

# Image Segmentation

Hugues Talbot (CVN, CentraleSupélec)

January 2021

# Goal of Image Segmentation

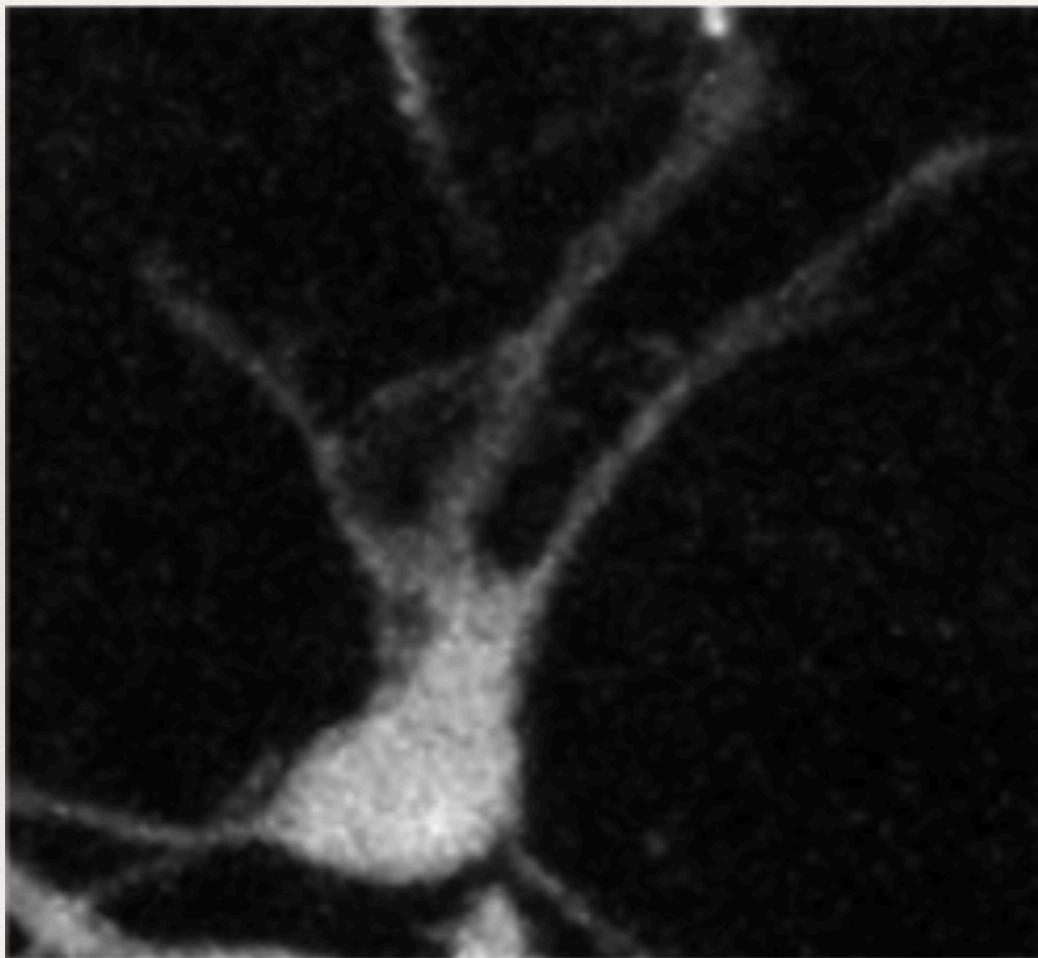
- ❖ The goal of image segmentation is to identify and delineate semantically coherent objects in images. E.g:
  - ❖ People, every day objects, etc
  - ❖ Cells in biology
  - ❖ Organs in medical imaging
  - ❖ Fields and roads in remote sensing...
- ❖ The task is similar to classification in machine learning, with the difference that spatial coherence is important.
- ❖ Defining spatial coherence is difficult.

# Learning-based segmentation

- ❖ Today, if there is \*enough\* training data, learning-based segmentation is the state of the art:
  - ❖ U-Net and its variation
  - ❖ Mask-RCNN and its variation
- ❖ But if the data is large, 3D or more and there is little or no annotation, it is still useful to know how to segment images without resorting to deep-learning
- ❖ DL methods also sometime fail inexplicably (sight change of input modality for example)

# Simple thresholding

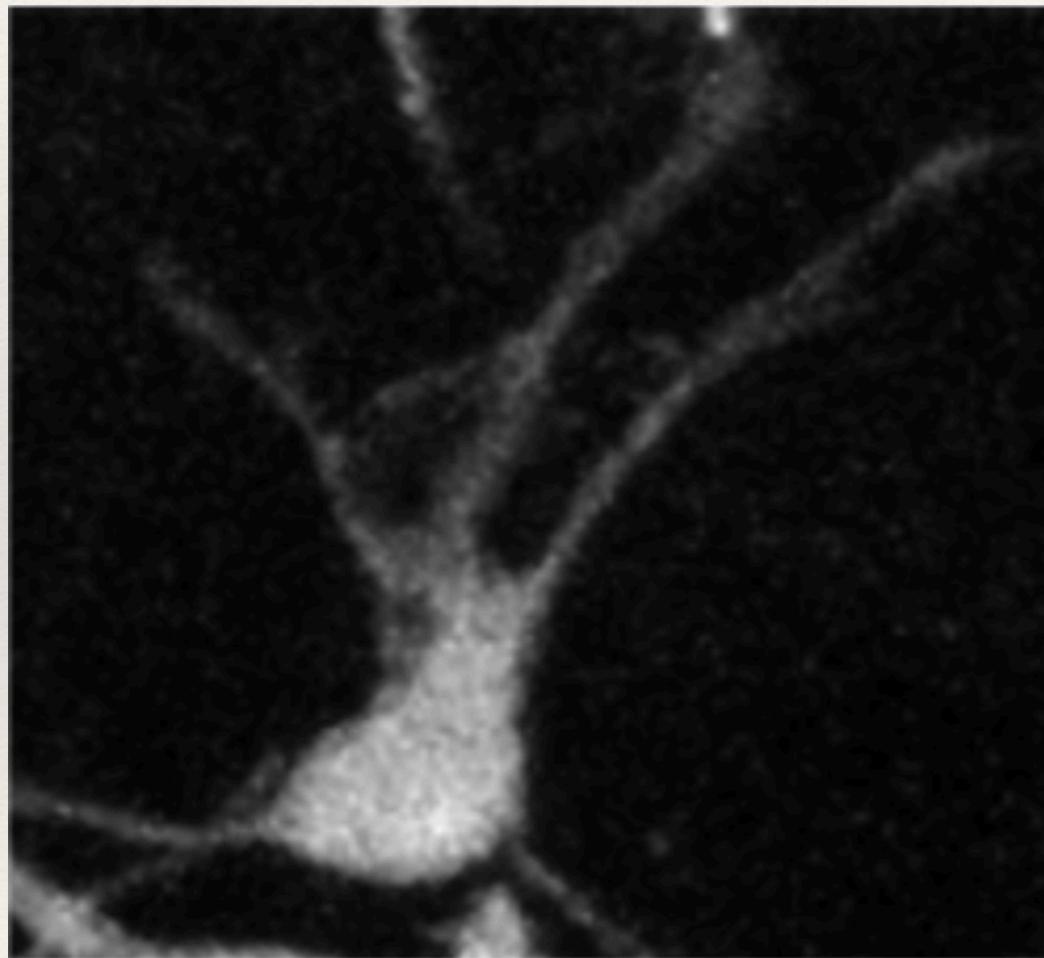
- ❖ An image of a neuron in confocal microscopy



Pure data term

# Shortest-perimeter prior

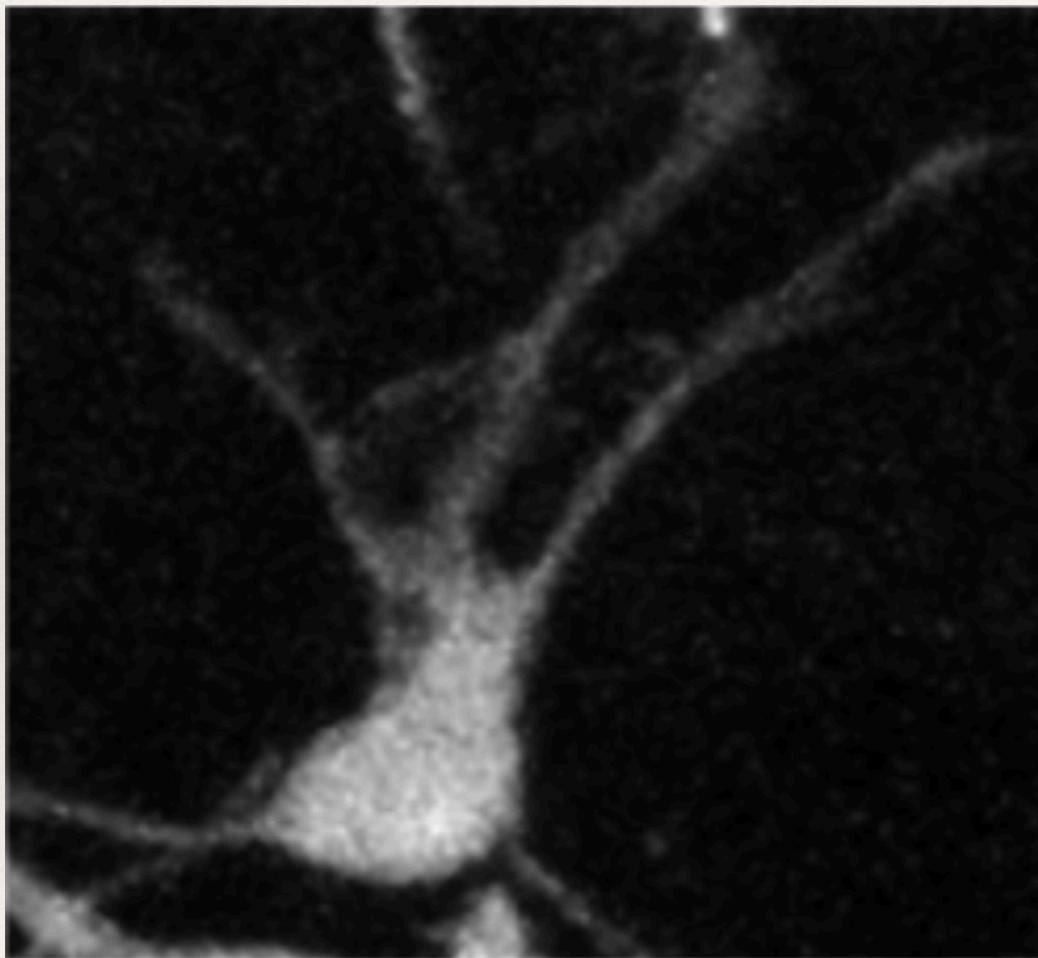
- ❖ Effect of the most commonly used coherence term



Perimeter term (TV)

# Use of a curvature term

- ❖ Use of a more advance prior (Euler's elastica)

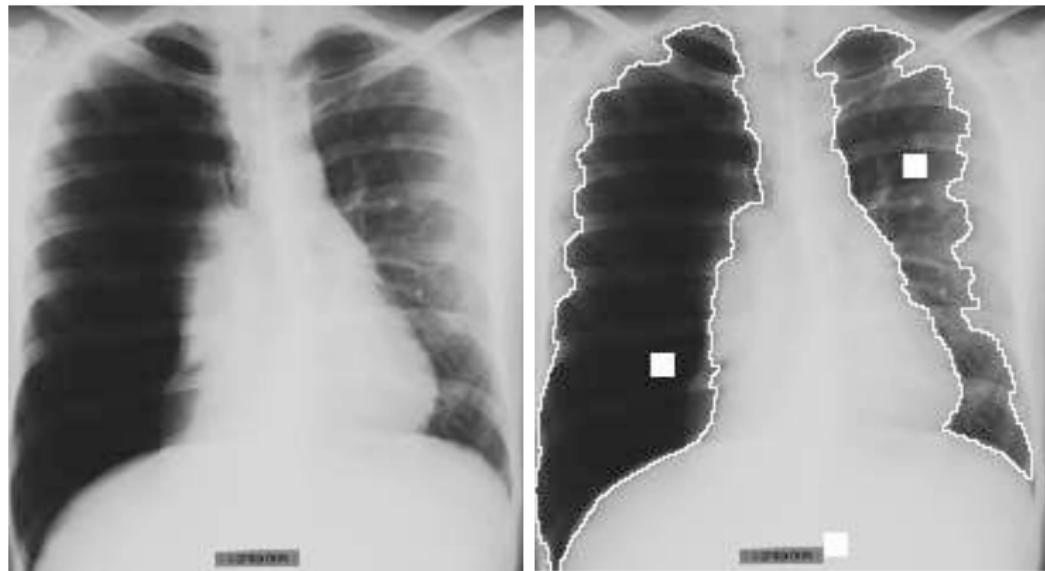


Perimeter + curvature term (elastica)

# What is segmentation

## Definition

- Segmentation is the process of finding and delineating regions of interests in images



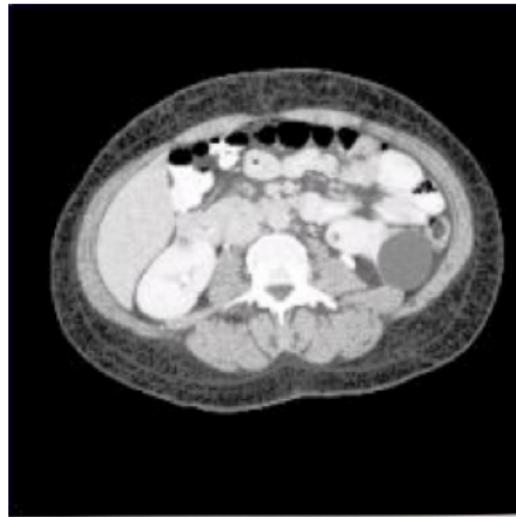
We assume the following:

### Assumptions

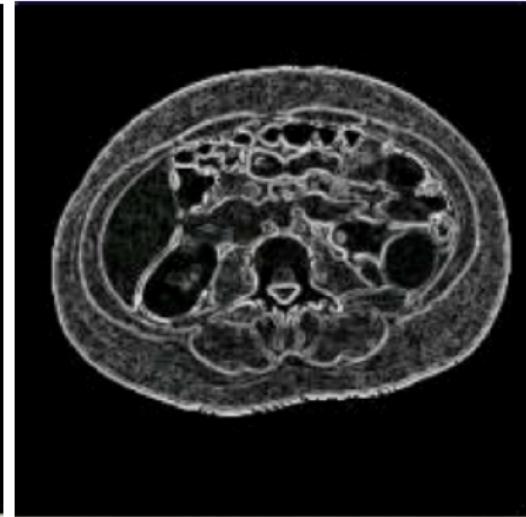
- Objects can be described by both their contour and their content ;
- There exists a *metric* allowing us describe the content of an object.  
It can be based on texture, colour, intensity, etc.

This metric is unspecified in general, because it is problem-dependent.  
We can provide many examples, but not general rules.

# Case of a simple model



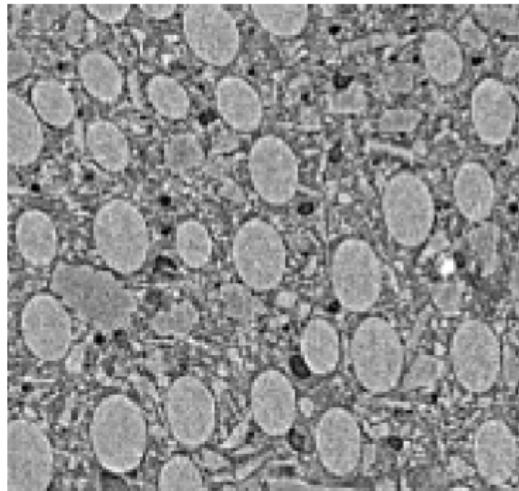
1.3: An MRI slice



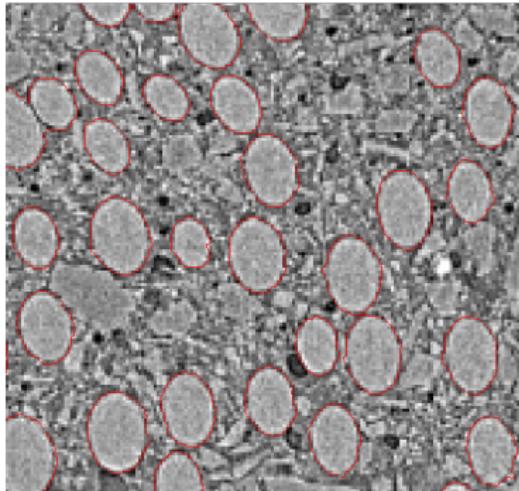
1.4: Contours

Here the similarity is given by the grey-levels. Contours are estimated by a gradient. Problem: the gradient may be weak and noisy.

# Case of a more complex model



1.5: Fibres slice



1.6: Segmentation

Here the similarity is given by texture, which itself can be derived from granulometries for instance [6].

# There are many segmentation methods

## A few segmentation methods

- Based on contour
  - Marr, Canny, ...
  - Active contour ...
- Based on regions
  - Watershed, Region Growing...
  - Thresholding, Split & Merge, MRFs,
- Based on optimization
  - Discrete: Graph Cuts, Trees ReWeighted, Belief Propagation...
  - Continuous: Level Sets, Total Variation, Continuous Maximum Flows...

More than 1000 methods published, many ad-hoc, few have survived the test of time.

# Specific challenges of Segmentation with Tomography images?

- Tomography data is large, microtomography even larger ( $2048^3$ );
- We may have to deal with time series;
- Materials imaging is typically dense (concrete, sand);
- Contrary to medical imaging, anatomical data is not available;
- Many modalities are noisy, e.g. nano-tomography, many artifacts;
- Imaging is a means to an end, materials scientists do not have the time to become imaging/segmentation specialists

# Properties of a good segmentation method?

A segmentation method needs to be

- Fast;
- Effective;
- Interactive;
- Easy to use;
- Works in 3D;
- Easy to understand;
- Easily available to users.

All components are essential.

# Interactive segmentation model

An interactive segmentation method consists of:

- ① A measure of similarity indicating the content of objects, from which the metric is derived;
- ② High-level knowledge for the placement of objects, derived from one or more of:
  - user interaction (manual seed placement);
  - pre-processing of the image (application-dependent);
  - machine learning (long, but can be largely automated).
- ③ An optimization method for the precise placement of contours.

So such methods are not really “interactive”, but they correspond to a philosophy [13].

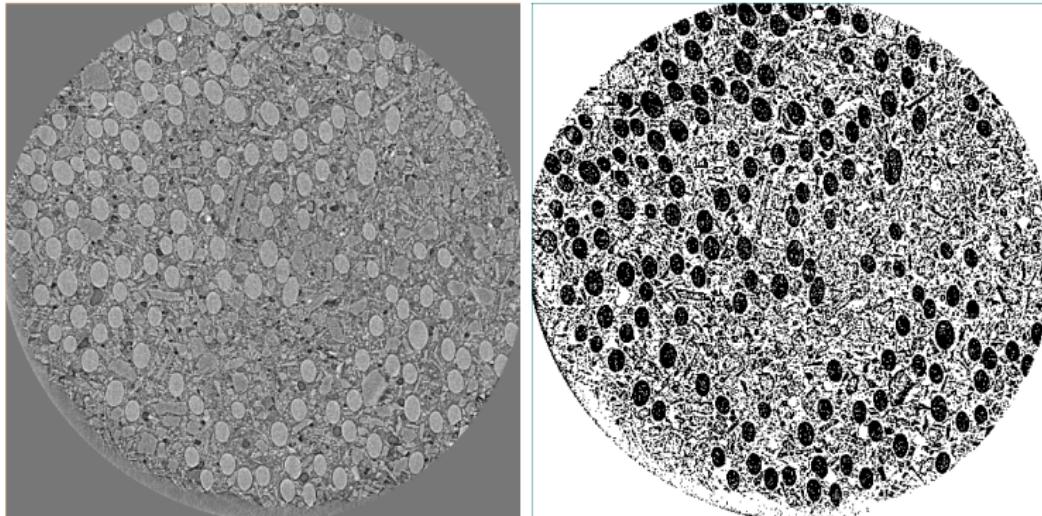
# Why bother about history?

- Because otherwise we are truly condemned to repeating it;
- Because both the vision and the image analysis communities have developed methods that are worth taking a look at;
- Many segmentation methods focus on the optimization step, this is only part of the story.

So we will *briefly* present a number of method and show that they converge.

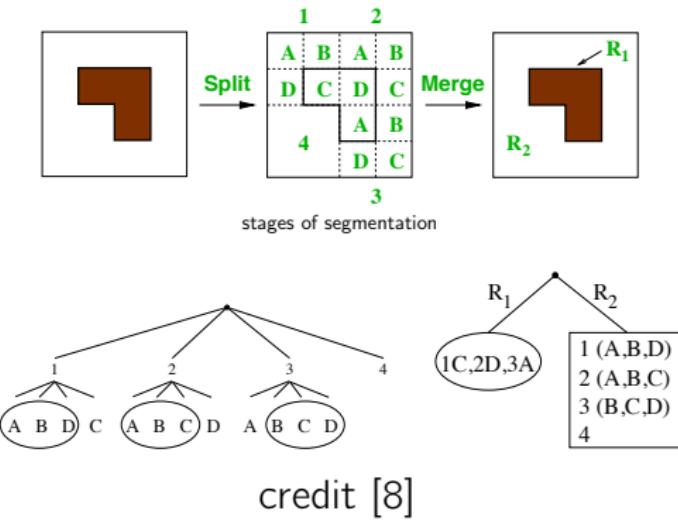
# from pixels to region statistics

Region-based segmentation consider the statistics of a given region, which may be of any shape. The challenge is to specify the region as well as the statistics. Thresholding considers the simplest region: individual pixels, and the simplest statistic: the grey-level intensity.



# from pixels to region statistics

Split-and-merge techniques consider multi-resolution version of images using quad-trees (2D) or oct-trees (3D) instead of pixels. They also consider more advanced statistics measures like mean and variance. Regions may be split or merged according to the result of statistical tests.



# from pixels to region statistics

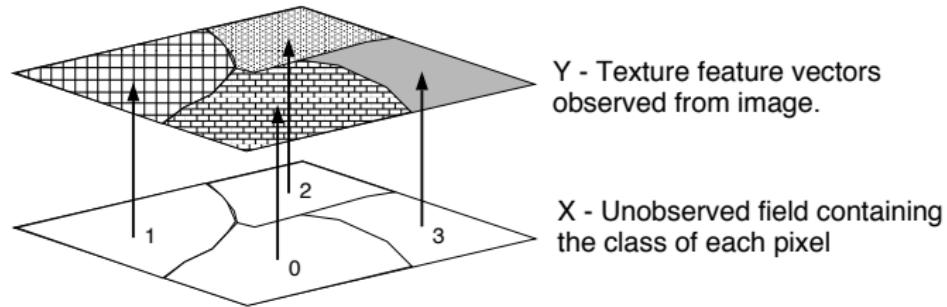
Because of the vertical and horizontal split and merge windows, split-and-merge methods tend to exhibit strong grid bias.



credit [10]

# from pixels to region statistics

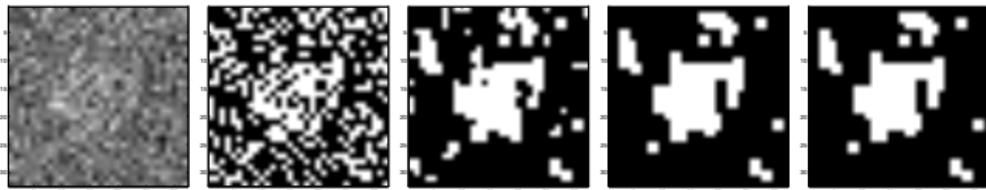
Markov Random Fields endeavour to build up statistically consistent regions from an initial state (e.g. all the pixels are different regions) and by considering *clique* statistics and random sampling.



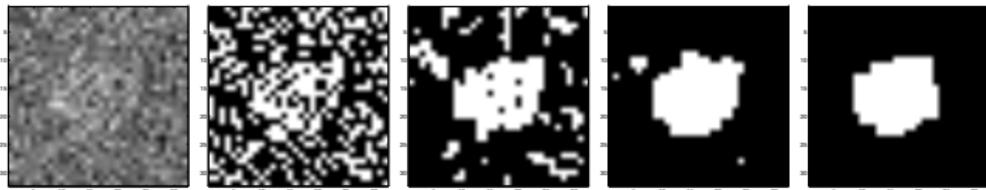
# from pixels to region statistics

Markov Random Fields endeavour to build up statistically consistent regions from an initial state (e.g. all the pixels are different regions) and by considering *clique* statistics and random sampling.

- Iterated Conditional Modes (ICM): ML ; ICM 1; ICM 5; ICM 10



- Simulated Annealing (SA): ML ; SA 1; SA 5; SA 10



credit [2]

# from pixels to region statistics

Example with the Maximum A-Posteriori estimation

$$\hat{X}_{\text{MAP}} = \operatorname{argmax}_x p_{x|y}(x|Y) \quad (1)$$

$$= \operatorname{argmax}_x \log \frac{p_{y,x}(Y, x)}{p_y(Y)} \quad (2)$$

$$= \operatorname{argmax}_x \{\log p(Y|x) + \log p(x)\} \quad (3)$$

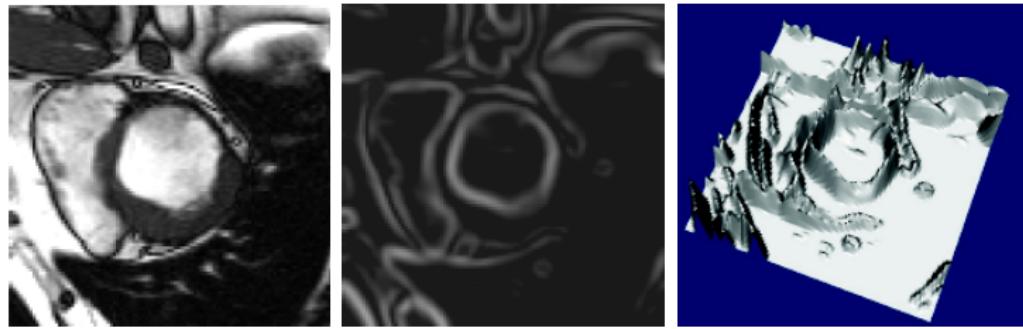
with  $p_x(x_i) = \frac{1}{Z} \exp\{-\beta \sum_{x_j \in \mathcal{N}(x_i)} \delta(x_i \neq x_j)\}$  we eventually find

$$\hat{X}_{\text{MAP}} = \operatorname{argmin}_x \left\{ \sum_{x_i \in E} I(y_i|x_i) + \beta \sum_{x_j \in \mathcal{N}(x_i)} \delta(x_i \neq x_j) \right\} \quad (4)$$

Optimisation is possible in the general case, with Gibbs Sampler and Simulated annealing, but slow. However special case: binary segmentation.

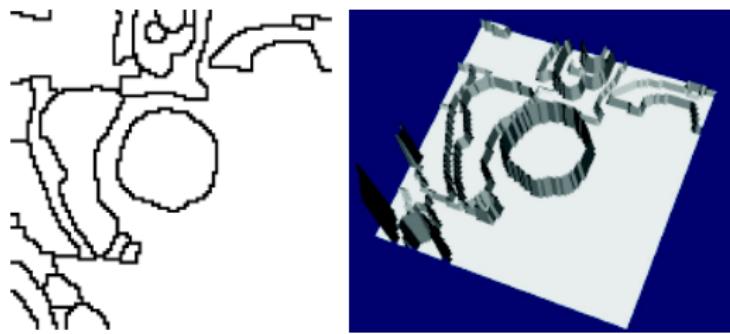
# from pixels to region statistics

The watershed and related region growing methods are often considered a region-based method, but this is quite arguable. It can be explained in terms of a drop of water falling onto the “terrain” of the image and ending on a local minimum of the image. The associated “catchment basins” provide a segmentation.



# from pixels to region statistics

The watershed and related region growing methods are often considered a region-based method, but this is quite arguable. It can be explained in terms of a drop of water falling onto the “terrain” of the image and ending on a local minimum of the image. The associated “catchment basins” provide a segmentation.



There are many variations of the WT: hierarchical, topological, etc.

# From edge detection to level-sets

One of the earliest approach to edge detection was to use Laplacian zero-crossing detection [12]. Theory shows that this *should* yield closed contours.



# From edge detection to level-sets

One of the earliest approach to edge detection was to use Laplacian zero-crossing detection [12]. Theory shows that this *should* yield closed contours.

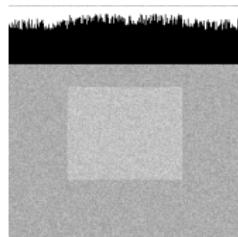


credit: [20]

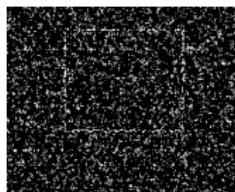
The contours are closed but not very interesting.

# From edge detection to level-sets

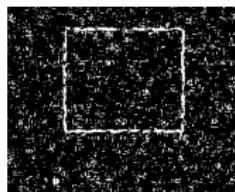
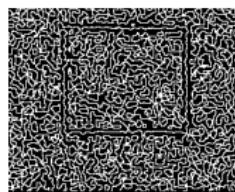
A more successful formulation was the computational approach of Canny [4]. It is optimal in 1D for finding a step function corrupted by Gaussian noise. It uses Gaussian blurring, hysteresis thresholding, and non-maxima suppression.



1.64: Orig



1.65: Sobel

1.66: Prewitt  
credit: [18]

1.67: DoG-ZC



1.68: Canny

# From edge detection to level-sets

This kind of result made people think the edge detection problem had been solved, but in fact, this is not so:



1.89: Orig



1.90: Canny



1.91: Snake

## From edge detection to level-sets

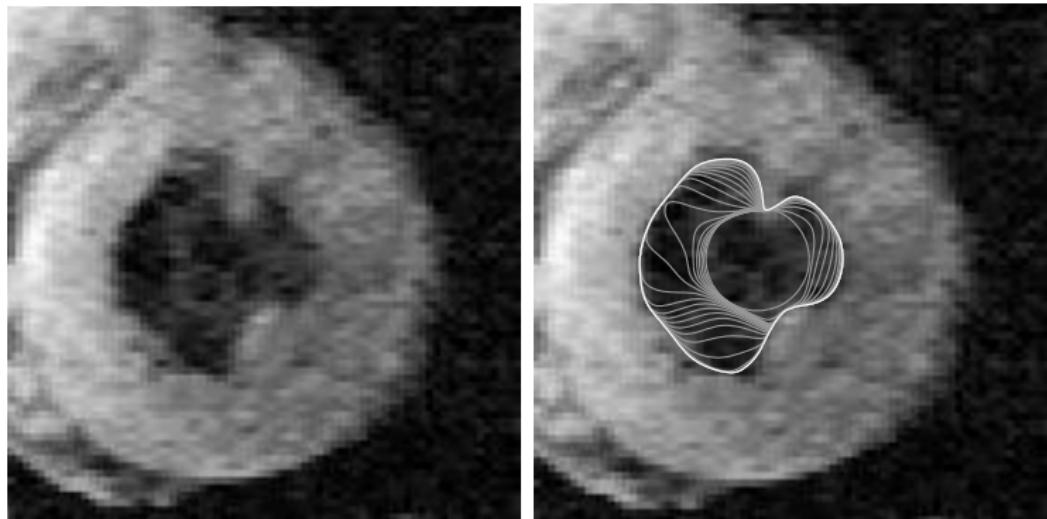
Still more successful were the *active contour* approaches of the late 80s [11]. They featured a Lagrangian contour/surface discretization and variational continuous optimisation. The models were complex and hard to manipulate, but at last we had closed contours/surfaces.

$$E_{\text{snake}} = \int_0^1 E_{\text{internal}}(\mathbf{v}(s)) + E_{\text{data}}(\mathbf{v}(s)) + E_{\text{constraints}}(\mathbf{v}(s)) ds \quad (1)$$

- Term 1 is the internal energy, including curvature and elasticity, optionnally including kinetic energy too.
- Term 2 is the data energy, which attracts the contours towards zone of high gradient for instance.
- Term 3 defines zones of attraction and repulsion

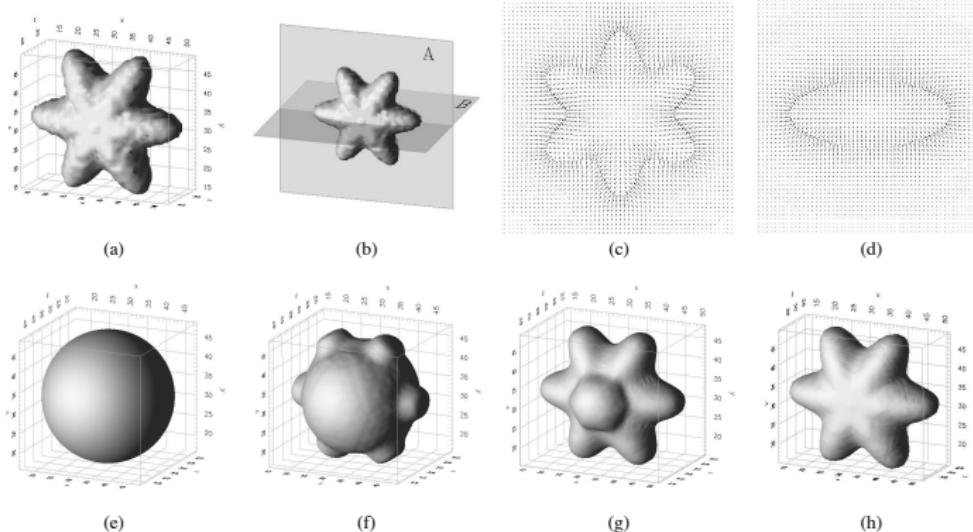
# From edge detection to level-sets

With active contours (a.k.a. Snakes), an initial placement of the contour is made, and the contour evolves according to a gradient descent of this energy, providing a local minimum energy. Many variants were proposed.



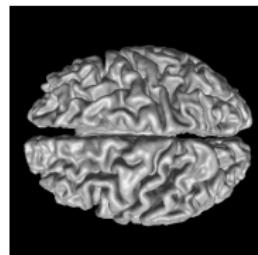
# From edge detection to level-sets

Extention to 3D is complex but feasible

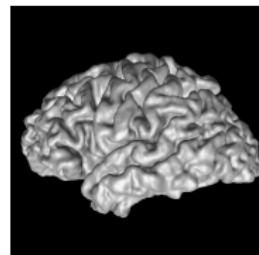


# From edge detection to level-sets

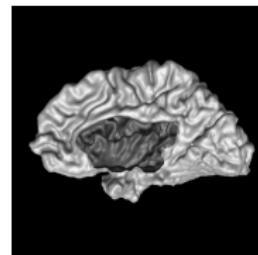
Extention to 3D is complex but feasible, even with complex topologies. It is used today in FreeSurfer.



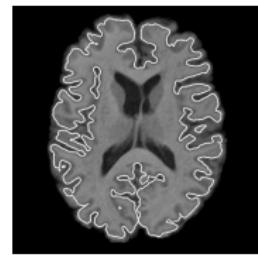
(a)



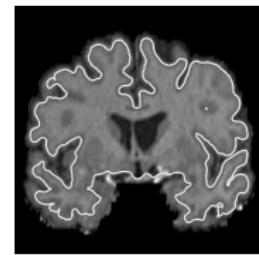
(b)



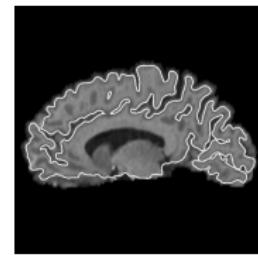
(c)



(d)



(e)

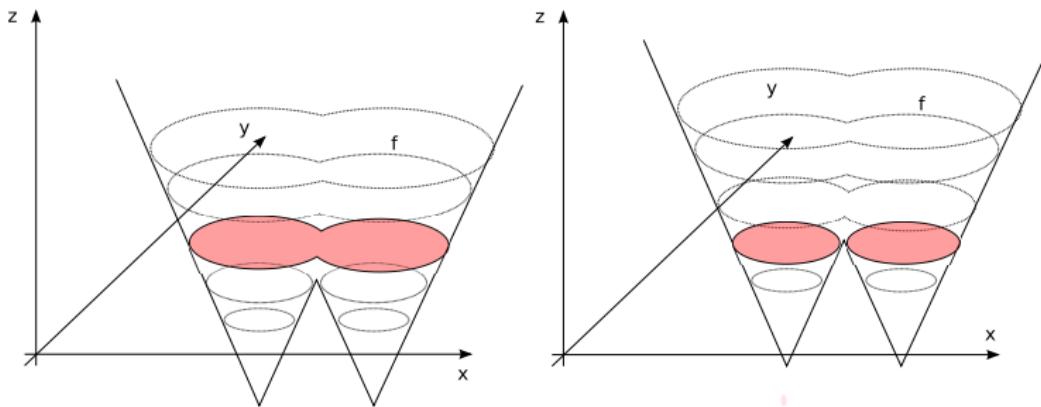


(f)

credit [19].

# From edge detection to level-sets

Level-sets methods were invented at the same time as active contours, but used in imaging only about 10 years later [16]. They represent a Eulerian, whole-space discretisation of the same kinds of equations as active contours via a dense function, and represent the contour/surface as a particular level of that function.



## From edge detection to level-sets

This formulation enables new behaviours for contours, like topology changes, but these methods are very slow. They add one dimension to the problem. An important contribution for segmentation was the Geodesic Active Contour, which minimizes the following equation:

$$\operatorname{argmin}_s \int_{\Omega} g(s) ds \quad (2)$$

with  $g(s) = \exp(-\alpha(G_\sigma * \nabla I)^2)$  for instance. This equation is important, because it is like a focal point.

LS methods are able to optimize locally many complex models, but the simpler ones are the ones that are used most often.

# Optimization and imaging

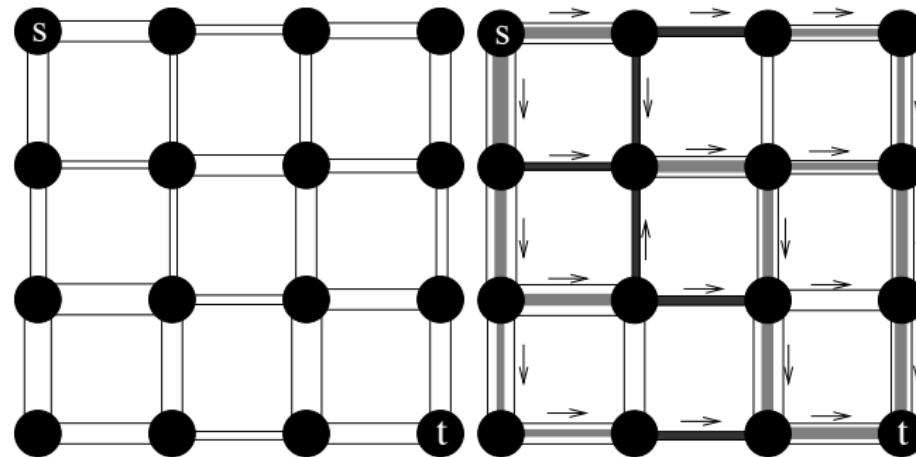
Until about the year 2000, models were more important and their optimization a secondary concern. This lead to very much suboptimal solution.

This is true for both discrete and continuous methods, for region or contour-based method.

The focus changed gradually when researchers found it was possible to solve the GAC equation exactly, first in 2D, then in arbitrary dimension, both using discrete and continuous methods.

# Graph Cuts principle

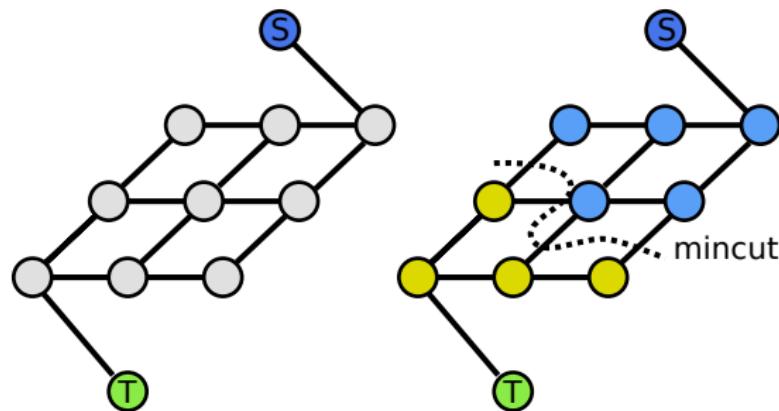
Graph cuts methods are based on the maxflow-mincut result of Ford and Fulkerson, 1956 [9]



This result can be viewed in several ways: it is a binary segmentation, it is also optimizing a sum of weights on the cut. It solves very efficiently a specific linear programming problem.

# Graph cuts: the partitionning viewpoint

The partitionning viewpoint leads to this particular graph construction:

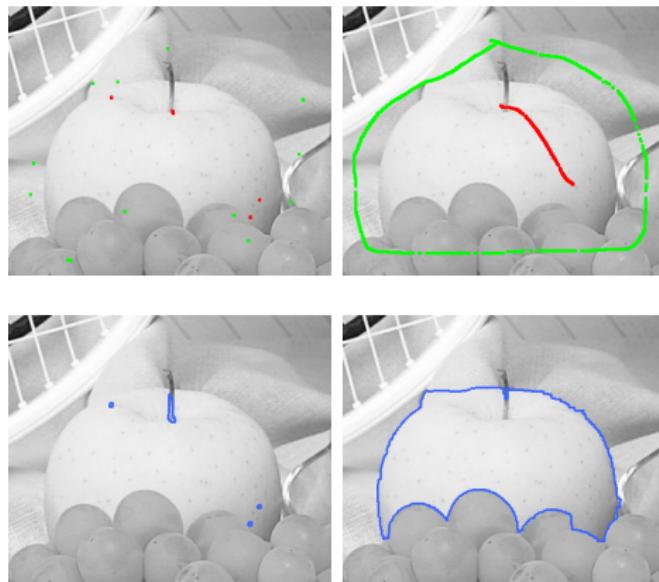


This solve the discrete GAC equation exactly, in arbitrary dimension:

$$\operatorname{argmin}_C \sum_{e_{i,j} \in C} w_{e_{i,j}} \quad (3)$$

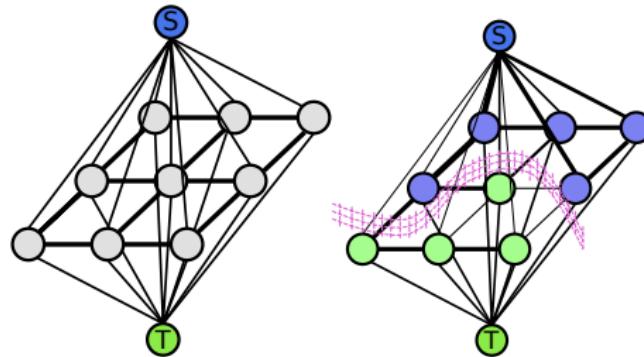
# Graph cuts: the partitionning viewpoint

A typical result looks like this, note size and grid bias:



# Graph cuts: the $\ell_1$ optimization viewpoint

To ameliorate the grid bias, a different graph construction can be proposed (Boykov-Jolly 2002) [3]

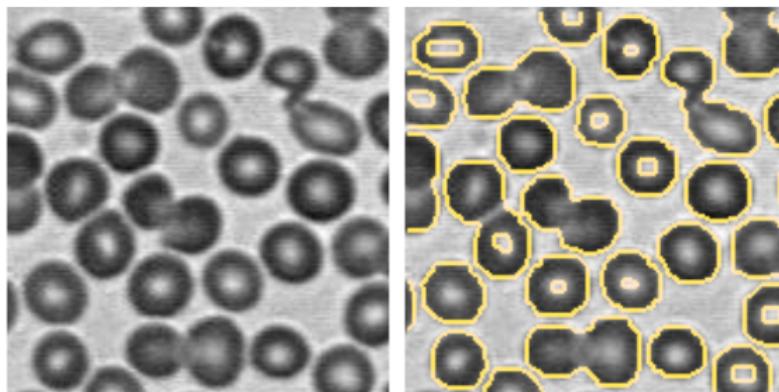


Graph cuts optimize the MRF energy exactly in the binary case.

$$\operatorname{argmin} \hat{E}(G) = \sum_{v_i \in V} w_i(V_i) + \lambda \sum_{e_{ij} \in \bar{E}} w_{ij} \delta_{V_i \neq V_j} \quad (4)$$

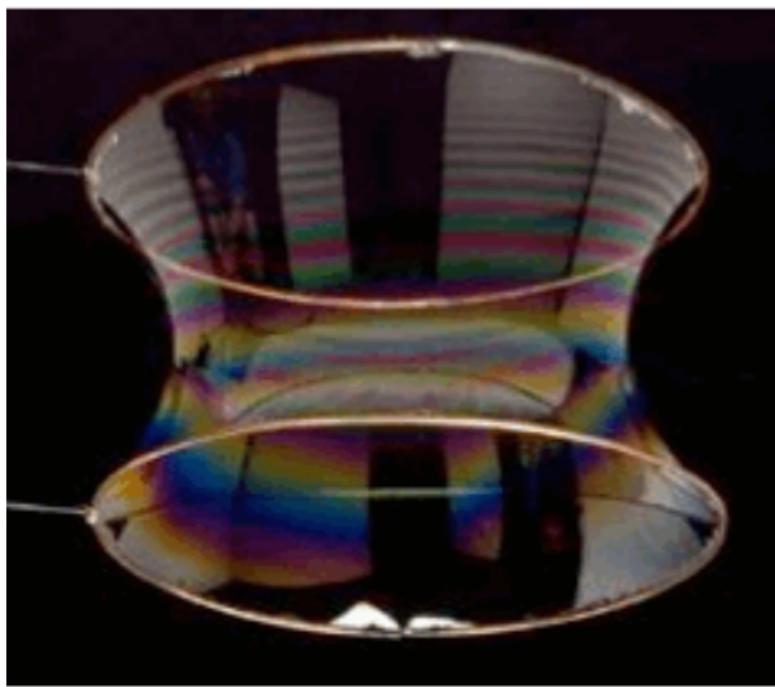
# Graph cuts: the $\ell_1$ optimization viewpoint

A typical segmentation may look like this



Is this the Holy grail ? no, due artefacts: grid bias, size bias, slow in more than 2D, hard to parallelize.

Solving the GAC is like computing a minimal surface



# Combinatorial Continuous Maximum Flow (CCMF)

- Incidence matrix of a graph noted  $A$

## Graph Cuts formulation

MaxFlow / GraphCuts

$$\begin{aligned} \max_{F} \quad & F_{st} \\ \text{s.t.} \quad & A^T F = 0, \\ & |F| \leq g \end{aligned}$$

$g$  defined on edges

## Convex optimization formulation

$$\begin{aligned} \max_{\vec{F}} \quad & F_{st} \\ \text{s.t.} \quad & \nabla \cdot \vec{F} = 0, \\ & \|\vec{F}\| \leq g. \end{aligned}$$

$g$  defined on vertices

- The Continuous Max Flow problem is convex
- Resolution by primal-dual methods.

# Resolution by a system of PDEs

The following system, discretized by finite differences, leads to a primal-dual resolution:

$$\frac{\partial \vec{F}}{\partial \tau} = -\nabla P \quad (5)$$

$$\frac{\partial P}{\partial \tau} = -\nabla \cdot \vec{F} \quad (6)$$

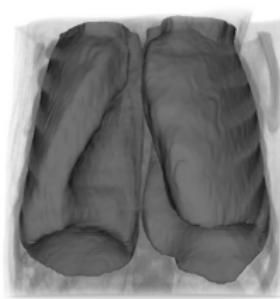
$$\|\vec{F}\| \leq g \quad (7)$$

At convergence, this system solves the GAC equation exactly [1].

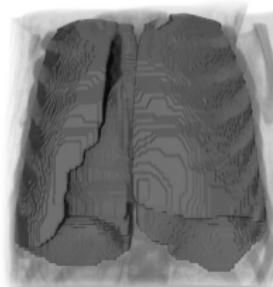
# Examples of solutions



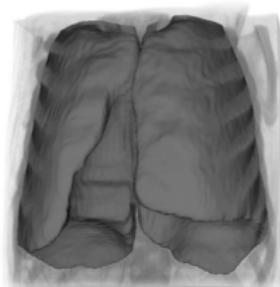
1.230: Orig



1.231: GAC-LS

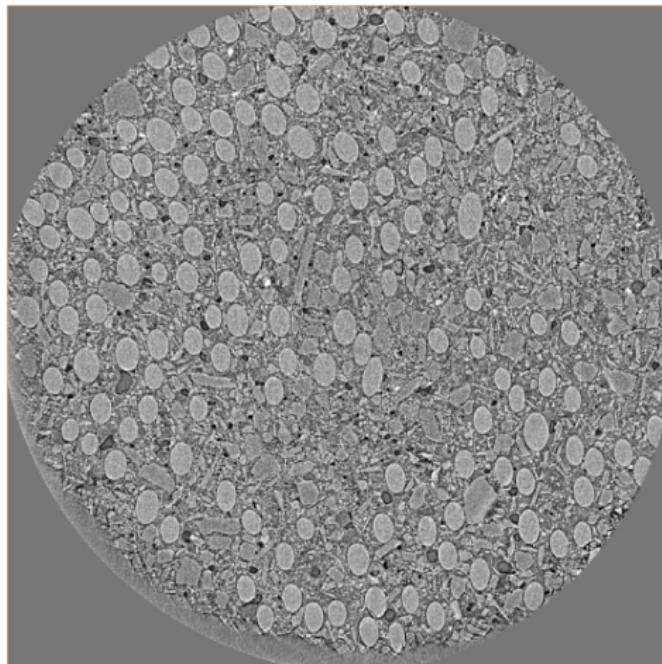


1.232: Graph cuts

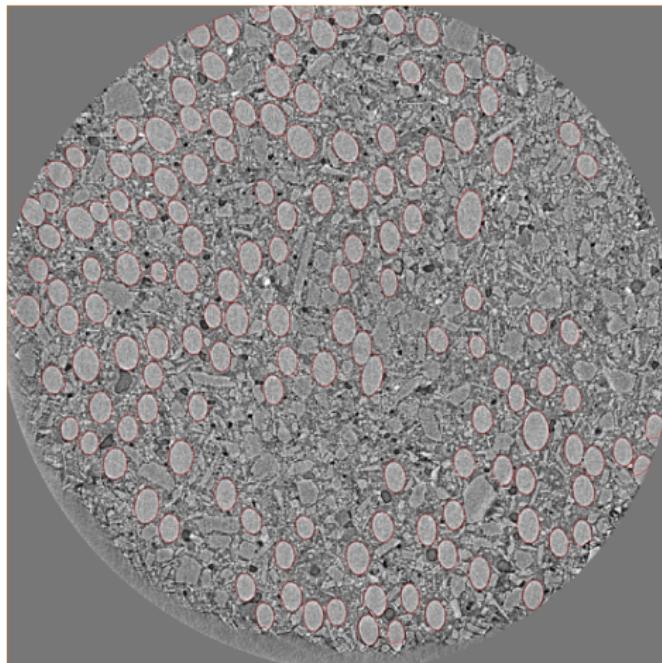


1.233: CMF

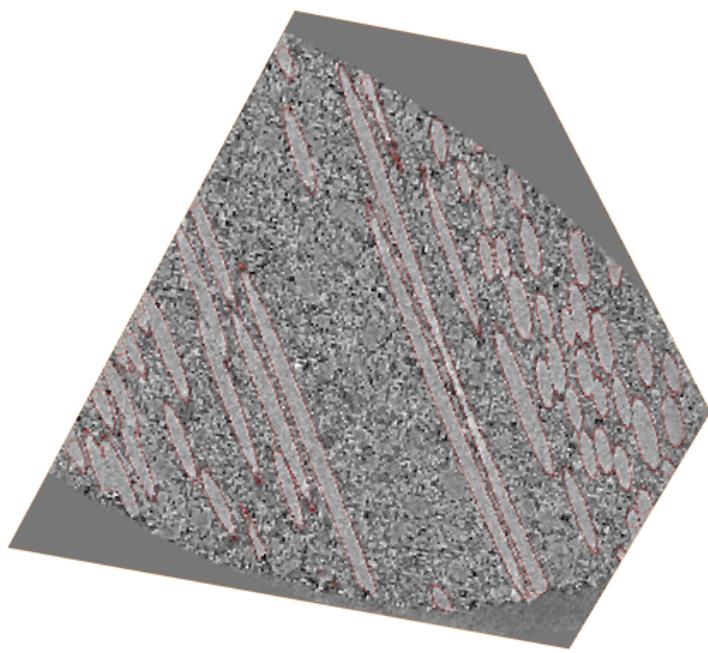
# Finally some 3D materials science images!



# Finally some 3D materials science images!

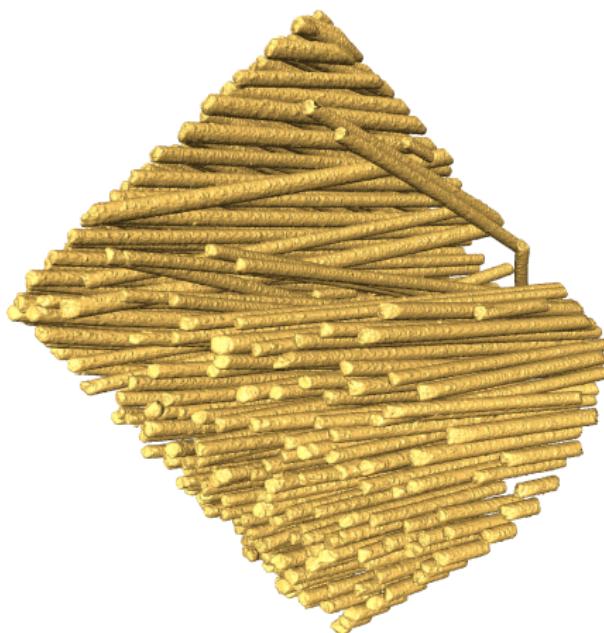


# Finally some 3D materials science images!

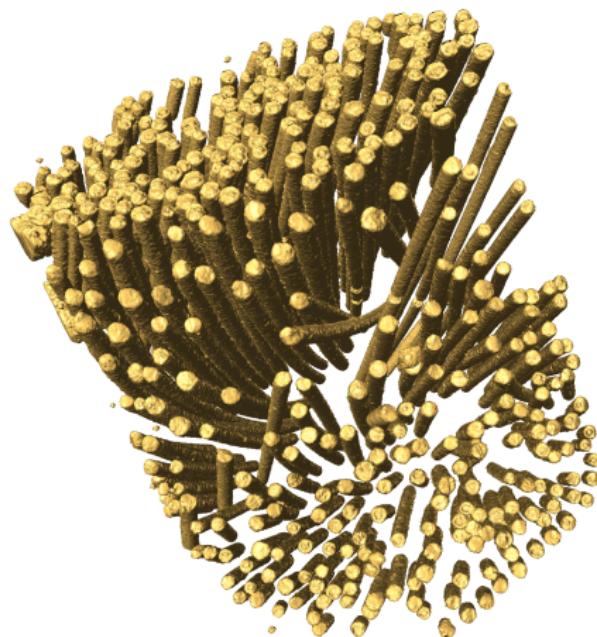


↑

# Finally some 3D materials science images!



# Finally some 3D materials science images!



# Finally some 3D materials science images!

## Highlights

- First effective solution of the GAC problem
- Convex optimization
- Isotropic
- Globally optimal
- Fast : for  $(512^3)$  voxels: less than a minute on GPU.
- Freely available: <http://www.pinkhq.com>

# Continuous MRF and more

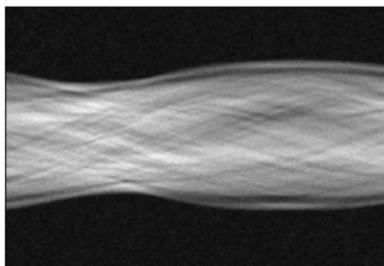
To solve the equivalent problem as MRF but in the continuous domain, we need to consider the Lagrangian dual of the continuous maximum flow problem.

This was formulated by Strang in 1983 [17] and found to be the Total Variation minimization problem. This was considered by many famous authors: Mumford and Shah 1989 [14], Rudin, Osher and Fatemi 1992 [15]. A first effective solution was found by Chambolle in 2004 [5]. Thanks to a similar approach, can now solve (Couprie et al 2011) [7]:

$$\min_x \underbrace{\max_{|A^T|F^2 \leq g^2} F^\top(Ax)}_{\text{regularization}} + \underbrace{\frac{1}{2\lambda} \|Hx - f\|_2^2}_{\text{data fidelity}} \quad (8)$$

# Joint tomographic reconstruction and segmentation

In practical terms,  $H$  can be the linear tomographic projection operator. Equation (8) can solve the inverse problem jointly with segmentation.



Noisy sinogram (25 dB)



Reconstructed image



Original image (Detail)



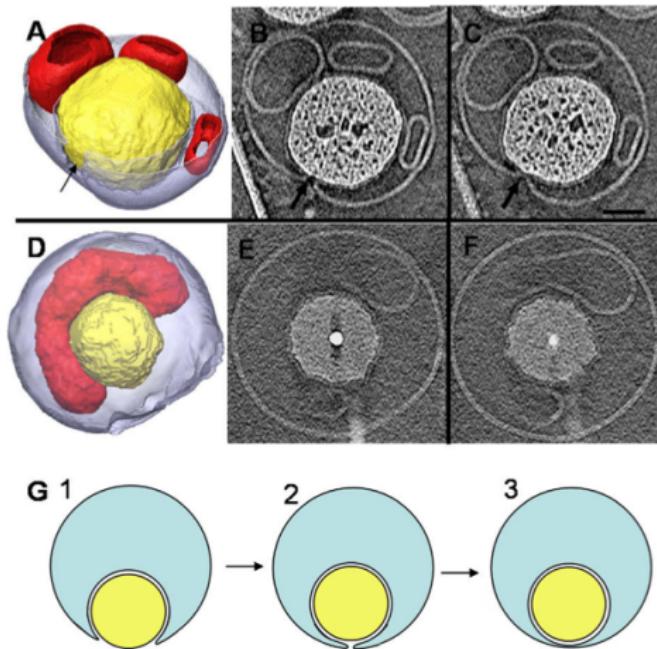
Proposed non-convex



Smooth convex

# Electron tomography: biological sample

This works also in the case where we have incomplete data, e.g. in electron tomography of flat samples (missing wedge) [?].



# Electron tomography: Cerium oxyde sample

We are working on incorporating shape constraints into this framework

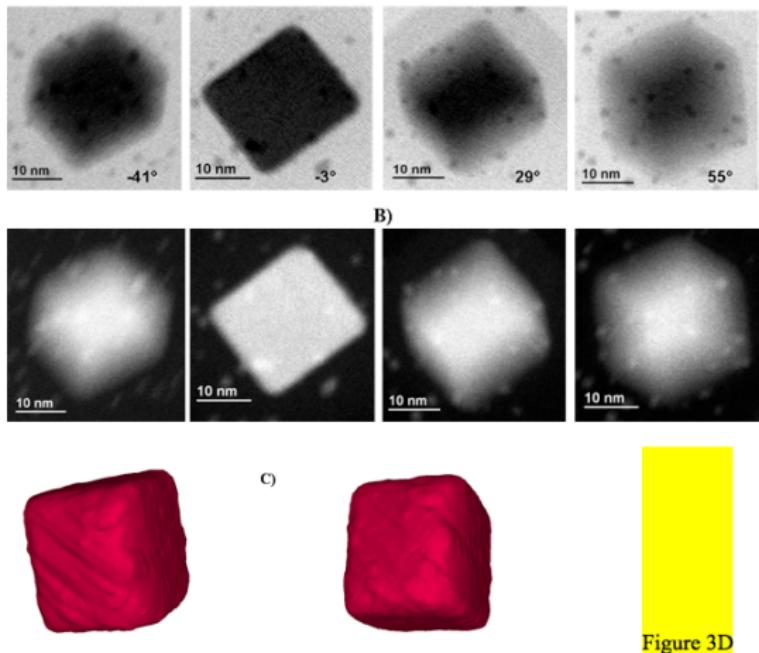
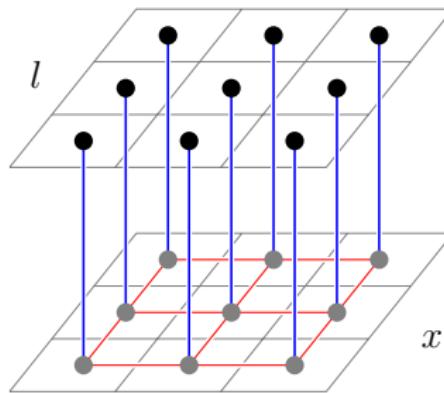


Figure 3D

# Power Watersheds: An energy minimization framework

$$\min_x \underbrace{\sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q}_{\text{Smoothness term}} + \underbrace{\sum_{v_i \in V} w_i^p |x_i - l_i|^q}_{\text{Data term}}$$



# Power Watersheds: An energy minimization framework

$$\min_x \underbrace{\sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q}_{\text{Smoothness term}} + \underbrace{\sum_{v_i \in V} w_i^p |x_i - l_i|^q}_{\text{Data term}}$$

$p \backslash q$	0	finite	$\infty$
1	Reduction to seeds	Graph cuts	Max Spanning Forest (watershed) [Allène et al. 07]
2	$\ell_2$ -norm Voronoi	Random walker	Power watershed [Couprie et al. 09]
$\infty$	$\ell_1$ -norm Voronoi	$\ell_1$ -norm Voronoi	Shortest Path [Sinop et al. 07]

# Power Watersheds: An energy minimization framework

$$\min_x \underbrace{\sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q}_{\text{Smoothness term}} + \underbrace{\sum_{v_i \in V} w_i^p |x_i - l_i|^q}_{\text{Data term}}$$

$p \backslash q$	0	finite	$\infty$
1	Reduction to seeds	Graph cuts	Max Spanning Forest (watershed) [Allène et al. 07]
2	$\ell_2$ -norm Voronoi	Random walker	Power watershed [Couprie et al. 09]
$\infty$	$\ell_1$ -norm Voronoi	$\ell_1$ -norm Voronoi	Shortest Path [Sinop et al. 07]

# Power Watersheds: An energy minimization framework

$$\min_x \underbrace{\sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q}_{\text{Smoothness term}} + \underbrace{\sum_{v_i \in V} w_i^p |x_i - l_i|^q}_{\text{Data term}}$$

$p \backslash q$	0	finite	$\infty$
1	Reduction to seeds	Graph cuts	Max Spanning Forest (watershed) [Allène et al. 07]
2	$\ell_2$ -norm Voronoi	Random walker	Power watershed [Couprie et al. 09]
$\infty$	$\ell_1$ -norm Voronoi	$\ell_1$ -norm Voronoi	Shortest Path [Sinop et al. 07]

# Power Watersheds: An energy minimization framework

$$\min_x \underbrace{\sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q}_{\text{Smoothness term}} + \underbrace{\sum_{v_i \in V} w_i^p |x_i - l_i|^q}_{\text{Data term}}$$

$p \backslash q$	0	finite	$\infty$
1	Reduction to seeds	Graph cuts	Max Spanning Forest (watershed) [Allène et al. 07]
2	$\ell_2$ -norm Voronoi	Random walker	Power watershed [Couprie et al. 09]
$\infty$	$\ell_1$ -norm Voronoi	$\ell_1$ -norm Voronoi	Shortest Path [Sinop et al. 07]

# Power Watersheds: An energy minimization framework

$$\min_x \underbrace{\sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q}_{\text{Smoothness term}} + \underbrace{\sum_{v_i \in V} w_i^p |x_i - l_i|^q}_{\text{Data term}}$$

$p \backslash q$	0	finite	$\infty$
1	Reduction to seeds	Graph cuts	Max Spanning Forest (watershed) [Allène et al. 07]
2	$\ell_2$ -norm Voronoi	Random walker	Power watershed [Couprie et al. 09]
$\infty$	$\ell_1$ -norm Voronoi	$\ell_1$ -norm Voronoi	Shortest Path [Sinop et al. 07]

# Power Watersheds: An energy minimization framework

$$\min_x \underbrace{\sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q}_{\text{Smoothness term}} + \underbrace{\sum_{v_i \in V} w_i^p |x_i - l_i|^q}_{\text{Data term}}$$

$p \backslash q$	0	finite	$\infty$
1	Reduction to seeds	Graph cuts	Max Spanning Forest (watershed) [Allène et al. 07]
2	$\ell_2$ -norm Voronoi	Random walker	Power watershed [Couprie et al. 09]
$\infty$	$\ell_1$ -norm Voronoi	$\ell_1$ -norm Voronoi	Shortest Path [Sinop et al. 07]

[Couprie-Grady-Najman-Talbot, ICCV 2009, PAMI 2011]

# Power watershed for image segmentation

- Simplification for algorithms comparison: only seeds used in the data fidelity term

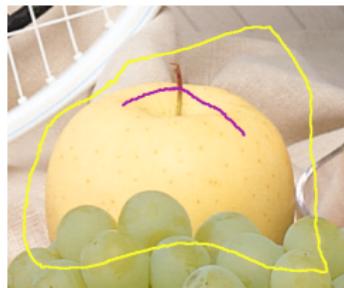
$$\min_x \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q$$

$$\text{s.t. } x(F) = 1, \quad x(B) = 0$$

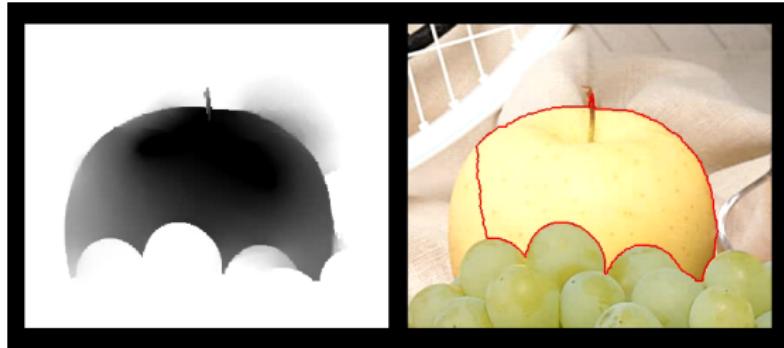
- Result: segmentation  $s$  defined  $\forall i$  by  $s_i = \begin{cases} 1 & \text{if } x_i \geq \frac{1}{2}, \\ 0 & \text{if } x_i < \frac{1}{2}. \end{cases}$

# Convergence of RW when $p \rightarrow \infty$ toward PW

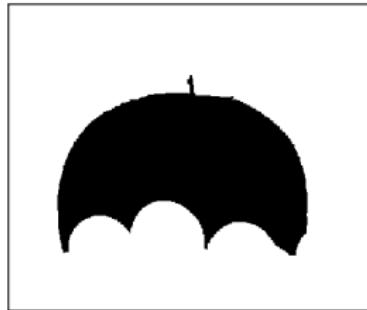
Input seeds



Random Walker  $p = 1$

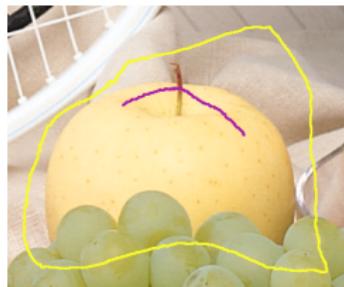


PowerWatershed  $q = 2$

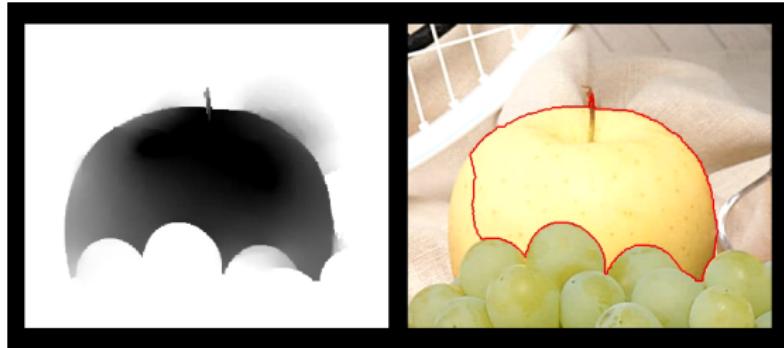


# Convergence of RW when $p \rightarrow \infty$ toward PW

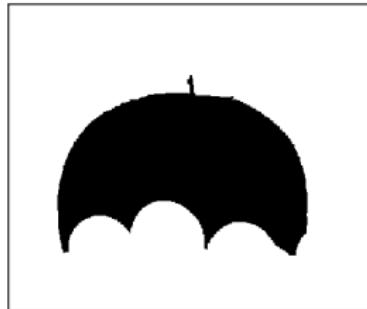
Input seeds



Random Walker  $p = 2$

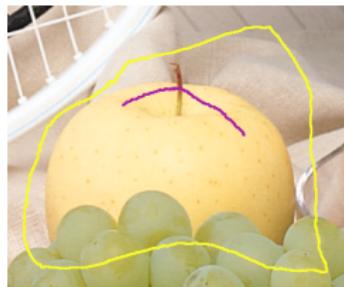


PowerWatershed  $q = 2$



# Convergence of RW when $p \rightarrow \infty$ toward PW

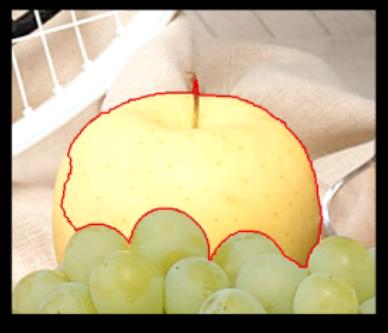
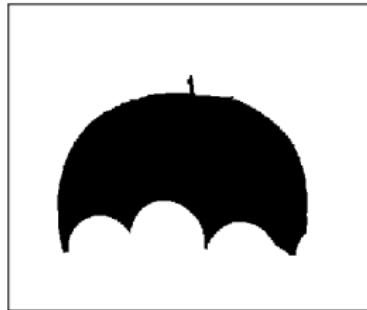
Input seeds



Random Walker  $p = 3$

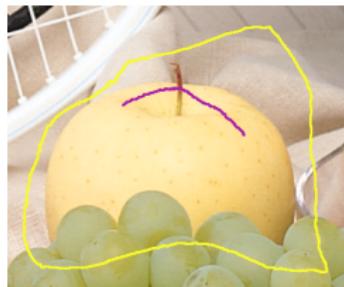


PowerWatershed  $q = 2$



# Convergence of RW when $p \rightarrow \infty$ toward PW

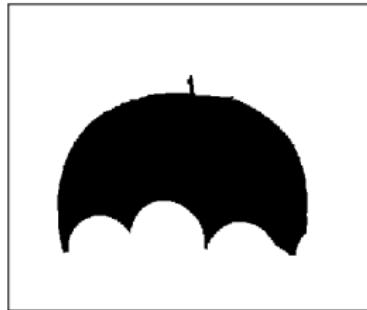
Input seeds



Random Walker  $p = 5$

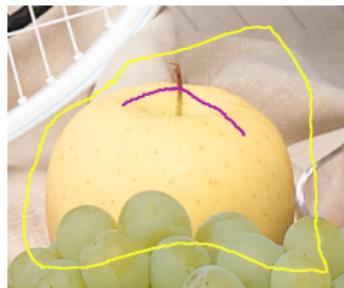


PowerWatershed  $q = 2$



# Convergence of RW when $p \rightarrow \infty$ toward PW

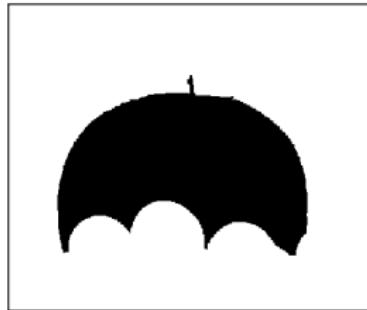
Input seeds



Random Walker  $p = 7$

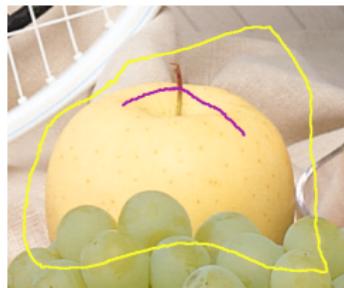


PowerWatershed  $q = 2$



# Convergence of RW when $p \rightarrow \infty$ toward PW

Input seeds



Random Walker  $p = 10$

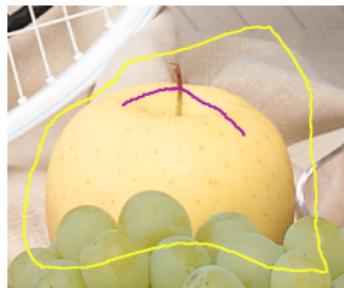


PowerWatershed  $q = 2$



# Convergence of RW when $p \rightarrow \infty$ toward PW

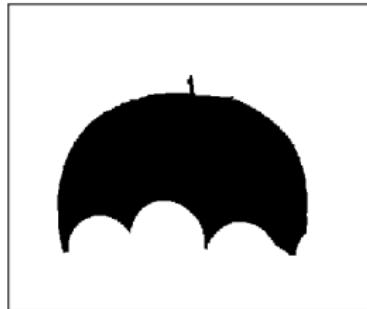
Input seeds



Random Walker  $p = 13$

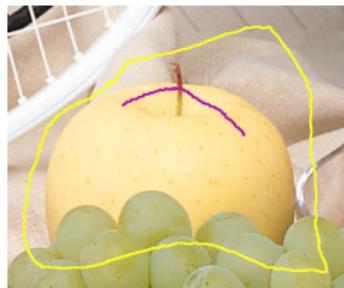


PowerWatershed  $q = 2$



# Convergence of RW when $p \rightarrow \infty$ toward PW

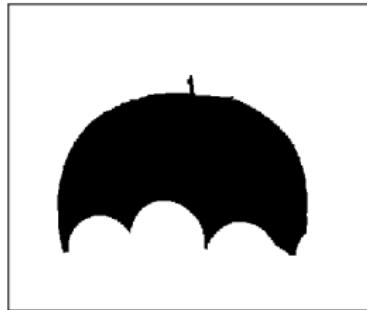
Input seeds



Random Walker  $p = 16$

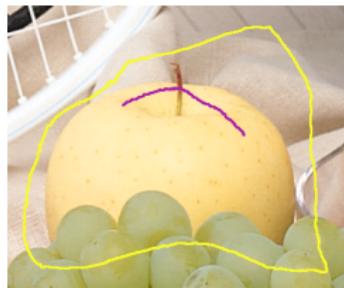


PowerWatershed  $q = 2$



# Convergence of RW when $p \rightarrow \infty$ toward PW

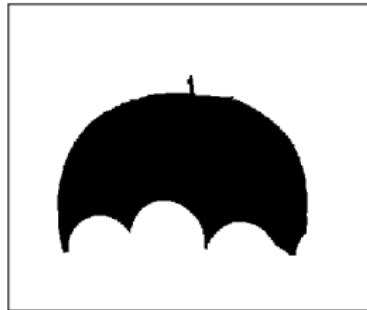
Input seeds



Random Walker  $p = 20$

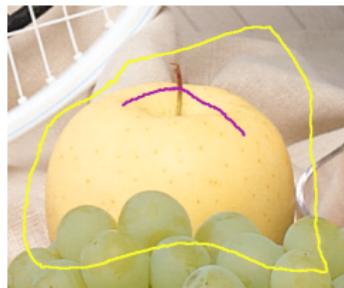


PowerWatershed  $q = 2$



# Convergence of RW when $p \rightarrow \infty$ toward PW

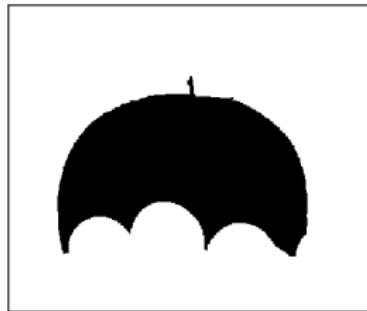
Input seeds



Random Walker  $p = 24$

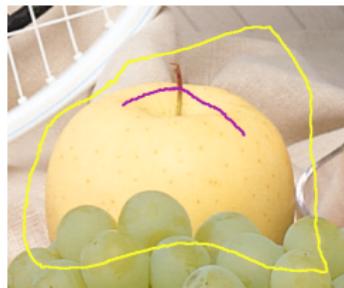


PowerWatershed  $q = 2$



# Convergence of RW when $p \rightarrow \infty$ toward PW

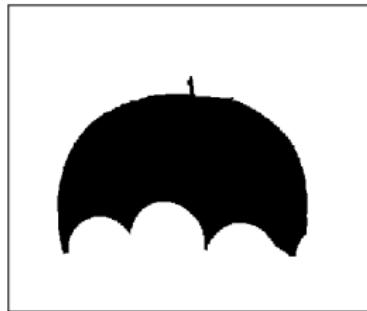
Input seeds



Random Walker  $p = 30$



PowerWatershed  $q = 2$



# Algorithm for the case $p \rightarrow \infty$ , variable $q$

- Compute  $x$  minimizing

$$\lim_{p \rightarrow \infty} \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q$$

subject to boundary conditions.

- We construct an MSF outside of plateaus, and optimize

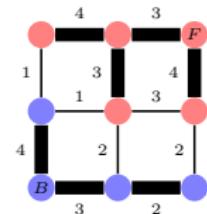
$$\min_x \sum_{e_{ij} \in \text{plateau}} |x_i - x_j|^q$$

on the plateaus.

- We call this algorithm “Power watershed”

# Link with watershed and unicity of solution

- Max Spanning Forest (MSF) : maximize the sum of edge weights of a forest spanning the graph.
- Watershed: points where a drop of water could flow toward different catchment basins [Cousty *et al.* 07].
- If seeds are the maxima of the weight function, every MSF cut is a watershed cut [Cousty *et al* 07].

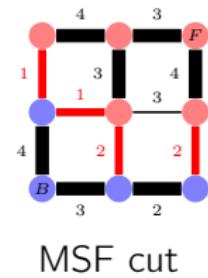


## Theorems

- The cut obtained by the power watershed algorithm is a MSF cut (and a watershed cut if seeds are the minima of the weights).
- When  $q > 1$ , the solution  $x$  obtained by the power watershed algorithm is unique.

# Link with watershed and unicity of solution

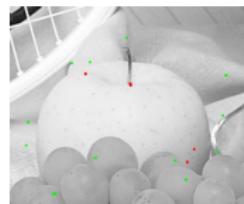
- Max Spanning Forest (MSF) : maximize the sum of edge weights of a forest spanning the graph.
- Watershed: points where a drop of water could flow toward different catchment basins [Cousty *et al.* 07].
- If seeds are the maxima of the weight function, every MSF cut is a watershed cut [Cousty *et al* 07].



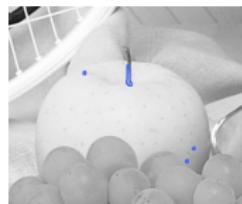
## Theorems

- The cut obtained by the power watershed algorithm is a MSF cut (and a watershed cut if seeds are the minima of the weights).
- When  $q > 1$ , the solution  $x$  obtained by the power watershed algorithm is unique.

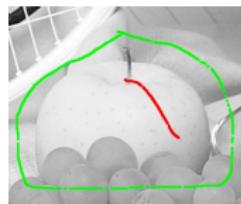
# Comparison of results



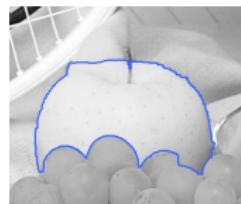
Input seeds



GraphCut



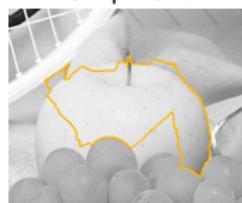
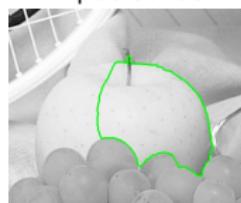
RandWalk



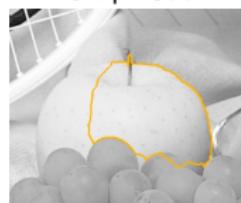
ShtPath



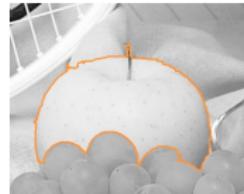
MaxSF

PW  $q = 2$ 

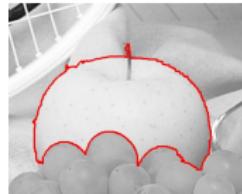
RandWalk



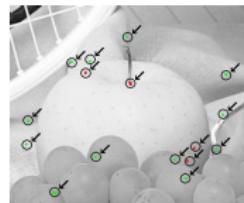
ShtPath



MaxSF

PW  $q = 2$

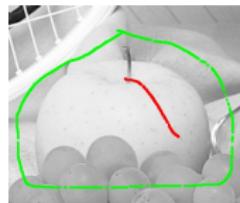
# Comparison of results



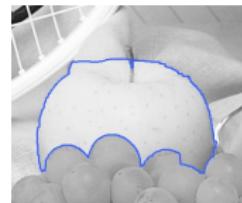
Input seeds



GraphCut



Input seeds



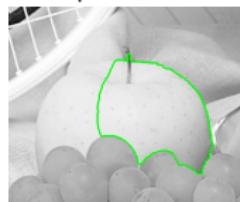
GraphCut



RandWalk



ShtPath



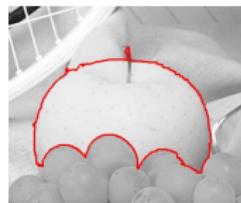
RandWalk



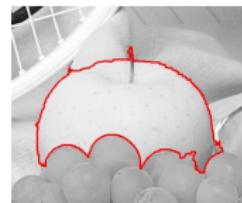
ShtPath



MaxSF

PW  $q = 2$ 

MaxSF

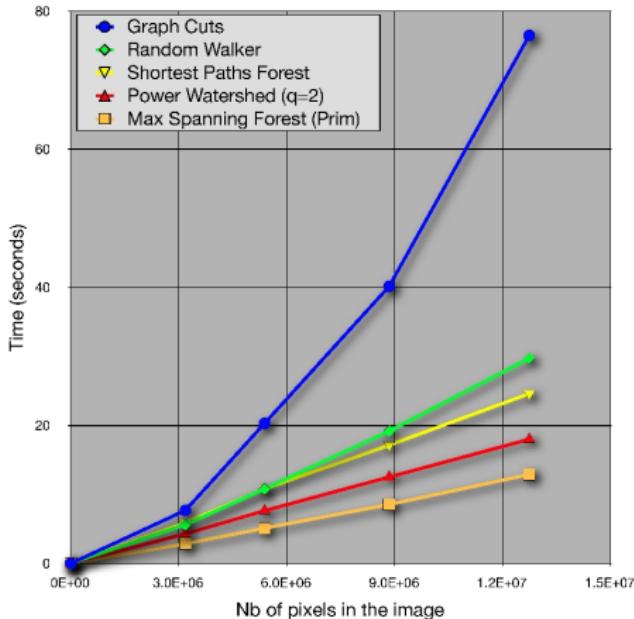
PW  $q = 2$

# Algorithms comparison

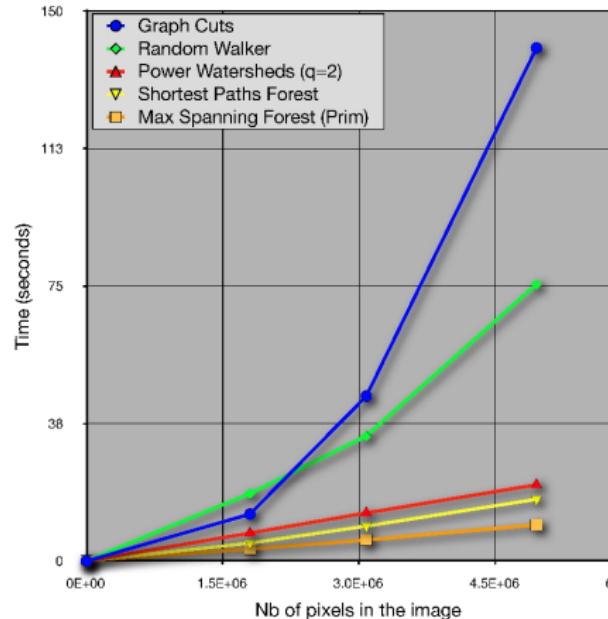
- Evaluation on GrabCut database
- 2 sets of seeds to study robustness to seeds centering
  - ① seeds well centered around boundaries:  
Best performer : Shrt path, worst performer : GraphCuts
  - ② seeds less centered around boundaries: From best to worst :  
GraphCuts, PWshed, Random Walker, MaxSF, Shrt path

# Computation time

**Computation times 2D**



**Computation times 3D**



# Conclusion

## Conclusion

- Many approaches were presented, from various horizons
- There are surprisingly many common elements
- Many issues remain: integrating higher order energy terms (e.g. curvature), shape information, etc.
- The best optimization strategy and the fastest algorithms don't solve problems by themselves, but we hope they contribute.

# Questions?



B. Appleton and H. Talbot.

Globally minimal surfaces by continuous maximal flows.

*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):106–118, 2006.



Charles A. Bouman.

Application of mrf's to segmentation.

EE641 Digital Image Processing II, December 2010.



Y. Boykov and M.-P. Jolly.

Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images.

In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, pages 105–112, Vancouver, BC , Canada, 2002.



J. Canny.

A computational approach to edge detection.

*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.



A. Chambolle.

An algorithm for total variation minimization and applications.

*J. Math. Imaging Vis.*, 20(1–2):89–97, 2004.



Nicolas Combaret and Hugues Talbot.

Robust 3d segmentation of composite materials fibres.

In Gerald Jean Francis Banon, Junior Barrera, Ulisses de Mendonça Braga-Neto, and Nina Sumiko Tomita Hirata, editors, *Proceedings of ISMM 2007*, volume 2, pages 25–26, São José dos Campos, October 10–13 2007. Universidade de São Paulo (USP), Instituto Nacional de Pesquisas Espaciais (INPE).



C. Couplie, H. Talbot, J.C. Pesquet, L. Najman, and L. Grady.

Dual constrained TV-based regularization.

In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 22–27 2011.



Dmitrij Csetverikov.

Basic algorithms for digital image analysis: a course.



L. R. Ford and D. R. Fulkerson.

Maximal flow through a network.

*Canadian Journal of Mathematics*, 8:399–404, 1956.



Lena Luisa Heß.

Split-and-merge.

Web page.



M. Kass, A. Witkin, and D. Terzopoulos.

Snakes: active contour models.

*International Journal of Computer Vision*, 1:321–331, 1988.



D. Marr and E. Hildreth.

Theory of edge detection.

*Proceedings of the Royal Society of London. Series B, Biological Sciences*, pages 187–217, 1980.



Fernand Meyer and Serge Beucher.

Morphological segmentation.

*Journal of Visual Communication and Image Representation*, 1(1):21–46, September 1990.



D. Mumford and J. Shah.

Optimal approximations by piecewise smooth functions and associated variational problems.

*Communications on pure and applied mathematics*, 42(5):577–685, 1989.



Leonid I. Rudin, Stanley Osher, and Emad Fatemi.

Nonlinear total variation based noise removal algorithms.

*Phys. D*, 60(1-4):259–268, 1992.



J.A. Sethian.

*Level set methods and fast marching methods*.

Cambridge University Press, 1999.

ISBN 0-521-64204-3.



Gilbert Strang.

Maximal flow through a domain.

*Mathematical Programming*, 26:123–143, 1983.



Ruye Wang.

Computer image processing and analysis (e161).

<http://fourier.eng.hmc.edu/e161/>.



Chenyang Xu and Jerry L. Prince.

Gradient vector flow deformable models.

In Isaac Bankman, editor, *Handbook of Medical Imaging*. Academic Press, 2000.



David Young.

Gaussian masks, scale space and edge detection.

[http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/YOUNG/vision3.html](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/YOUNG/vision3.html), January 1994.

Advanced topic: use of curvature

# Elastica

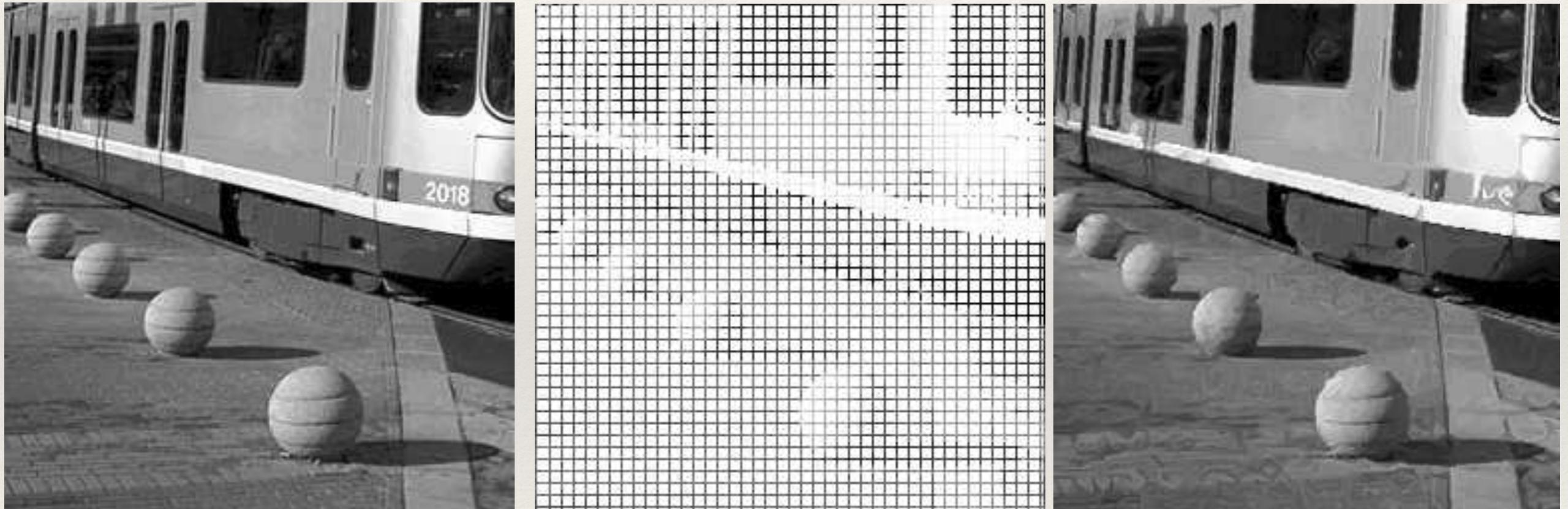
- ❖ Use curvature regularization in image processing tasks, i.e in inpainting, segmentation, stereo matching, etc.
- ❖ How? We aim at minimizing the functional

$$u^* = \arg \min_{u_s \in \Omega} \int_{\Omega} \|u - u_s\|^2 \, dx + \int_{\partial u_s} \alpha + \beta \kappa^2 \, ds$$

- ❖ Studied by Mumford [6], Chen et al [7], Masnou et al.[8], El Zehiri et al [9], Nieuwenhuis et al [10] and many others.

# Motivation

- ❖ Why curvature?

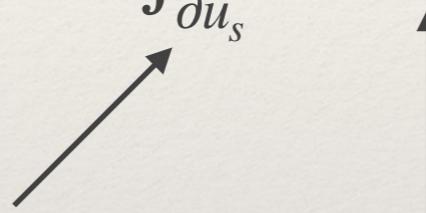


Inpainting example (S. Masnou)

# Challenge

- ❖ Why is this challenging ?

$$u^\star = \arg \min_{u_s \in \Omega} \int_{\Omega} \|u - u_s\|^2 \, dx + \int_{\partial u_s} \alpha + \beta \kappa^2 \, ds$$



Undefined border   Non-convex term

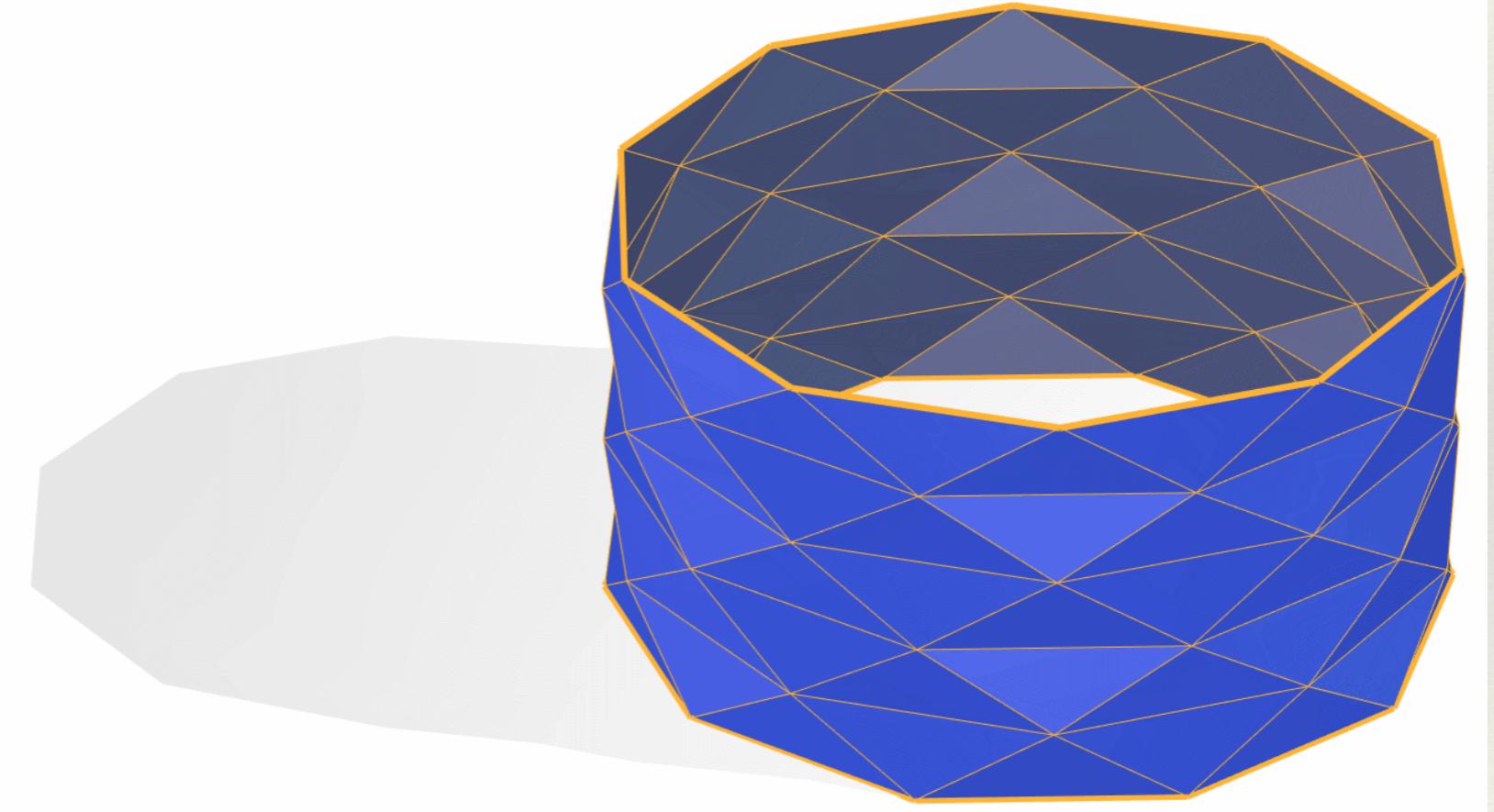
Difficult to optimize

2nd order term, careful with discretization

# Estimating discrete measurements is hard

N: number of  
axial slices  
M: number of radial  
vertices

**$N \rightarrow M^2$**   
**M: 6**  
**N: 10**  
**area: 6.362709035**



Schwarz Lantern counter-example: Surface area cannot be defined as the supremum of inscribed polyhedral surfaces [3]

# Multigrid convergence

**Definition 1 (Multigrid convergence for local geometric quantites)** A local discrete geometric estimator  $\hat{u}$  of some geometric quantity  $u$  is (uniformly) multigrid convergent for the family  $\mathbb{X}$  if and only if, for any  $X \in \mathbb{X}$ , there exists a grid step  $h_X > 0$  such that the estimate  $\hat{u}(D_h(X), \hat{x}, h)$  is defined for all  $\hat{x} \in \partial_h X$  with  $0 < h < h_X$ , and for any  $x \in \partial X$ ,

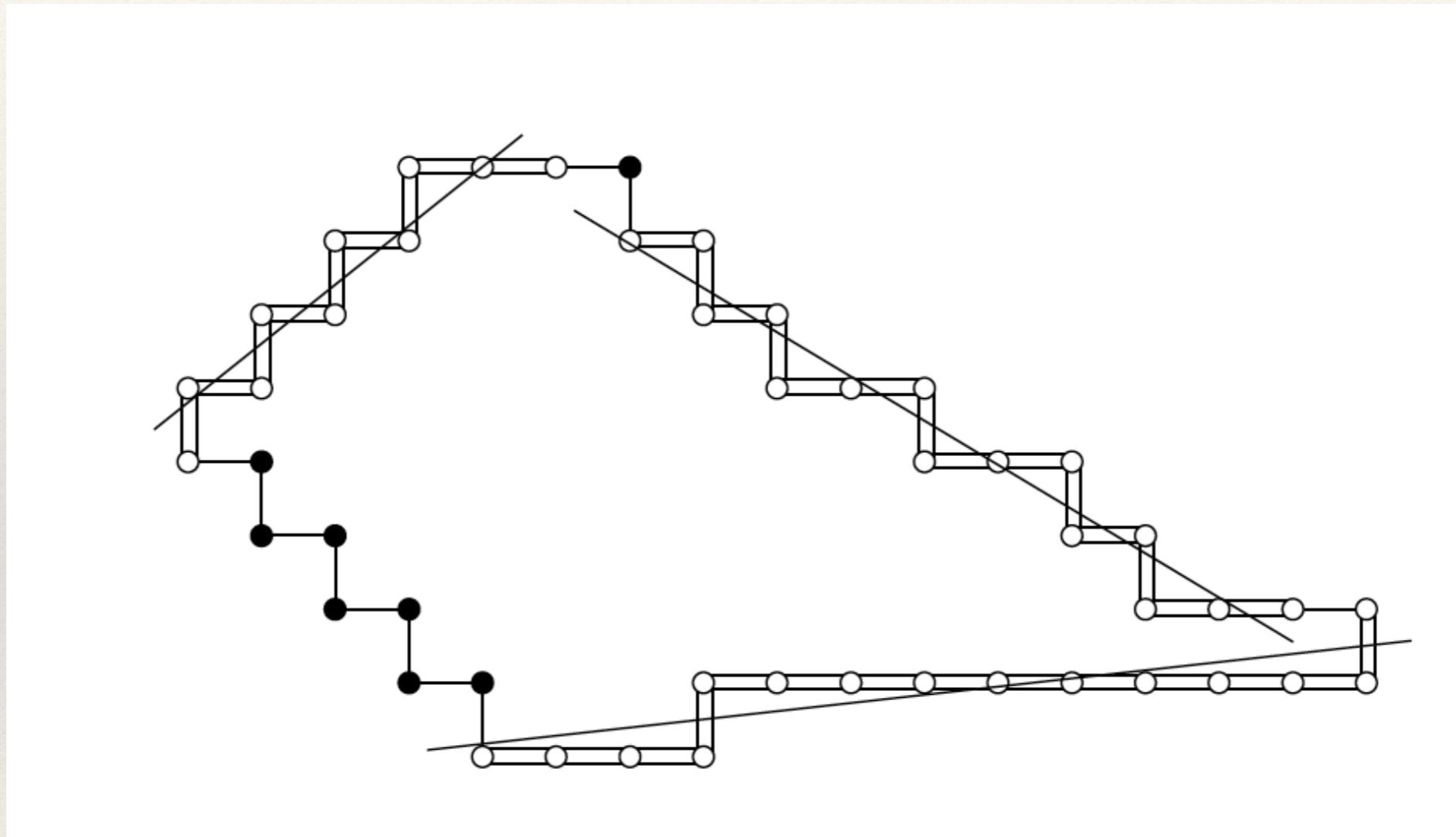
$$\forall \hat{x} \in \partial_h X \text{ with } \|\hat{x} - x\|_\infty \leq h, \|\hat{u}(D_h(X), \hat{x}, h) - u(X, x)\| \leq \tau_X(h),$$

where  $\tau_X : \mathbb{R}^+ \setminus \{0\} \rightarrow \mathbb{R}^+$  has null limit at 0. This function defines the speed of convergence of  $\hat{u}$  towards  $u$  for  $X$ .

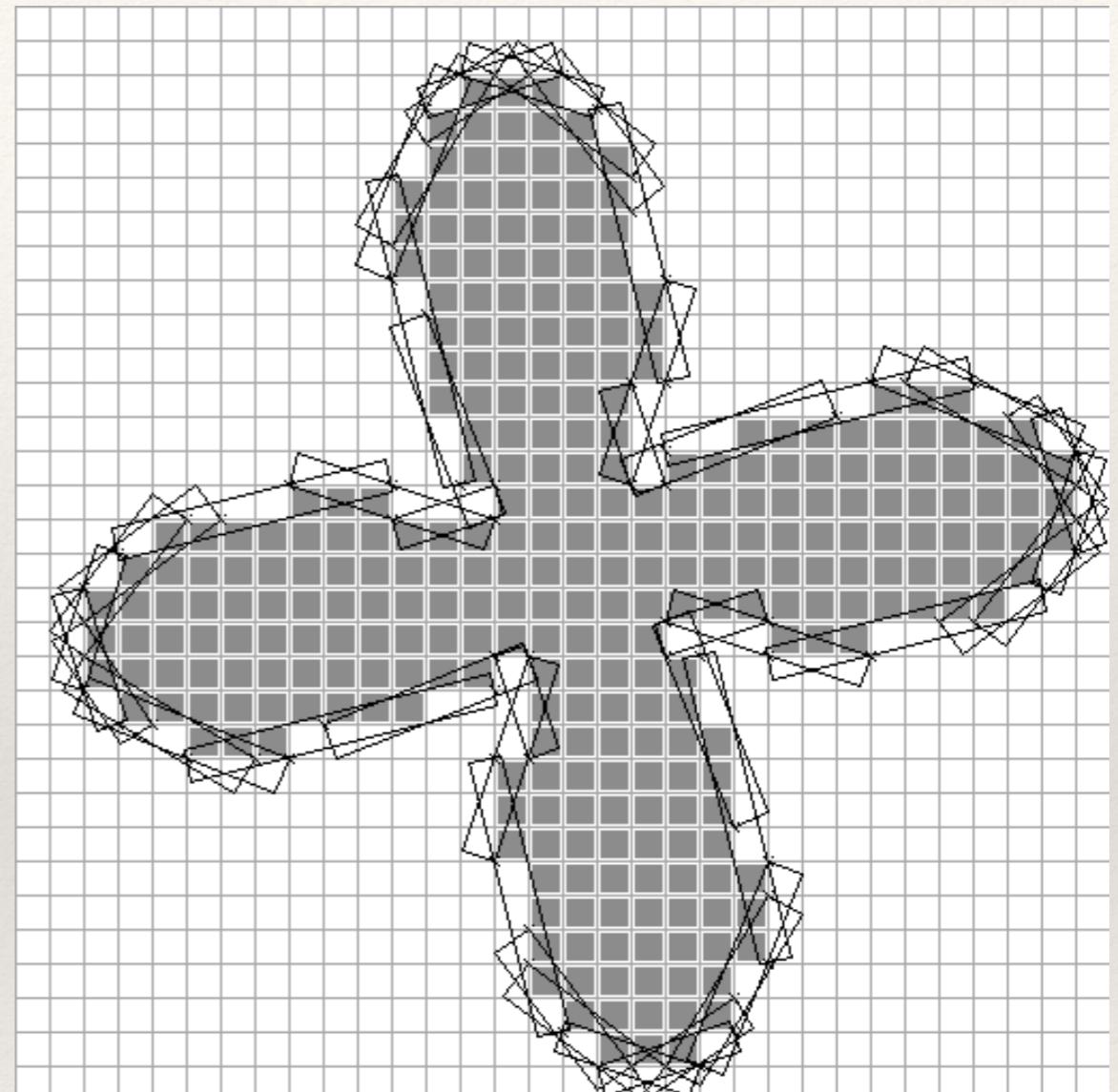
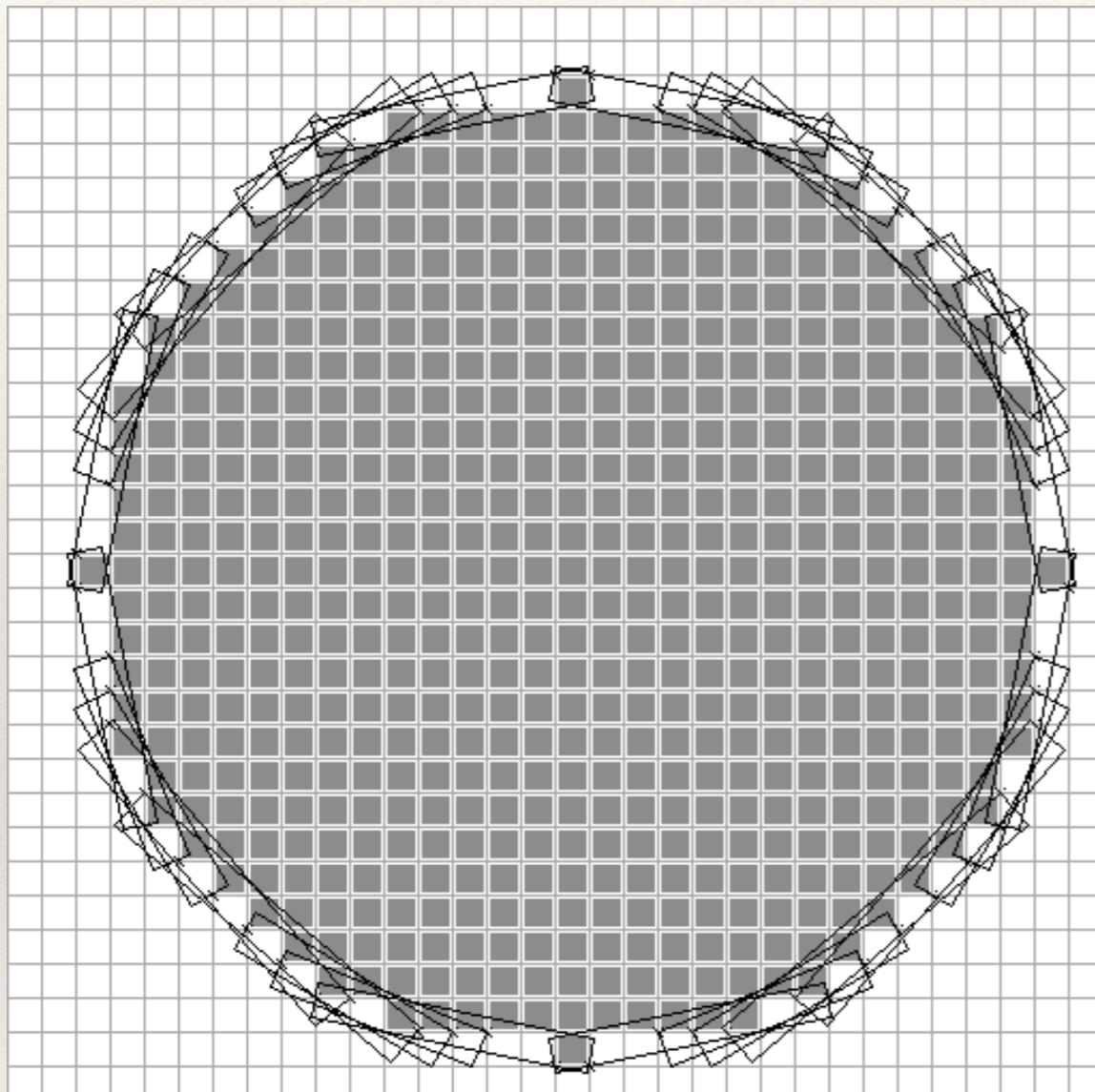
Gauss digitization:  $D_h(X) = X \cap (h\mathbb{Z})^2$

h-boundary:  $\partial_h X$

# Multigrid convergent estimators of perimeter

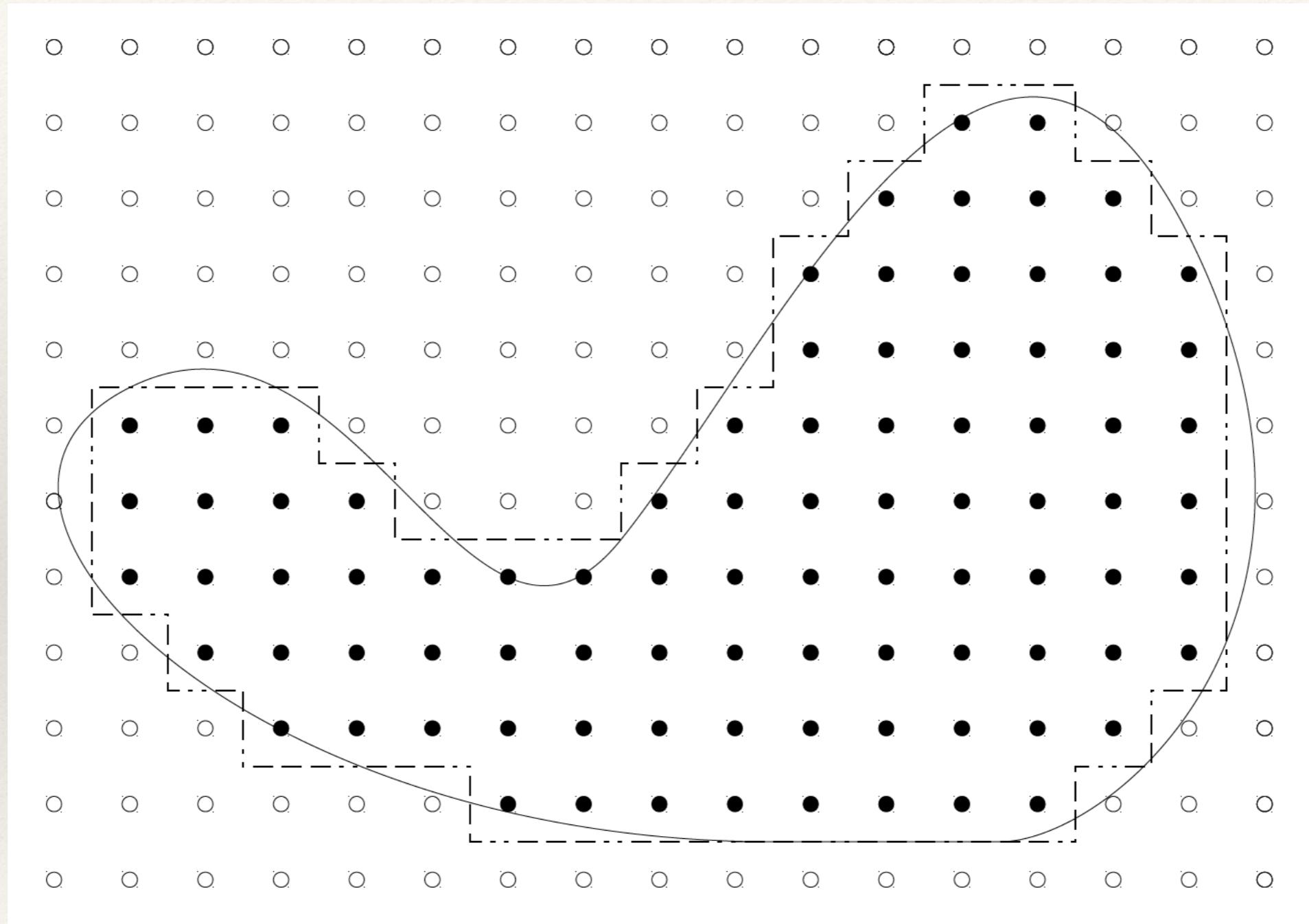


# Multigrid convergent estimators of perimeter

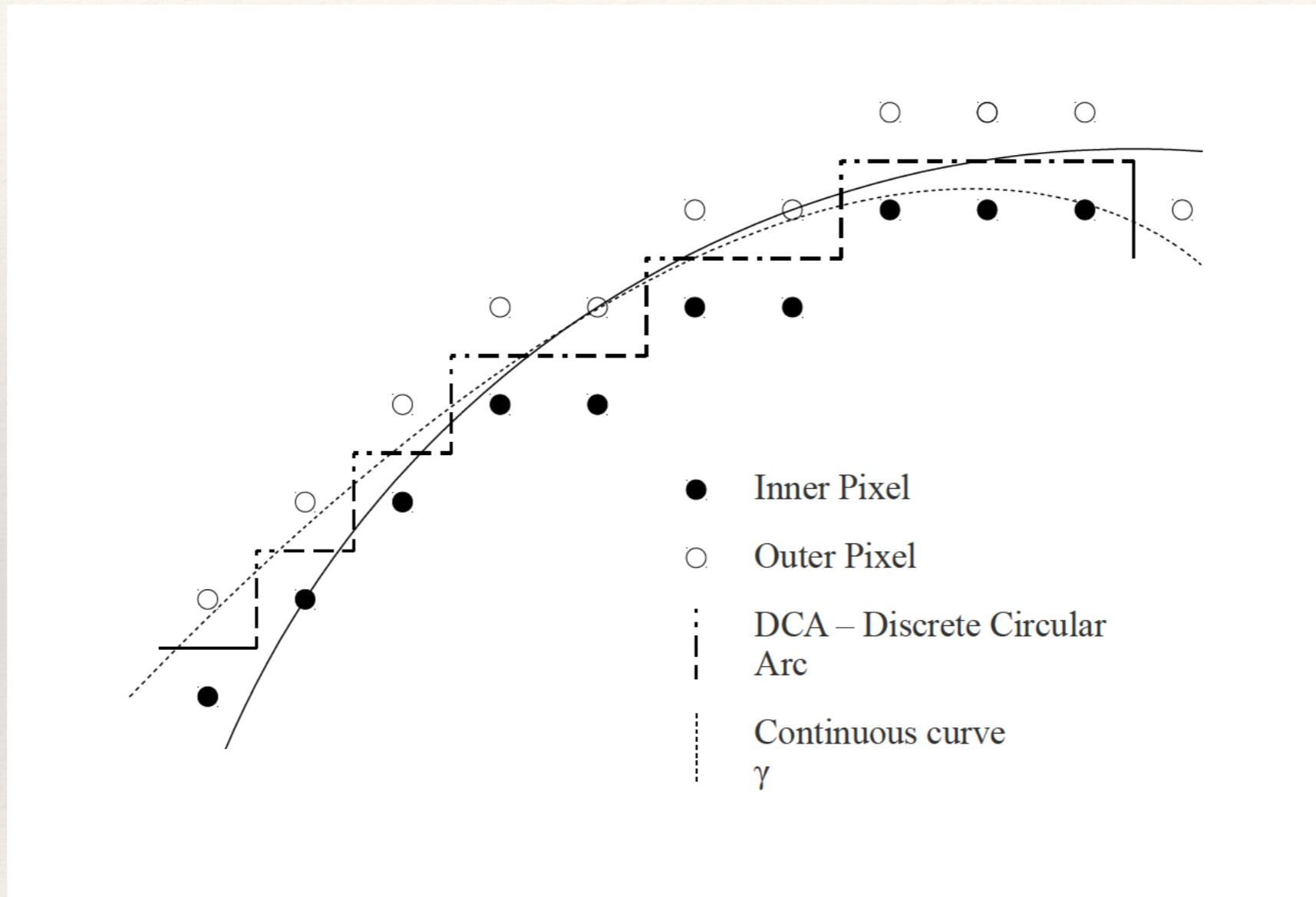


See [1]

# Multigrid convergent estimators of curvature

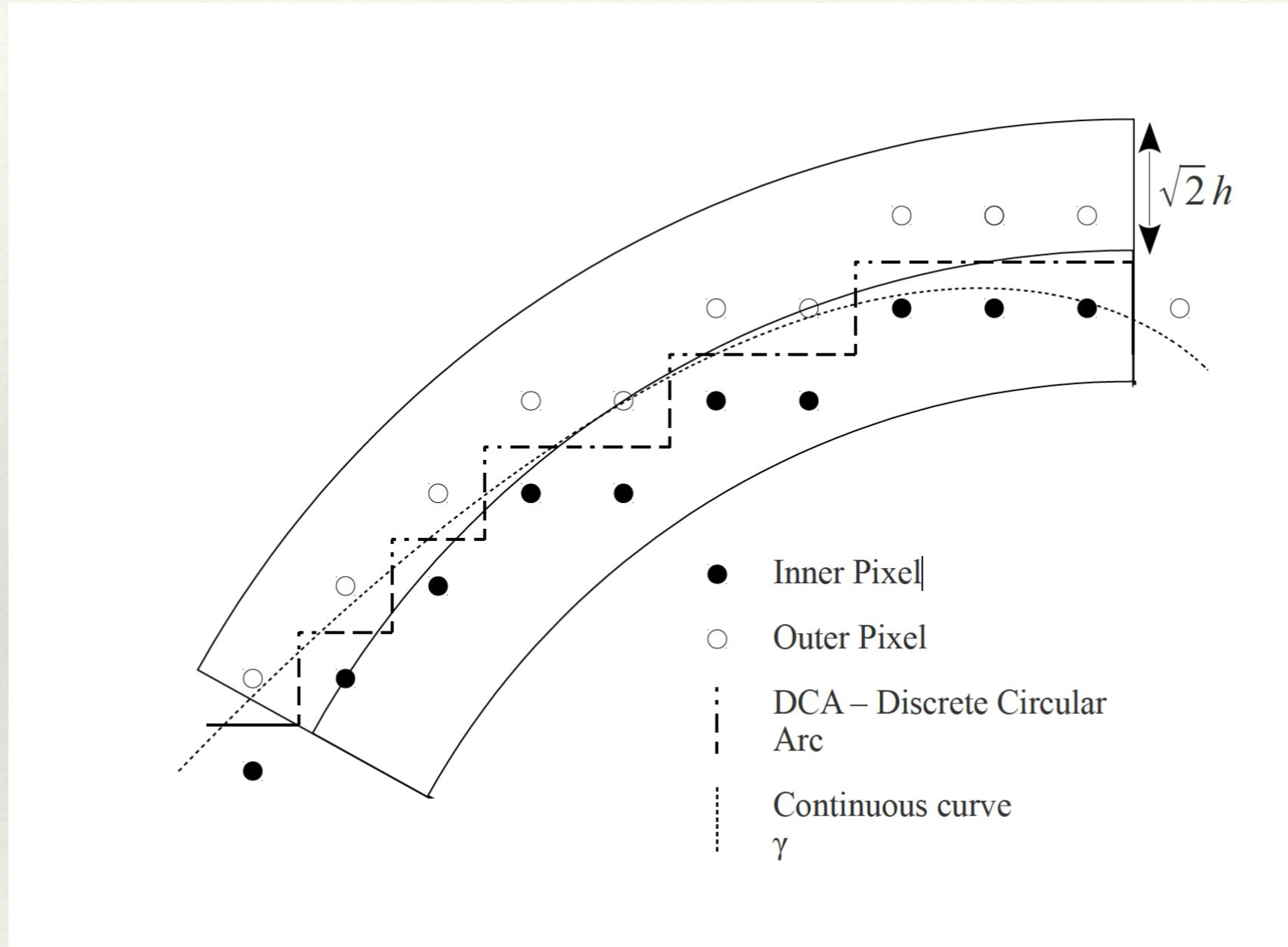


# Maximal Digital Circular Arc



MDCA is a multigrid-convergent estimator of curvature [2]

# Maximal Digital Circular Arc



MDCA is a multigrid-convergent estimator of curvature [2] in  $\mathcal{O}(h^{\frac{1}{3}})$

# Using MDCA for optimizing curvature

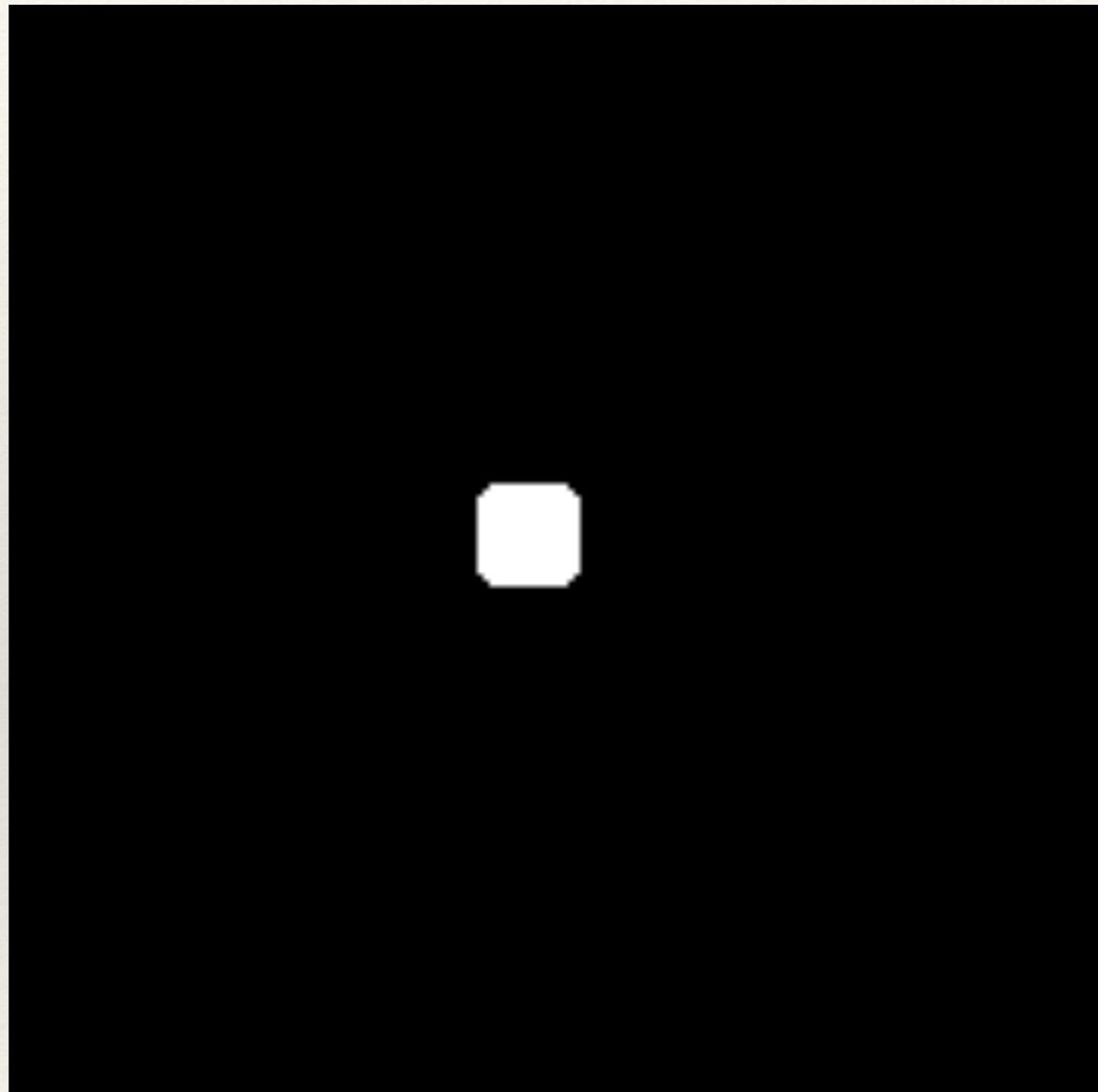
- ❖ Grid cell representation
- ❖ Binary variables for pixels and lines

$$\min_{x_p, x_e \in \Omega} \sum_{x_p \in \Omega} (u_p - x_p)^2 + \sum_{x_e \in \Omega} \hat{\kappa}^2(x_e) \cdot x_e$$

s.t  $x \in \{0,1\}, T(\Omega)$

- ❖ Relaxation, consistency constraints...

# Using MDCA for optimizing curvature



# Using MDCA for optimizing curvature

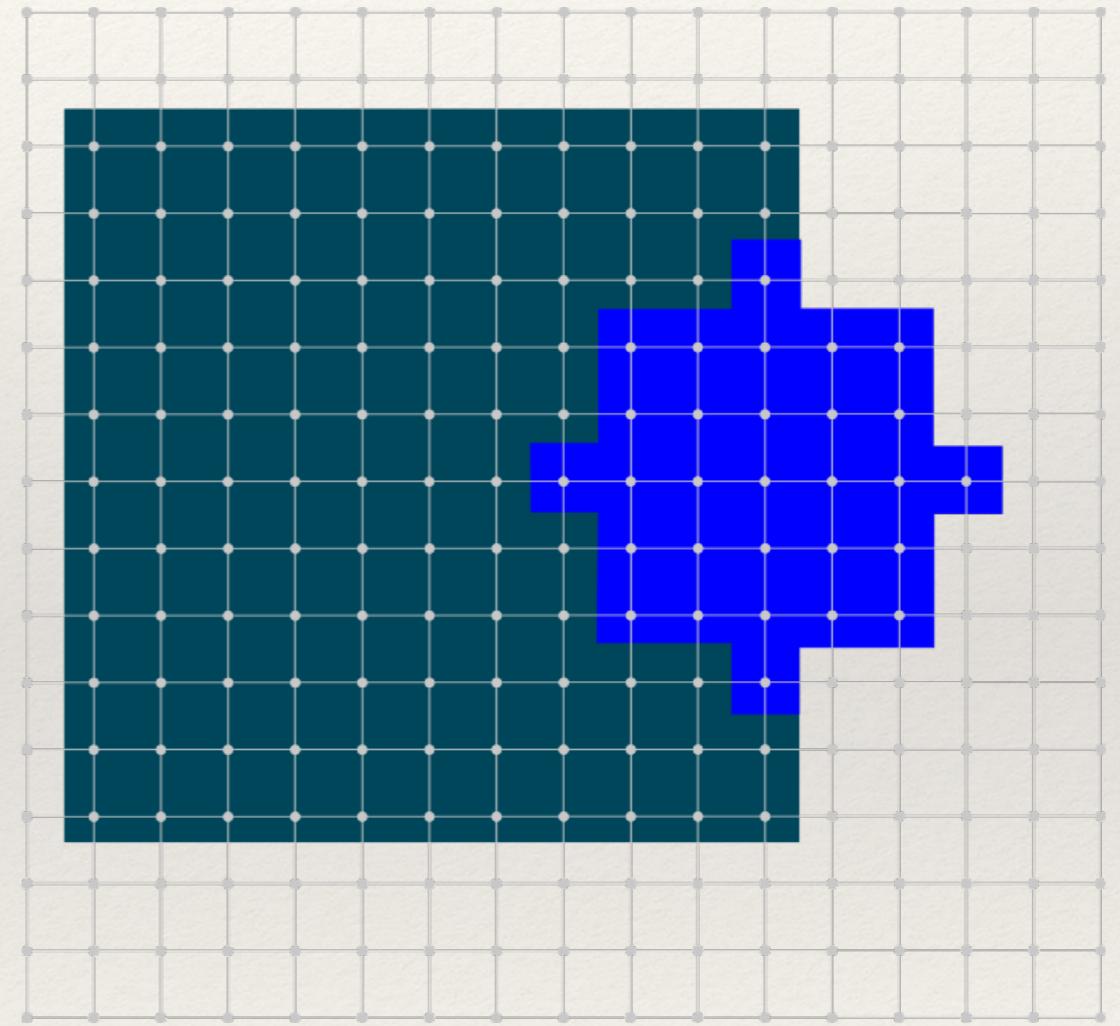
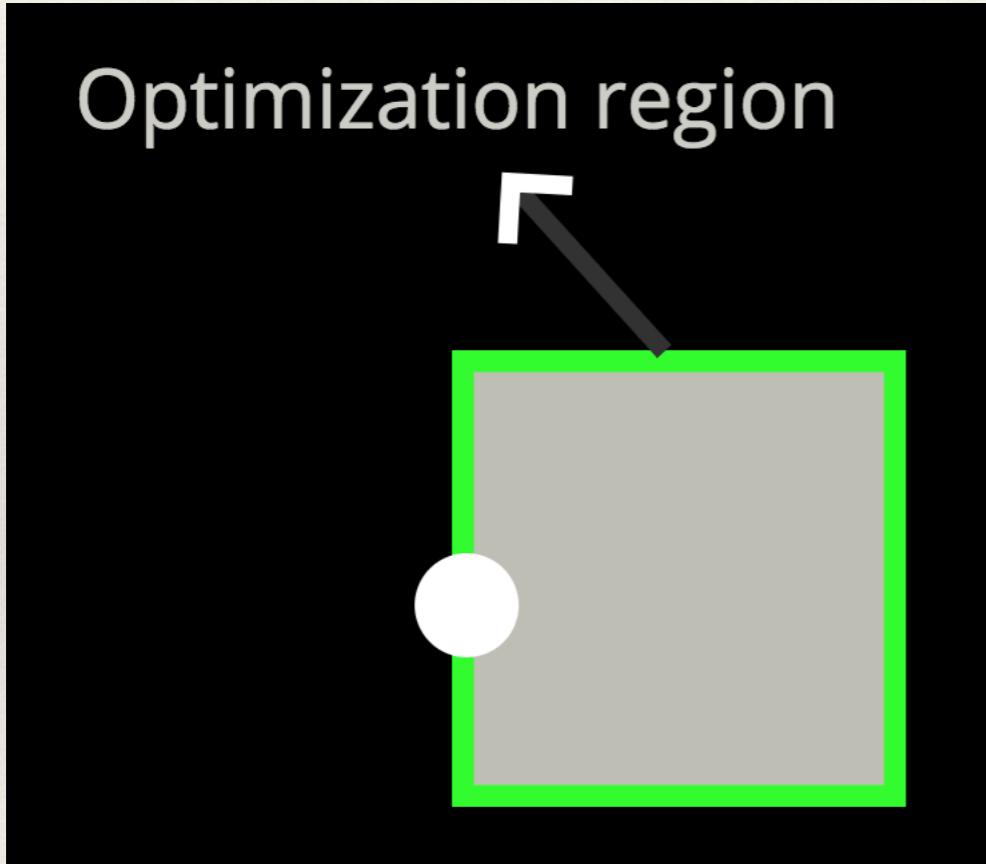
- ❖ MDCA is too local
- ❖ Optimisation is difficult: long running times or poor relaxation.

# Integral area invariant

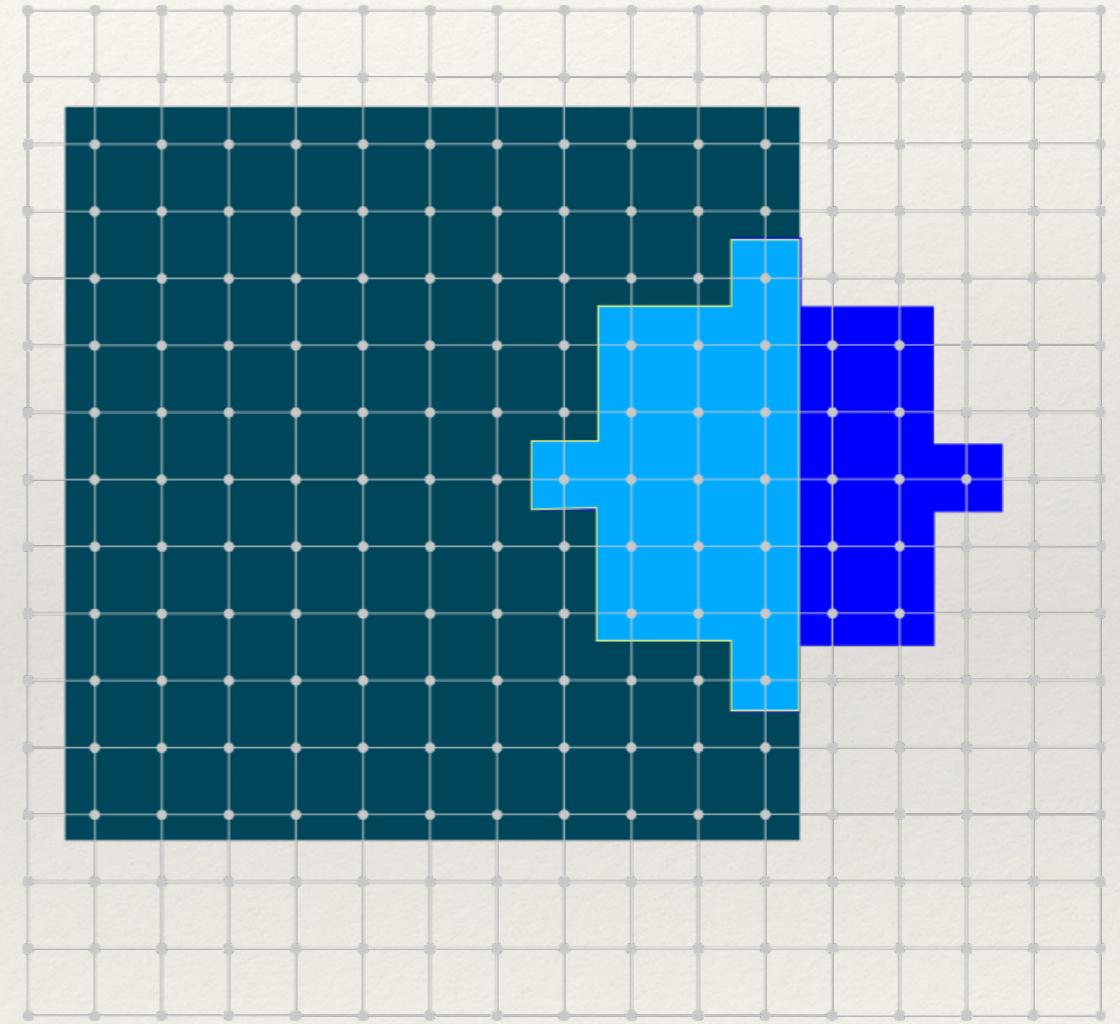
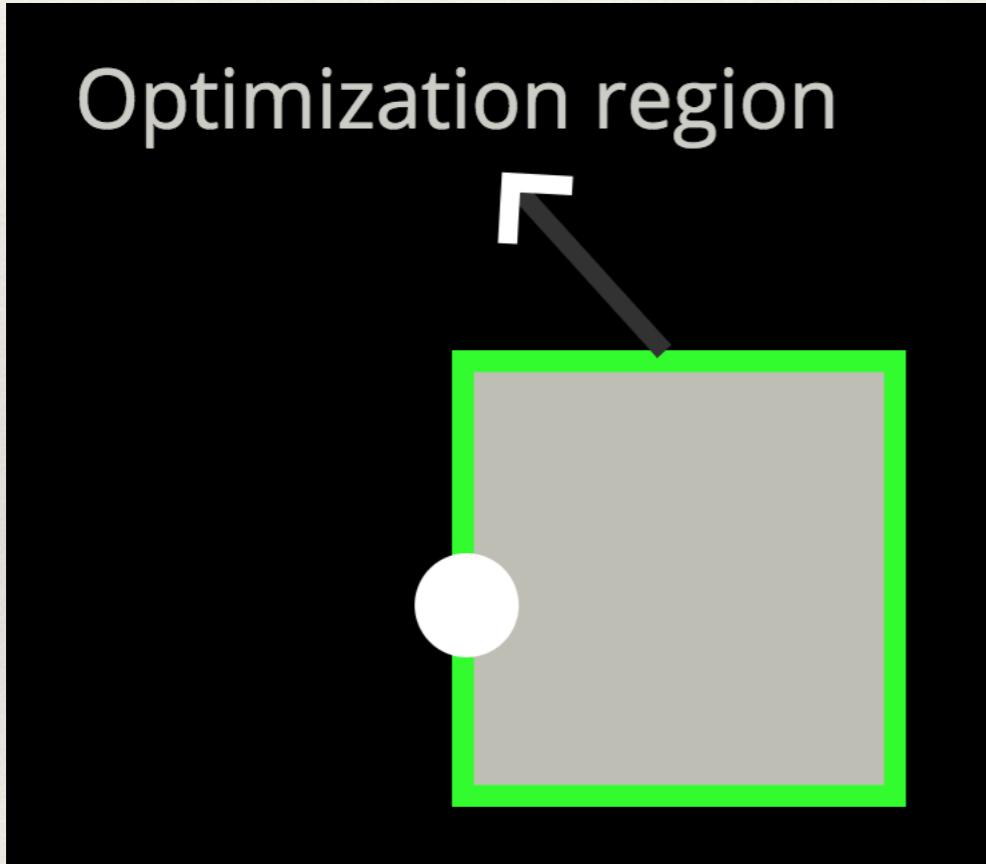
- ❖ Let  $\sigma_{X,r}(p)$  be the intersection area of a ball  $B_r(p)$  with shape  $X$
- ❖ Taylor expansion:  $\sigma_{X,r} = \frac{\pi}{2}r^2 - \frac{\kappa(X, r)}{3}r^3 + O(r^4)$
- ❖ a MC curvature estimator is (see [4]):

$$\tilde{\kappa} := \frac{3}{r^3} \left( \frac{\pi r^2}{2} - \sigma_{X,r}(p) \right)$$

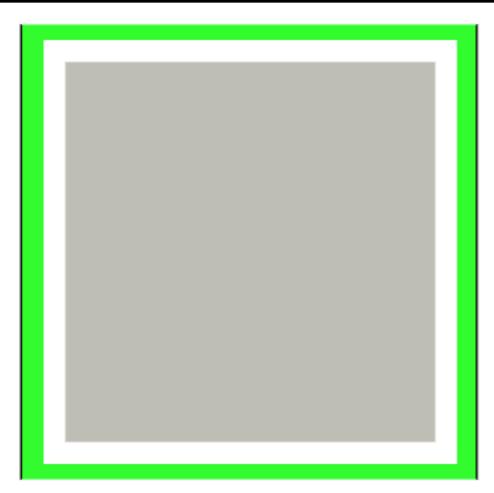
# Discrete implementation



# Discrete implementation



# Optimisation



Optimization region

Estimator evaluation region

$$Y^* = \arg \min_{Y \in \{0,1\}^{|O|}} \sum_{p \in A} \hat{\kappa}_{II}(p)^2$$

$$P(D) = \{ x \mid x \in D, |\mathcal{N}_4(x) \cap D| < 4 \}$$

$O$	$= P(D)$	Optimization region.
$F$	$= D - P(D)$	Trust foreground.
$B$	$= \Omega - D$	Trust background.
$A$	$= P(F) \cup P(B)$	Computation region.

# Optimisation

Define constants  $c_1 = (3/r^3)^2$ ,  $c_2 = \pi r^2/2$ . Hence,

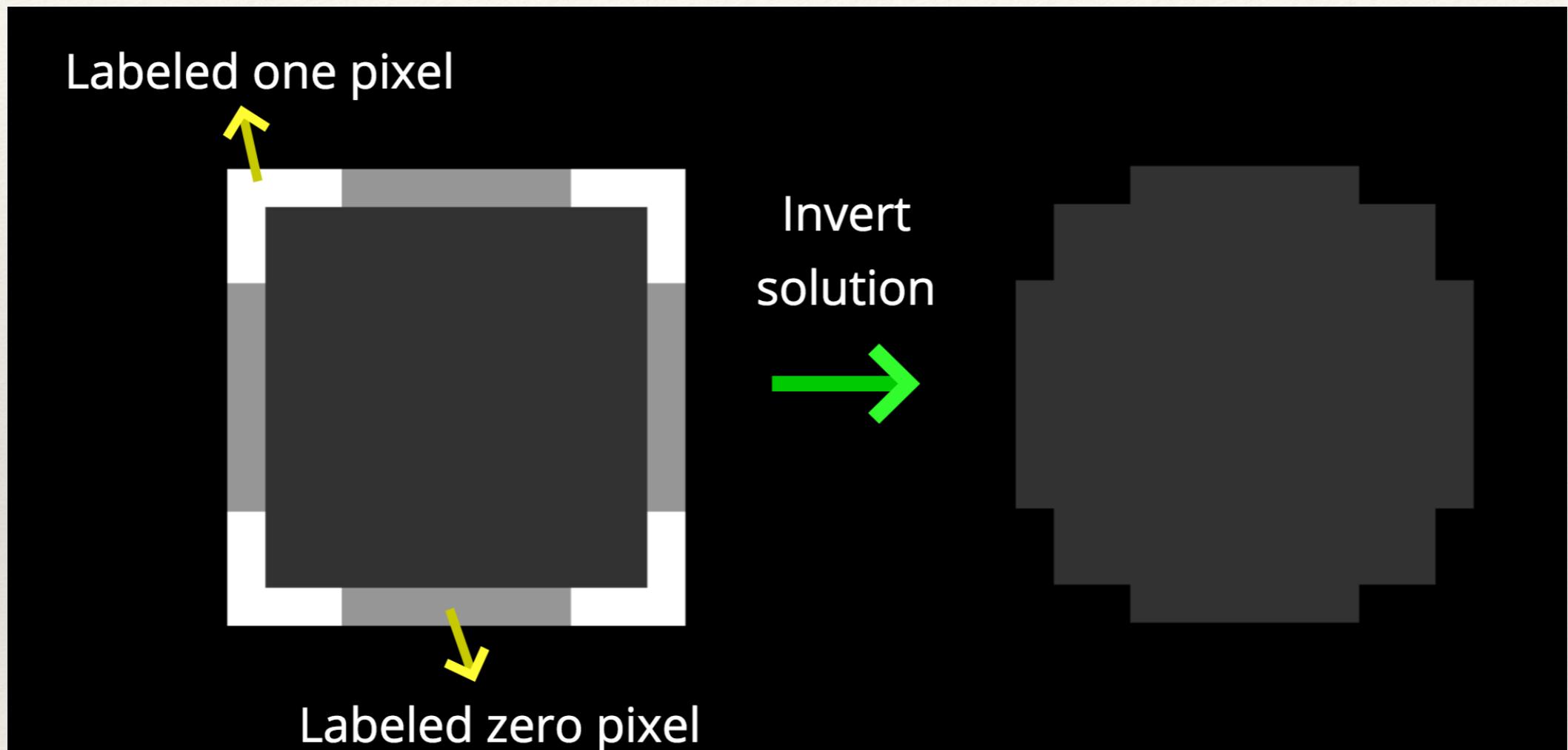
$$\begin{aligned}\hat{\kappa}_r^2(p) &= c_1 \cdot (c_2 - \sigma_{D,r}(p))^2 \\ &= c_1 \cdot (c_2^2 - 2c_2\sigma_{D,r}(p) + \sigma_{D,r}(p)^2).\end{aligned}$$

Denoting  $F_r(p)$  the intersection set with the foreground

$$Y^* = \arg \min_{Y \in \{0,1\}^{|O|}} \sum_{p \in A} \left( (1 + |F_r(p)| - c_2) \cdot \sum_{y_i \in Y_r(p)} y_i + \sum_{\substack{y_i, y_j \in Y_r(p) \\ i < j}} y_i y_j \right).$$

This problem is a non-linear sum of products of binary terms.  
It can be formulated as an LP. We use QPBO [5] as the solver.

# Optimisation



# Curvature flow

- ❖ Does it work?

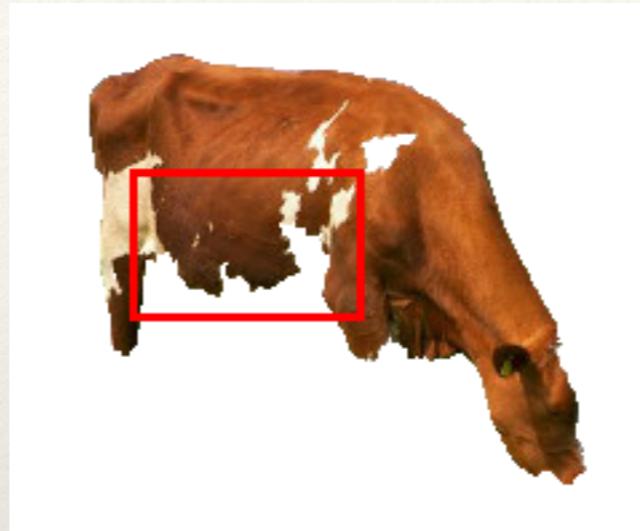
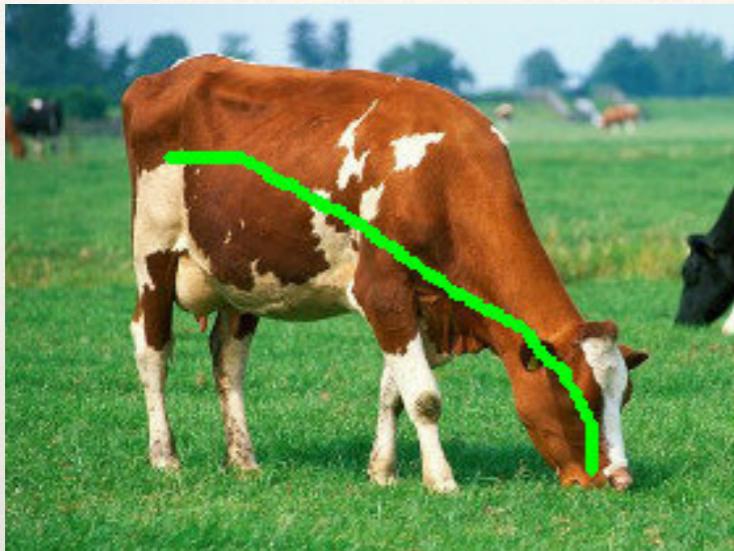


# Curvature flow

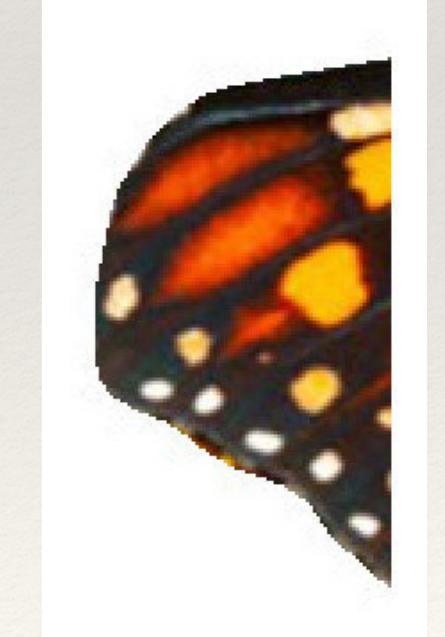
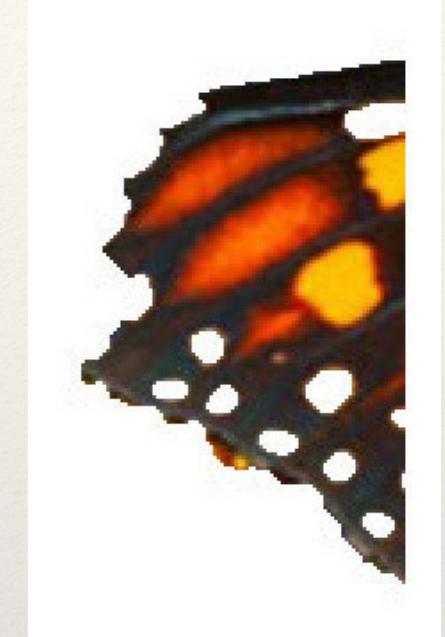
- ❖ Does it work?



# Segmentation



# Segmentation



# Bibliography

- ❖ [1] Lachaud, Jacques-Olivier, Anne Vialard, and François de Vieilleville. "Fast, accurate and convergent tangent estimation on digital contours." *Image and Vision Computing* 25.10 (2007): 1572-1587.
- ❖ [2] Schindele, Andreas, Peter Massopust, and Brigitte Forster. "Multigrid Convergence for the MDCA Curvature Estimator." *Journal of Mathematical Imaging and Vision* 57.3 (2017): 423-438.
- ❖ [3] Schwarz, H. A. (1890). *Gesammelte Mathematische Abhandlungen von H. A. Schwarz*. Verlag von Julius Springer. pp. 309–311.
- ❖ [4] Coeurjolly, David, Jacques-Olivier Lachaud, and Jérémie Levallois. "Integral based curvature estimators in digital geometry." *International Conference on Discrete Geometry for Computer Imagery*. Springer, Berlin, Heidelberg, 2013.
- ❖ [5] Kolmogorov, Vladimir, and Carsten Rother. "Minimizing nonsubmodular functions with graph cuts-a review." *IEEE transactions on pattern analysis and machine intelligence* 29.7 (2007).

# Bibliography (2)

- ❖ [6] Mumford, David. "Elastica and computer vision." *Algebraic geometry and its applications*. Springer, New York, NY, 1994. 491-506.
- ❖ [7] Shen, Jianhong, Sung Ha Kang, and Tony F. Chan. "Euler's elastica and curvature-based inpainting." *SIAM journal on Applied Mathematics* 63.2 (2003): 564-592.
- ❖ [8] El-Zehiry, Noha Youssry, and Leo Grady. "Contrast driven elastica for image segmentation." *IEEE Transactions on Image Processing* 25.6 (2016): 2508-2518.
- ❖ [9] Masnou, Simon, and Giacomo Nardi. "A coarea-type formula for the relaxation of a generalized elastica functional." *arXiv preprint arXiv:1112.2090* (2011).
- ❖ [10] Nieuwenhuis, Claudia, et al. "Efficient squared curvature." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014.