Morphological and Statistical Techniques for the Analysis of 3D Images

Enric Meinhardt-Llopis

3–3–2011

Outline

Outline

Motivation: 2D/3D differences

The 3D tree of shapes

Applications of the tree of shapes

Finsler-Cheeger Sets and Affine Invariants

Conclusion

Motivation: differences between 2D and 3D

A large part of this thesis is the extension of the tree of shapes to 3D images, and its applications.

Differences 2D/3D: semantics, visualization, topology, statistics, noise

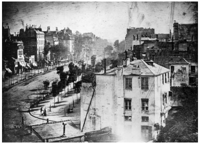
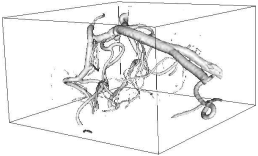
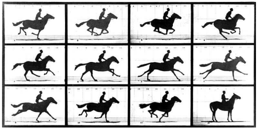


photo (x, y)

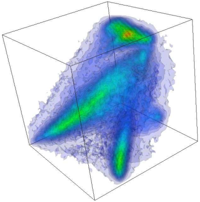


tomography (x, y, z)

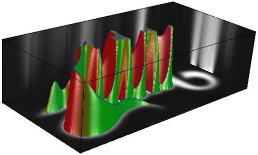


video (x, y, t)

X-ray (x, y)



RGB histogram (r, g, b)

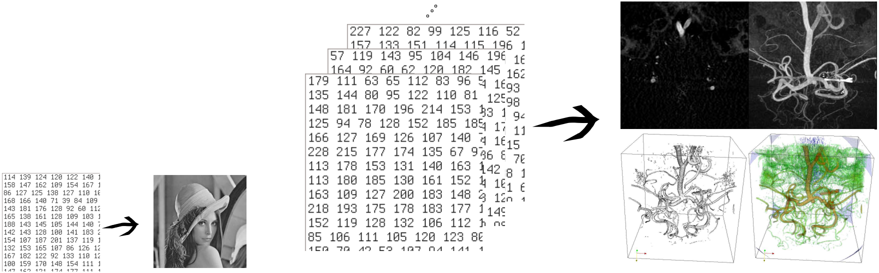


scale-space (x, y, s)

Different visualization of 2D and 3D images

The first difference is the need for visualization.

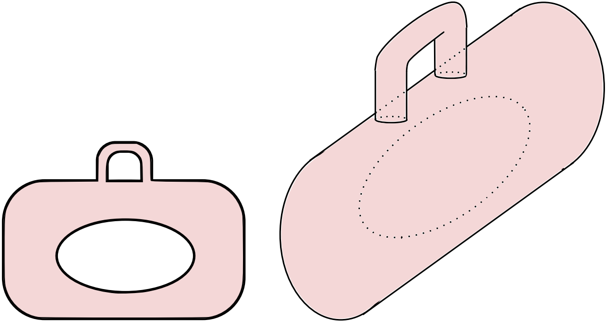
- 2D, trivial: simply look at them (array of pixels)
- 3D, tricky: slices, projections,isosurfaces, volume rendering



Topological differences between 2D and 3D

The second difference is topological behaviour.

- 2D: cavities and handles are the same thing
- 3D: cavities and handles are different things



Due to this fact, for example, the number of cavities can be computed locally in 2D but not in 3D.

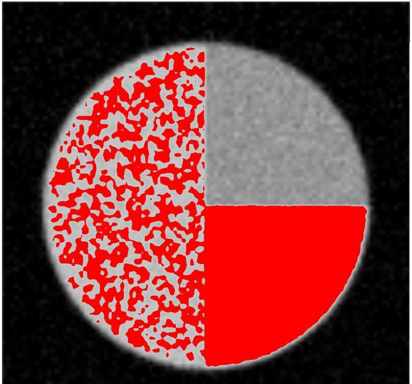
Statistical differences between 2D and 3D

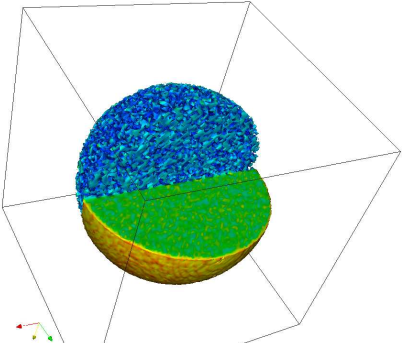
Critical levels for bound percolation:

- 2D grid: 0.5
- 3D grid: 0.2488

This affects, for example, the behaviour of level sets of white noise (at the mean value):

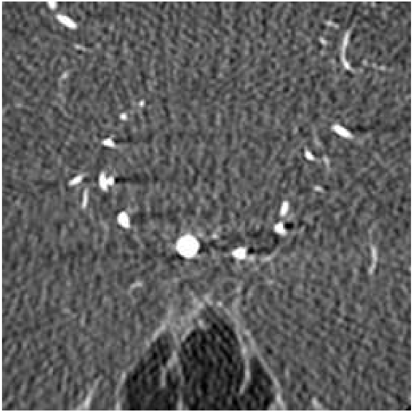
- in 2D: many connected components
- in 3D: a single connected component





Different kinds of noise between 2D and 3D images

- 2D: noise from cameras is independent from pixel to pixel
- 3D: noise from CT has anisotropic artifacts



Slice of a tomography

Outline

Data structures for images

There are many data structures for storing digital images:

- Array of pixel values
- Coefficients on a linear basis (e.g., Fourier representation)
- Tree-like data structures:
  - Quadtrees/octtrees [Samet'79, Meagher'82]
  - Region merging trees [Koepfler *et al.* '94, Salembier *et al.* '00]
  - Tree of (upper or lower) level sets [Cox *et al.* '03, Carr *et al.* '03]
  - Tree of shapes [Monasse *et al.* '02]

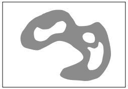
Each representation favors some operations.

Mathematical definition of the tree of shapes

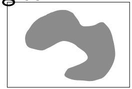
Intuition: Shapes $=$ connected components of level surfaces
$=$ external boundaries of saturations

Notation:

- Image $f : \Omega \to \mathbf{R}$
- $[f \geq \lambda] := \{x \in \Omega \ : \ f(x) \geq \lambda\}$
- $cc(A, x) :=$ connected component of A containing x
- $Sat(A, p_\infty) := \Omega \setminus cc(\Omega \setminus A, p_\infty)$



A



Sat(A)

Trees of level sets:

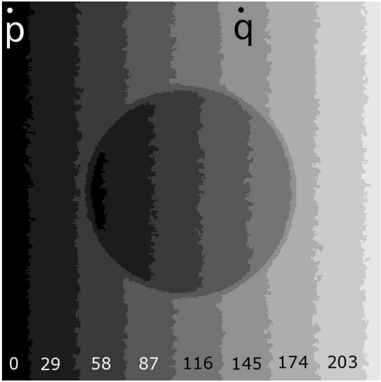
- $\mathcal{ULT}(f) := \{cc([f(x) \geq \lambda], x) \ : \ x \in \Omega, \lambda \in \mathbf{R}\}$
- $\mathcal{LLT}(f) := \{cc([f(x) < \lambda], x) \ : \ x \in \Omega, \lambda \in \mathbf{R}\}$

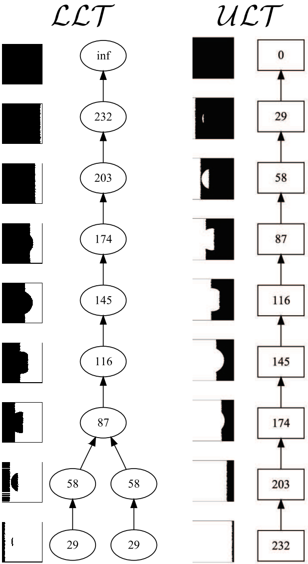
Tree of shapes:

- $\mathcal{TOS}(f) := \{Sat(r), \ r \in \mathcal{ULT}(f) \cup \mathcal{LLT}(f)\}$

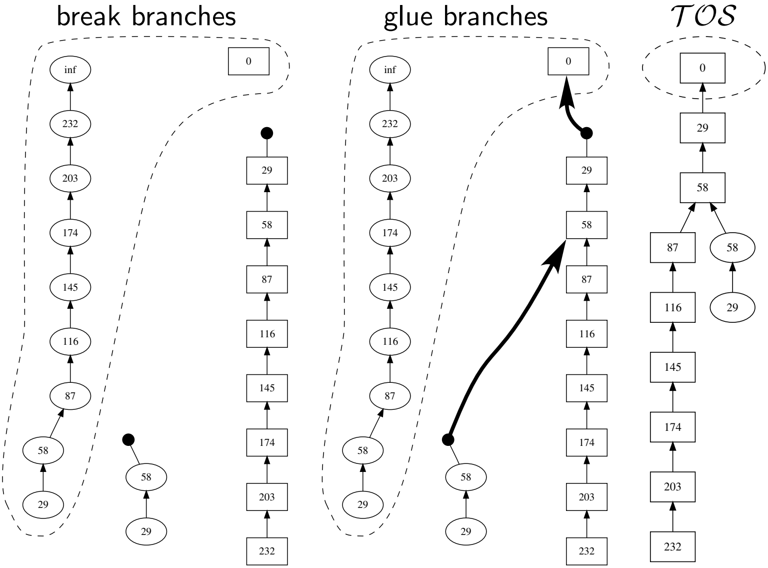
Tree of shapes as fusion of upper and lower trees

image





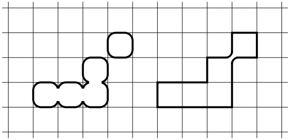
Tree of shapes as fusion of upper and lower trees



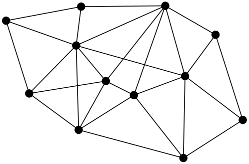
Implementation

Discretization of the image domain, two choices:

semi-continuous interpolation



graph-based



Algorithms:

- Building the upper tree: pixel merging
- Joining both trees: branch gluing
- Where to glue: find the smallest “outer adjacent” region

Outline

Applications of the 3D tree of shapes

- Morphological operations (tree pruning)
- Image visualization (tree navigation)
- RGB histogram analysis
- Video analysis
- 3D edge detection

Morphological filtering

Grain filtering: removal of small leafs

original image

opening

closing

grain filter

original image
18351 shapes

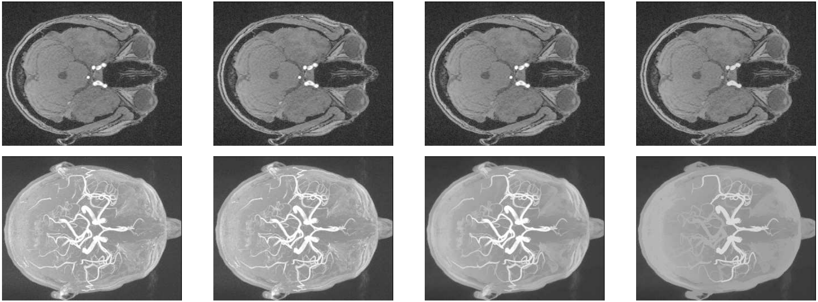
t=2
11173 (61%)

t=10
4924 (27%)

t=1000
715 (4%)

Morphological filtering

Grain filtering: removal of small leafs



original image	t=2	t=100	t=10000
295937 shapes	141005 (47%)	2414 (0.8%)	124 (0.04%)

Observation: it does not seem to affect the texture.

Morphological filtering

Adaptive quantization: removal of nodes inside long branches

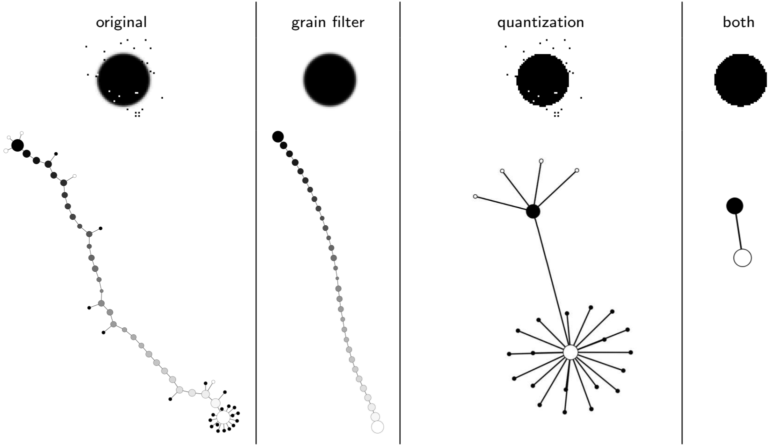
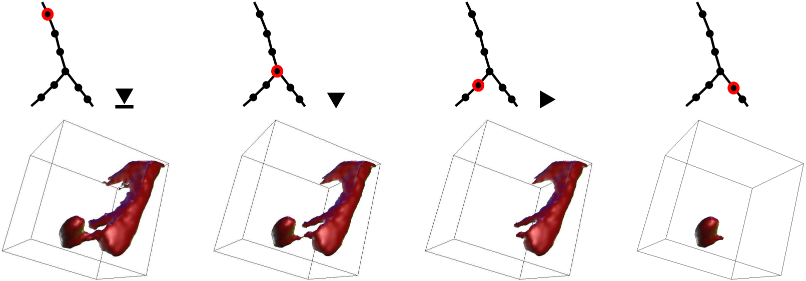


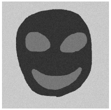
Image visualization

Navigating the tree (with keyboard or mouse):

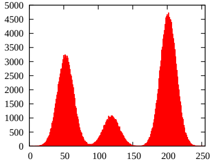


RGB histogram analysis

gray image

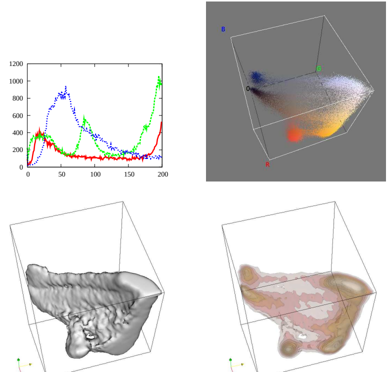


its 1D histogram

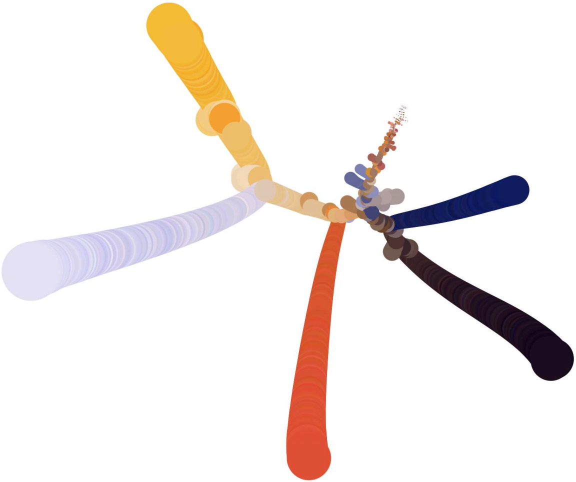


color image

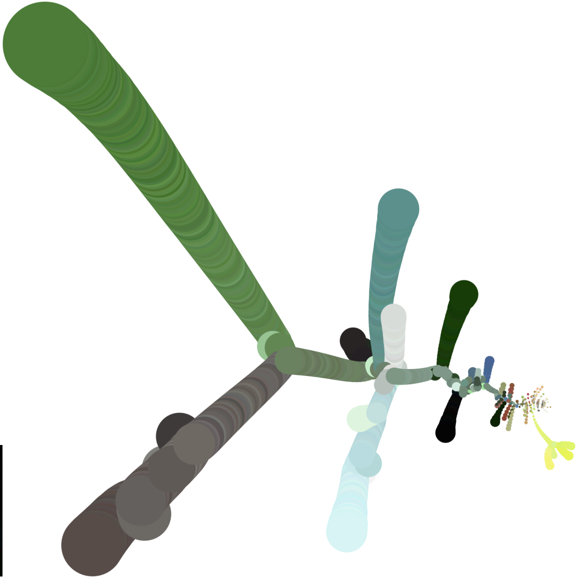
visualizations of its 3D histogram



RGB histogram analysis



RGB histogram analysis



Video analysis

1. Build the structure
   - Input: video & optical flow
   - Pre-processing: graph of optical flow connectivity
   - Structure: level sets of video data on the graph

2. Work with it:
   - Grain filtering: remove spots, small objects
   - Select a region: tracking
   - Automatic segmentation from scribbles
   - Monocular depth estimation

Video processing using the tree of shapes

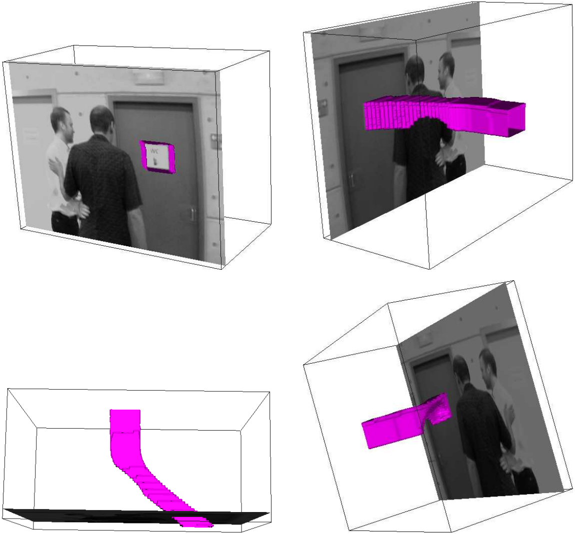
Grain filtering and branch simplification turn the video into a small set of "tubes"

input video
$2s \approx 6.7 \cdot 10^6$ voxels

simplified tree of shapes
617 shapes

Tracking using the tree of shapes

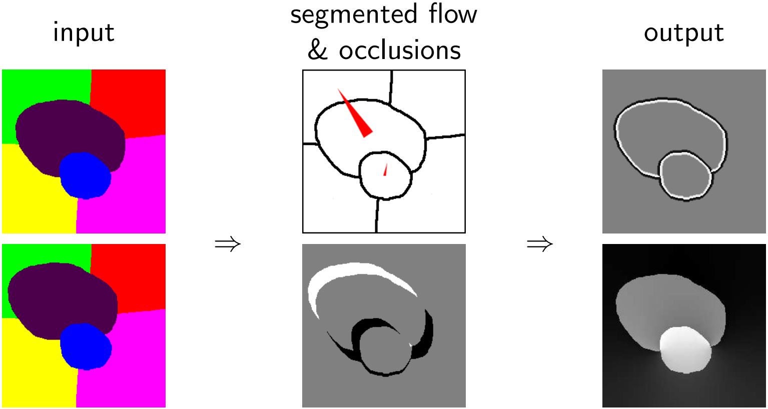
Tracking means querying the structure:



one shape of a video sequence

Monocular depth estimation

The discontinuities of the flow along occlusions are a monocular depth cue.



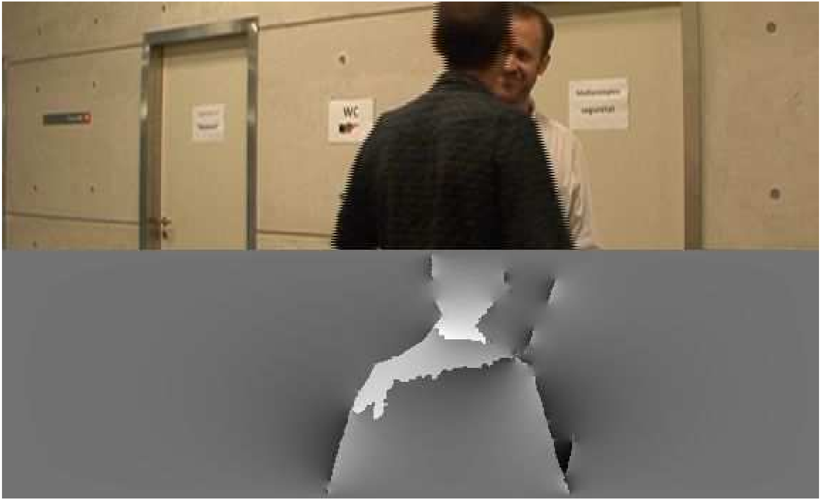
Monocular depth estimation

(monocular depth estimation video)

Monocular depth estimation

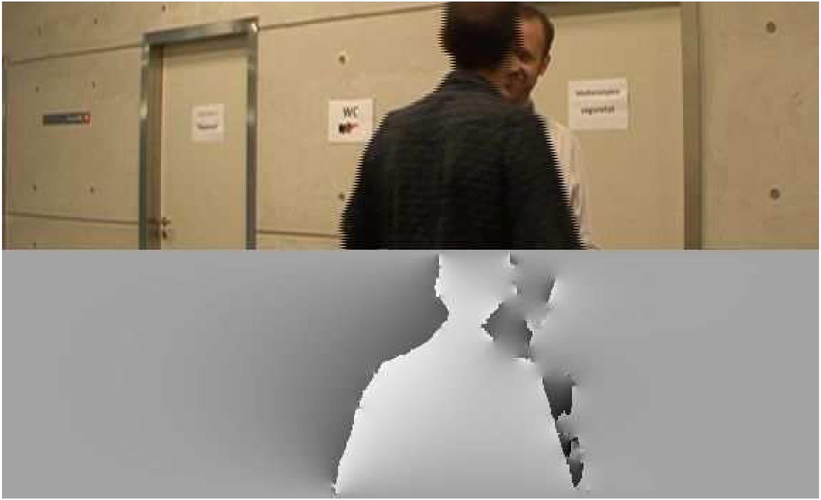
1/14

Monocular depth estimation



2/14

Monocular depth estimation



3/14

Monocular depth estimation

4/14

Monocular depth estimation

5/14

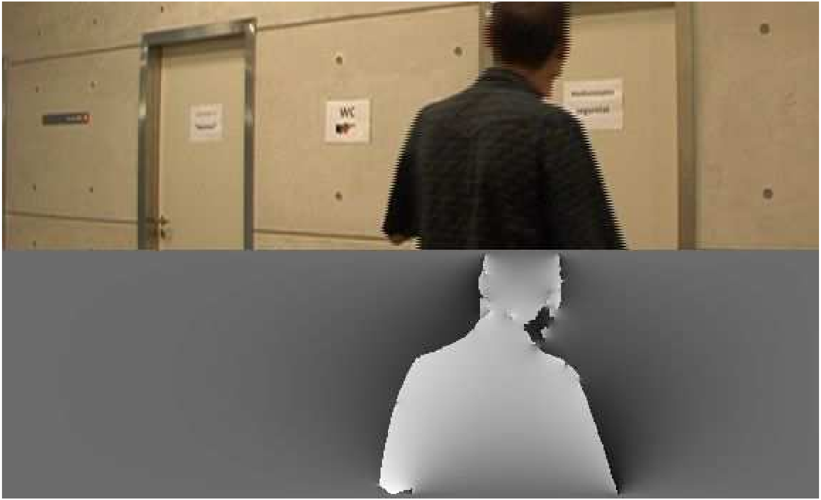
Monocular depth estimation

6/14

Monocular depth estimation

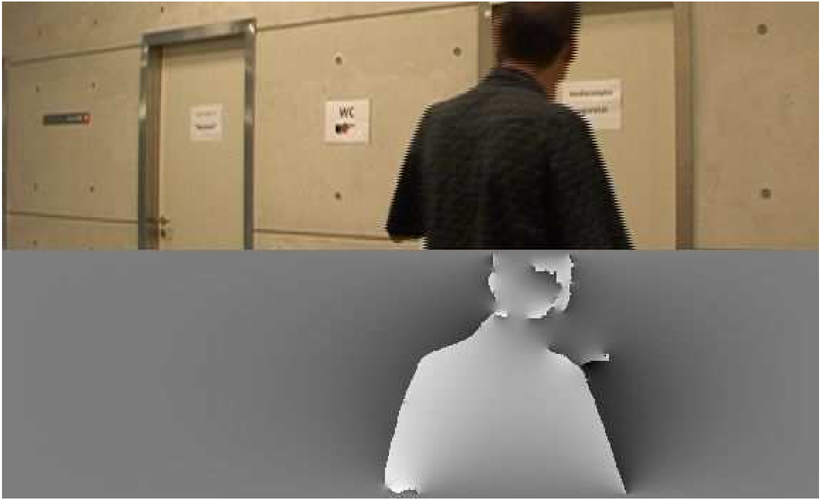
7/14

Monocular depth estimation



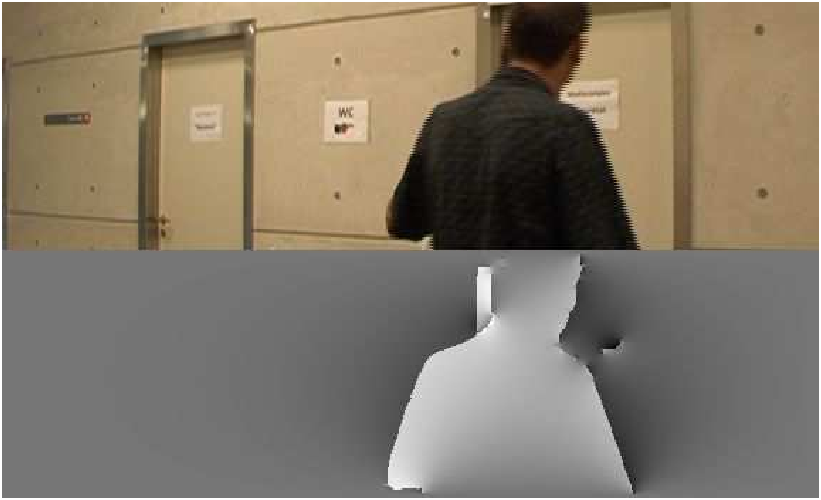
8/14

Monocular depth estimation



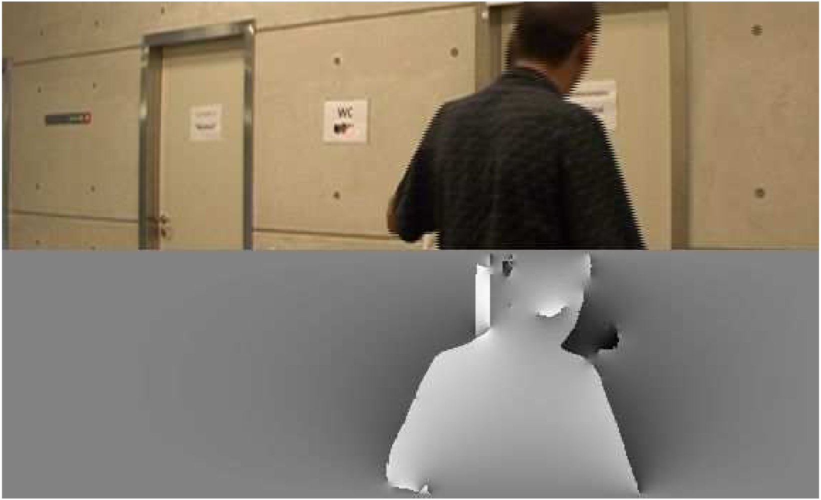
9/14

Monocular depth estimation



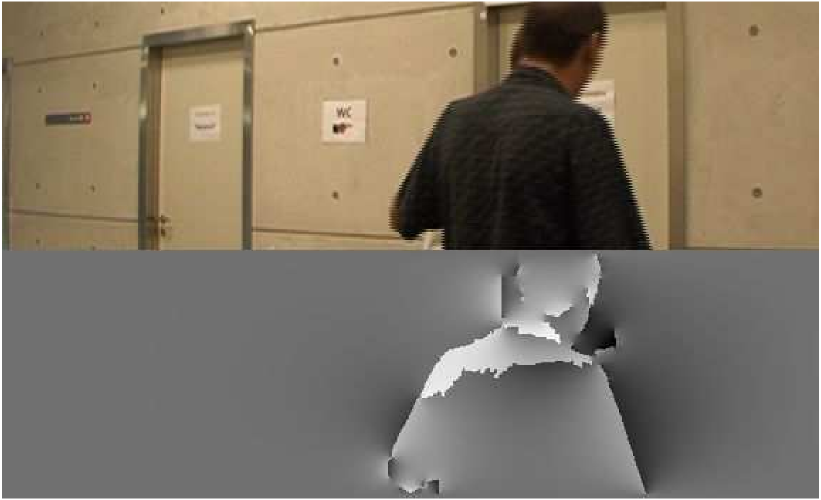
10/14

Monocular depth estimation



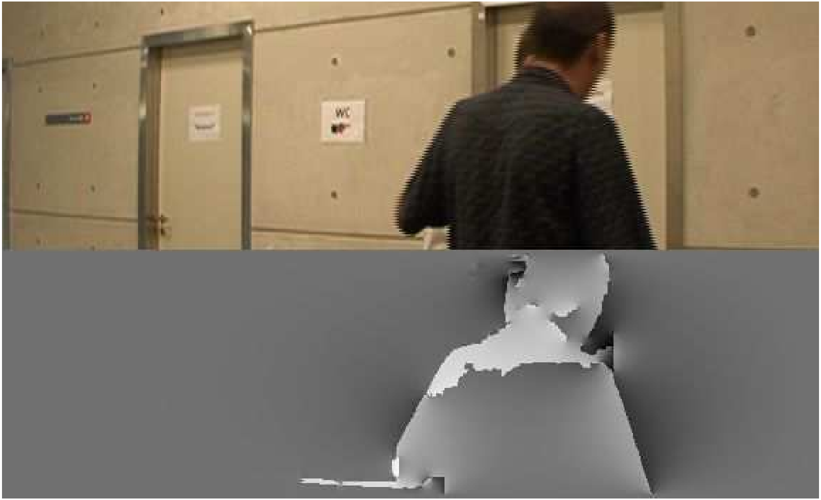
11/14

Monocular depth estimation



12/14

Monocular depth estimation



13/14

Monocular depth estimation

14/14

3D Edge detection

Extend to 3D images the 2D detector based on Helmholtz principle. [Desolneux *et al.* '01]

- In 2D: *select* connected pieces of level curves which are well-contrasted
- In 3D: *select* connected pieces of level surfaces which are well-contrasted

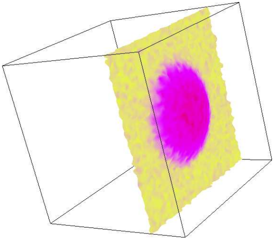
✓ No priors

✓ No parameters

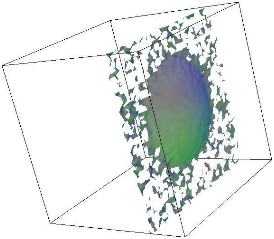
✗ Connected subsets of a given surface can be very complicated!

Mumford-Shah segmentation on surfaces

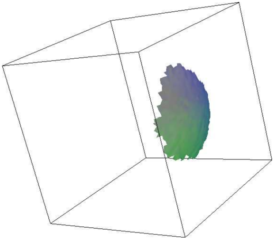
Solution: restrict the selection to the Mumford-Shah hierarchy of the "contrast" on each level surface.



contrast on
level surface



best subset
among all

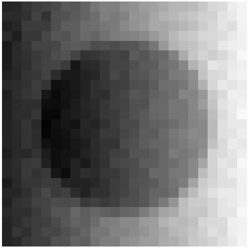


best subset
on MS hierarchy

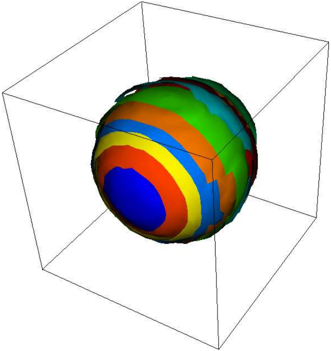
3D edge detection by Helmholtz principle

Algorithm:

* For each level surface S of the image:
  * Build a Mumford-Shah hierarchy of the contrast on S
  * For each patch P in the hierarchy
    * if P is well-contrasted, select P



input image
(slice)



selected patches

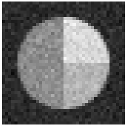
Hypothesis of the proposed edge detector

The desired boundaries are formed by large pieces of level surface of the original image.

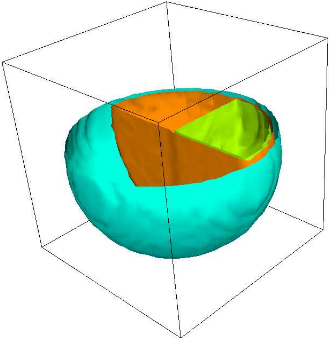
Observation: this hypothesis is more sound for 3D medical images than for 2D photographs.

Experimental results of edge detection

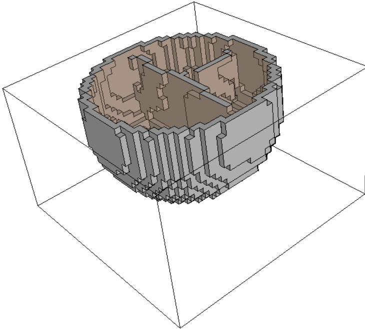
Synthetic best case:



input



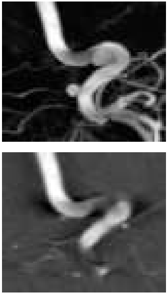
output
(3 patches)



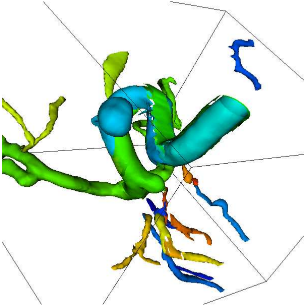
Canny

Experimental results of edge detection

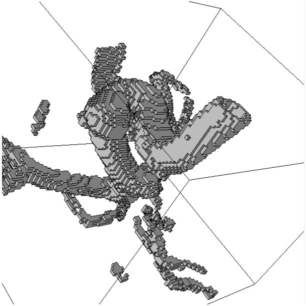
Real CTA image:



input



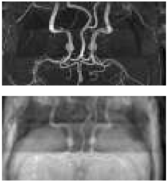
output
(11 patches)



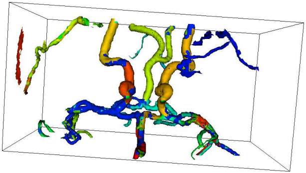
Canny

Experimental results of edge detection

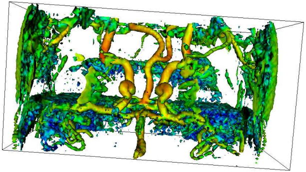
Real MRI image:



input



output
(17 patches)



isosurface

Pre and post-processing of the edge detector

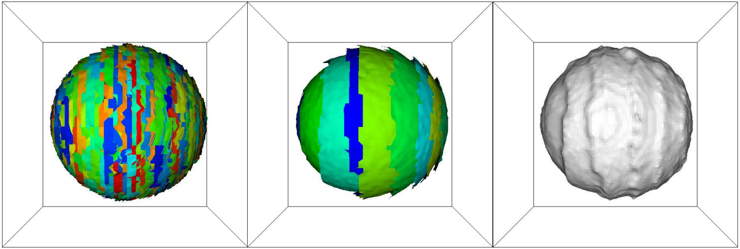
The performance of the method can be improved in several ways

Pre-processing:

- Grain filter (reduces the number of small shapes)
- Median filtering (reduces the complexity of noisy shapes)
- Quantization (reduces the number of nested shapes)
- Canny instead of gradient norm (improves localization)

Post-processing:

- Exclusion principle (allow only one output patch per voxel)
- Surface joining (join overlapping patches into one)

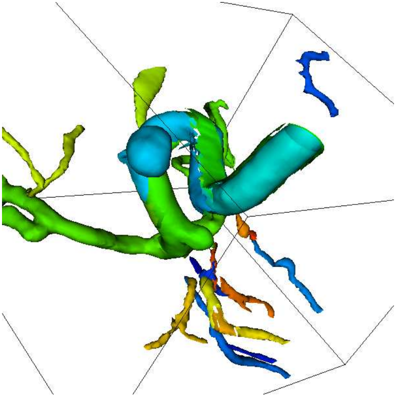


Surface joining (or 3D edge linking)

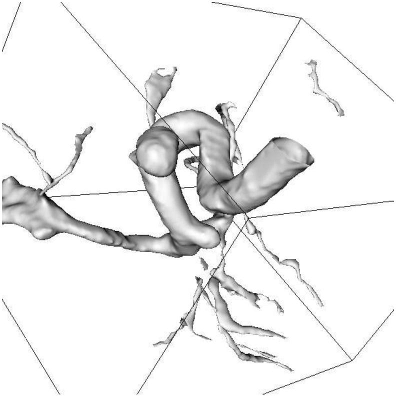
Some functionals for edge linking data S with model Γ:

- Osher-Zhao: $E_0(\Gamma) = \int_\Gamma d_S$
- "reverse Osher-Zhao": $E_1(\Gamma) = \int_S |d_\Gamma|$
- Adjustment of distance functions: $E_2(\Gamma) = \int_{\mathbf{R}^3} |d_S - |d_\Gamma||$
- Combinations of the above ($E_0 + E_1$, etc.)

Surface joining (Osher-Zhao)

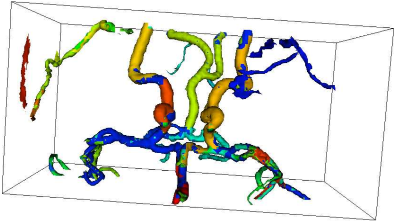


input patches

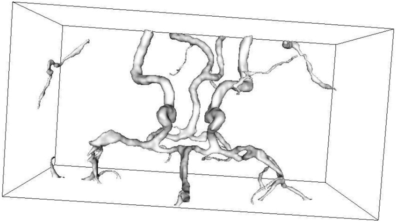


joined surface

Surface joining (Osher-Zhao)



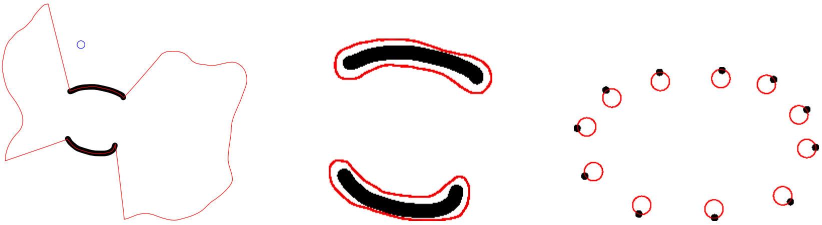
input patches



joined surface

Trivial global minima

Problem: all the functionals above have trivial global minima



Solution: adding an "inflating force" to the functional may improve the situation.

$$E(\Gamma) = E_i(\Gamma) - \lambda \cdot \mathrm{volume}(\Gamma)$$

A happy connection

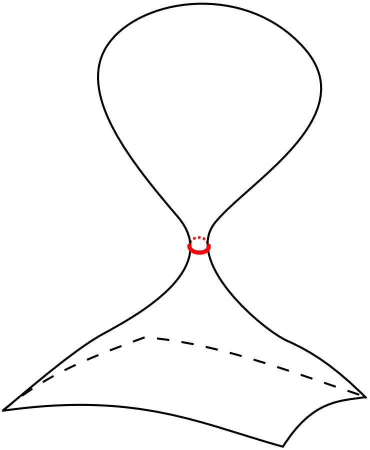
Edge linking with an inflating force can be formulated in terms of anisotropic Cheeger sets.

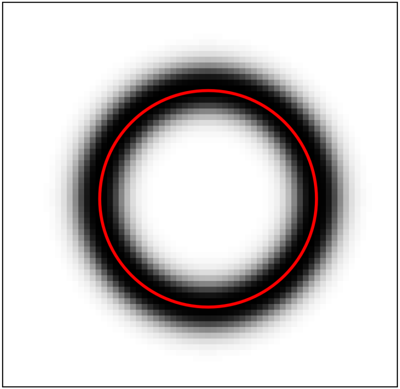
Let us forget about edge linking, and focus on Finsler-Cheeger sets.

Outline

Finsler metrics in image processing

Finsler (or Riemannian) metrics are widely used in image processing, and elsewhere.





geodesic active contours, anisotropic diffusion...

Euclidean Cheeger sets

- Cheeger constant of Ω: $\quad C_\Omega := \min_{A \subseteq \Omega} \frac{|\partial A|}{|A|}$
- Cheeger set of Ω: set A where the minimum is attained
- Equivalent definition: minimizer of $|\partial A| - C_\Omega |A|$

Images from http://www.ceremade.dauphine.fr/˜peyre/cheeger/

let us move to Finsler spaces

Finsler-Cheeger sets

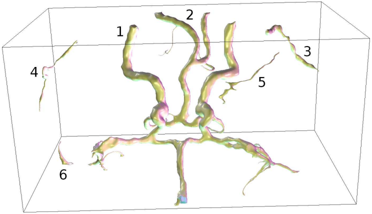
Let us weigh perimeters by φ and volumes by h.

- Cheeger constant of Ω: $C_\Omega := \min_{A \subseteq \Omega} \frac{|\partial A|_\varphi}{|A|_h}$
- Cheeger set of Ω: set A where the minimum is attained
- Equivalent definition: minimizer of $|\partial A|_\varphi - C_\Omega |A|_h$

Finsler-Cheeger sets

Let us weigh perimeters by φ and volumes by h.

- Cheeger constant of Ω: $C_\Omega := \min_{A \subseteq \Omega} \frac{|\partial A|_\varphi}{|A|_h}$
- Cheeger set of Ω: set A where the minimum is attained
- Equivalent definition: minimizer of $|\partial A|_\varphi - C_\Omega |A|_h$



How to compute these objects?

Cheeger sets and total variation

Euclidean case:

- Variational problem:
  $$E_\lambda(u) := \int_\Omega |Du| + \frac{\lambda}{2}\int_\Omega (u-1)^2 + \int_{\partial\Omega} |u|$$
- Cheeger sets arise as level sets of the solution: $[u \geq |u|_\infty]$

Finslerian case:

- Variational problem:
  $$E_{\varphi,h,\lambda}(u) := \int_\Omega |Du|_\varphi + \frac{\lambda}{2}\int_\Omega (u-1)^2 h + \int_{\partial\Omega} \varphi(x,\nu^\Omega)|u|$$
- φ-Cheeger sets arise as level sets of the solution: $[u \geq |u|_\infty]$

Numerical solution of the variational problem

- Euclidean case: Chambolle algorithm

$$p^{t+1} = \frac{p^n + \tau \nabla \mathrm{div}(p^n)}{1 + \tau |\nabla \mathrm{div}(p^n)|}$$

- Finsler case: Chambolle algorithm with suitable weights

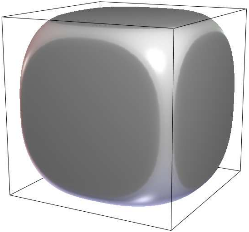
$$p^{t+1} = \frac{p^n + \tau A_x \cdot \nabla \left(h^{-1} \mathrm{div}(A_x \cdot p^n) \right)}{1 + \tau \left| A_x \cdot \nabla \left(h^{-1} \mathrm{div}(A_x \cdot p^n) \right) \right|}$$

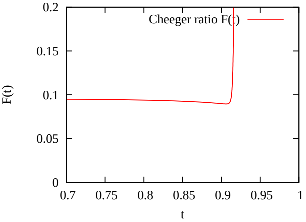
Once these iterations converge, the Cheeger set is found among the level sets of the resulting image $\mathrm{div}(p)$.

Numerical computation of the Cheeger set

We recover our old friend, the tree of shapes.

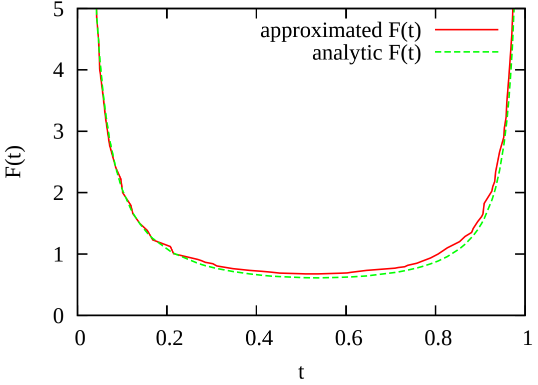
- The Cheeger ratio is defined for every shape
- Find a minimum of the Cheeger ratio defined on the tree of shapes (which is a topological space)

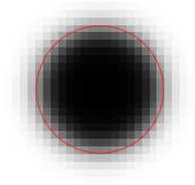




Euclidean Cheeger set of a cube

Computation of Finsler-Cheeger ratio





Finsler-Cheeger ratio numeric/analytic

Applications of Finsler-Cheeger sets

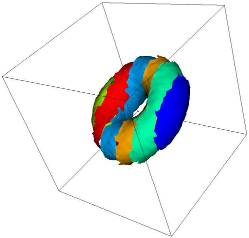
By fine-tuning the Finsler metric, we can make the Cheeger sets to have any shape that we want.

(For example, by letting the metric vanish on a given curve)

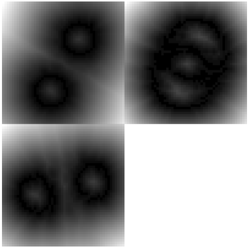
Application to edge linking

The φ-Cheeger is a global optimum of Osher-Zhao with an inflating force

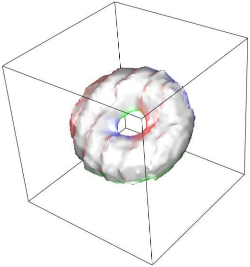
"Distance Cheegers": $\varphi =$ distance to data



S



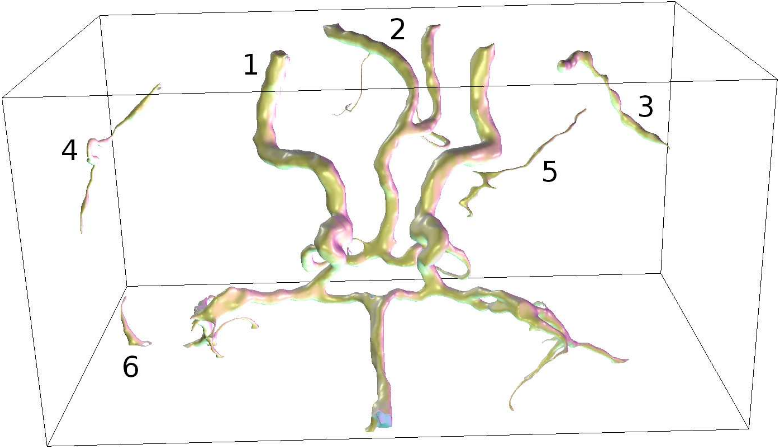
slices of d_S



d_S-Cheeger

Application to edge linking

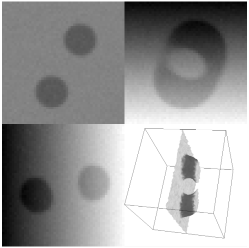
d_S-Cheeger set of the whole image domain



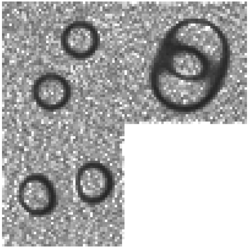
Application to active contours

The φ-Cheeger is a global optimum of active contours with inflating force

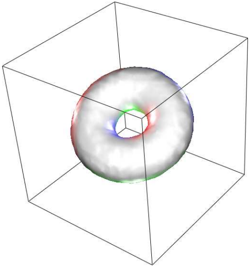
"Gradient Cheegers": $\varphi = \frac{1}{|\nabla I|}$



slices of I



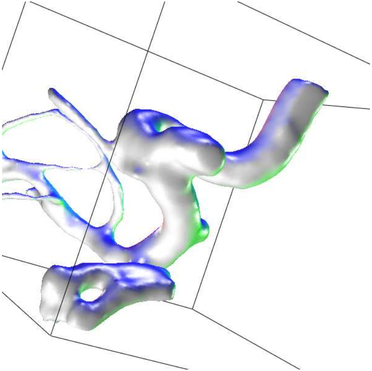
slices of $\frac{1}{|\nabla I|}$



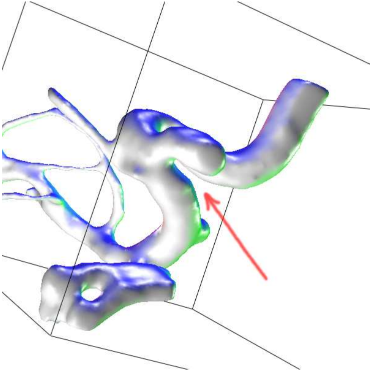
$\frac{1}{|\nabla I|}$-Cheeger set

Application to active contours

The φ-Cheeger is a global optimum of active contours with inflating force



$\frac{1}{|\nabla u|}$-Cheeger set of the whole image domain



$\frac{1}{|\nabla u|}$-Cheeger set of the image domain minus some manually selected voxels

Finsler-Cheeger sets and MSER

There is a connection between $\frac{1}{|\nabla u|}$-Cheegers and MSER.

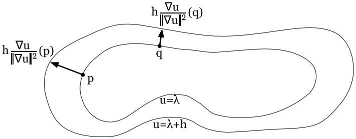
MSER [Matas *et al.* '04] are defined as minimizers of

$$F(\lambda) := \frac{|[u < \lambda + \delta]| - |[u < \lambda - \delta]|}{|[u < \lambda]|}$$

By co-area formula,

$$F(\lambda) = 2\delta \frac{\int_{\{u=\lambda'\}} \frac{1}{|\nabla u|} d\mathcal{H}^{N-1}}{|[u < \lambda]|}$$

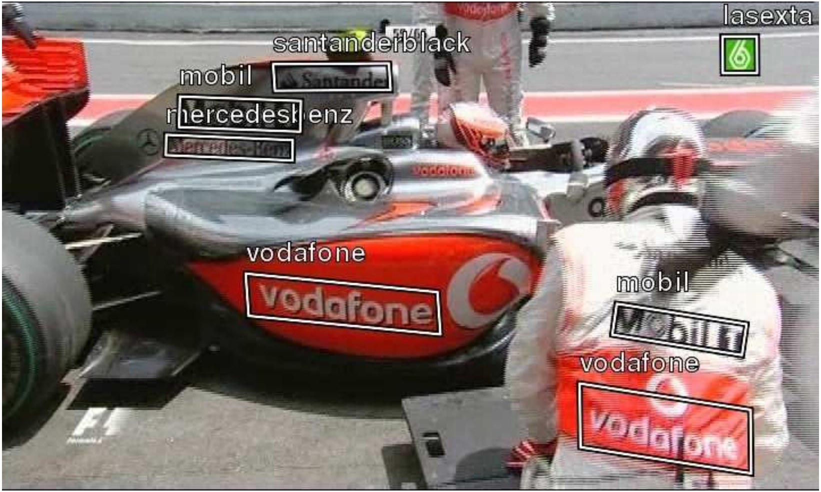
Which is almost exactly the $\frac{1}{|\nabla u|}$-Cheeger ratio!



MSER can be regarded as a crude approximation to (local) $\frac{1}{|\nabla u|}$*-Cheegers*

Application to logo detection

Match MSER and other features between a logo database and a video frame.



Outline

Contributions of this thesis

- Algorithm for 3D tree of shapes
- Basic applications (grain filters, visualization, RGB histograms, video analysis)

- 3D edge detection (exclusion principle, MS hierarchies, consistent Marching Cubes)

- Numerical computation of Finsler-Cheeger sets
- Application of Finsler-Cheeger sets to edge linking
- MSER are local Finsler-Cheeger sets

Future work

- User interface for tree of shapes navigation and editing
- Joining the trees of level sets defined on a graph
- Quantitative comparison of edge detectors
- Quantitative comparison of surface joiners
- Development of object detection based on the new affine invariants
- Faster schemes for Finsler-Cheeger sets

Publications

Published:

- **Constructing the Tree of Shapes of an Image by Fusion of the Trees of Connected Components of Upper and Lower Level Sets**
  V. Caselles, E. Meinhardt and P. Monasse
  Positivity, Vol. 12, Num. 1, pp. 55–73, 2008
- **Edge Detection by Selection of Pieces of Level Lines**
  E. Meinhardt
  International Conference on Image Processing 2008
- **3D Edge Detection By Selection of Level Surface Patches**
  E. Meinhardt, E. Zacur, A.F. Frangi and V. Caselles
  Journal of Mathematical Imaging and Vision,
  Vol. 32, Num. 1, pp. 1–16, 2009
- **Anisotropic Cheeger Sets and Applications**
  V. Caselles, G. Facciolo and E. Meinhardt
  SIAM Journal on Imaging Sciences,
  Vol. 2, Num. 4, pp. 1211–1254, 2009
- **A Robust Pipeline for Logo Detection**
  E. Meinhardt, C. Constantinopoulos, V. Caselles
  International Conference on Multimedia and Expo 2011

Under review:

- **Relative Depth Estimation from Monocular Video**
  E. Meinhardt, V .Caselles
  Submitted to *International Conference on Image Processing 2011*
- **On Affine Invariant Descriptors Related to SIFT**
  R. Sadek, C. Constantinopoulos, E. Meinhardt, C. Ballester, V. Caselles
  Submitted to *SIAM Journal on Imaging Sciences* on 2010