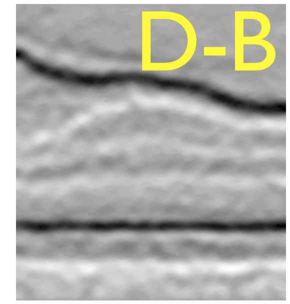
1



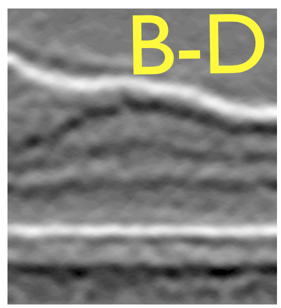
2



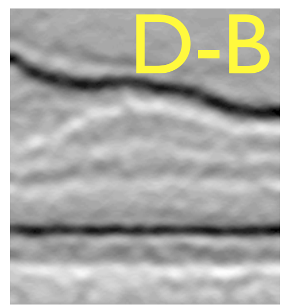
3



4



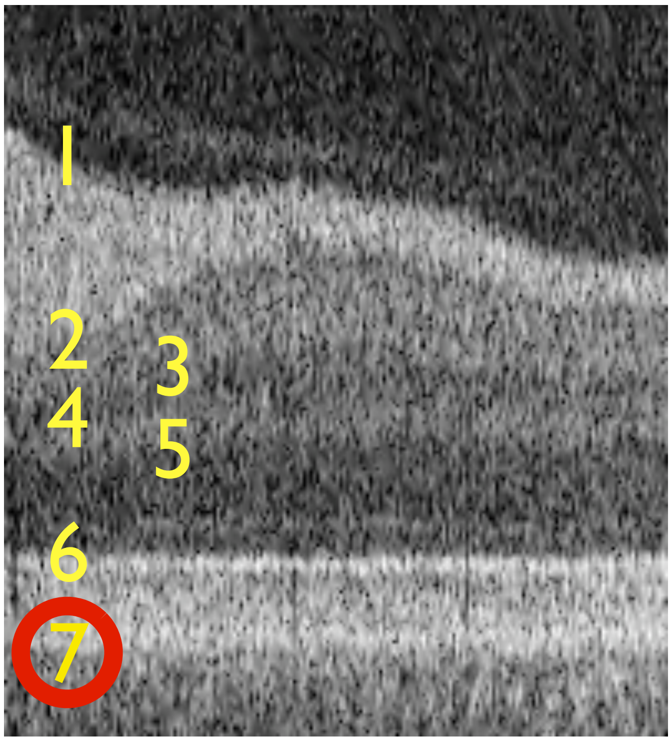
5



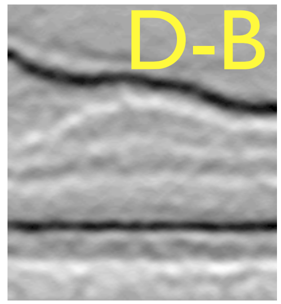
6

Example using only on-surface costs

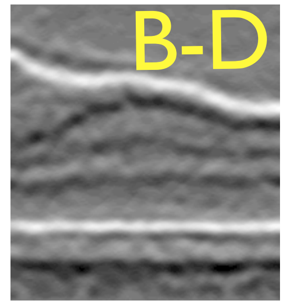
Seven cost images:



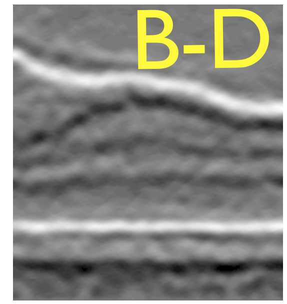
OCT Image



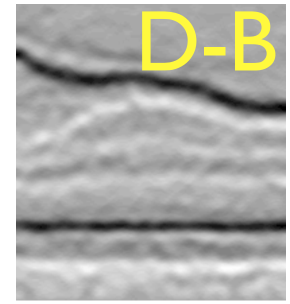
1



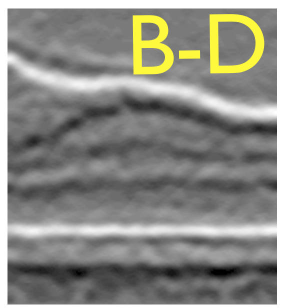
2



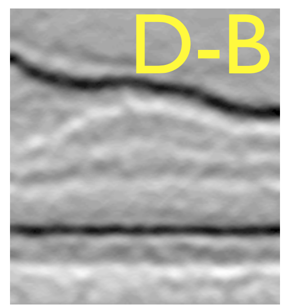
3



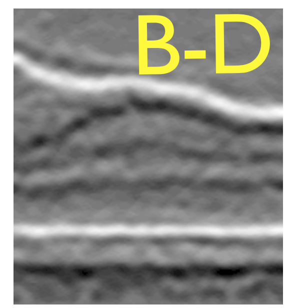
4



5

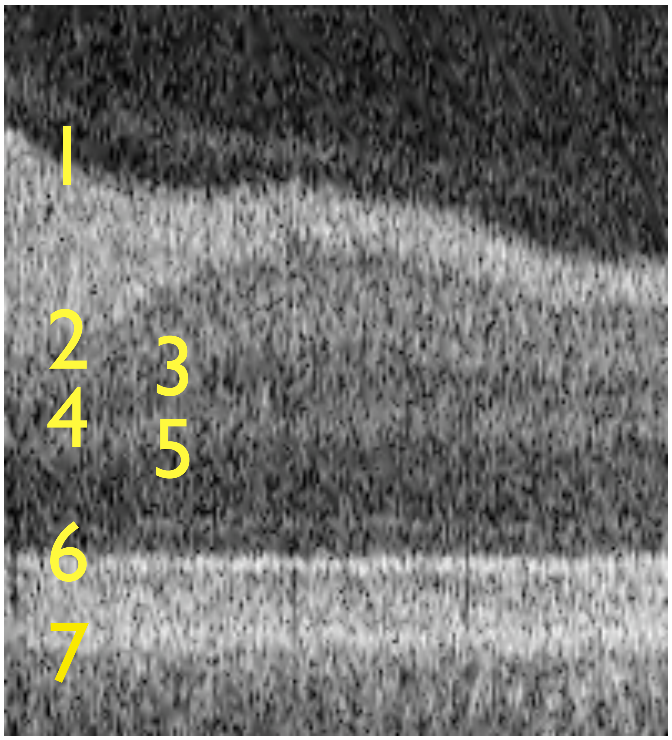


6



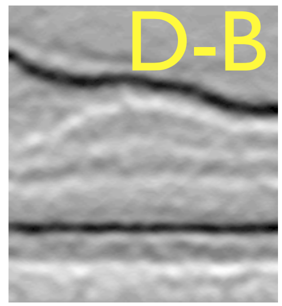
7

Example using only on-surface costs

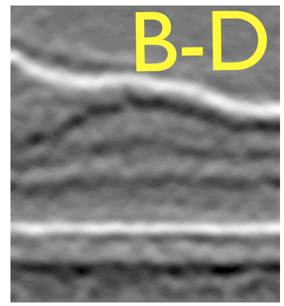


OCT Image

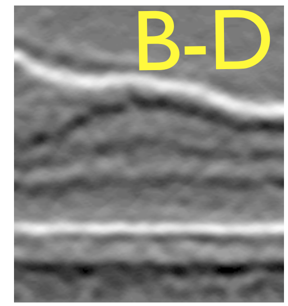
Seven cost images:



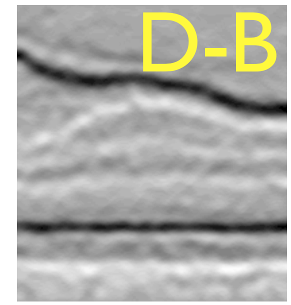
1



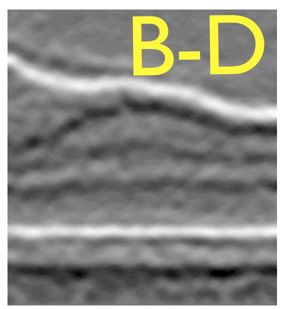
2



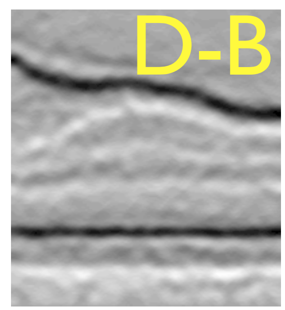
3



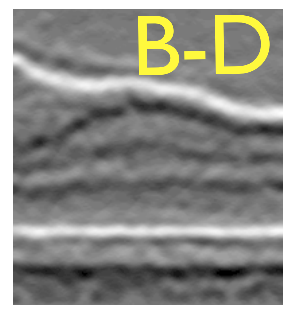
4



5



6



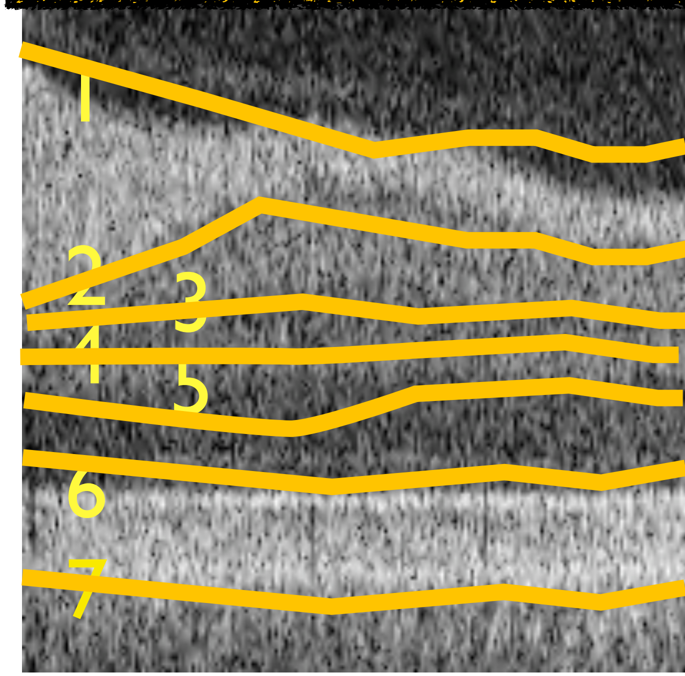
7

+ smoothness constraints

+ thickness constraints

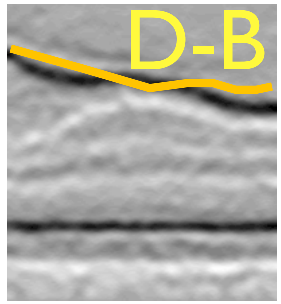
Example using only on-surface costs

A non-optimal set of surfaces

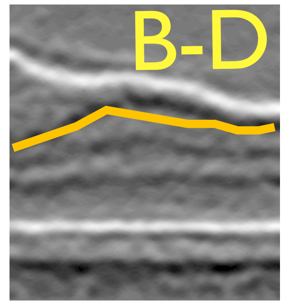


OCT Image

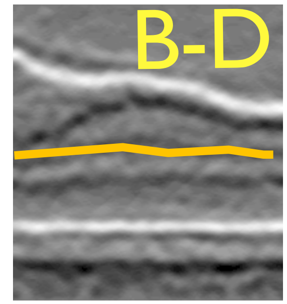
Seven cost images:



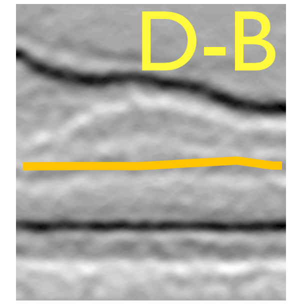
1



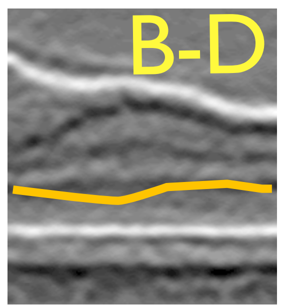
2



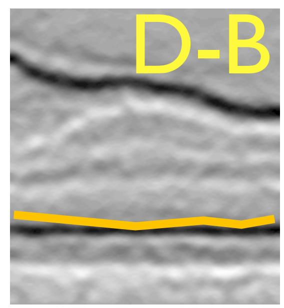
3



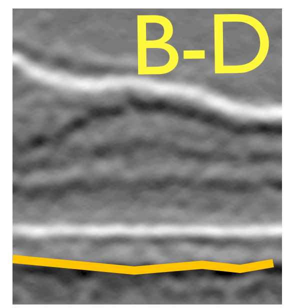
4



5

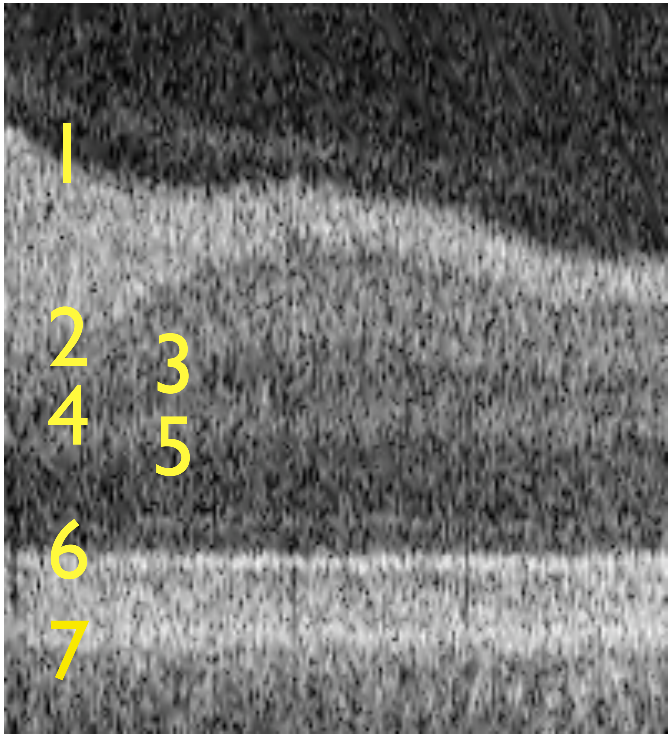


6

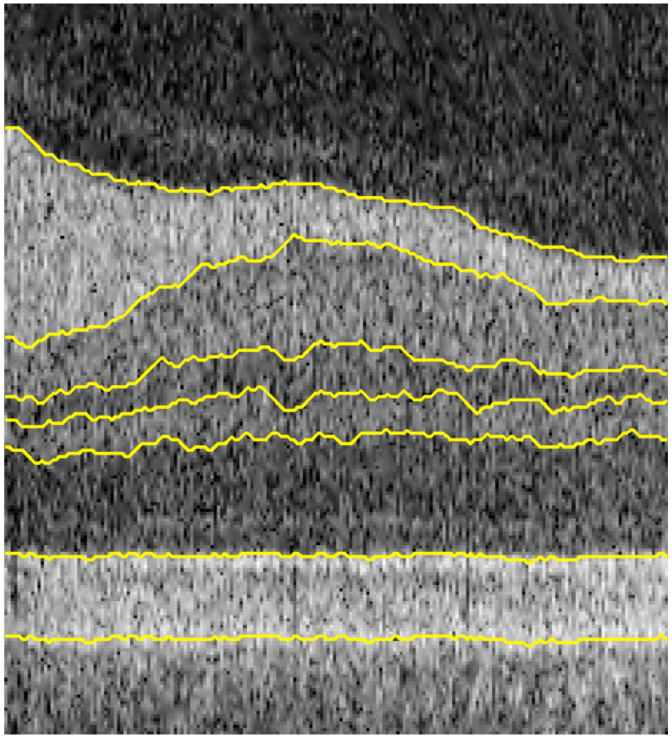


7

Example using only on-surface costs



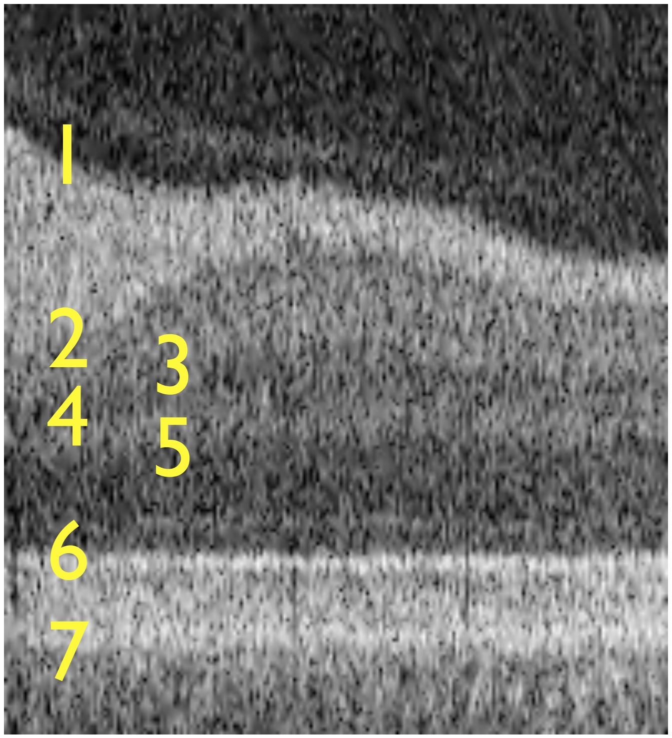
OCT Image



Result

The optimal set of surfaces

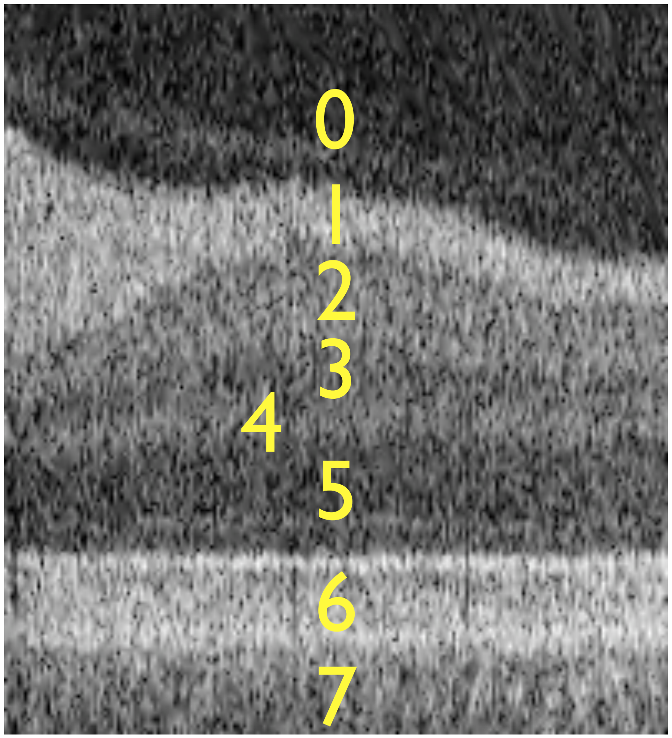
Example using only in-region costs



OCT Image

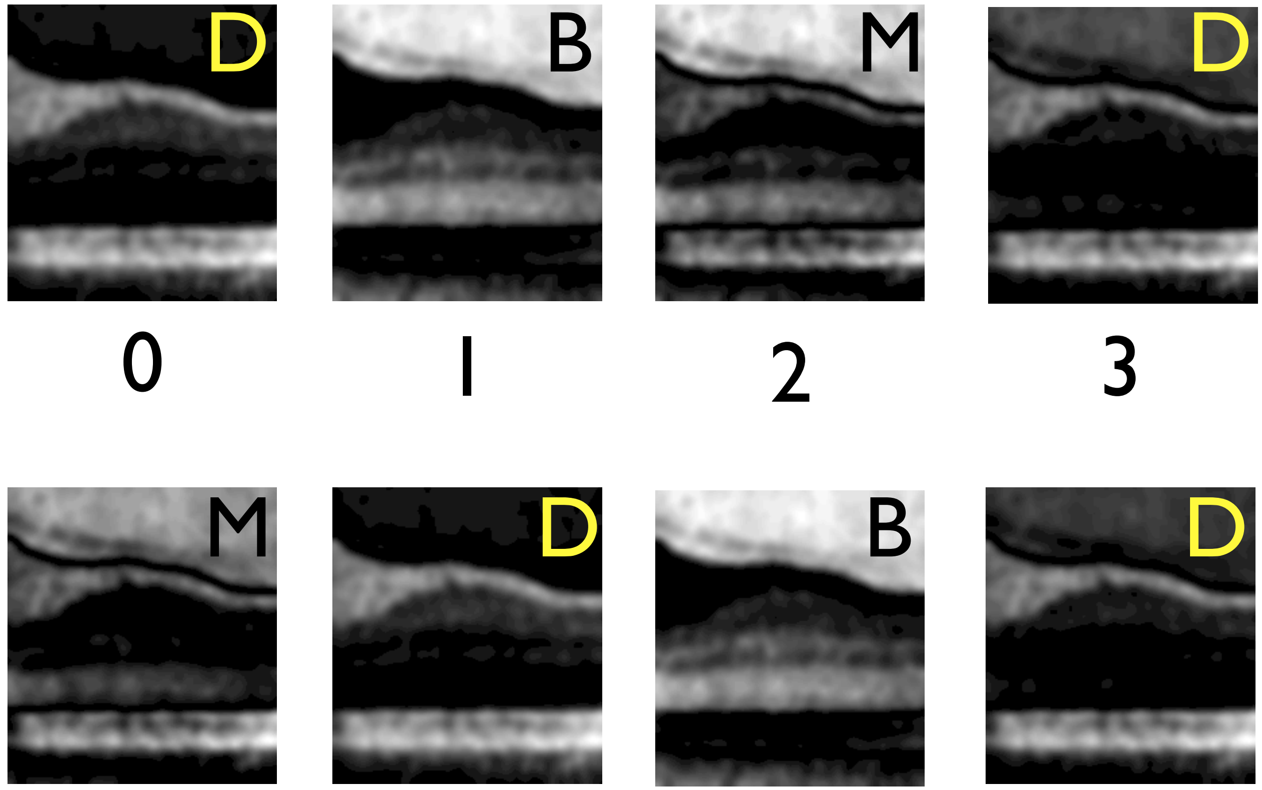
Find indicated 7 surfaces in OCT image using only in-region costs.

Example using only in-region costs



OCT Image
(labeled regions)

Eight cost images:



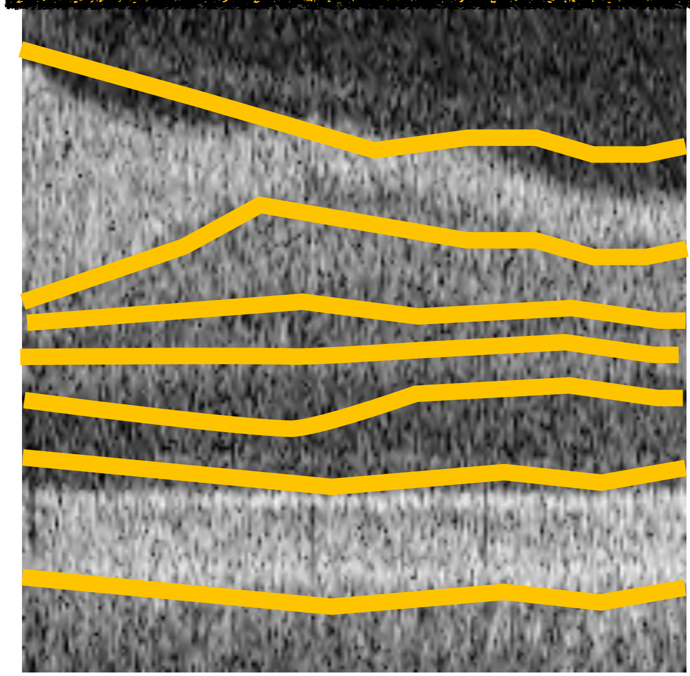
0 1 2 3

4 5 6 7

+ smoothness constraints
+ thickness constraints

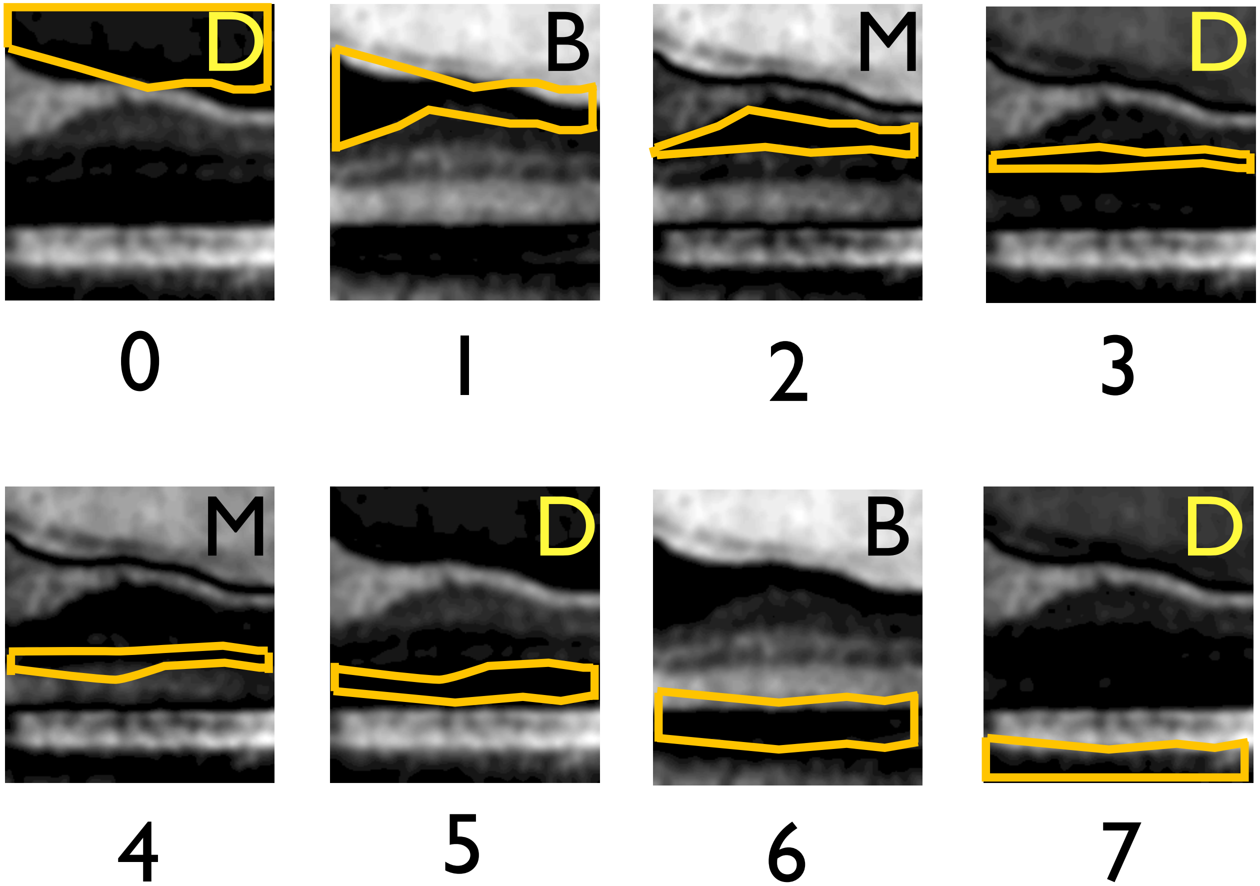
Example using only in-region costs

A non-optimal set of surfaces

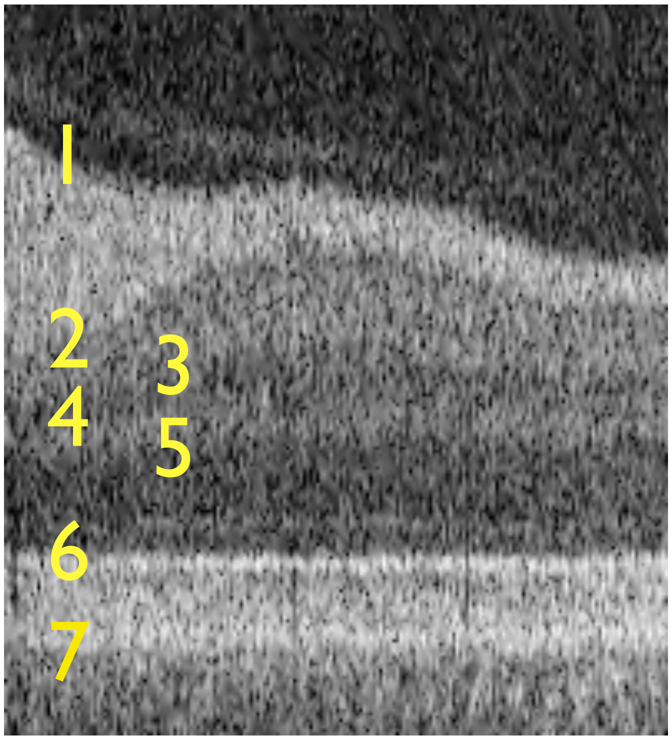


OCT Image

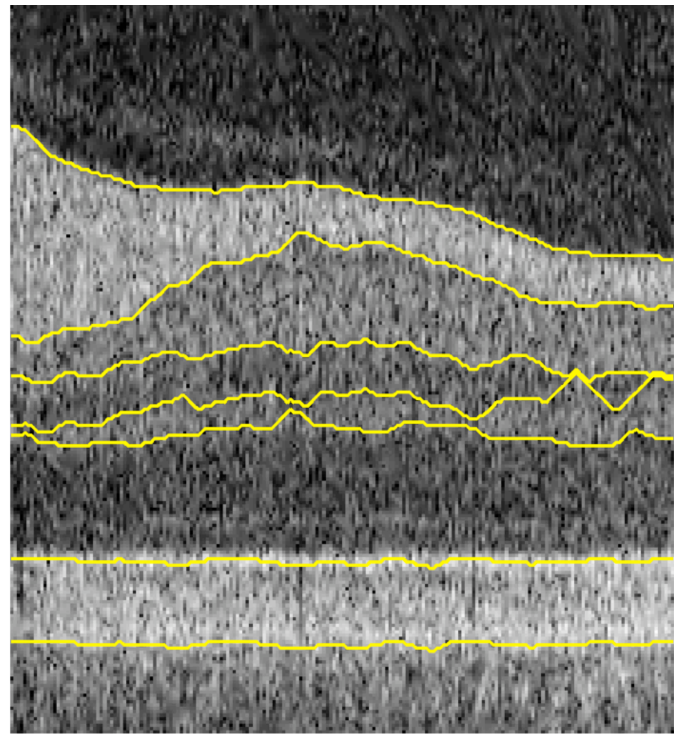
Eight cost images:



Example using only in-region costs



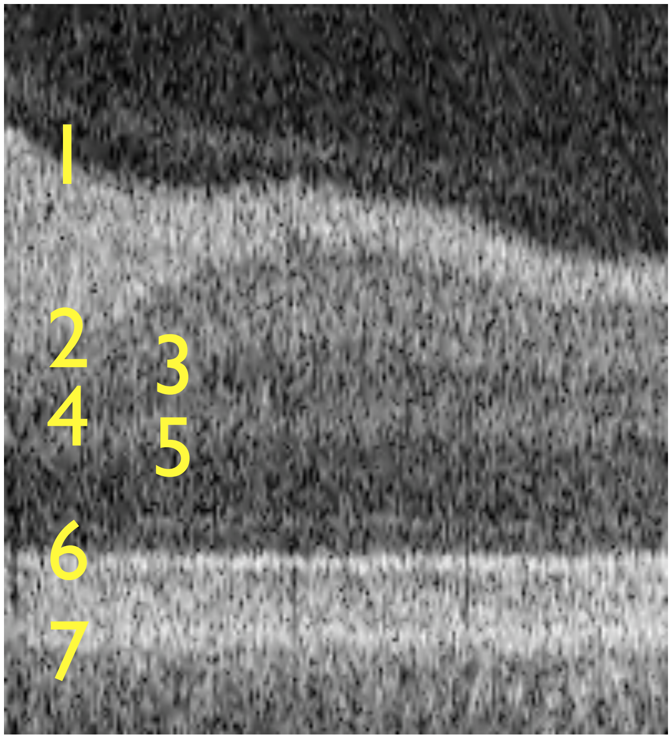
OCT Image



Result

The optimal set of surfaces

Example using both on-surface and in-region costs

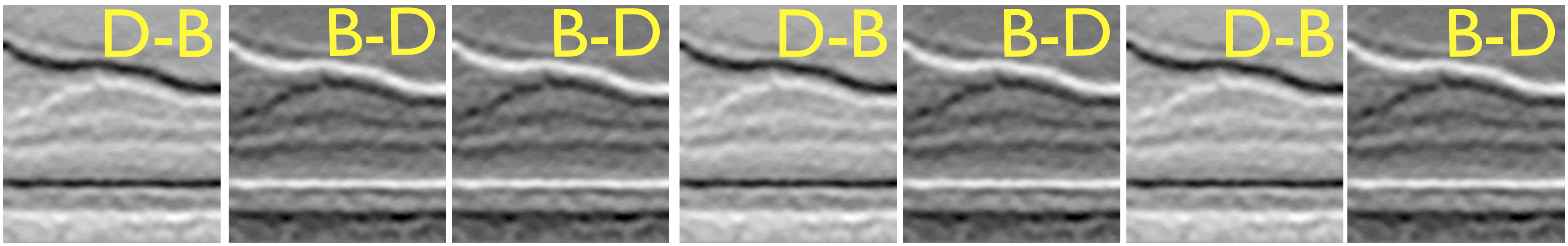


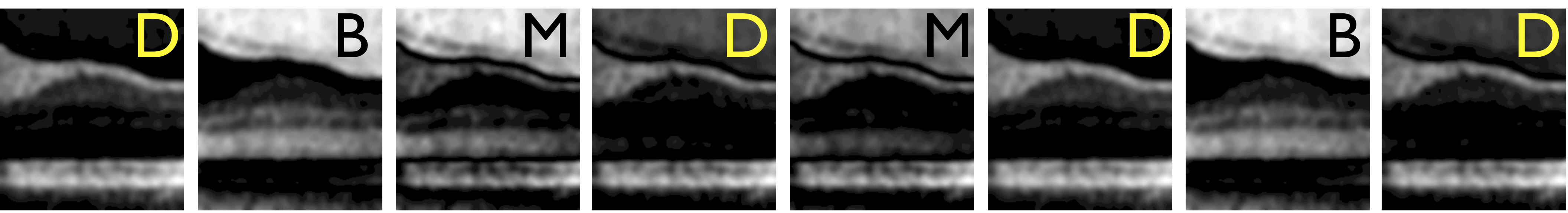
OCT Image

Find indicated 7 surfaces in OCT image using both on-surface and in-region costs.

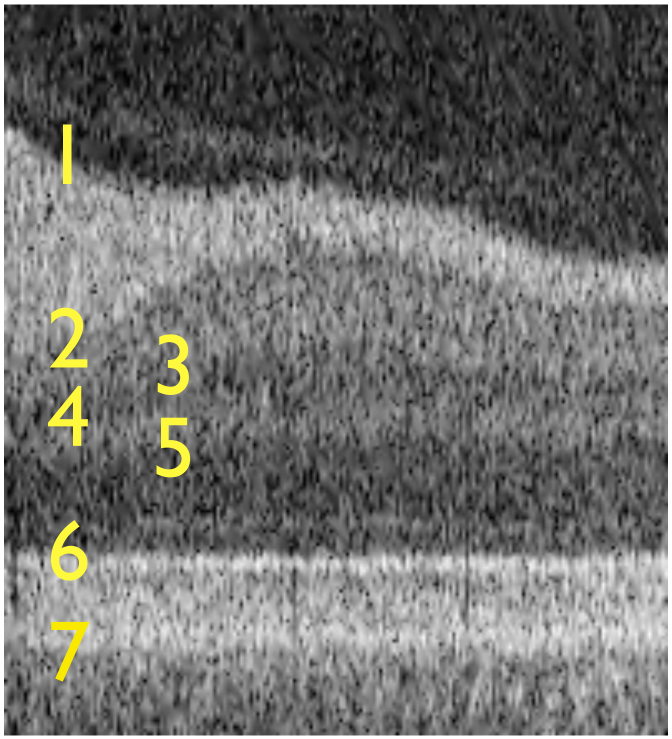
Example using both on-surface and in-region costs

Seven on-surface and eight in-region cost images:

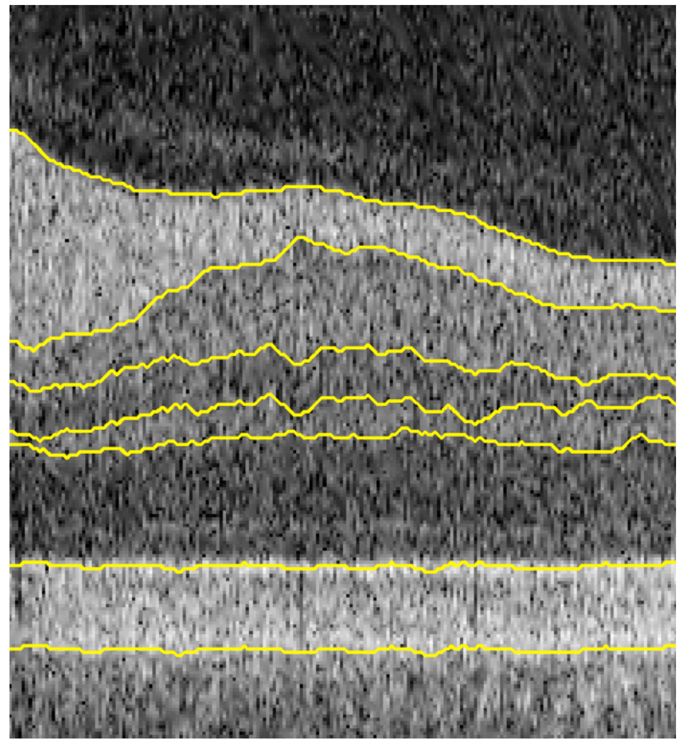




Example using both on-surface and in-region costs



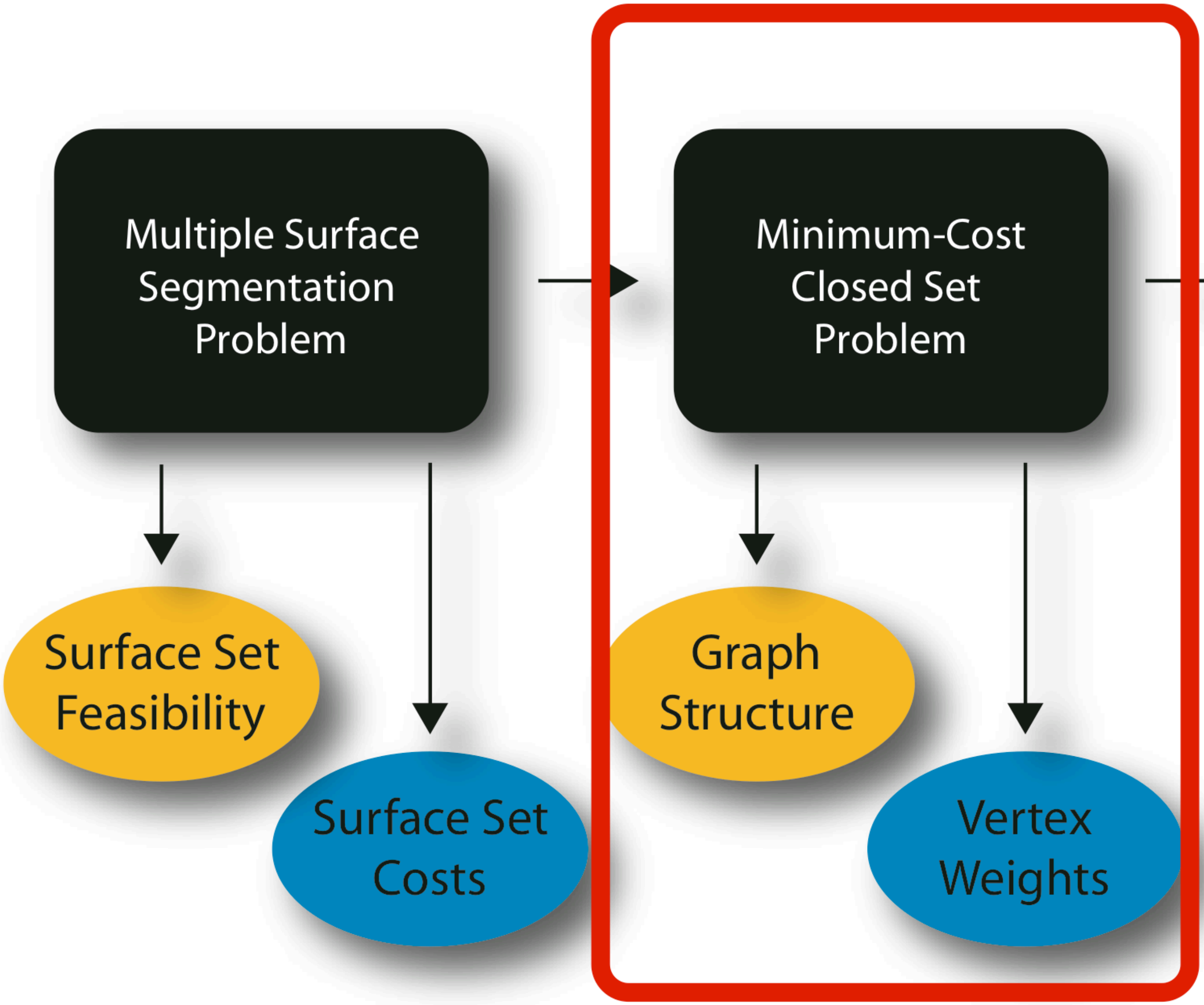
OCT Image



Result

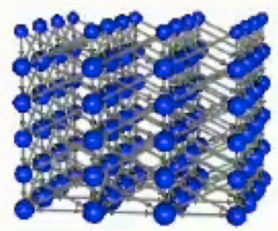
The optimal set of surfaces

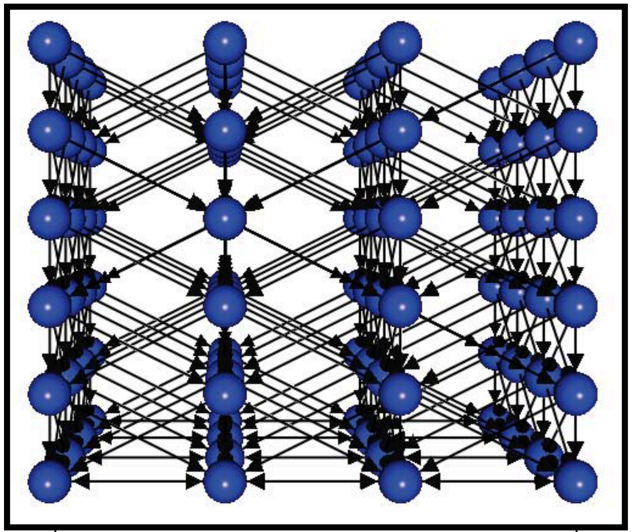
The graph structure ensures surface set feasibility. The assigned vertex weights ensure the optimal feasible surface set will be found.



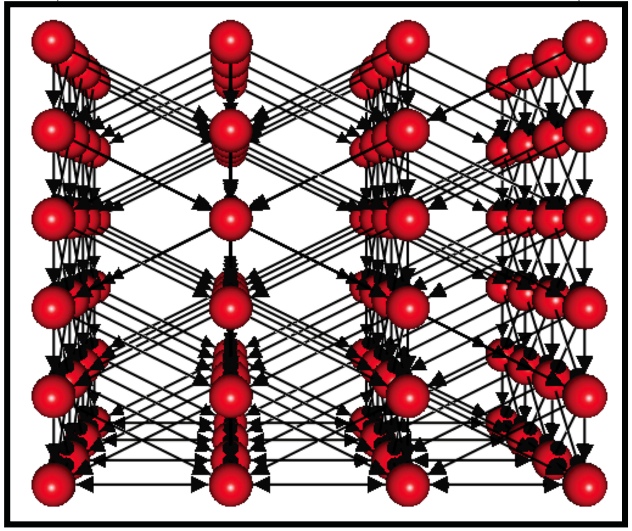
K. Li et al., PAMI 2006, extensions: M. Garvin et al., TMI 2009

Each set of feasible surfaces corresponds to a non-empty closed set in the constructed graph (and vice versa)



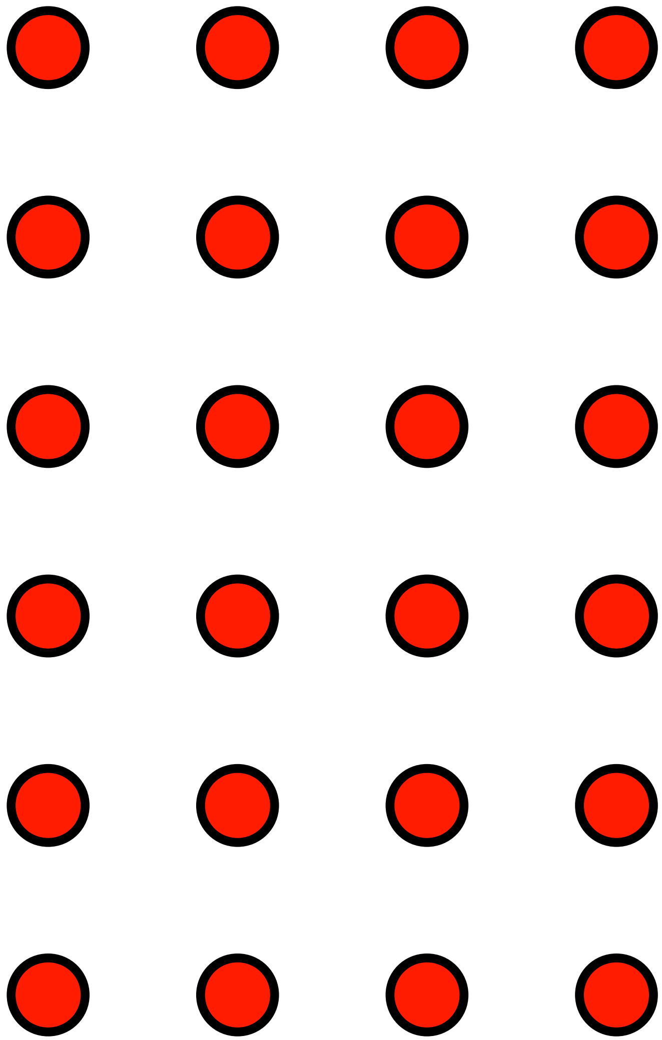


Intersurface edges



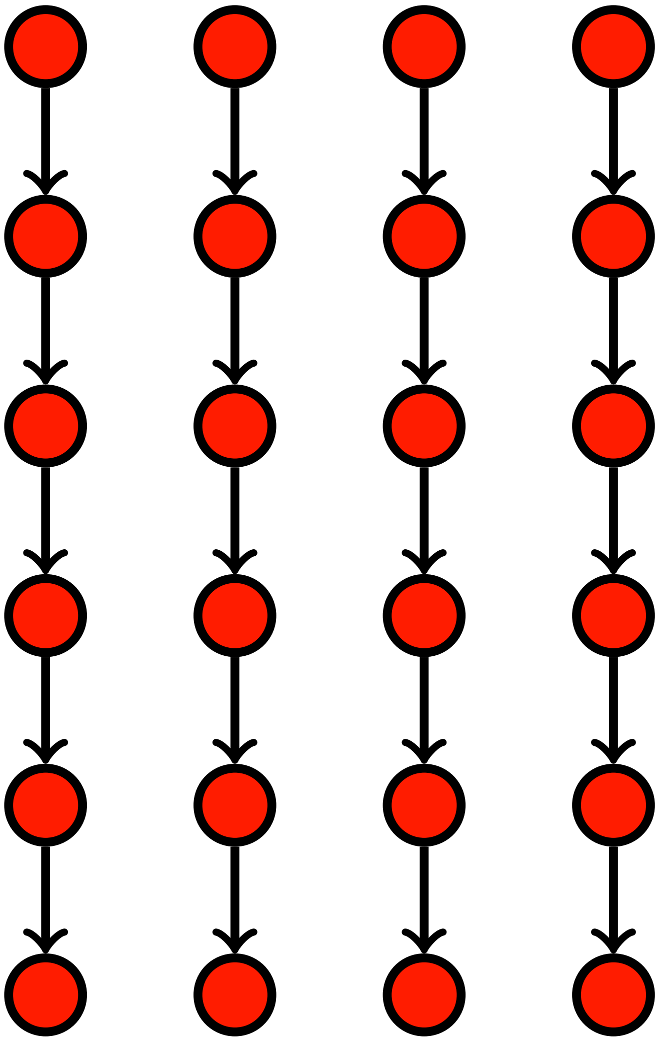
Intracolumn and intercolumn edges

Intracolumn and intercolumn edges enforce smoothness constraints



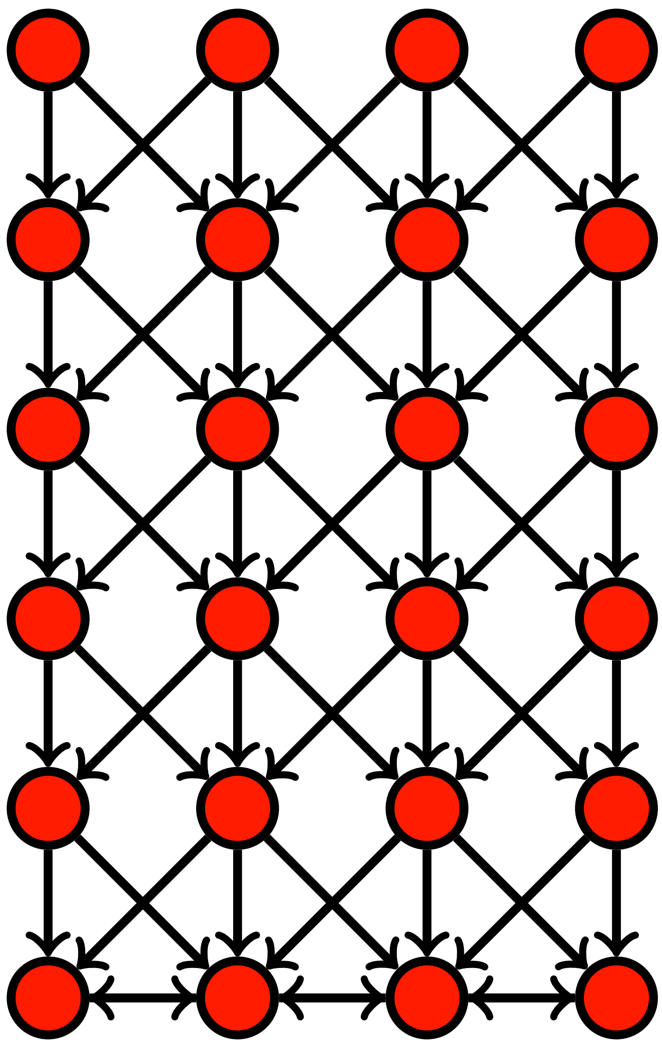
Graph nodes

Intracolumn and intercolumn edges enforce smoothness constraints



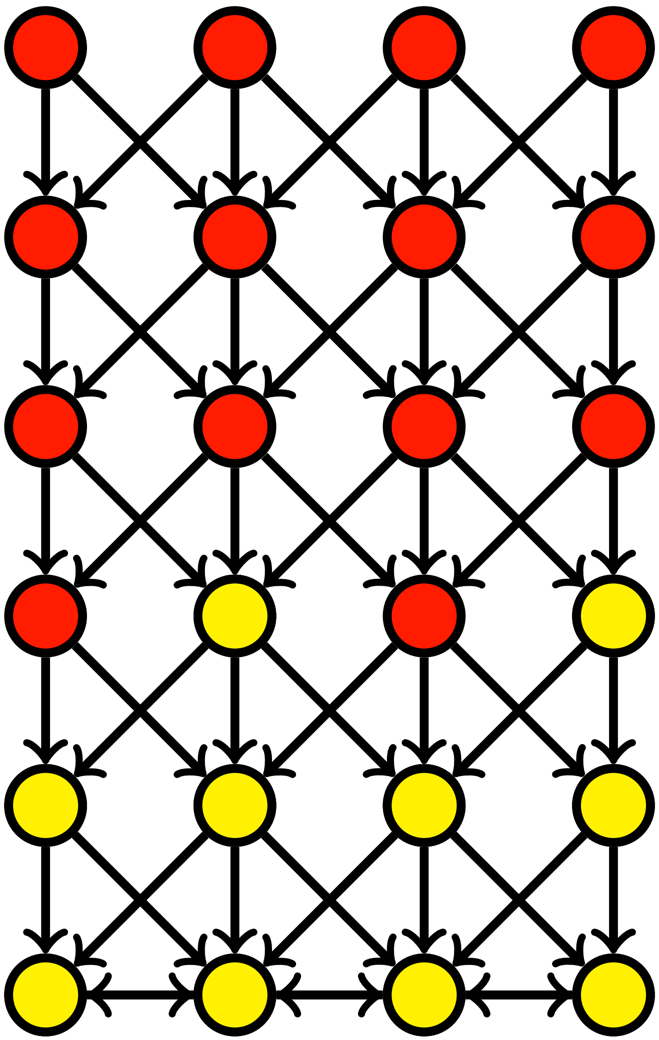
Add intracolumn edges

Intracolumn and intercolumn edges enforce smoothness constraints



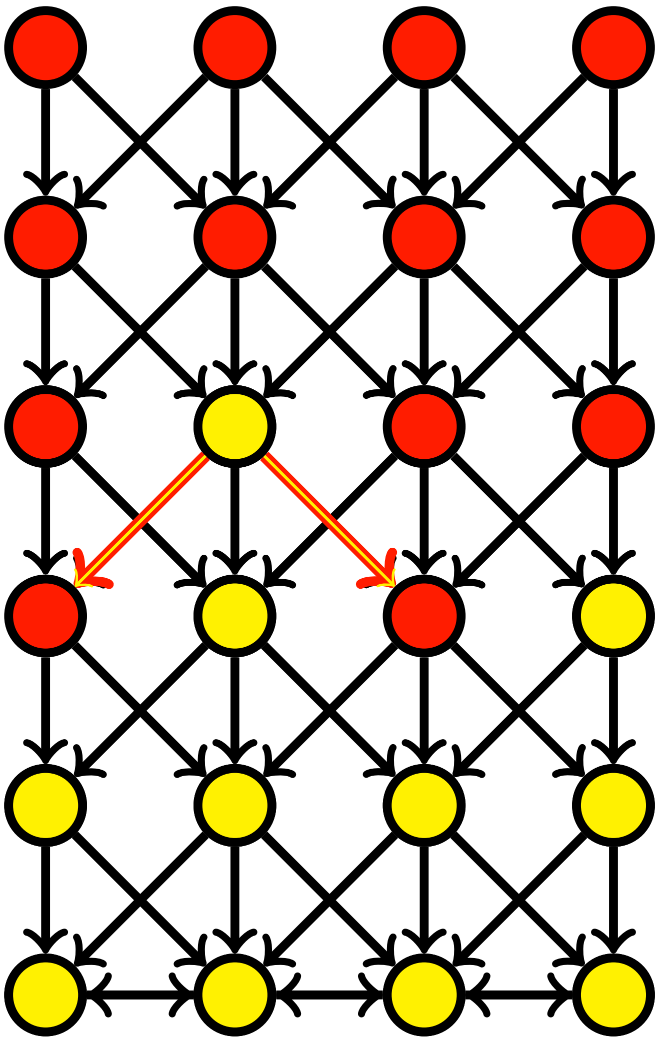
Add intercolumn edges

Intracolumn and intercolumn edges enforce smoothness constraints



A closed set

Intracolumn and intercolumn edges enforce smoothness constraints

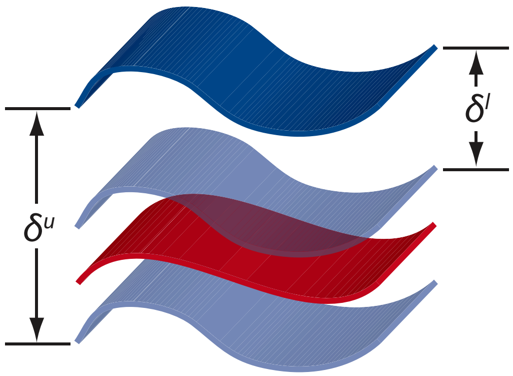


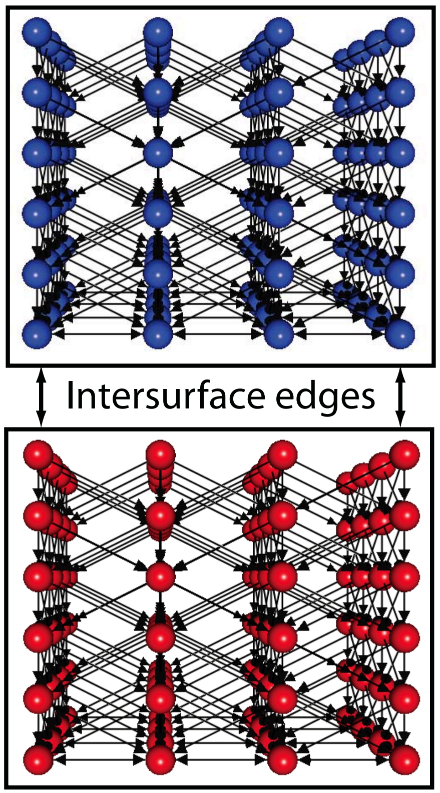
Not a closed set

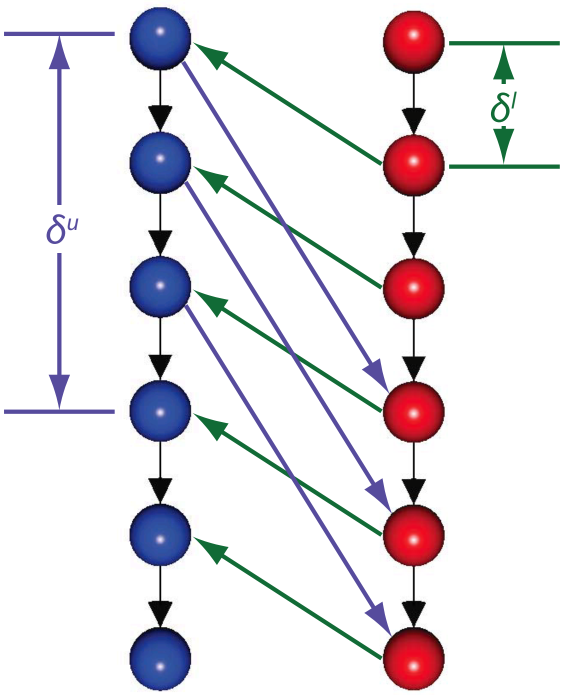
Intersurface edges enforce surface interaction constraints

Surface interaction constraints:

$$\delta^l(x,y) \leq f_i(x,y) - f_j(x,y) \leq \delta^u(x,y)$$







The cost of each surface set corresponds (within a constant) to the cost of the corresponding closed set

Surface set cost:

$$\sum_{i=1}^{n} C_{f_i(x,y)} + \sum_{i=0}^{n} C_{R_i}$$

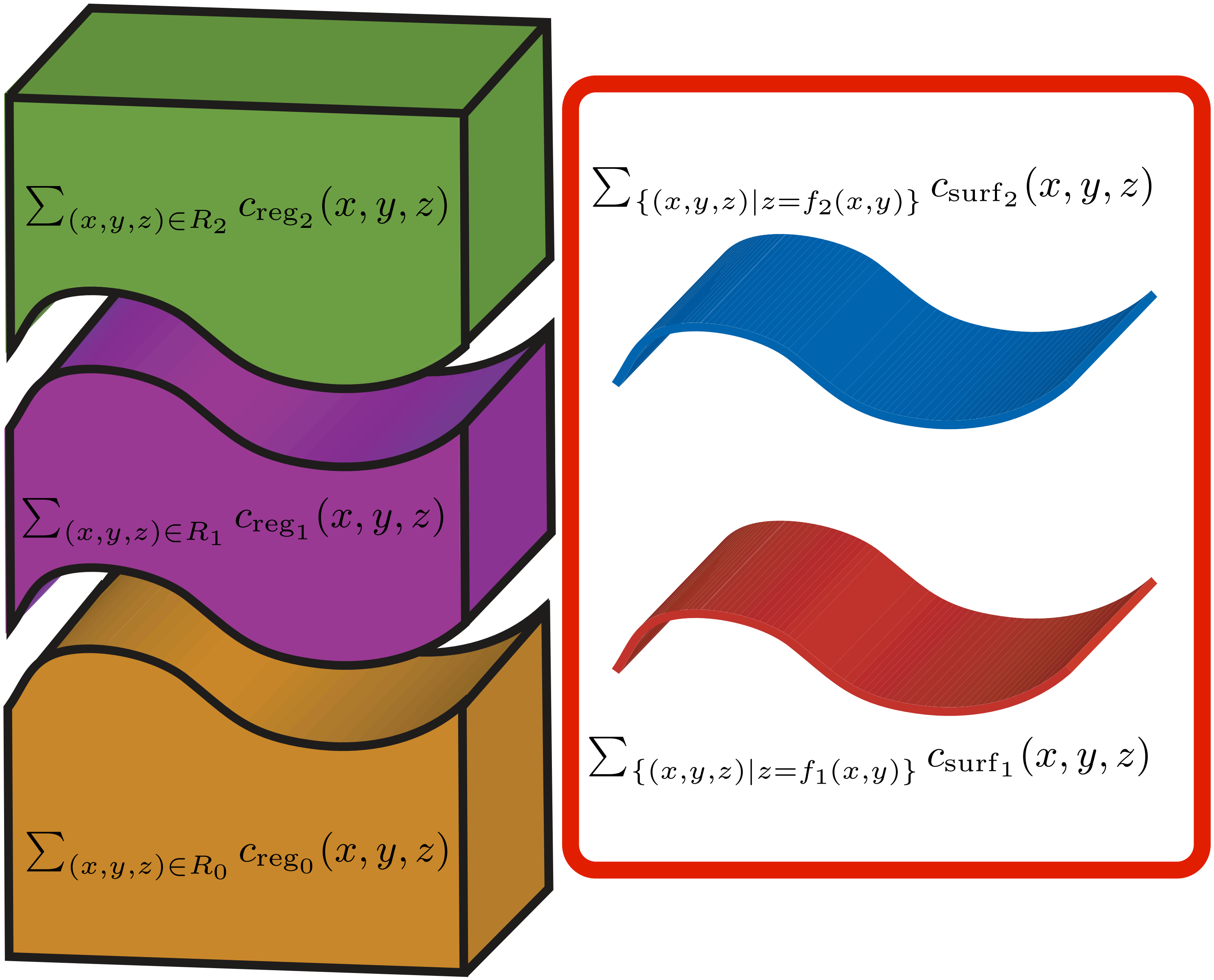
Closed set cost:

$$\sum_{i=1}^{n} C_{f_i(x,y)} + \sum_{i=0}^{n} C_{R_i} + K$$

↓

Minimum closed set → Minimum surface set cost

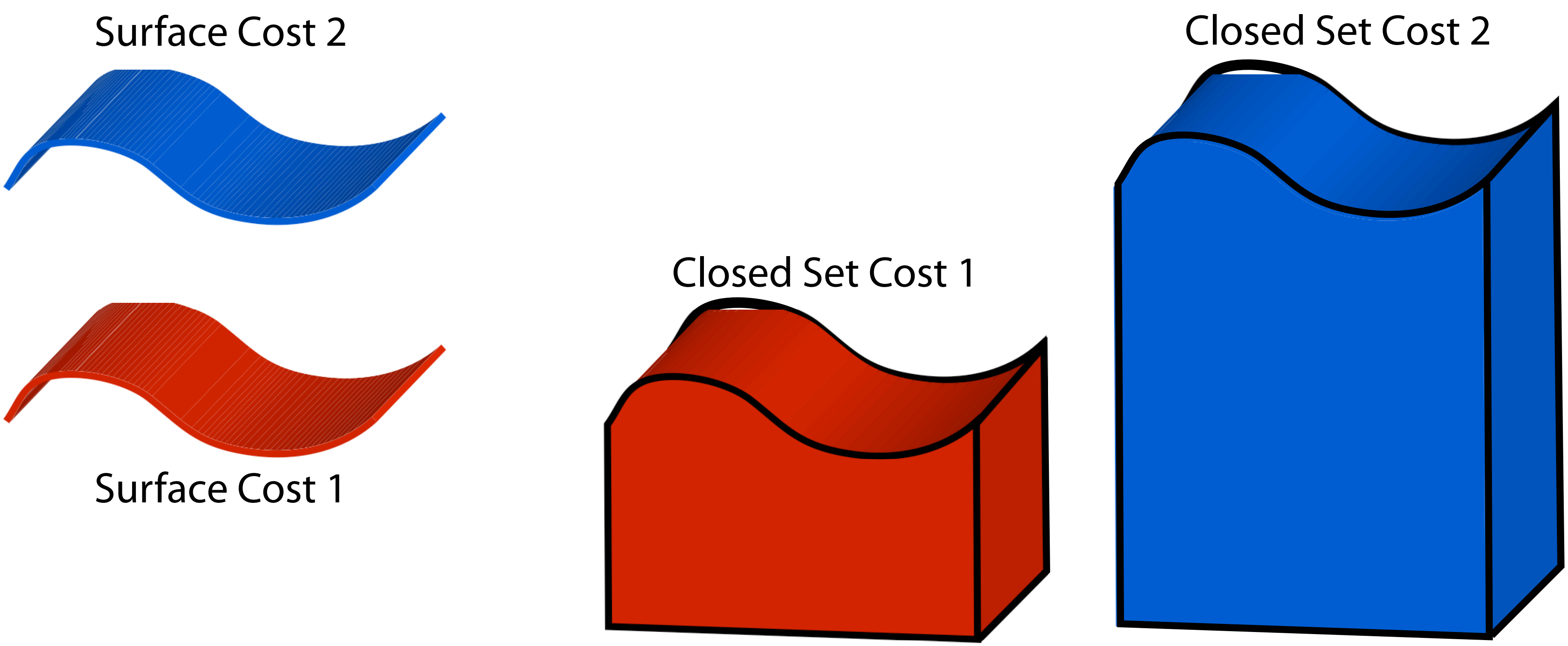
On-surface cost representation



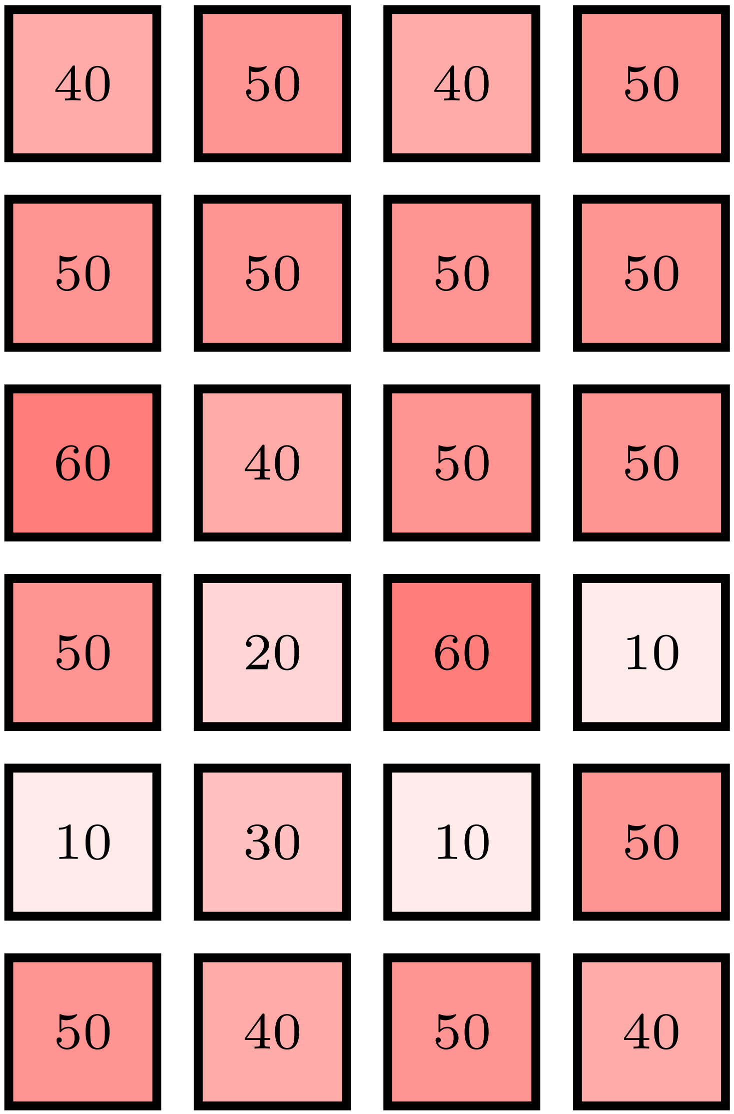
On-surface cost representation

node weight:

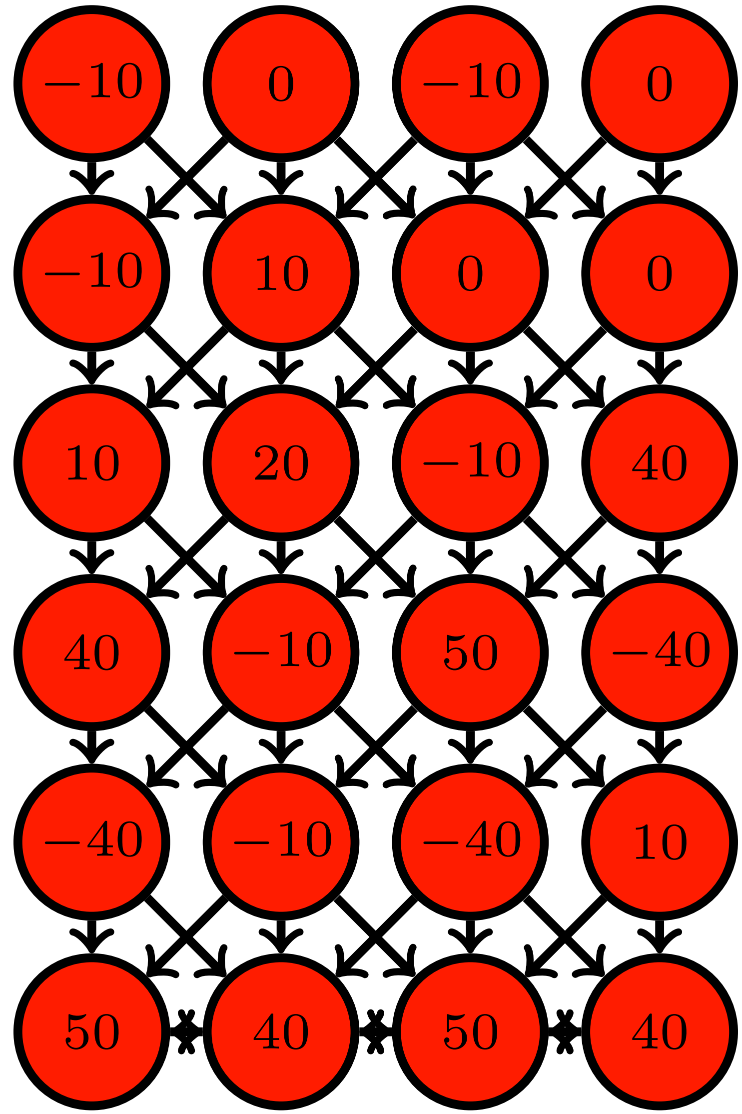
$$w_{\text{on-surf}_i}(x,y,z) = \begin{cases} c_{\text{surf}_i}(x,y,z) & \text{if} \quad z=0 \\ c_{\text{surf}_i}(x,y,z) - c_{\text{surf}_i}(x,y,z-1) & \text{otherwise} \end{cases}$$



Graph representation of on-surface costs (toy example)

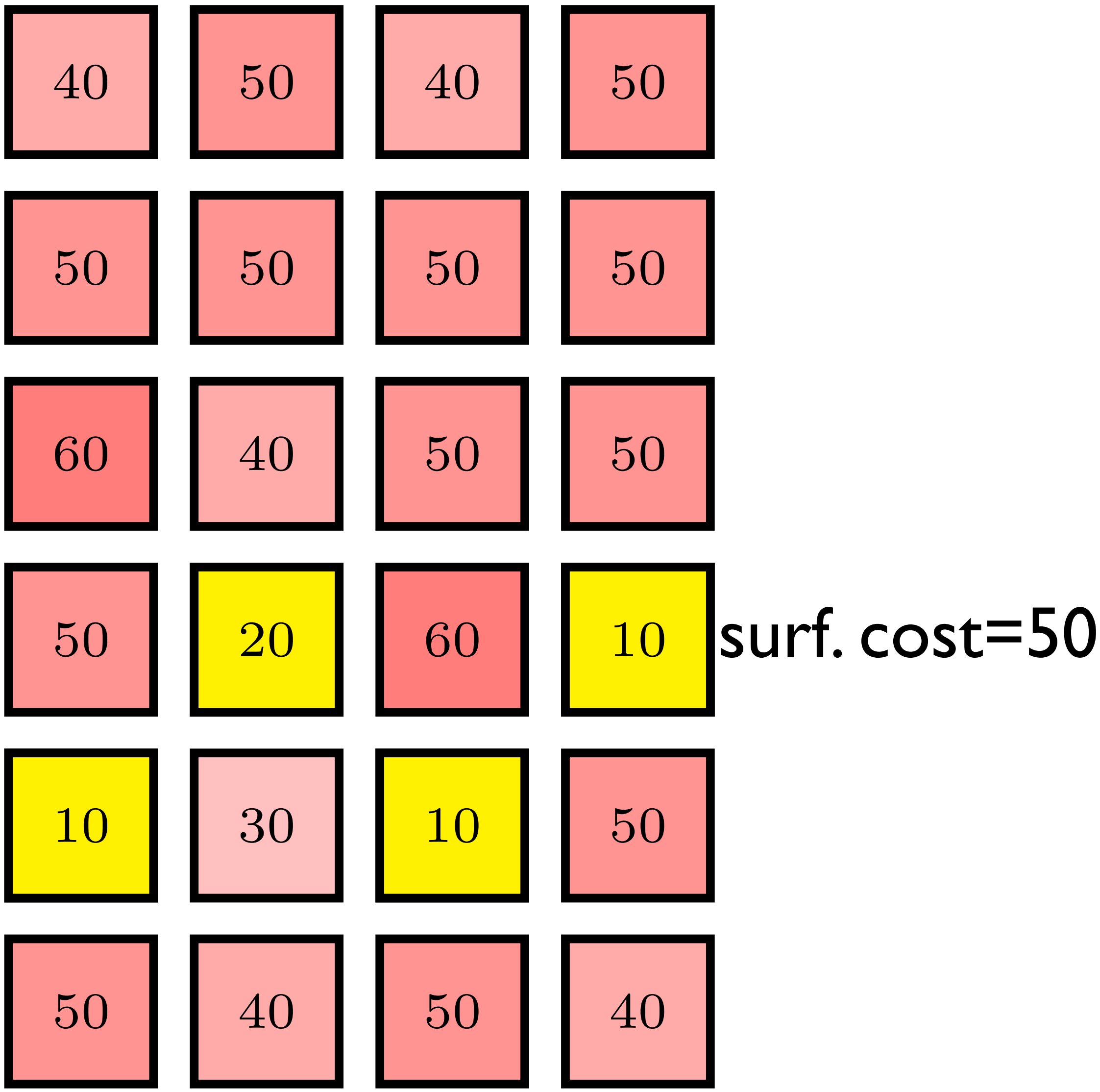


cost image

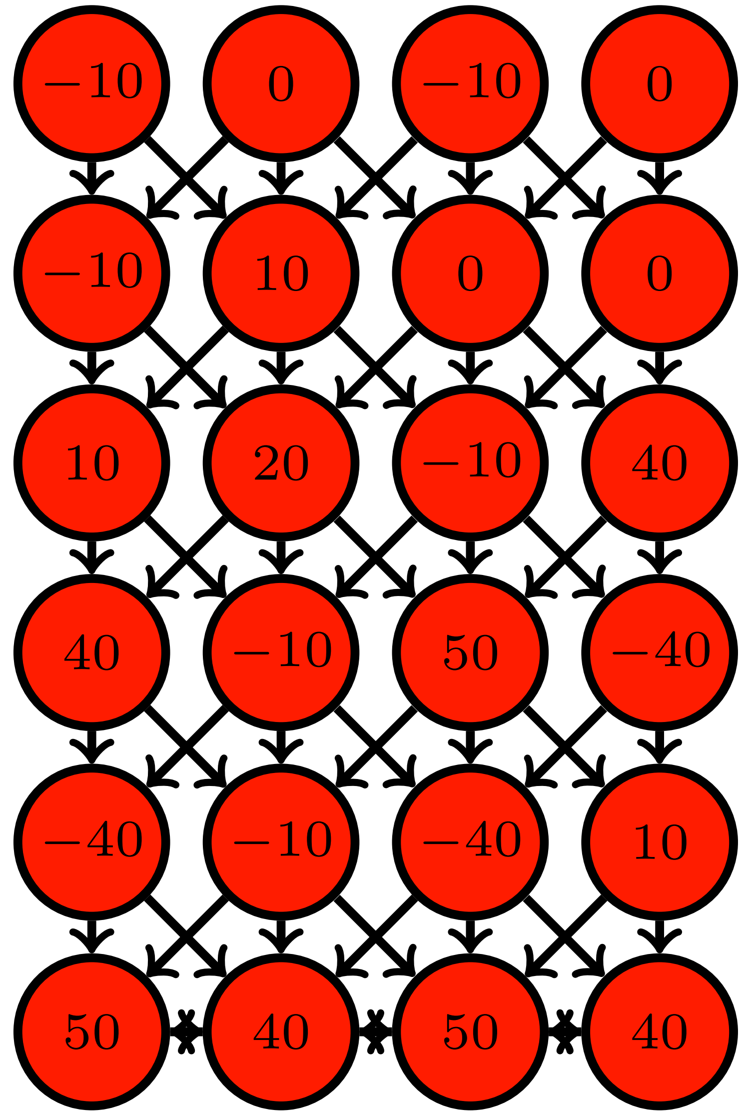


graph representation

Graph representation of on-surface costs (toy example)

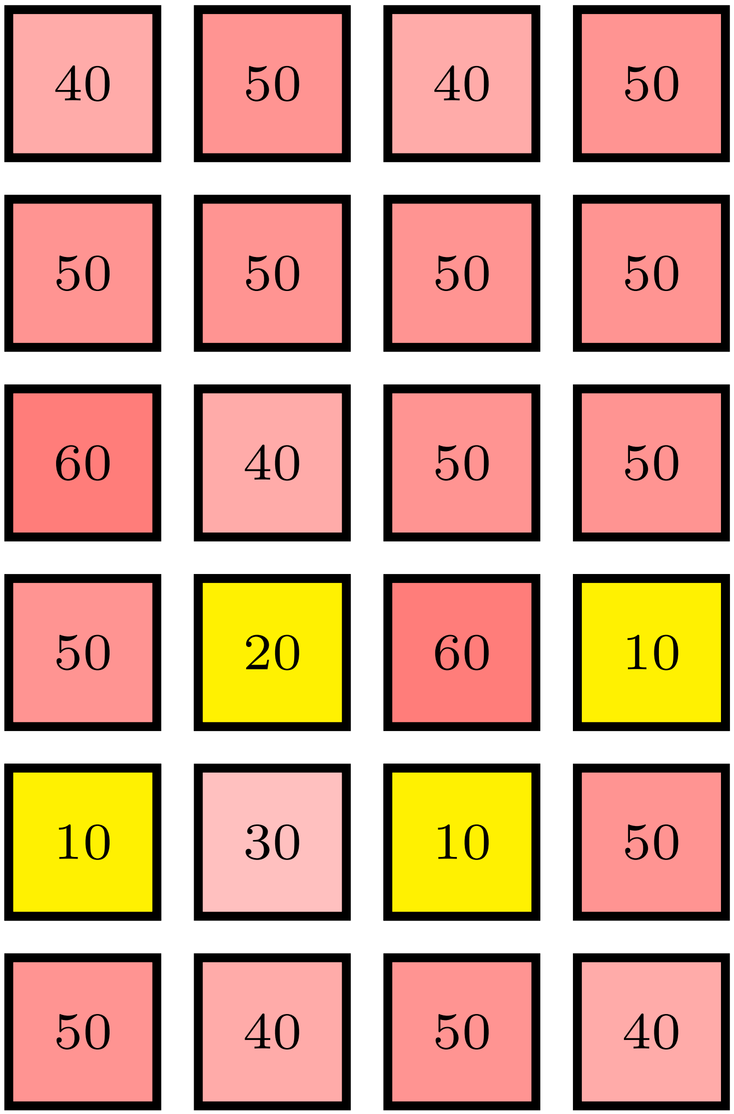


cost image



graph representation

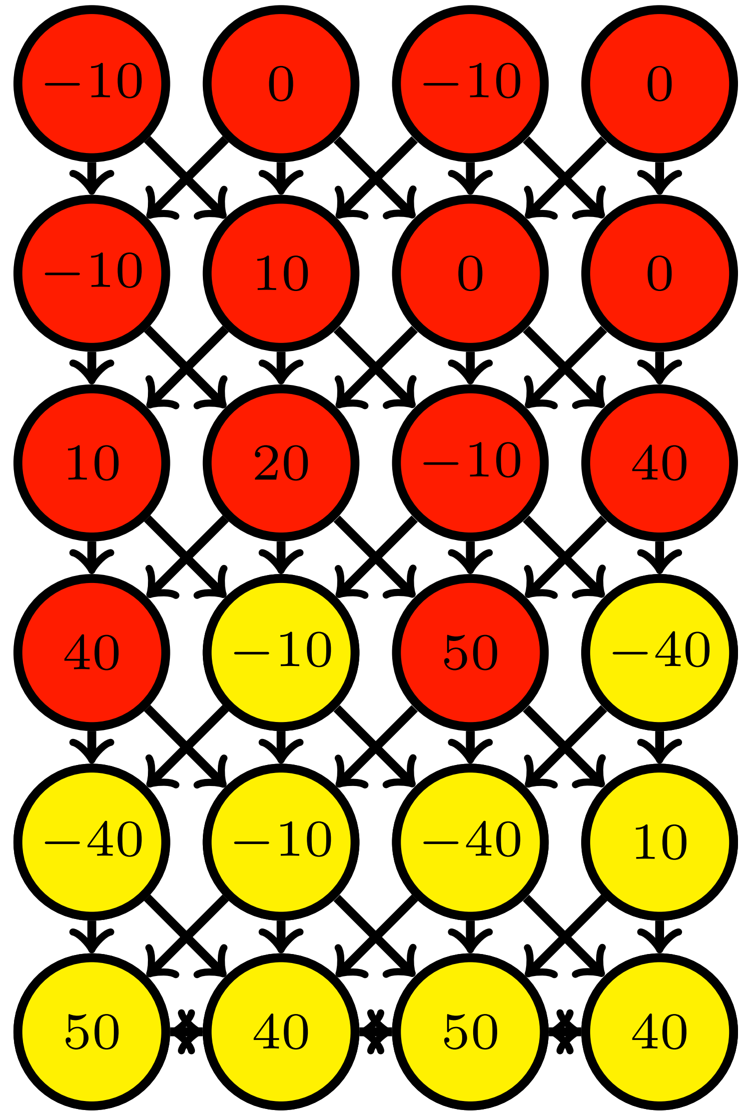
Graph representation of on-surface costs (toy example)



cost image

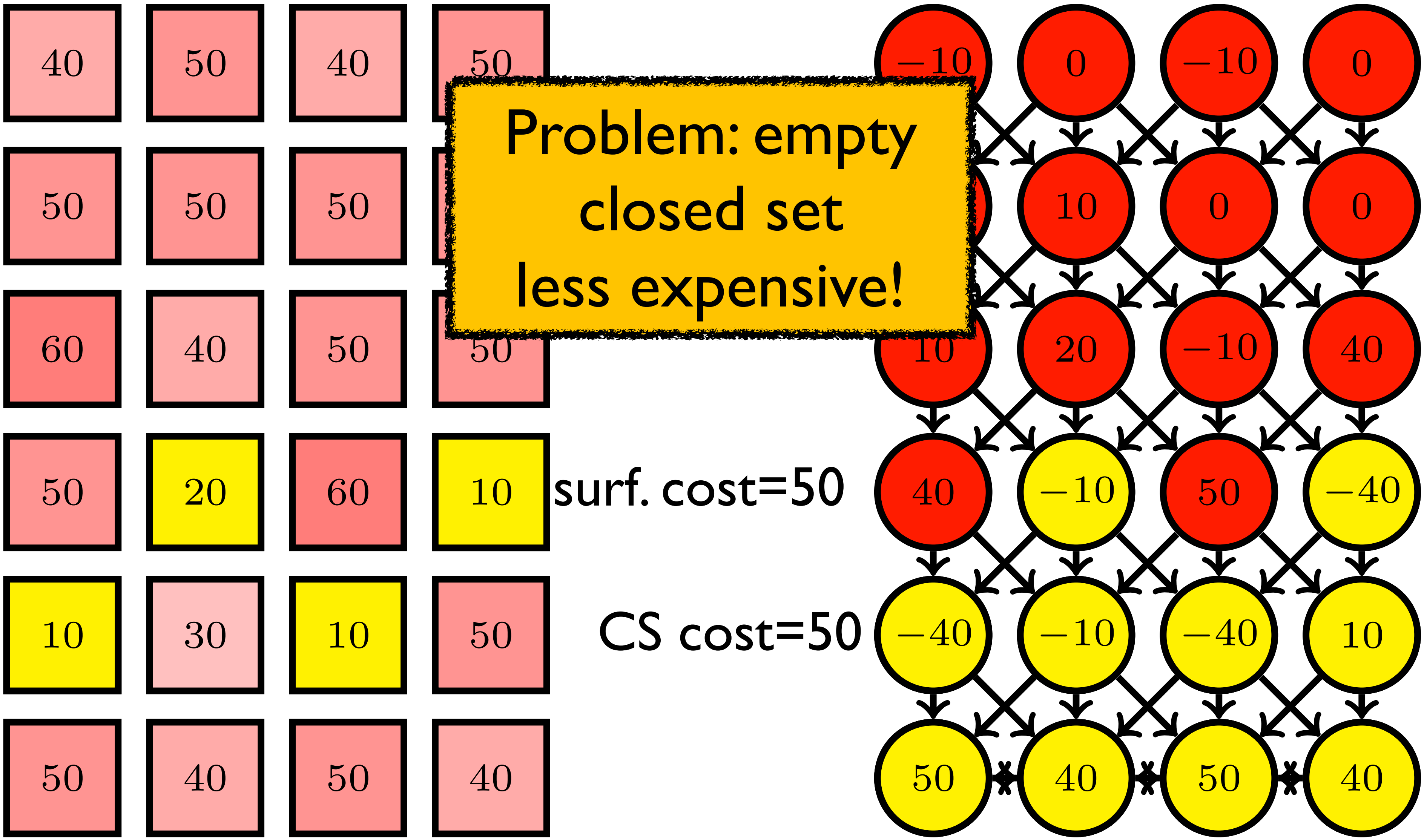
surf. cost=50

CS cost=50



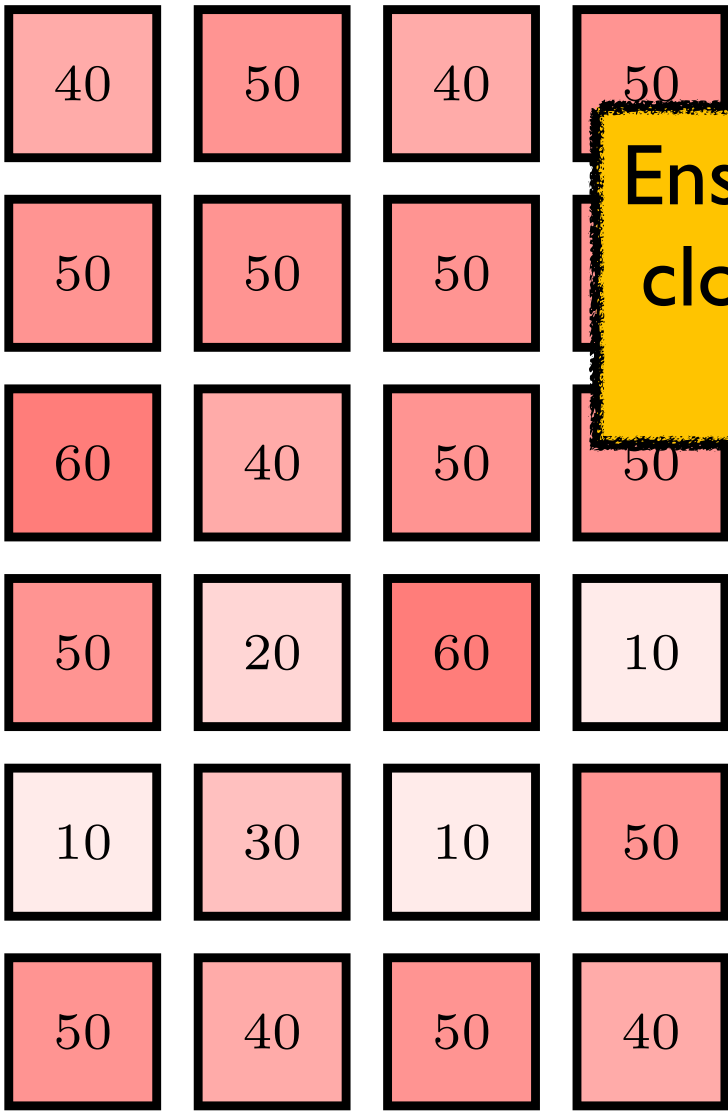
graph representation

Graph representation of on-surface costs (toy example)



cost image   graph representation

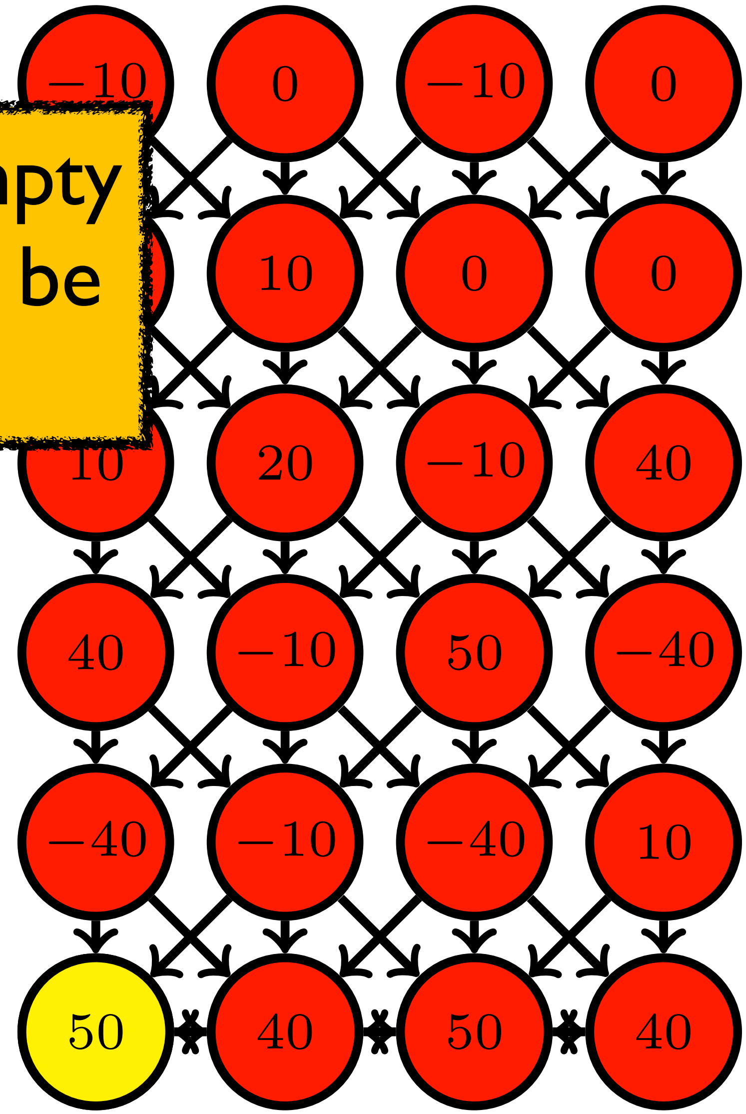
Graph representation of on-surface costs (toy example)



cost image

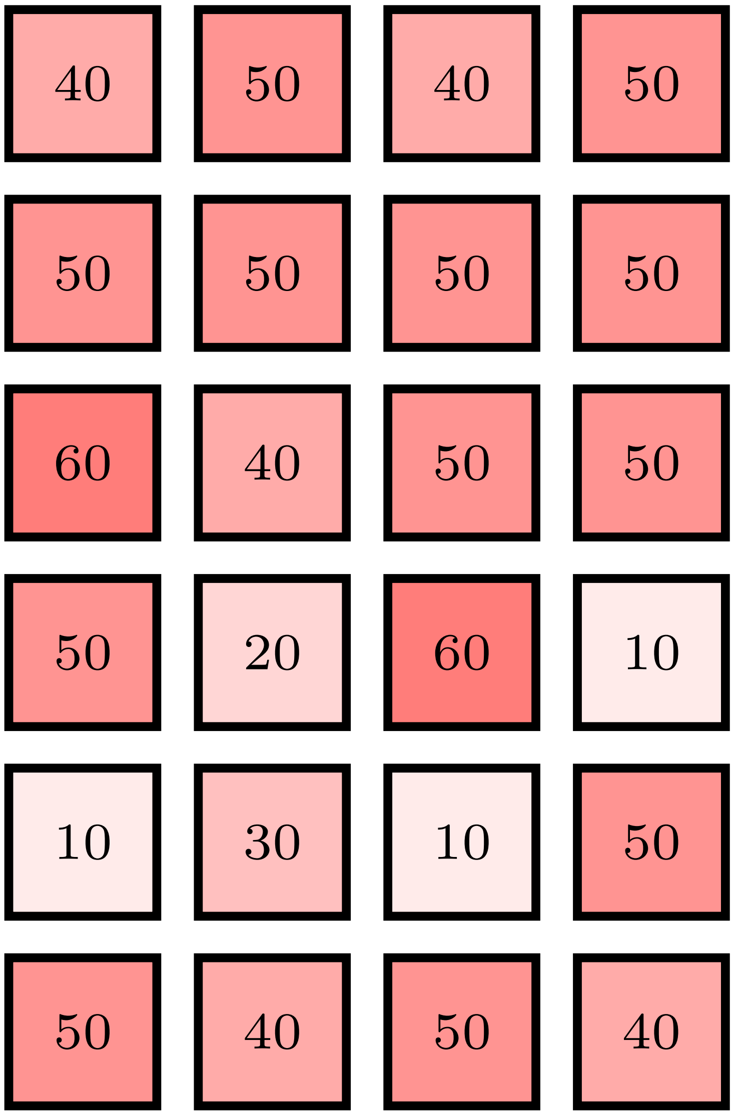
Ensure non-empty closed set will be minimum.

subtract (sum of last row + 1) = 181



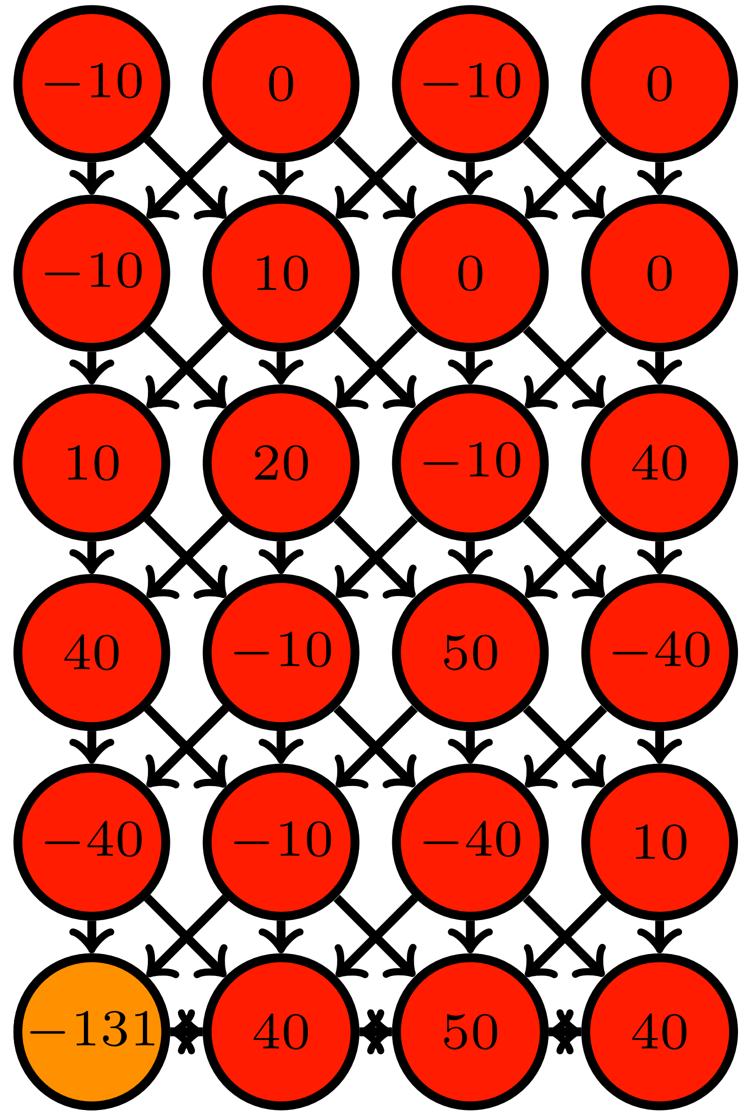
graph representation

Graph representation of on-surface costs (toy example)



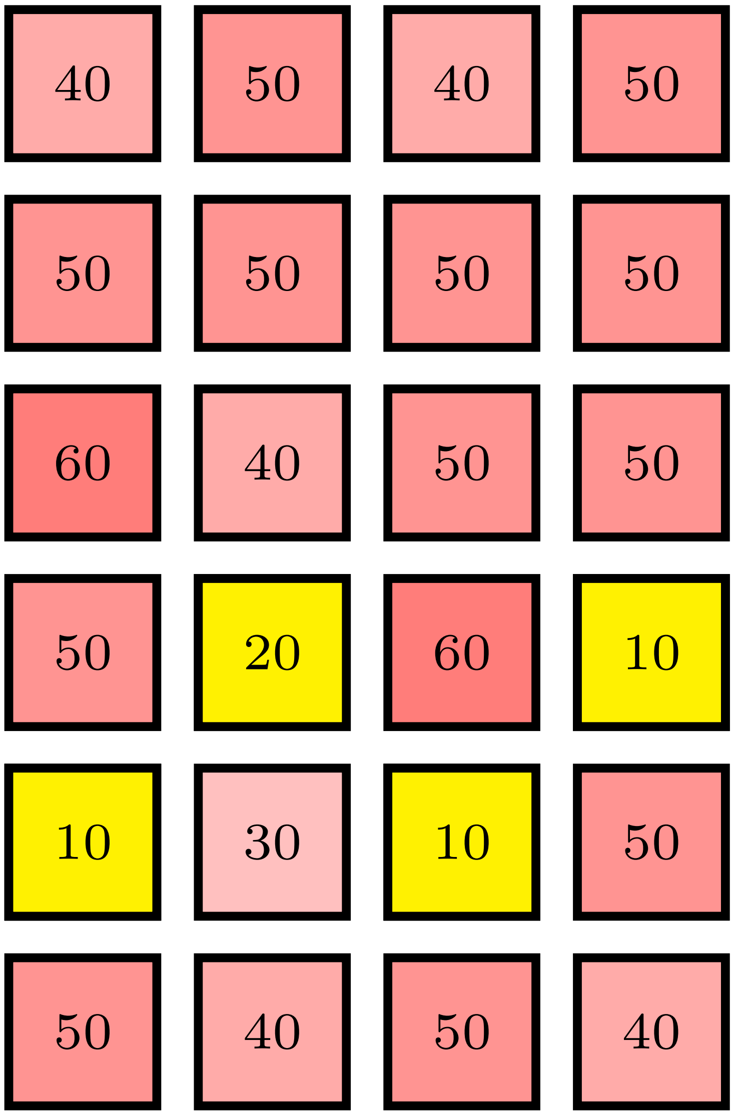
cost image

50 - 181



graph representation

Graph representation of on-surface costs (toy example)

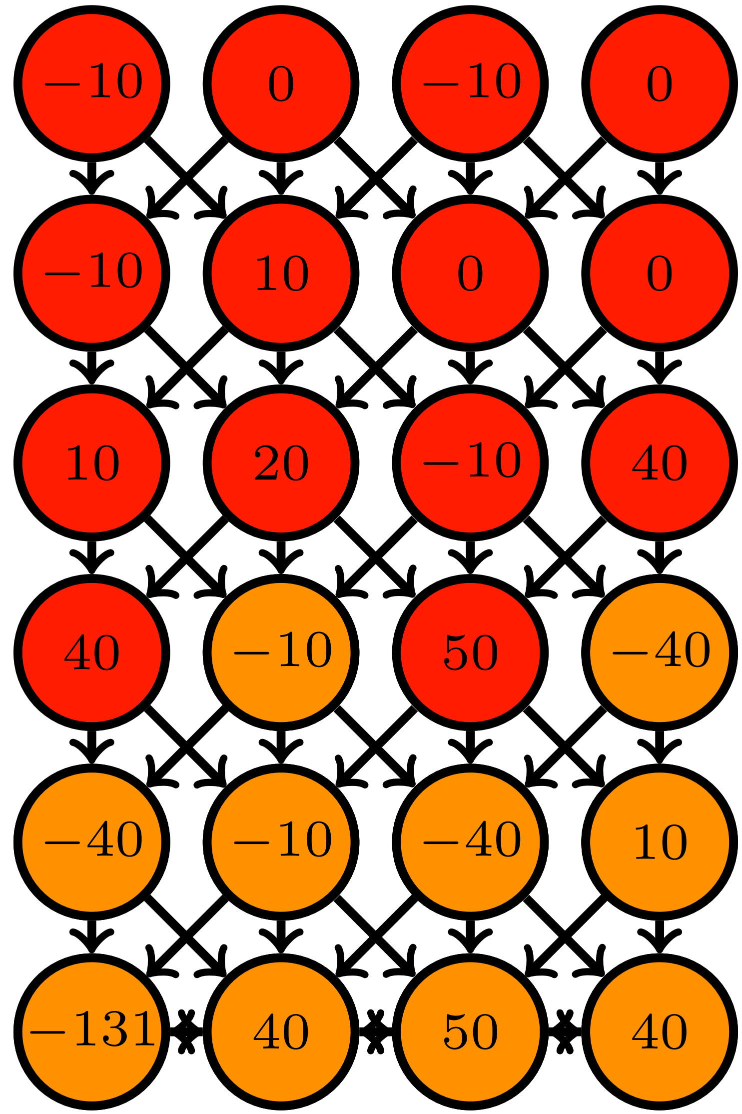


cost image

surf. cost=50

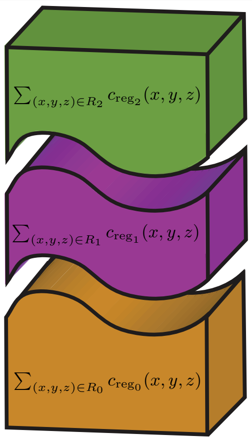
CS cost=
K + 50

(K = -181)



graph representation

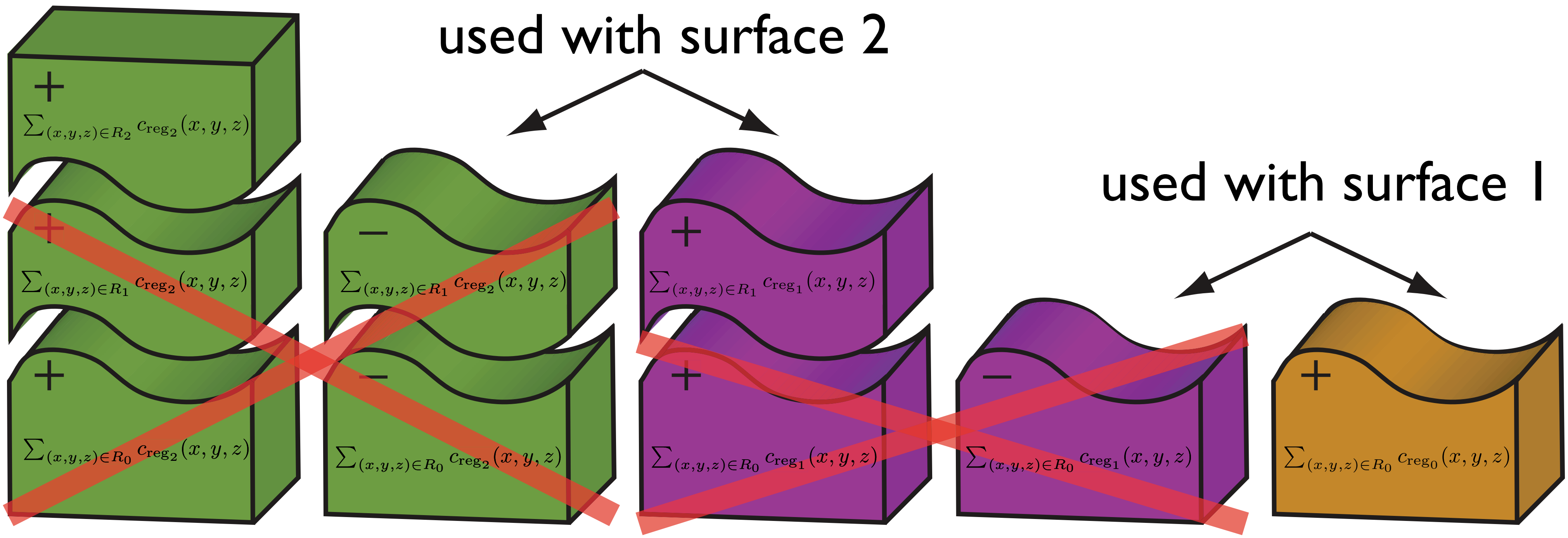
In-region cost representation



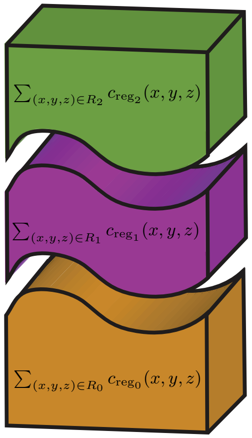
node weight (in subgraph associated with surface *i*)

$$w_{\mathrm{in-reg}_i}(x,y,z) = c_{\mathrm{reg}_{i-1}}(x,y,z) - c_{\mathrm{reg}_i}(x,y,z)$$

(region below) (region above)



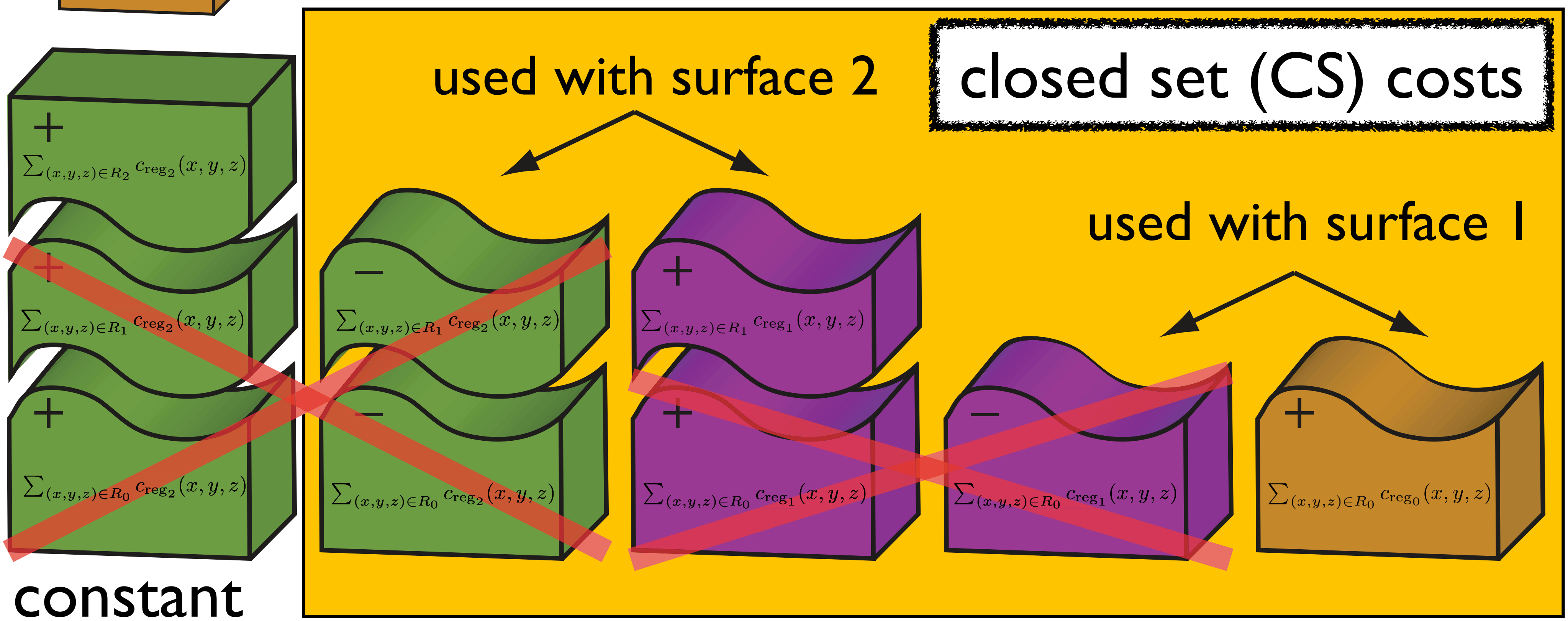
In-region cost representation



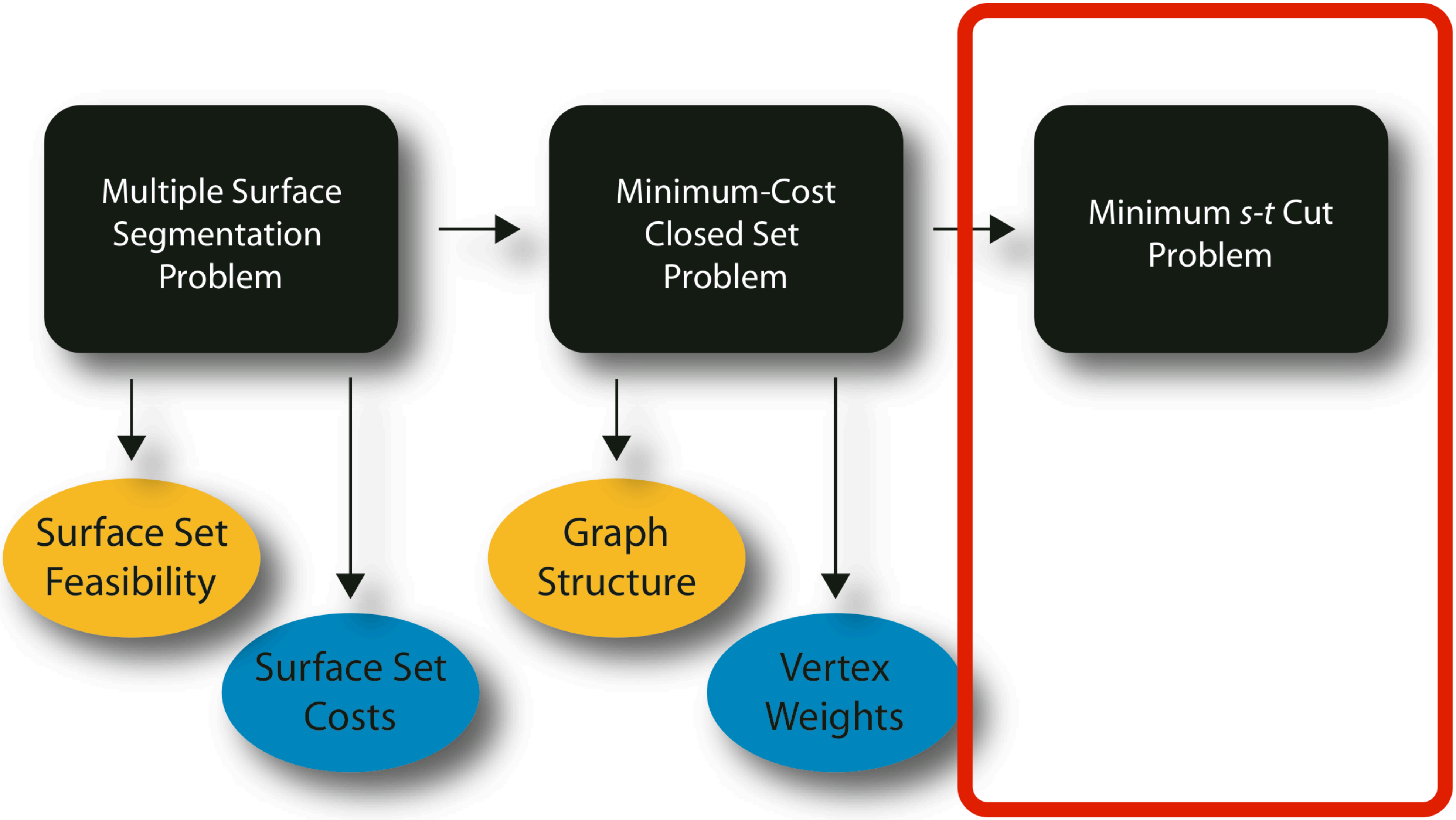
node weight (in subgraph associated with surface *i*)

$$w_{\text{in-reg}_i}(x,y,z) = c_{\text{reg}_{i-1}}(x,y,z) - c_{\text{reg}_i}(x,y,z)$$

(region below) (region above)



The graph structure ensures surface set feasibility. The assigned vertex weights ensure the optimal feasible surface set will be found.



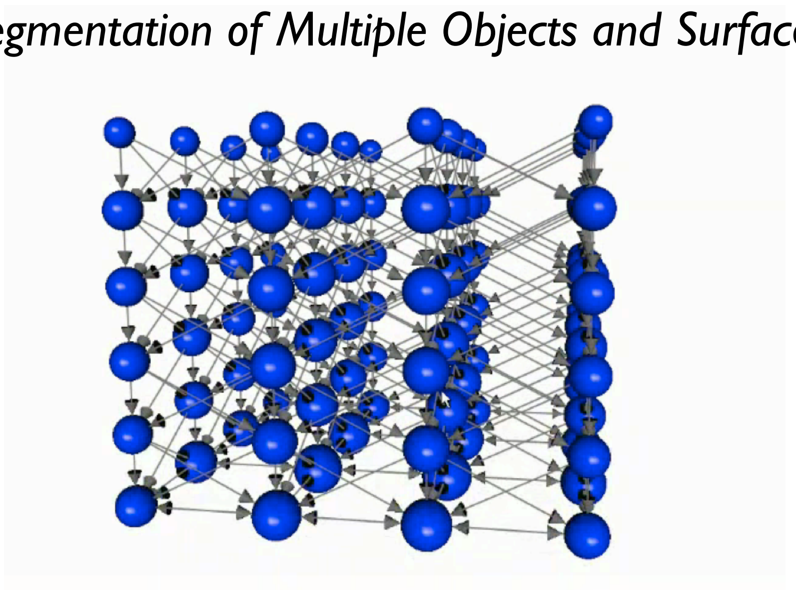
K. Li et al., PAMI 2006, extensions: M. Garvin et al., TMI 2009

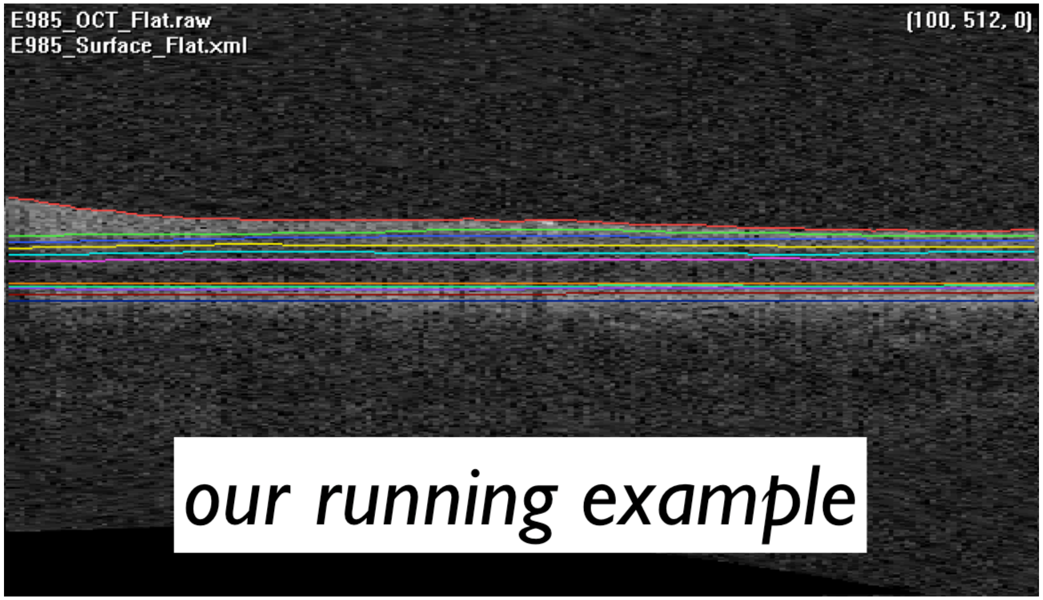
Outline

LOGISMOS approach

Intraretinal layer segmentation

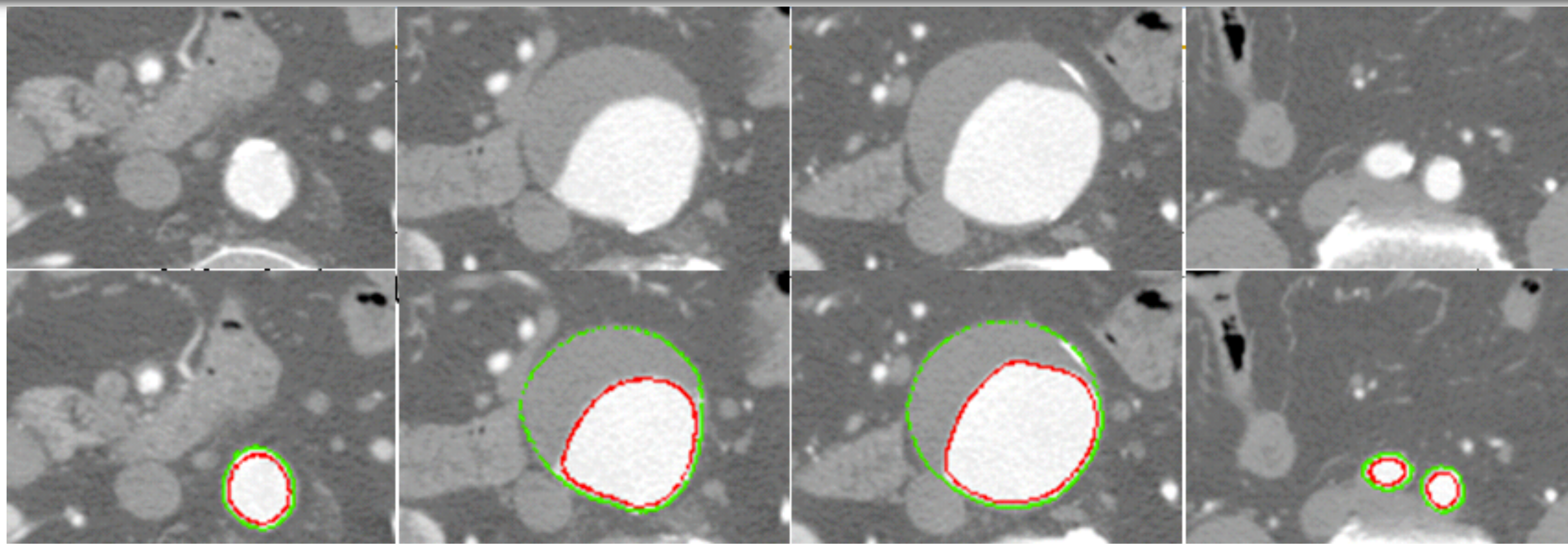
LOGISMOS = Layered Optimal Graph Image Segmentation of Multiple Objects and Surfaces

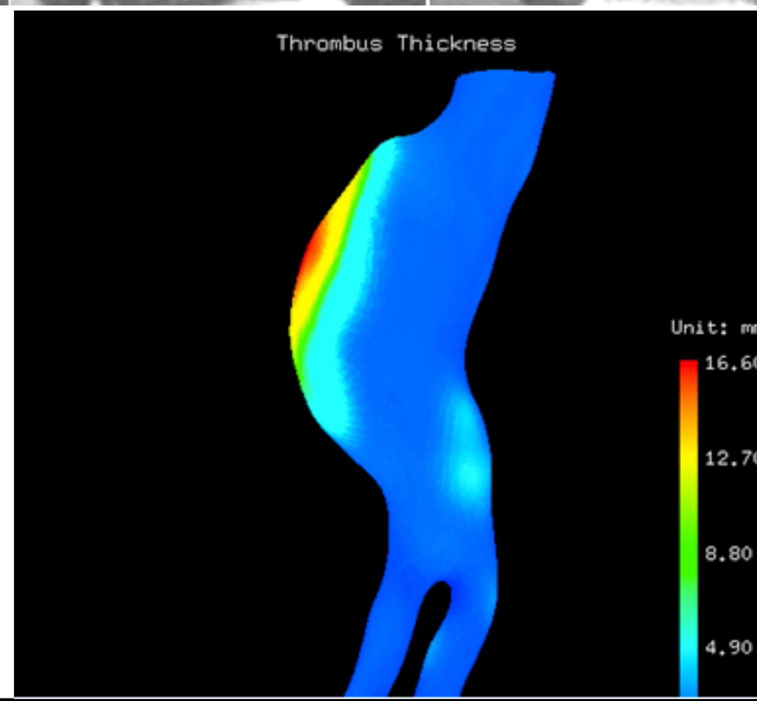




Other applications and future directions

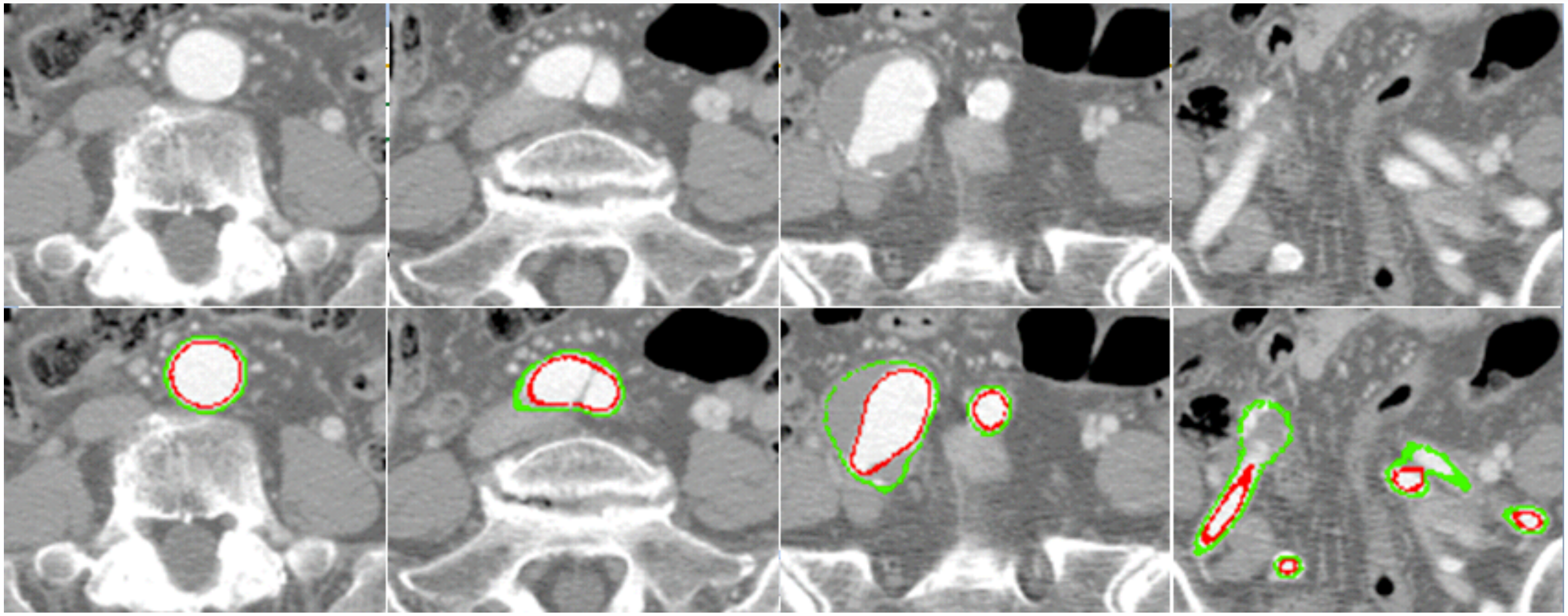
Example other LOGISMOS applications: thrombus segmentation in abdominal aortic aneurysms



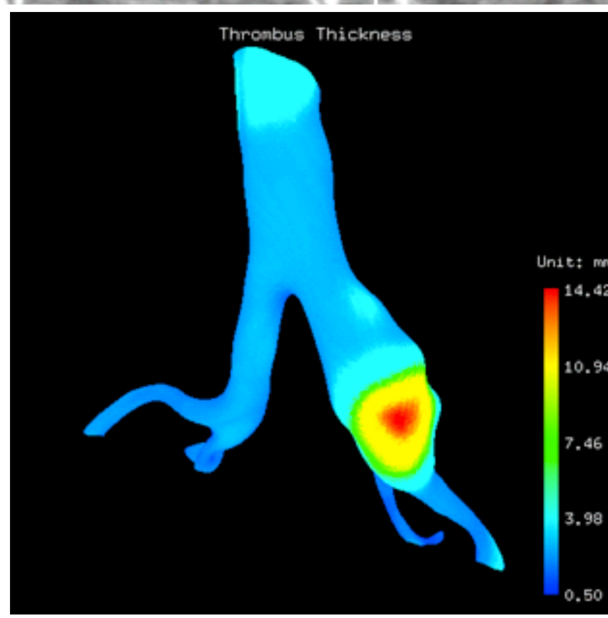


Lee, et al., Comput Biol Med, 2010

Example other LOGISMOS applications: thrombus segmentation in abdominal aortic aneurysms



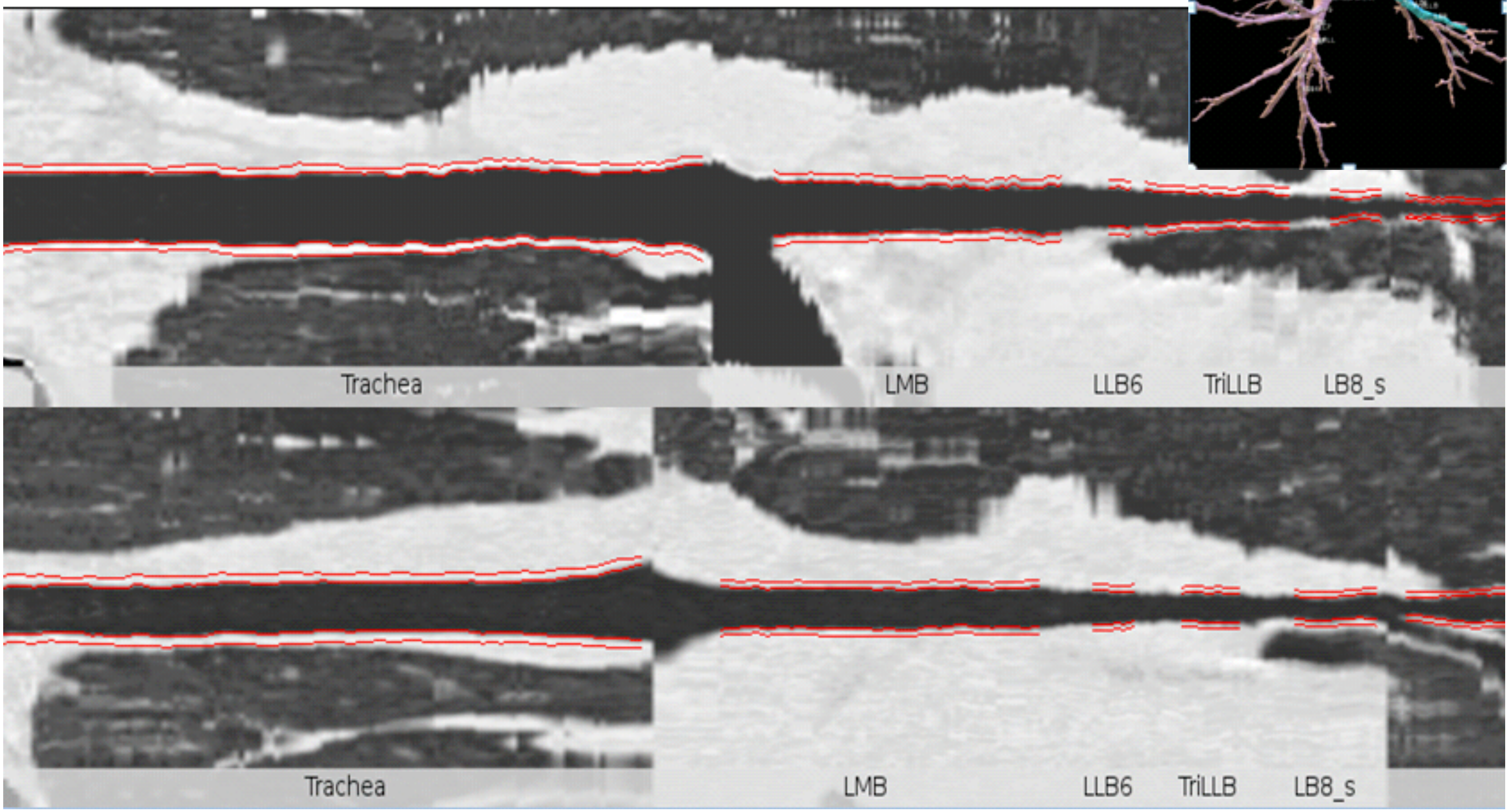
(from Sonka, et al.)



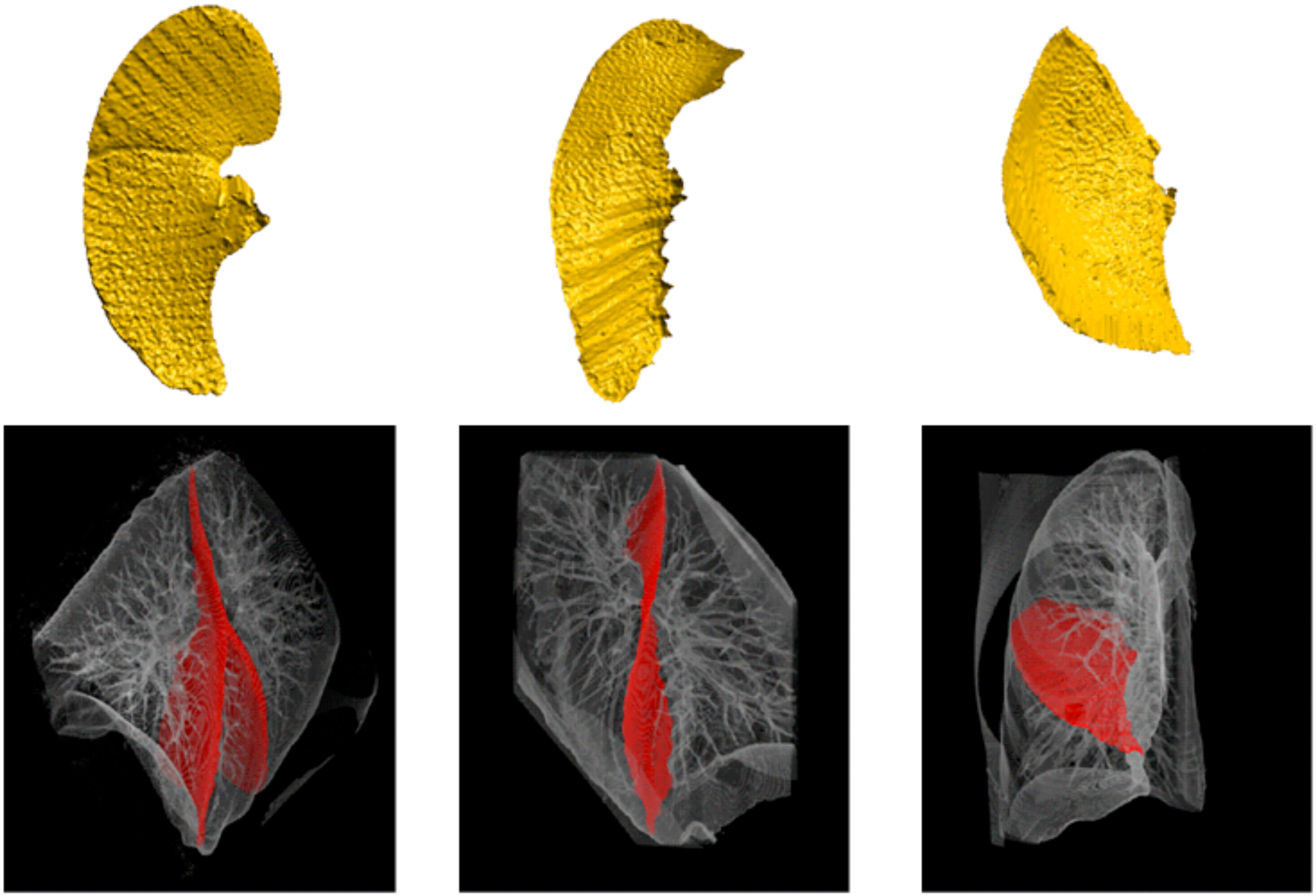
Lee, et al., Comput Biol Med, 2010

Example other LOGISMOS applications: airway tree

(from Sonka, et al.)

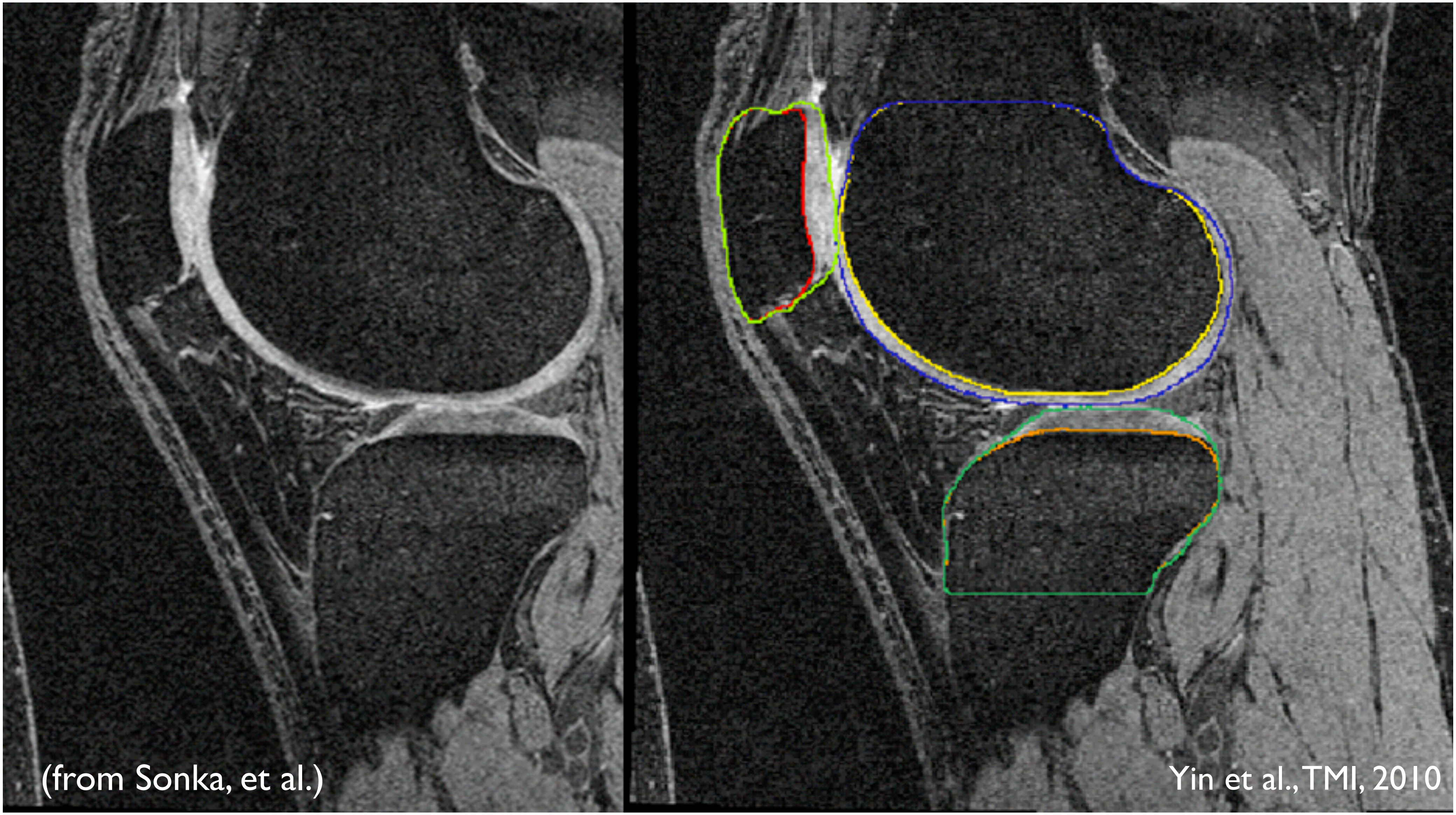


Example other LOGISMOS applications: pulmonary fissures

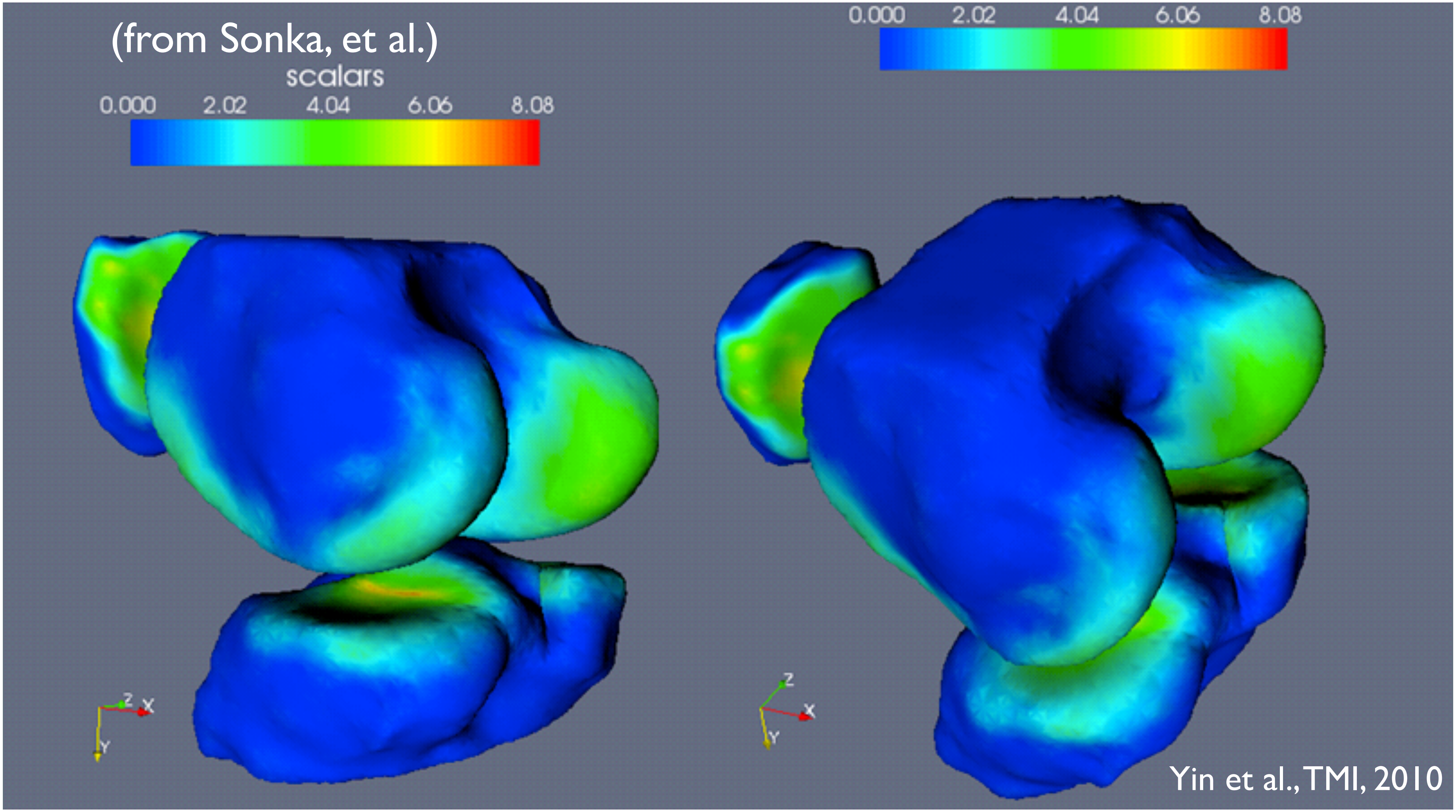


(from Reinhardt, et al.)

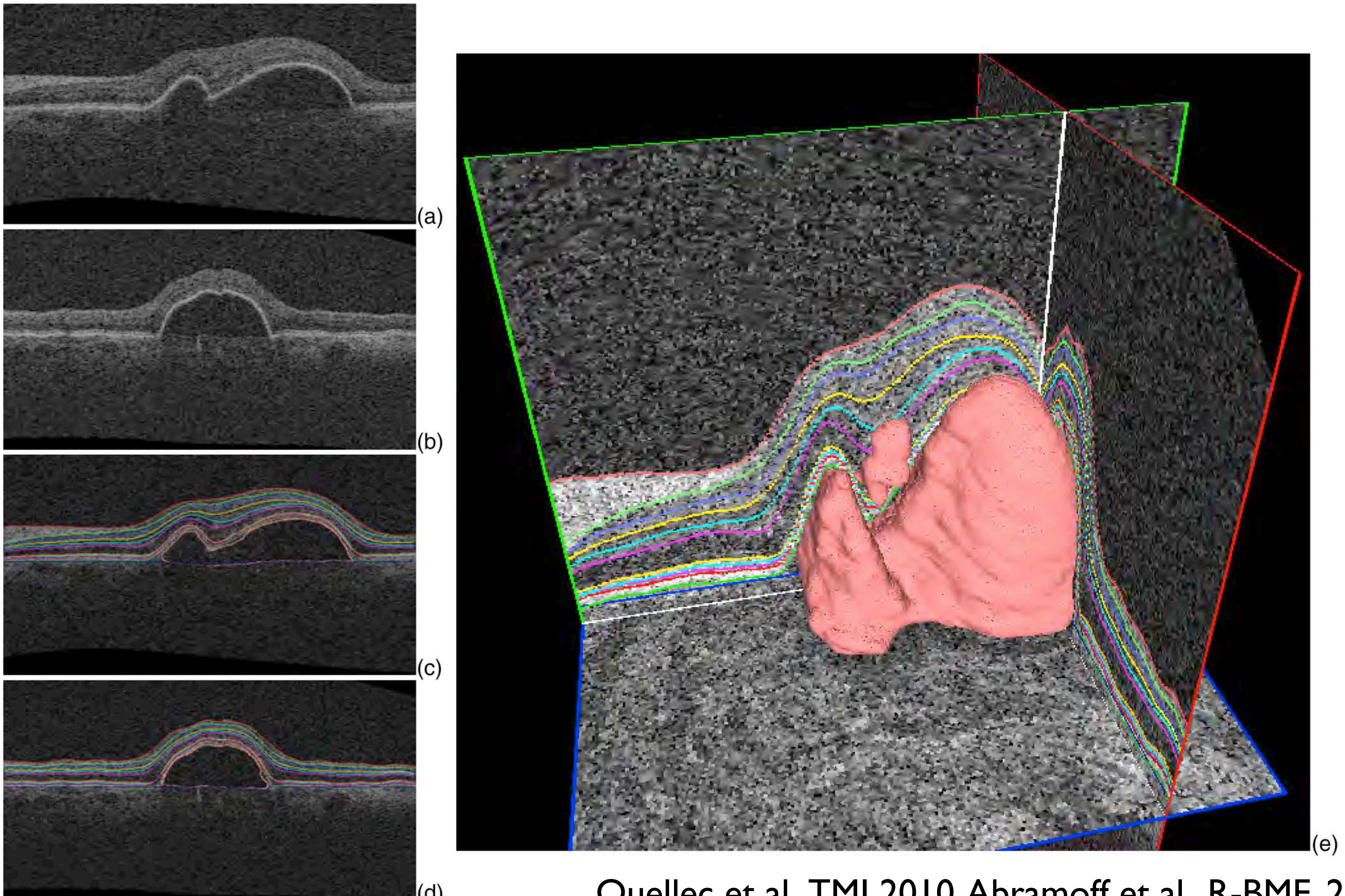
Example other LOGISMOS applications: determining cartilage thickness



Example other LOGISMOS applications: determining cartilage thickness



Challenge: layers with disruptions



Quellec et al., TMI 2010, Abramoff et al., R-BME, 2010

Summary

- The LOGISMOS approach enables the optimal and simultaneous segmentation of multiple surfaces and/or objects in polynomial time.
- Example applications include intraretinal layer segmentation, knee cartilage segmentation, vascular segmentation, airway segmentation, ...