Graph-based variational optimization

and applications in computer vision

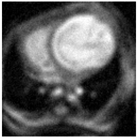
Camille Couprie, joint work with
Laurent Najman*, Hugues Talbot* and Leo Grady+.
*LIGM, Université Paris Est, ESIEE, France
+Siemens Corporate Research, Princeton, USA

20 October 2011

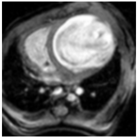
Introduction

Image Segmentation

Image restoration



→ ?

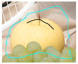


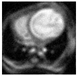
Stereo-vision reconstruction

→ ?

Regularity hypothesis

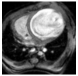
Classical formulation for solving our problems :





data f

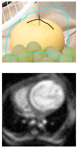
$$\rightarrow \boxed{\min_x \underbrace{R(x)}_{\text{Regularization}} + \underbrace{\mathcal{D}(x,f)}_{\text{Data fidelity}}} \rightarrow$$



solution x

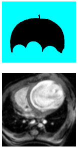
Regularity hypothesis

Classical formulation for solving our problems :



data f

$$\rightarrow \boxed{\min_x \int_\Omega \underbrace{||\nabla x(z)|| \ \ dz}_{\text{Regularization}} + \underbrace{\mathcal{D}(x, f)}_{\text{Data fidelity}}} \rightarrow$$

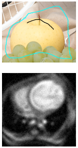


solution x

Example : minimization of the total variation

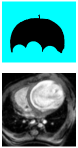
Regularity hypothesis

Classical formulation for solving our problems :



data f

$$\rightarrow \boxed{\min_x \int_\Omega \underbrace{||\nabla x(z)|| \ \ dz}_{\text{Regularization}} + \underbrace{\mathcal{D}(x, f)}_{\text{Data fidelity}}} \rightarrow$$



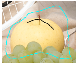
solution x

Example : minimization of the total variation

The solution x may be a labeling :

Regularity hypothesis

Classical formulation for solving our problems :



data f

$$\rightarrow \boxed{\min_x \int_\Omega \underbrace{||\nabla x(z)|| \quad dz}_{\text{Regularization}} + \underbrace{\mathcal{D}(x, f)}_{\text{Data fidelity}}} \rightarrow$$

solution x

Example : minimization of the total variation

The solution x may be a labeling :

- partitionning an image in different regions

Regularity hypothesis

Classical formulation for solving our problems :

data f

$$\rightarrow \boxed{\min_x \int_\Omega \underbrace{||\nabla x(z)|| \quad dz}_{\text{Regularization}} + \underbrace{\mathcal{D}(x, f)}_{\text{Data fidelity}}} \rightarrow$$

solution x

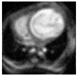
Example : minimization of the total variation

The solution x may be a labeling :

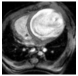
- partitionning an image in different regions
- estimating a depth map for stereo-vision reconstruction

Regularity hypothesis

Classical formulation for solving our problems :



data f $\rightarrow$ $$\boxed{\min_x \int_\Omega \underbrace{||\nabla x(z)|| \quad dz}_{\text{Regularization}} + \underbrace{||x - f||_2^2}_{\text{Data fidelity}}}$$ $\rightarrow$



solution x

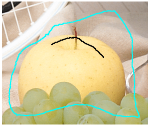
Example : minimization of the total variation

The solution x may be a labeling :

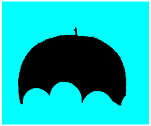
- partitionning an image in different regions
- estimating a depth map for stereo-vision reconstruction
- restored intensities of an image f [ROF model, 1992]

Genericity of graph-based methods

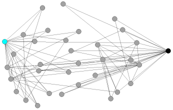
Seeded segmentation



→ ?



Classification



→ ?

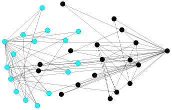
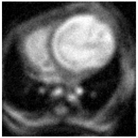
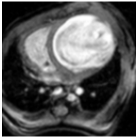


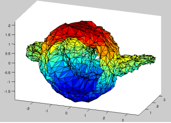
Image restoration



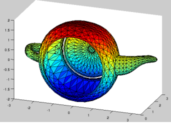
→ ?



Mesh denoising



→ ?



Outline

- **I - Standard graph-based methods**
- **II - Flow based methods**
  1. Segmentation : Combinatorial Continuous Maximum Flow
  2. Restoration : Dual constrained TV-based regularization
- **III - Power watershed**
  1. A new graph-based optimization framework
  2. Image segmentation
  3. Image filtering (nonconvex optimization)
  4. Surface reconstruction
- **IV - Conclusion**

Outline

- **I - Standard graph-based methods**

Some graph-based segmentation tools

- Watershed [Beucher-Lantuéjoul 1979, Vincent-Soille 1991]

Advantages

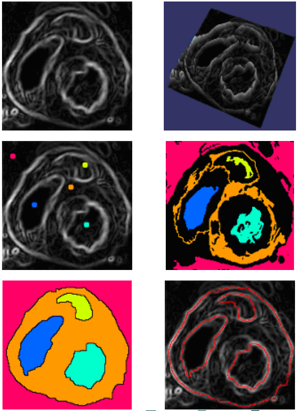
- Fast

Some graph-based segmentation tools

- Watershed [Beucher-Lantuéjoul 1979, Vincent-Soille 1991]

Advantages

- Fast
- Multilabel

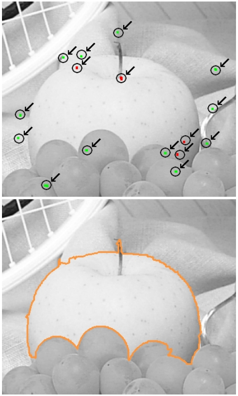


Some graph-based segmentation tools

- Watershed [Beucher-Lantuéjoul 1979, Vincent-Soille 1991]

Advantages

- Fast
- Multilabel
- Robust to markers size



Some graph-based segmentation tools

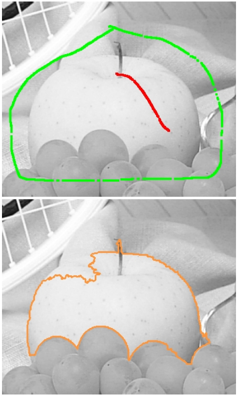
- Watershed [Beucher-Lantuéjoul 1979, Vincent-Soille 1991]

Advantages

- Fast
- Multilabel
- Robust to markers size

Drawbacks

- Leaking effect



Some graph-based segmentation tools

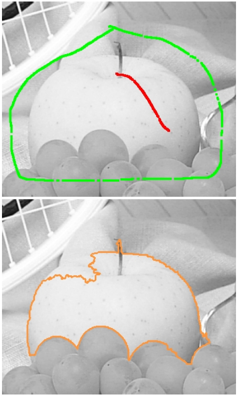
- Watershed [Beucher-Lantuéjoul 1979, Vincent-Soille 1991]

Advantages

- Fast
- Multilabel
- Robust to markers size

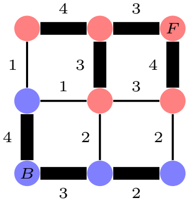
Drawbacks

- Leaking effect
- Non unique solution (difficult to get a non algorithmically dependent result)



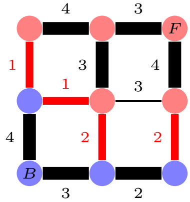
Watershed and Maximum Spanning Forest equivalence

- MSF : set of trees
  - spanning all nodes
  - not connecting different seeds
  - such that the total sum of their weights is maximum.
- If seeds are the maxima of the weight function, every MSF cut on the weight function is a watershed cut [Cousty *et al* 07, the drop of water principle]

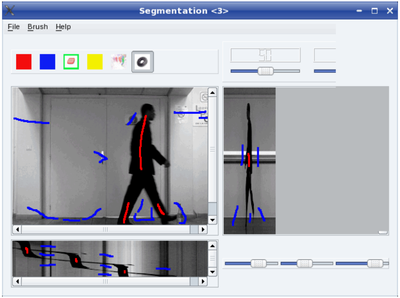


Watershed and Maximum Spanning Forest equivalence

- MSF : set of trees
  - spanning all nodes
  - not connecting different seeds
  - such that the total sum of their weights is maximum.
- If seeds are the maxima of the weight function, every MSF cut on the weight function is a watershed cut [Cousty *et al* 07, the drop of water principle]



Max Spanning Forest (Watershed)



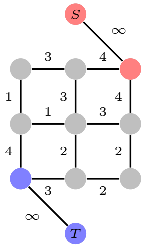
Some graph-based segmentation tools : Graph Cuts

- Graph cuts / Max flow

Advantages

- Energy formulation → extends to a large class of problems

[Ford & Fulkerson 60s, Boykov-Joly 1998]



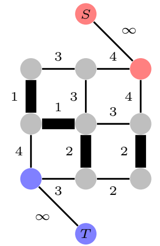
Some graph-based segmentation tools : Graph Cuts

- Graph cuts / Max flow

Advantages

- Energy formulation → extends to a large class of problems

[Ford & Fulkerson 60s, Boykov-Joly 1998]



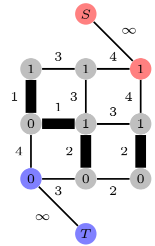
Some graph-based segmentation tools : Graph Cuts

- Graph cuts / Max flow

Advantages

- Energy formulation → extends to a large class of problems

[Ford & Fulkerson 60s, Boykov-Joly 1998]



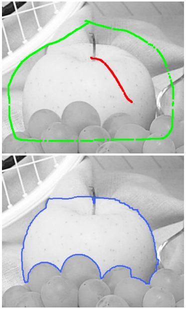
Some graph-based segmentation tools : Graph Cuts

- Graph cuts / Max flow

Advantages

- Energy formulation → extends to a large class of problems
- Robust to markers placement

[Ford & Fulkerson 60s, Boykov-Joly 1998]



Some graph-based segmentation tools : Graph Cuts

- Graph cuts / Max flow

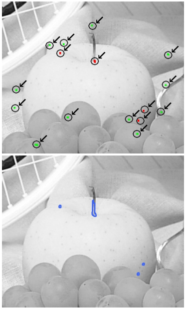
Advantages

- Energy formulation → extends to a large class of problems
- Robust to markers placement

Drawbacks

- Bias toward small contours

[Ford & Fulkerson 60s, Boykov-Joly 1998]



Some graph-based segmentation tools : Graph Cuts

- Graph cuts / Max flow

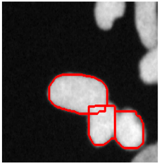
Advantages

- Energy formulation → extends to a large class of problems
- Robust to markers placement

Drawbacks

- Bias toward small contours
- Block artifacts

[Ford & Fulkerson 60s, Boykov-Joly 1998]



Some graph-based segmentation tools : Graph Cuts

- Graph cuts / Max flow

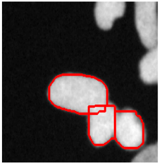
Advantages

- Energy formulation → extends to a large class of problems
- Robust to markers placement

Drawbacks

- Bias toward small contours
- Block artifacts
- Super-linear complexity

[Ford & Fulkerson 60s, Boykov-Joly 1998]



Some graph-based segmentation tools : Graph Cuts

- Graph cuts / Max flow

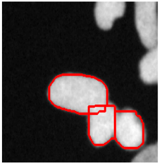
Advantages

- Energy formulation $\rightarrow$ extends to a large class of problems
- Robust to markers placement

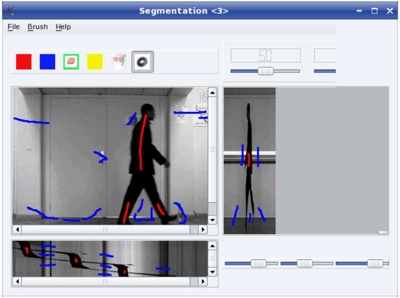
Drawbacks

- Bias toward small contours
- Block artifacts
- Super-linear complexity
- Limited to binary (2 labels) segmentation

[Ford & Fulkerson 60s, Boykov-Joly 1998]

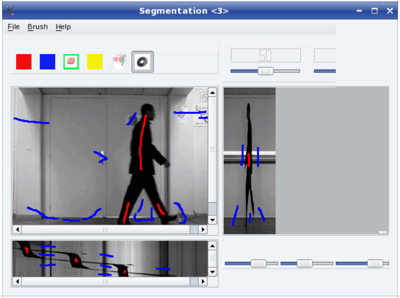


Some graph-based segmentation tools : Graph cuts



Graph cuts segmentation

Some graph-based segmentation tools : Graph cuts

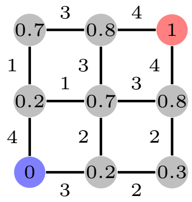


Graph cuts segmentation

Some graph-based segmentation tools : Random Walker

- Combinatorial Dirichlet problem. Seeded segmentation [Grady 2006]
- Resolution of system of linear equations.

Advantages

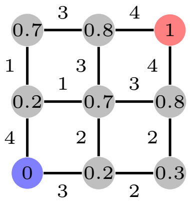


Some graph-based segmentation tools : Random Walker

- Combinatorial Dirichlet problem. Seeded segmentation [Grady 2006]
- Resolution of system of linear equations.

Advantages

- Energy formulation → extends to a large class of problems

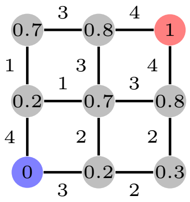


Some graph-based segmentation tools : Random Walker

- Combinatorial Dirichlet problem. Seeded segmentation [Grady 2006]
- Resolution of system of linear equations.

Advantages

- Energy formulation → extends to a large class of problems
- No blocking artefacts



Some graph-based segmentation tools : Random Walker

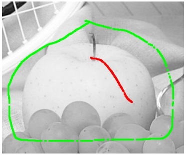
- Combinatorial Dirichlet problem. Seeded segmentation [Grady 2006]
- Resolution of system of linear equations.

Advantages

- Energy formulation → extends to a large class of problems
- No blocking artefacts

Drawbacks

- Requires a more centered markers placement



Some graph-based segmentation tools : Random Walker

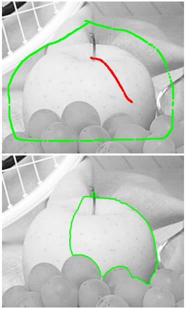
- Combinatorial Dirichlet problem. Seeded segmentation [Grady 2006]
- Resolution of system of linear equations.

Advantages

- Energy formulation → extends to a large class of problems
- No blocking artefacts

Drawbacks

- Requires a more centered markers placement



Some graph-based segmentation tools : Random Walker

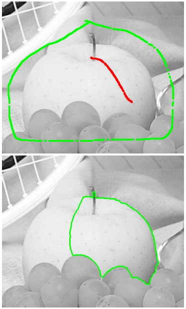
- Combinatorial Dirichlet problem. Seeded segmentation [Grady 2006]
- Resolution of system of linear equations.

Advantages

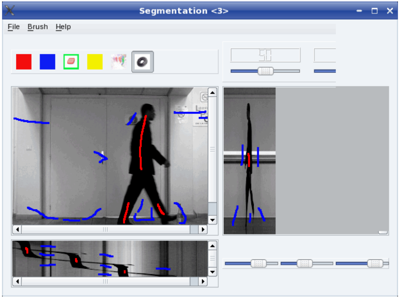
- Energy formulation → extends to a large class of problems
- No blocking artefacts

Drawbacks

- Requires a more centered markers placement
- Super-linear complexity



Some graph-based segmentation tools : Random Walker



Random Walker segmentation

Outline

- **I - Standard graph-based methods**
- **II - Flow based methods**
  1. Segmentation : Combinatorial Continuous Maximum Flow
  2. Restoration : Dual constrained TV-based regularization
- **III - Power watershed**
  1. A new graph-based optimization framework
  2. Image segmentation
  3. Image filtering (nonconvex optimization)
  4. Surface reconstruction
- **IV - Conclusion**

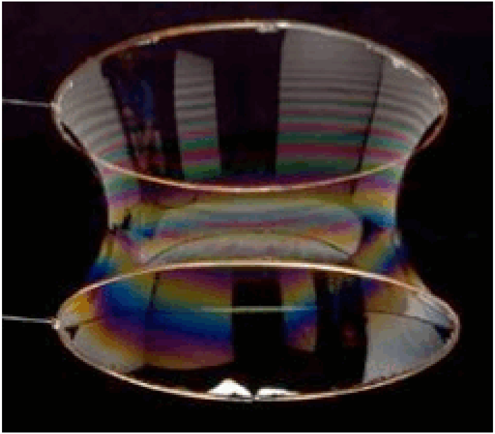
Outline

- **II - Flow based methods**
  1. Segmentation : Combinatorial Continuous Maximum Flow
  2. Restoration : Dual constrained TV-based regularization

Outline

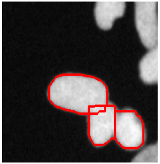
- **II - Flow based methods**
  1. Segmentation : Combinatorial Continuous Maximum Flow
  2. Restoration : Dual constrained TV-based regularization

Minimal surfaces

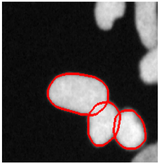


Motivation

- In the continuum : Minimal cut (surface in 3D) is dual of continuous maximum flow [Strang 1983]
- In the classic discrete case min-cut (= "Graph cuts")/ max flow duality but grid bias in the solution
- Recent trend : employ a spatially *continuous* maximum flow to produce solutions with no grid bias



Max Flow (Graph Cuts)



Continuous Max Flow [Appleton-Talbot 2006]

Motivation

- [Appleton-Talbot 2006, generalized by Unger-Pock-Bishof 2008] Fastest known continuous max-flow algorithm has **no stopping criteria** and **no converge proof**.

Our contribution : Combinatorial Continuous Maximum Flow

- a new discrete isotropic formulation
- **avoids blockiness artifacts**
- is **proved to converge**, is **fast**
- **generalizes to arbitrary graphs**

[In SIAM Journal on Imaging Sciences, 2011]

Combinatorial Continuous Maximum Flow (CCMF)

- Incidence matrix of a graph noted A

Continuous MaxFlow

$$\max_{\vec{F}} \quad \vec{F}_{st}$$
$$\text{s.t.} \quad \nabla \cdot \vec{F} = 0, \quad ||\vec{F}|| \leq g.$$

Combinatorial formulation

$$\max_{F} \quad F_{st}$$
$$\text{s.t.} \quad A^T F = 0, \quad |A^T| F^2 \leq g^2$$

g defined on nodes

MaxFlow, GraphCuts

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

g defined on edges

- CCMF : convex problem
- Resolution by an interior point method.

Combinatorial Continuous Maximum Flow (CCMF)

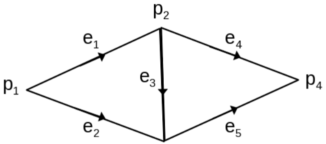
- Incidence matrix of a graph noted A

Continuous MaxFlow	Combinatorial formulation	MaxFlow, GraphCuts
$\max_{\vec{F}} \; \vec{F}_{st}$ s.t. $\nabla \cdot \vec{F} = 0,$ $\|\vec{F}\| \leq g.$	$\max_{F} \; F_{st}$ s.t. $A^T F = 0,$ $\lvert A^T \rvert F^2 \leq g^2$	$\max_{F} \; F_{st}$ s.t. $A^T F = 0,$ $\lvert F \rvert \leq g$
	g defined on nodes	g defined on edges

- CCMF : convex problem
- Resolution by an interior point method.

Discrete formulation on graphs - notations

Graph of N vertices, M edges



Incidence matrix $A \in \mathbb{R}^{M \times N}$

$$A = \begin{array}{c|cccc|} & p_1 & p_2 & p_3 & p_4 \\ \hline e_1 & -1 & 1 & 0 & 0 \\ e_2 & -1 & 0 & 1 & 0 \\ e_3 & 0 & -1 & 1 & 0 \\ e_4 & 0 & -1 & 0 & 1 \\ e_4 & 0 & 0 & -1 & 1 \end{array}$$

- A gradient operator
- $A^\top$ divergence operator
- allows general formulation of problems on arbitrary graphs

Combinatorial Continuous Maximum Flow (CCMF)

- Incidence matrix of a graph noted A

Continuous MaxFlow	Combinatorial formulation	MaxFlow, GraphCuts
$$\max_{\vec{F}} \quad F_{st}$$ $$\text{s.t.} \quad \nabla \cdot \vec{F} = 0, \quad \|\vec{F}\| \leq g.$$	$$\max_{F} \quad F_{st}$$ $$\text{s.t.} \quad A^T F = 0, \quad \|A^T\|F^2 \leq g^2$$ g defined on nodes	$$\max_{F} \quad F_{st}$$ $$\text{s.t.} \quad A^T F = 0, \quad \|F\| \leq g$$ g defined on edges

- CCMF : convex problem
- Resolution by an interior point method.

Combinatorial Continuous Maximum Flow (CCMF)

- Incidence matrix of a graph noted A

Continuous MaxFlow	Combinatorial formulation	MaxFlow, GraphCuts
$\max_{\vec{F}} \quad F_{st}$ s.t. $\nabla \cdot \vec{F} = 0,$ $\|\vec{F}\| \leq g.$	$\max_{F} \quad F_{st}$ s.t. $A^T F = 0,$ $\|A^T\| F^2 \leq g^2$ g defined on nodes	$\max_{F} \quad F_{st}$ s.t. $A^T F = 0,$ $\|F\| \leq g$ g defined on edges

- CCMF : convex problem
- Resolution by an interior point method.

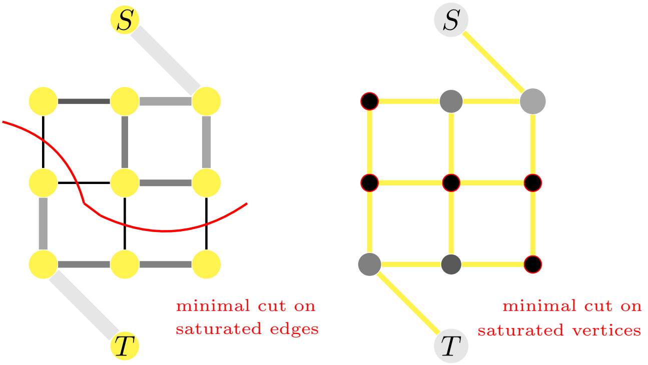
Combinatorial Continuous Maximum Flow (CCMF)

- Incidence matrix of a graph noted A

Continuous MaxFlow	Combinatorial formulation	MaxFlow, GraphCuts
$\max_{\vec{F}} \; F_{st}$ s.t. $\nabla \cdot \vec{F} = 0$, $\|\vec{F}\| \leq g$.	$\max_{F} \; F_{st}$ s.t. $A^T F = 0$, $\lvert A^T\rvert F^2 \leq g^2$	$\max_{F} \; F_{st}$ s.t. $A^T F = 0$, $\lvert F\rvert \leq g$
	g defined on nodes	g defined on edges

- CCMF : convex problem
- Resolution by an interior point method.

Graph Cuts vs CCMF



Scale of weight intensity : 1 ... ∞

CCMF dual problem

- The dual of the CCMF problem is

$$\min_{\lambda\geq 0,\nu} \sum_{v_i\in V} \underbrace{\lambda_i g_i^2}_{\text{weighted cut}} + \underbrace{\frac{1}{4}\sum_{e_{ij}\in E\setminus\{s,t\}} \frac{(\nu_i-\nu_j)^2}{\lambda_i+\lambda_j}}_{\text{smoothness term}} + \underbrace{\frac{1}{4}\, \frac{(\nu_s-\nu_t-1)^2}{\lambda_s+\lambda_t}}_{\text{source-sink enforcement}}$$

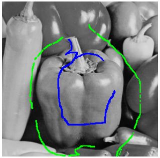


Image with seeds


λ

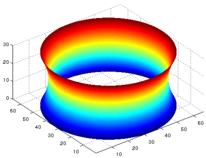

ν


Threshold of ν at .5

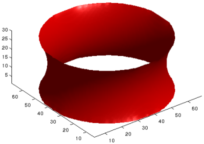
Minimal surfaces

Catenoid test problem :

- source constituted by two full circles
- sink by the remaining boundary of the image, constant metric g



analytic minimal surface

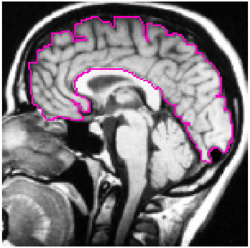


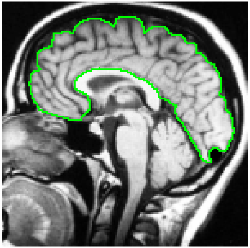
CCMF result isosurface of ν

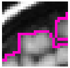
Root Mean Square Error between the surfaces : 0.75

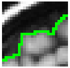
(Appleton-Talbot error : 1.98)

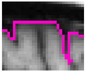
Comparison with Graph cuts

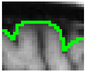

Graph cuts result

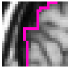

CCMF result

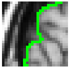

GC


CCMF

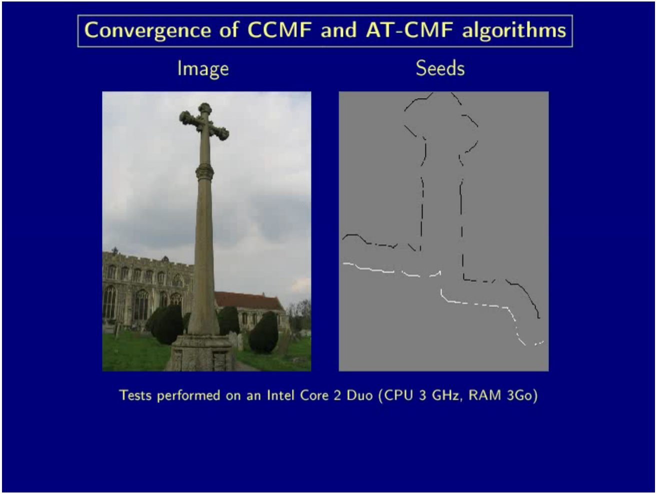

GC


CCMF

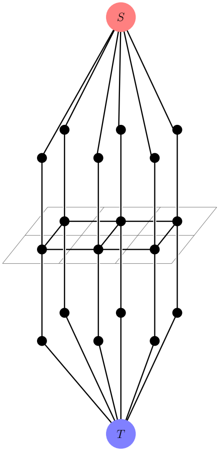

GC

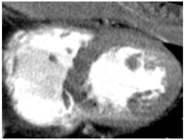

CCMF

Comparison with Appleton-Talbot method

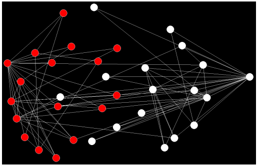


Genericity of the method



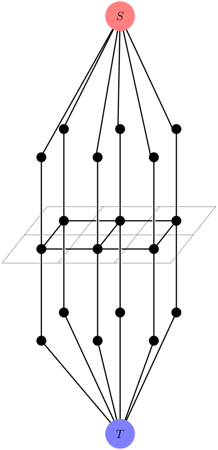


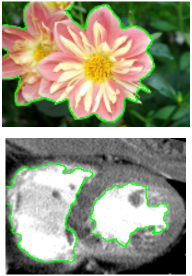
Unseeded segmentation



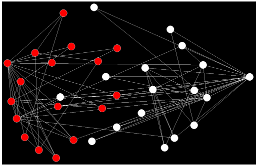
Classification

Genericity of the method





Unseeded segmentation



Classification

Dual constrained TV based formulation

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

- $f \in \mathbb{R}^Q$ observed image
- $x \in \mathbb{R}^N$ restored image
- $F \in \mathbb{R}^M$ flow, projection vector
- $H \in \mathbb{R}^{Q \times N}$ linear operator (e.g. degradation matrix)

Joint work with Jean-Christophe Pesquet

[Couprie-Talbot-Pesquet-Najman-Grady, ICASSP 2011]

Dual constrained TV based formulation

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

- $f \in \mathbb{R}^Q$ observed image
- $x \in \mathbb{R}^N$ restored image
- $F \in \mathbb{R}^M$ flow, projection vector
- $H \in \mathbb{R}^{Q \times N}$ linear operator (e.g. degradation matrix)

Joint work with Jean-Christophe Pesquet
[Couprie-Talbot-Pesquet-Najman-Grady, ICASSP 2011]

Dual constrained TV based formulation

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

- $f \in \mathbb{R}^Q$ observed image
- $x \in \mathbb{R}^N$ restored image
- $F \in \mathbb{R}^M$ flow, projection vector
- $H \in \mathbb{R}^{Q \times N}$ linear operator (e.g. degradation matrix)

Joint work with Jean-Christophe Pesquet
[Couprie-Talbot-Pesquet-Najman-Grady, ICASSP 2011]

Dual constrained TV based formulation

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

- $f \in \mathbb{R}^Q$ observed image
- $x \in \mathbb{R}^N$ restored image
- $F \in \mathbb{R}^M$ flow, projection vector
- $H \in \mathbb{R}^{Q \times N}$ linear operator (e.g. degradation matrix)

Joint work with Jean-Christophe Pesquet
[Couprie-Talbot-Pesquet-Najman-Grady, ICASSP 2011]

Dual constrained TV based formulation

$$\min_{x} \underbrace{\max_{\mathbf{F} \in \mathbf{C}} \quad F^{\top}(Ax)}_{\text{regularization}} + \underbrace{\frac{1}{2\lambda} \| Hx - f \|_2^2}_{\text{data fidelity}}$$

- $f \in \mathbb{R}^Q$ observed image
- $x \in \mathbb{R}^N$ restored image
- $F \in \mathbb{R}^M$ flow, projection vector
- $H \in \mathbb{R}^{Q \times N}$ linear operator (e.g. degradation matrix)

Joint work with Jean-Christophe Pesquet
[Couprie-Talbot-Pesquet-Najman-Grady, ICASSP 2011]

Dual constrained TV based formulation

$$\min_{x} \underbrace{\max_{\mathbf{F} \in \mathbf{C}} \quad F^\top(Ax)}_{\text{regularization}} + \underbrace{\frac{1}{2\lambda}\|Hx - f\|_2^2}_{\text{data fidelity}}$$

- $f \in \mathbb{R}^Q$ observed image
- $x \in \mathbb{R}^N$ restored image
- $F \in \mathbb{R}^M$ flow, projection vector
- $H \in \mathbb{R}^{Q \times N}$ linear operator (e.g. degradation matrix)
- Combinatorial variant of TV with flexible choice for C

Joint work with Jean-Christophe Pesquet
[Couprie-Talbot-Pesquet-Najman-Grady, ICASSP 2011]

Dual constrained TV based formulation

$$\min_{x} \underbrace{\max_{\mathbf{F} \in \mathbf{C}} \quad F^{\top}(Ax)}_{\text{regularization}} + \underbrace{\frac{1}{2\lambda} \|Hx - f\|_2^2}_{\text{data fidelity}}$$

- $f \in \mathbb{R}^Q$ observed image
- $x \in \mathbb{R}^N$ restored image
- $F \in \mathbb{R}^M$ flow, projection vector
- $H \in \mathbb{R}^{Q \times N}$ linear operator (e.g. degradation matrix)
- Combinatorial variant of TV with flexible choice for C
- $C = \cap_{i=1}^{s} C_i$ decomposed in an intersection of convex sets

Joint work with Jean-Christophe Pesquet
[Couprie-Talbot-Pesquet-Najman-Grady, ICASSP 2011]

Dual problem

- Fenchel-Rockafellar dual problem :

$$\min_{F \in \mathbb{R}^M} \sum_{i=1}^{s} f_i(F) + f_{s+1}(F)$$

Dual problem

- Fenchel-Rockafellar dual problem :

$$\min_{F \in \mathbb{R}^M} \sum_{i=1}^{s} \iota_{C_i}(F) + f_{s+1}(F)$$

where ι_C : indicator function of convex C (=0 in C, $+\infty$ outside), $f_{s+1} : F \mapsto \frac{1}{2} F^\top A \Gamma A^\top F - F^\top A \Gamma H^\top f$, and $\Gamma = (H^\top H)^{-1}$.

Dual problem

- Fenchel-Rockafellar dual problem :

$$\min_{F \in \mathbb{R}^M} \sum_{i=1}^{s} \iota_{C_i}(F) + f_{s+1}(F)$$

  where ι_C : indicator function of convex C (=0 in C, $+\infty$ outside), $f_{s+1} : F \mapsto \frac{1}{2} F^\top A \Gamma A^\top F - F^\top A \Gamma H^\top f$, and $\Gamma = (H^\top H)^{-1}$.

- The primal problem admits a unique solution x^*.
- If F^* is a solution to the dual problem, $x^* = \Gamma \left(H^\top f - A^\top F^* \right)$.

Dual problem

- Fenchel-Rockafellar dual problem :

$$\min_{F \in \mathbb{R}^M} \sum_{i=1}^{s} \iota_{C_i}(F) + f_{s+1}(F)$$

where ι_C : indicator function of convex C (=0 in C, $+\infty$ outside), $f_{s+1}: F \mapsto \frac{1}{2} F^\top A \Gamma A^\top F - F^\top A \Gamma H^\top f$, and $\Gamma = (H^\top H)^{-1}$.

- The primal problem admits a unique solution x^*.
- If F^* is a solution to the dual problem, $x^* = \Gamma \left(H^\top f - A^\top F^* \right)$.
- Proximity operator : $\forall y \in \mathbb{R}^N$, $\operatorname{prox}_f y = \arg\min_{u \in \mathbb{R}^N} f(u) + \frac{1}{2}\|u - y\|^2$.

Parallel ProXimal Algorithm (PPXA) optimizing DCTV

[Pesquet, Combettes, 2008], minimize$_F \sum_{i=1}^{s} f_i(F) + f_{s+1}(F)$

$\gamma > 0, \nu \in]0, 2[$.

Repeat until convergence

- For (in parallel) $i = 1, \ldots, s+1$
  - $\pi_i = \operatorname{prox} \gamma f_i(y_i)$
- $z = \frac{2}{s+1}(\pi_1 + \cdots + \pi_{s+1}) - F$
- For (in parallel) $i = 1, \ldots, s+1$
  - $y_i = y_i + \nu(z - \pi_i)$
- $F = F + \frac{\nu}{2}(z - F)$

Parallel ProXimal Algorithm (PPXA) optimizing DCTV

[Pesquet, Combettes, 2008], minimize$_F \sum_{i=1}^{s} f_i(F) + f_{s+1}(F)$

$\gamma > 0, \nu \in]0, 2[$.

Repeat until convergence

$$
\left\lfloor
\begin{array}{l}
\text{For (in parallel) } i = 1, \dots, s+1 \\
\quad \left\lfloor \pi_i = \begin{cases} \mathrm{P}_{C_i}(y_i) & \text{if } i \leq s \\ (\gamma A \Gamma A^\top + I)^{-1}(\gamma A \Gamma H^\top f + y_{s+1}) & \text{otherwise} \end{cases} \right. \\
z = \frac{2}{s+1}(\pi_1 + \dots + \pi_{s+1}) - F \\
\text{For (in parallel) } i = 1, \dots, s+1 \\
\quad \lfloor\; y_i = y_i + \nu(z - \pi_i) \\
F = F + \frac{\nu}{2}(z - F)
\end{array}
\right.
$$

Parallel ProXimal Algorithm (PPXA) optimizing DCTV

[Pesquet, Combettes, 2008], minimize$_F \sum_{i=1}^{s} f_i(F) + f_{s+1}(F)$

$\gamma > 0, \nu \in]0,2[$.

Repeat until convergence

$$
\left\lfloor
\begin{array}{l}
\text{For (in parallel) } i = 1,\dots,s+1 \\
\quad \left\lfloor \pi_i = \begin{cases} \mathrm{P}_{C_i}(y_i) & \text{if } i \le s \\ (\gamma A \Gamma A^\top + I)^{-1}(\gamma A \Gamma H^\top f + y_{s+1}) & \text{otherwise} \end{cases} \right. \\
z = \frac{2}{s+1}(\pi_1 + \cdots + \pi_{s+1}) - F \\
\text{For (in parallel) } i = 1,\dots,s+1 \\
\quad \left\lfloor y_i = y_i + \nu(z - pi_i) \right. \\
F = F + \frac{\nu}{2}(z - F)
\end{array}
\right.
$$

- Simple projections onto hyperspheres

Parallel ProXimal Algorithm (PPXA) optimizing DCTV

[Pesquet, Combettes, 2008], minimize$_F \sum_{i=1}^{s} f_i(F) + f_{s+1}(F)$

$\gamma > 0, \nu \in]0, 2[$.
Repeat until convergence

For (in parallel) $i = 1, \ldots, s+1$

$$\pi_i = \begin{cases} \mathrm{P}_{C_i}(y_i) & \text{if } i \leq s \\ (\gamma A \Gamma A^\top + I)^{-1}(\gamma A \Gamma H^\top f + y_{s+1}) & \text{otherwise} \end{cases}$$

$z = \frac{2}{s+1}(\pi_1 + \cdots + \pi_{s+1}) - F$

For (in parallel) $i = 1, \ldots, s+1$

$y_i = y_i + \nu(z - pi_i)$

$F = F + \frac{\nu}{2}(z - F)$

- Linear system resolution

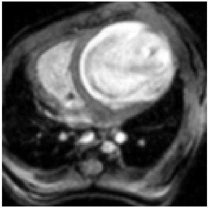
Results

- Applications in data restoration

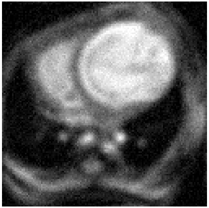
Results

- Applications in data restoration

- Image denoising and deblurring

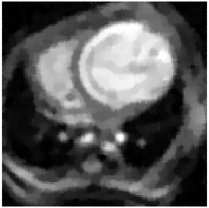


Original image



Noisy, blurry

SNR=24.3dB



DCTV

SNR=27.7dB

Results

- Applications in data restoration

- Image fusion

Original image

Noisy

SNR=17.3dB

blurry

SNR=23.9dB

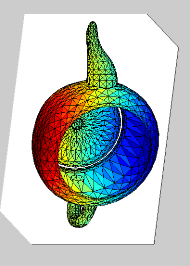
DCTV

SNR=26.5dB

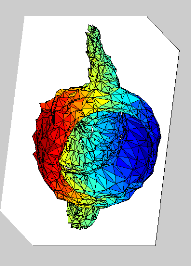
Results

- Applications in data restoration

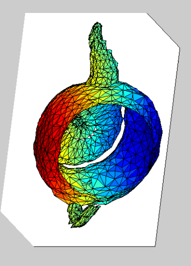
- Mesh denoising



Original mesh



Noisy mesh

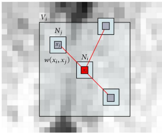


DCTV regularization

Results

- Applications in data restoration

- Image denoising using image patches



Nonlocal graph
Figure from
P. Coupé et al.

Original image

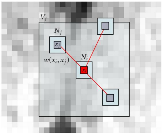
Noisy image
PSNR=28.1dB

Nonlocal DCTV
PSNR=35 dB

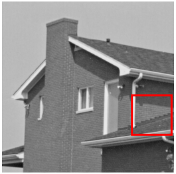
Results

- Applications in data restoration

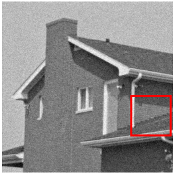
- Image denoising using image patches



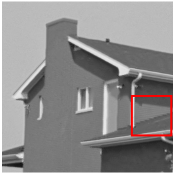
Nonlocal graph
Figure from
P. Coupé et al.



Original image



Noisy image
PSNR=28.1dB

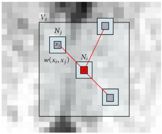


Nonlocal DCTV
PSNR=35 dB

Results

- Applications in data restoration

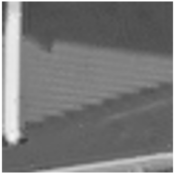
- Image denoising using image patches



Nonlocal graph
Figure from
P. Coupé et al.

Original image

Noisy image
PSNR=28.1dB



Nonlocal DCTV
PSNR=35 dB

Conclusion on the flow-based methods

- Discrete isotropic formulation of the max flow problem that avoids blockiness artifacts
- Guaranteed convergence of the Interior Point method
- Works on arbitrary graphs
- Extension to multi-label problems in a new framework : "dual constrained total variation"

Outline

- **I - Introduction**
- **II - Flow based methods**
  1. Segmentation : Combinatorial Continuous Maximum Flow
  2. Restoration : Dual constrained TV-based regularization
- **III - Power watershed**
  1. A new graph-based optimization framework
  2. Image segmentation
  3. Image filtering
  4. Surface reconstruction
- **IV - Conclusion**

Outline

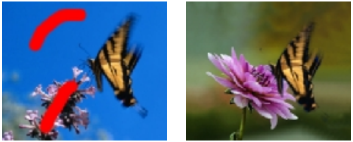
- **III - Power watershed**
  1. A new graph-based optimization framework
  2. Image segmentation
  3. Image filtering
  4. Surface reconstruction
- **IV - Conclusion**

Outline

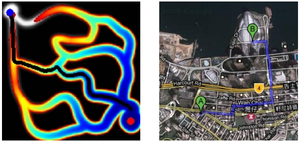
- **III - Power watershed**
  1. A new graph-based optimization framework
  2. Image segmentation
  3. Image filtering
  4. Surface reconstruction
- **IV - Conclusion**

What does all those algorithms have in common ?

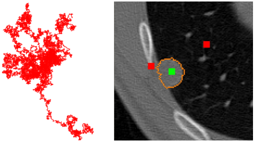
Graph cuts



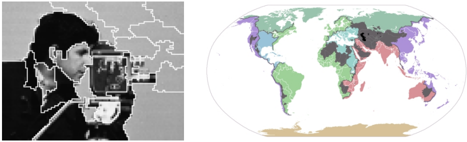
Shortest paths



Random walker

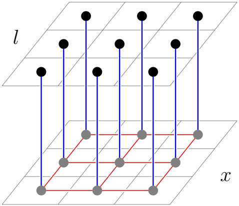


Watersheds



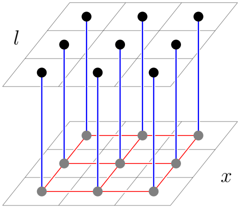
Previously established links

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



Previously established links

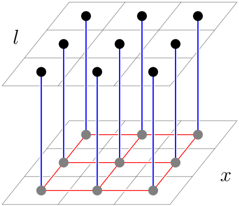
$$\arg\min_x \underbrace{\sum_{e_{ij}\in E} w_{ij} \; |x_i - x_j|}_{\text{Smoothness term}} + \underbrace{\sum_{v_i \in V} w_i \; |x_i - l_i|}_{\text{Data term}}$$



$q = 1$: Graph cuts [Boykov-Joly 2001 (only for 2 labels l)]

Previously established links

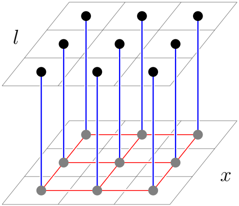
$$\arg\min_x \underbrace{\sum_{e_{ij}\in E} {w_{ij}}^2 |x_i - x_j|^2}_{\text{Smoothness term}} + \underbrace{\sum_{v_i \in V} {w_i}^2 |x_i - l_i|^2}_{\text{Data term}}$$



$q = 2$: Random walker [Grady 2006]

Previously established links

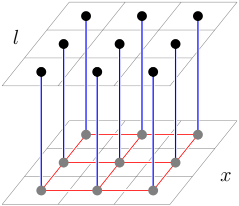
$$\lim_{q\to\infty} \arg\min_x \underbrace{\sum_{e_{ij}\in E} {w_{ij}}^q |x_i - x_j|^q}_{\text{Smoothness term}} + \underbrace{\sum_{v_i \in V} {w_i}^q |x_i - l_i|^q}_{\text{Data term}}$$



$q \to \infty$: Shortest paths [Sinop *et al* 2007]

Previously established links

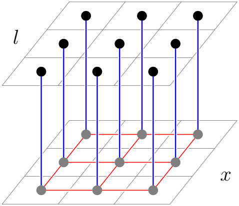
$$\lim_{p\to\infty} \arg\min_x \underbrace{\sum_{e_{ij}\in E} {w_{ij}}^p |x_i - x_j|}_{\text{Smoothness term}} + \underbrace{\sum_{v_i\in V} {w_i}^p |x_i - l_i|}_{\text{Data term}}$$



$p \to \infty$: MSF (Watershed) [Allène et al. 2007]

Power watershed framework

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



Power watershed framework

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

q \ p	0	finite	∞
1	Reduction to seeds	Graph cuts	Max Spanning Forest (watershed) [Allène et al. 07]
2	ℓ_2-norm Voronoi	Random walker	
∞	ℓ_1-norm Voronoi	ℓ_1-norm Voronoi	Shortest Path [Sinop et al. 07]

Power watershed framework

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

q \ p	0	finite	∞
1	Reduction to seeds	Graph cuts	Max Spanning Forest (watershed) [Allène et al. 07]
2	ℓ_2-norm Voronoi	Random walker	
∞	ℓ_1-norm Voronoi	ℓ_1-norm Voronoi	Shortest Path [Sinop et al. 07]

Power watershed framework

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

q \ p	0	finite	∞
1	Reduction to seeds	Graph cuts	Max Spanning Forest (watershed) [Allène et al. 07]
2	ℓ_2-norm Voronoi	Random walker	
∞	ℓ_1-norm Voronoi	ℓ_1-norm Voronoi	Shortest Path [Sinop et al. 07]

Power watershed framework

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

q \ p	0	finite	∞
1	Reduction to seeds	Graph cuts	Max Spanning Forest (watershed) [Allène et al. 07]
2	ℓ_2-norm Voronoi	Random walker	
∞	ℓ_1-norm Voronoi	ℓ_1-norm Voronoi	Shortest Path [Sinop et al. 07]

Power watershed framework

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

q \ p	0	finite	∞
1	Reduction to seeds	Graph cuts	Max Spanning Forest (watershed) [Allène et al. 07]
2	ℓ_2-norm Voronoi	Random walker	
∞	ℓ_1-norm Voronoi	ℓ_1-norm Voronoi	Shortest Path [Sinop et al. 07]

Power watershed framework

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

q \ p	0	finite	∞
1	Reduction to seeds	Graph cuts	Max Spanning Forest (watershed) [Allène et al. 07]
2	ℓ_2-norm Voronoi	Random walker	
∞	ℓ_1-norm Voronoi	ℓ_1-norm Voronoi	Shortest Path [Sinop et al. 07]

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

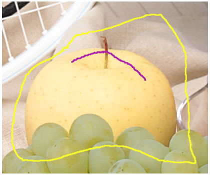
Power watershed framework

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

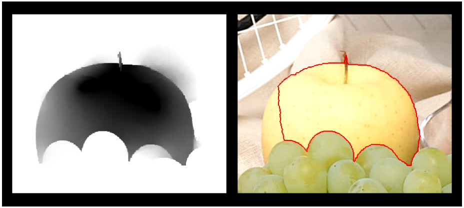
$$\bar{x} = \lim_{p\to\infty} x^*_{p,q}$$

Convergence of RW when $p \to \infty$ toward MSF cut

Input seeds



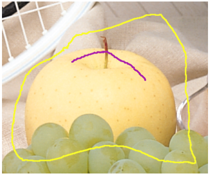
$$x^*_1 = \arg\min_x \underbrace{\sum_{e_{ij} \in E} w_{ij}^{\;1} |x_i - x_j|^2}_{\text{Smoothness term}} + \underbrace{\mathcal{D}(x)}_{\text{Data fidelity}}$$



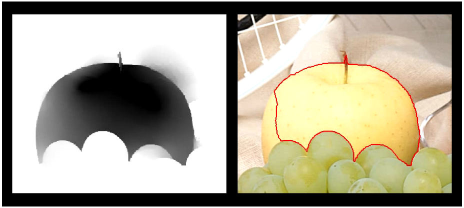
solution x^*_1 cut : threshold of x^*_1

Convergence of RW when $p \to \infty$ toward MSF cut

Input seeds



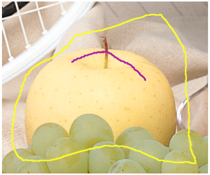
$$x^*_2 = \arg\min_x \underbrace{\sum_{e_{ij} \in E} w_{ij}^{\;2} |x_i - x_j|^2}_{\text{Smoothness term}} + \underbrace{\mathcal{D}(x)}_{\text{Data fidelity}}$$



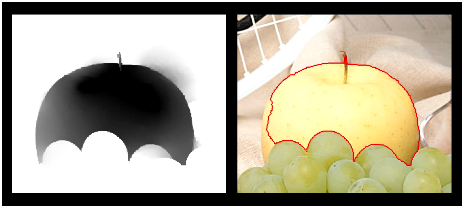
solution x^*_2 cut : threshold of x^*_2

Convergence of RW when $p \to \infty$ toward MSF cut

Input seeds



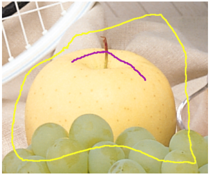
$$x^*_3 = \arg\min_x \underbrace{\sum_{e_{ij}\in E} w_{ij}{}^3 |x_i - x_j|^2}_{\text{Smoothness term}} + \underbrace{\mathcal{D}(x)}_{\text{Data fidelity}}$$



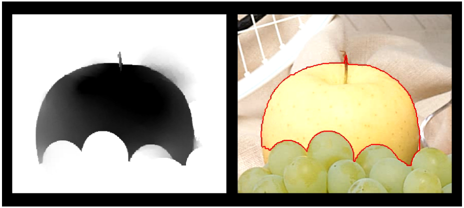
solution x^*_3 cut : threshold of x^*_3

Convergence of RW when $p \to \infty$ toward MSF cut

Input seeds



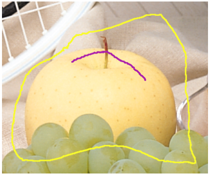
$$x^*_4 = \arg\min_x \underbrace{\sum_{e_{ij} \in E} w_{ij}{}^4 |x_i - x_j|^2}_{\text{Smoothness term}} + \underbrace{\mathcal{D}(x)}_{\text{Data fidelity}}$$



solution x^*_4 cut : threshold of x^*_4

Convergence of RW when $p \to \infty$ toward MSF cut

Input seeds

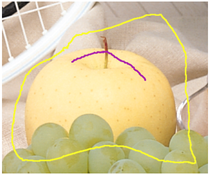


$$x^*_6 = \arg\min_x \underbrace{\sum_{e_{ij} \in E} w_{ij}{}^{6} |x_i - x_j|^2}_{\text{Smoothness term}} + \underbrace{\mathcal{D}(x)}_{\text{Data fidelity}}$$

solution x^*_6 cut : threshold of x^*_6

Convergence of RW when $p \to \infty$ toward MSF cut

Input seeds

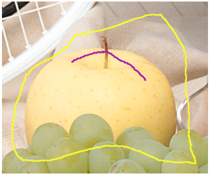


$$x^*_9 = \arg\min_x \underbrace{\sum_{e_{ij} \in E} w_{ij}^{\ 9} |x_i - x_j|^2}_{\text{Smoothness term}} + \underbrace{\mathcal{D}(x)}_{\text{Data fidelity}}$$

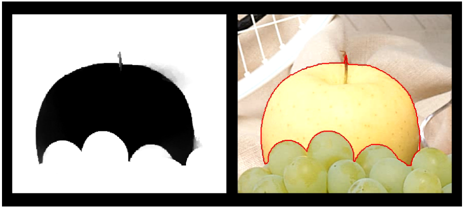
solution x^*_9 cut : threshold of x^*_9

Convergence of RW when $p \to \infty$ toward MSF cut

Input seeds



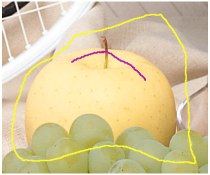
$$x_{13}{}^* = \arg\min_x \underbrace{\sum_{e_{ij} \in E} w_{ij}{}^{13} |x_i - x_j|^2}_{\text{Smoothness term}} + \underbrace{\mathcal{D}(x)}_{\text{Data fidelity}}$$



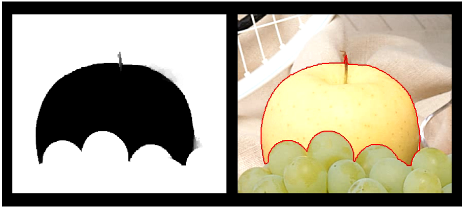
solution $x_{13}{}^*$ cut : threshold of $x_{13}{}^*$

Convergence of RW when $p \to \infty$ toward MSF cut

Input seeds



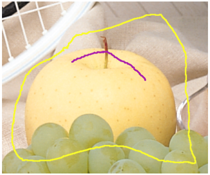
$$x_{18}{}^{*} = \arg\min_{x} \underbrace{\sum_{e_{ij} \in E} w_{ij}{}^{18} |x_i - x_j|^2}_{\text{Smoothness term}} + \underbrace{\mathcal{D}(x)}_{\text{Data fidelity}}$$



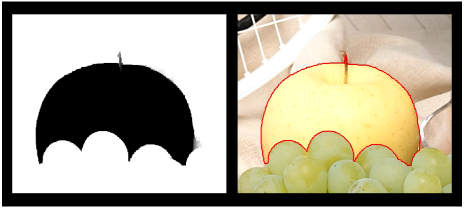
solution $x_{18}{}^{*}$ cut : threshold of $x_{18}{}^{*}$

Convergence of RW when $p \rightarrow \infty$ toward MSF cut

Input seeds



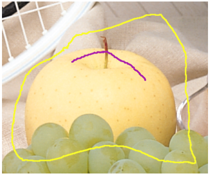
$$x_{24}{}^{*} = \arg\min_{x} \underbrace{\sum_{e_{ij} \in E} w_{ij}{}^{24} |x_i - x_j|^2}_{\text{Smoothness term}} + \underbrace{\mathcal{D}(x)}_{\text{Data fidelity}}$$



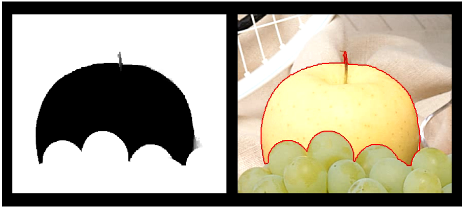
solution $x_{24}{}^{*}$ cut : threshold of $x_{24}{}^{*}$

Convergence of RW when $p \to \infty$ toward MSF cut

Input seeds



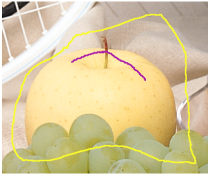
$$x_{30}{}^{*} = \arg\min_{x} \underbrace{\sum_{e_{ij} \in E} w_{ij}{}^{30} |x_i - x_j|^2}_{\text{Smoothness term}} + \underbrace{\mathcal{D}(x)}_{\text{Data fidelity}}$$



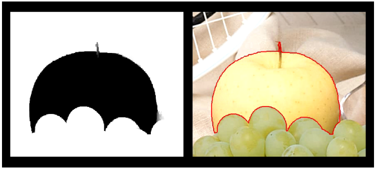
solution $x_{30}{}^{*}$ cut : threshold of $x_{30}{}^{*}$

Convergence of RW when $p \to \infty$ toward MSF cut

Input seeds



$$x_p{}^* = \arg\min_x \underbrace{\sum_{e_{ij}\in E} w_{ij}{}^p |x_i - x_j|^q}_{\text{Smoothness term}} + \underbrace{\mathcal{D}(x)}_{\text{Data fidelity}}$$



$\bar{x} = \lim_{p\to\infty} x_p^*$ cut : threshold of $\bar{x}$

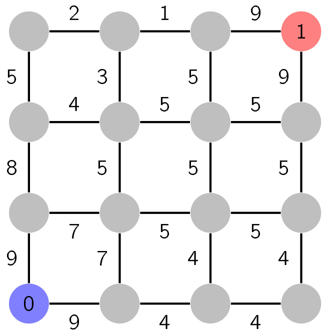
Theorems

When $p \to \infty$,

- the obtained cut is an MSF cut.
- when $q > 1$, the solution $\bar{x}$ is unique.

Power watershed algorithm

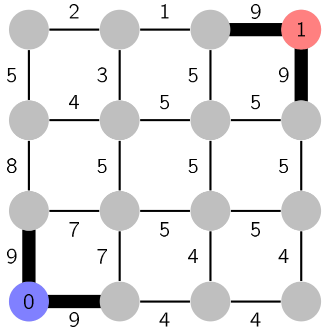
1. Choose an edge with maximal weight $e_{\max}$. Let S the set of edges connected to $e_{\max}$ with the same weight as $e_{\max}$.
2. If S does not contain vertices that have different labels, merge the nodes of S into one node, otherwise minimize $E_{1,q}$ on S.
3. Repeat steps 1 and 2 until all vertices are labeled.



$\bar{x} = \lim_{p\to\infty} \arg\min_x \sum_{e_{ij}\in E} w_{ij}^p |x_i - x_j|^q$

Power watershed algorithm

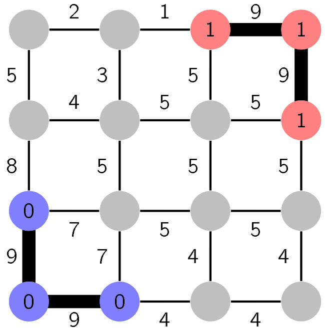
1. Choose an edge with maximal weight $e_{\max}$. Let S the set of edges connected to $e_{\max}$ with the same weight as $e_{\max}$.
2. If S does not contain vertices that have different labels, merge the nodes of S into one node, otherwise minimize $E_{1,q}$ on S.
3. Repeat steps 1 and 2 until all vertices are labeled.



$$\bar{x} = \lim_{p \to \infty} \arg\min_x \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q$$

Power watershed algorithm

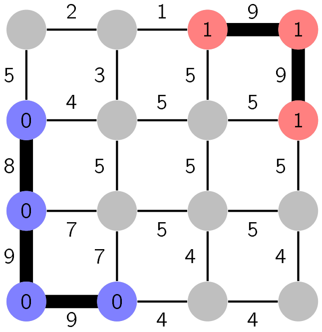
1. Choose an edge with maximal weight $e_{\max}$. Let S the set of edges connected to $e_{\max}$ with the same weight as $e_{\max}$.
2. If S does not contain vertices that have different labels, merge the nodes of S into one node, otherwise minimize $E_{1,q}$ on S.
3. Repeat steps 1 and 2 until all vertices are labeled.



$$\bar{x} = \lim_{p\to\infty} \arg\min_x \sum_{e_{ij}\in E} w_{ij}^p |x_i - x_j|^q$$

Power watershed algorithm

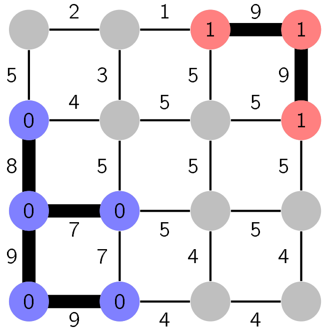
1. Choose an edge with maximal weight $e_{\max}$. Let S the set of edges connected to $e_{\max}$ with the same weight as $e_{\max}$.
2. If S does not contain vertices that have different labels, merge the nodes of S into one node, otherwise minimize $E_{1,q}$ on S.
3. Repeat steps 1 and 2 until all vertices are labeled.



$$\bar{x} = \lim_{p\to\infty} \arg\min_x \sum_{e_{ij}\in E} w_{ij}^p |x_i - x_j|^q$$

Power watershed algorithm

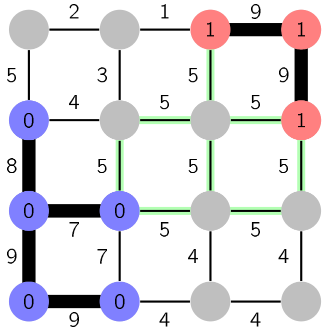
1. Choose an edge with maximal weight e_{max}. Let S the set of edges connected to e_{max} with the same weight as e_{max}.
2. If S does not contain vertices that have different labels, merge the nodes of S into one node, otherwise minimize $E_{1,q}$ on S.
3. Repeat steps 1 and 2 until all vertices are labeled.



$$\bar{x} = \lim_{p\to\infty} \arg\min_x \sum_{e_{ij}\in E} w_{ij}^p |x_i - x_j|^q$$

Power watershed algorithm

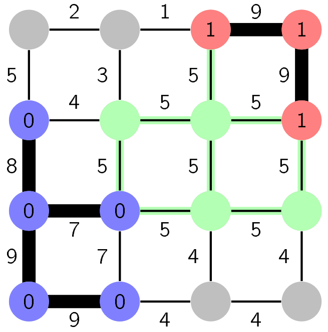
1. Choose an edge with maximal weight e_{max}. Let S the set of edges connected to e_{max} with the same weight as e_{max}.
2. If S does not contain vertices that have different labels, merge the nodes of S into one node, otherwise minimize $E_{1,q}$ on S.
3. Repeat steps 1 and 2 until all vertices are labeled.



$$\bar{x} = \lim_{p\to\infty} \arg\min_x \sum_{e_{ij}\in E} w_{ij}^p |x_i - x_j|^q$$

Power watershed algorithm

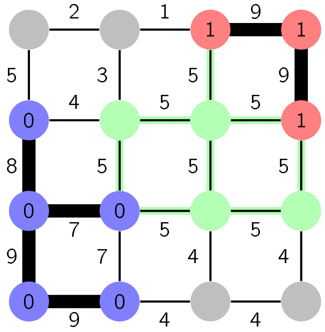
1. Choose an edge with maximal weight $e_{\max}$. Let S the set of edges connected to $e_{\max}$ with the same weight as $e_{\max}$.
2. If S does not contain vertices that have different labels, merge the nodes of S into one node, otherwise minimize $E_{1,q}$ on S.
3. Repeat steps 1 and 2 until all vertices are labeled.



$$\bar{x} = \lim_{p\to\infty} \arg\min_x \sum_{e_{ij}\in E} w_{ij}^p |x_i - x_j|^q$$

Power watershed algorithm

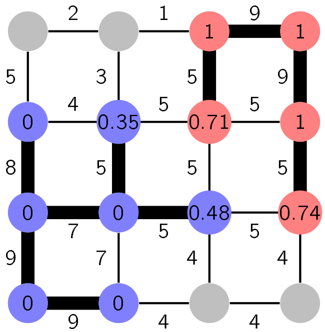
1. Choose an edge with maximal weight $e_{\max}$. Let S the set of edges connected to $e_{\max}$ with the same weight as $e_{\max}$.
2. If S does not contain vertices that have different labels, merge the nodes of S into one node, otherwise minimize $E_{1,q}$ on S.
3. Repeat steps 1 and 2 until all vertices are labeled.



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

Power watershed algorithm

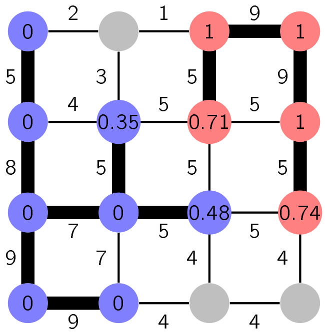
1. Choose an edge with maximal weight $e_{\max}$. Let S the set of edges connected to $e_{\max}$ with the same weight as $e_{\max}$.
2. If S does not contain vertices that have different labels, merge the nodes of S into one node, otherwise minimize $E_{1,q}$ on S.
3. Repeat steps 1 and 2 until all vertices are labeled.



$$\bar{x} = \lim_{p\to\infty} \arg\min_x \sum_{e_{ij}\in E} w_{ij}^p |x_i - x_j|^q$$

Power watershed algorithm

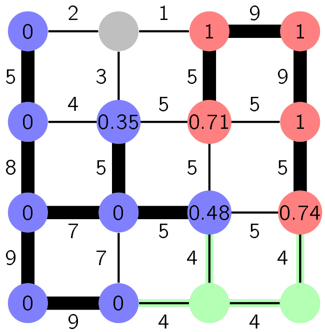
1. Choose an edge with maximal weight e_{max}. Let S the set of edges connected to e_{max} with the same weight as e_{max}.
2. If S does not contain vertices that have different labels, merge the nodes of S into one node, otherwise minimize $E_{1,q}$ on S.
3. Repeat steps 1 and 2 until all vertices are labeled.



$$\bar{x} = \lim_{p\to\infty} \arg\min_x \sum_{e_{ij}\in E} w_{ij}^p |x_i - x_j|^q$$

Power watershed algorithm

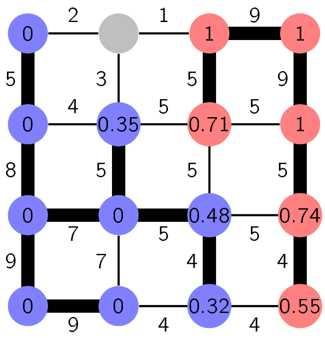
1. Choose an edge with maximal weight $e_{\max}$. Let S the set of edges connected to $e_{\max}$ with the same weight as $e_{\max}$.
2. If S does not contain vertices that have different labels, merge the nodes of S into one node, otherwise minimize $E_{1,q}$ on S.
3. Repeat steps 1 and 2 until all vertices are labeled.



$$\bar{x} = \lim_{p\to\infty} \arg\min_x \sum_{e_{ij}\in E} w_{ij}^p |x_i - x_j|^q$$

Power watershed algorithm

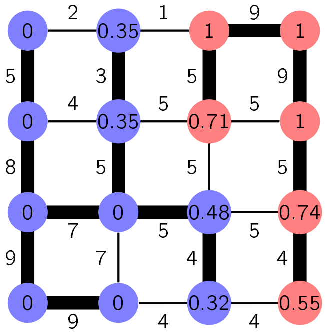
1. Choose an edge with maximal weight $e_{\max}$. Let S the set of edges connected to $e_{\max}$ with the same weight as $e_{\max}$.
2. If S does not contain vertices that have different labels, merge the nodes of S into one node, otherwise minimize $E_{1,q}$ on S.
3. Repeat steps 1 and 2 until all vertices are labeled.



$$\bar{x} = \arg\min_x \lim_{p\to\infty} \sum_{e_{ij}\in E} w_{ij}^p |x_i - x_j|^q$$

Power watershed algorithm

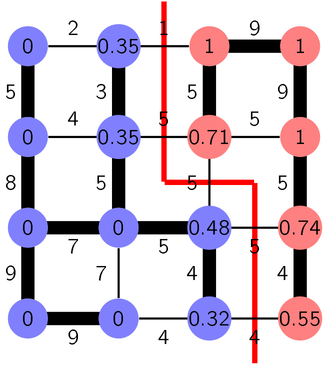
1. Choose an edge with maximal weight $e_{\max}$. Let S the set of edges connected to $e_{\max}$ with the same weight as $e_{\max}$.
2. If S does not contain vertices that have different labels, merge the nodes of S into one node, otherwise minimize $E_{1,q}$ on S.
3. Repeat steps 1 and 2 until all vertices are labeled.



$$\bar{x} = \arg\min_x \lim_{p\to\infty} \sum_{e_{ij}\in E} w_{ij}^p |x_i - x_j|^q$$

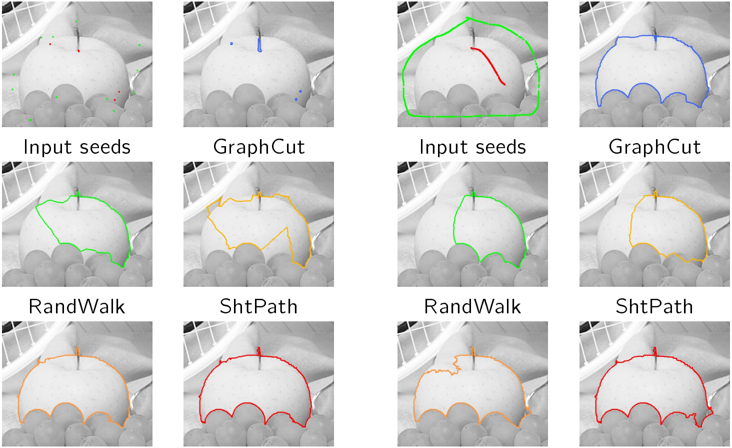
Power watershed algorithm

1. Choose an edge with maximal weight $e_{\max}$. Let S the set of edges connected to $e_{\max}$ with the same weight as $e_{\max}$.
2. If S does not contain vertices that have different labels, merge the nodes of S into one node, otherwise minimize $E_{1,q}$ on S.
3. Repeat steps 1 and 2 until all vertices are labeled.



$$\bar{x} = \arg\min_x \lim_{p \to \infty} \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q$$

Comparison of results

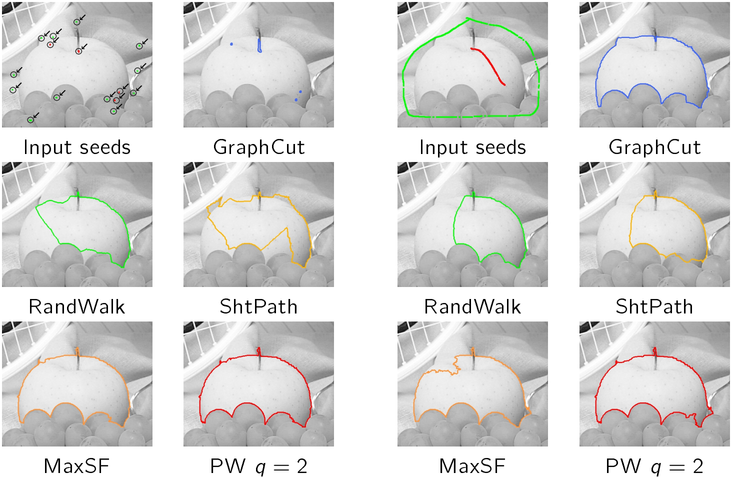


Input seeds | GraphCut | Input seeds | GraphCut

RandWalk | ShtPath | RandWalk | ShtPath

MaxSF | PW $q = 2$ | MaxSF | PW $q = 2$

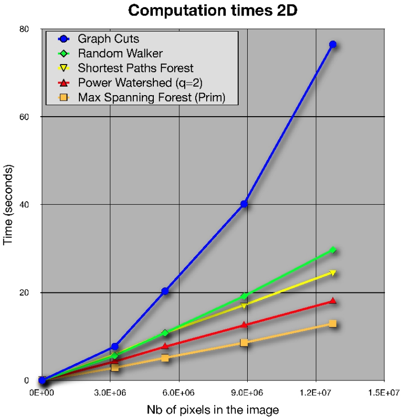
Comparison of results



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

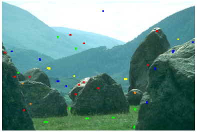


Optimal multilabels segmentation

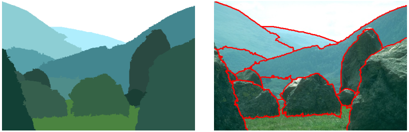
- l solutions $x^1, x^2, \ldots x^l$ computed
- x^k computed by enforcing $\begin{cases} x^k(l^k) = 1 \\ x^k(l^q) = 0 \text{ for all } q \neq k. \end{cases}$
- Each node i is affected to the label for which x_i^k is maximum :

$$s_i = \arg\max_k x_i^k$$

Input seeds

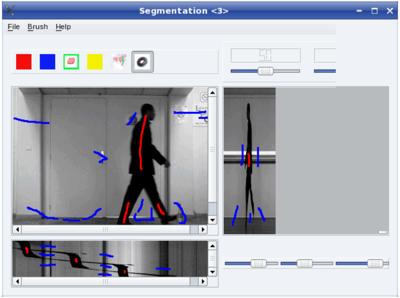


Segmentation by PowerWatershed ($q = 2$)



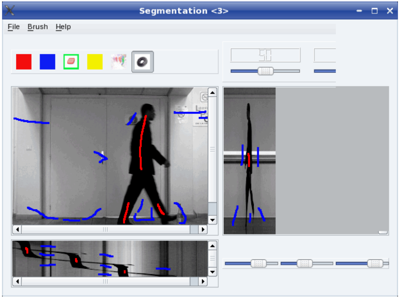
Video segmentation

Prim's algorithm of Max Spanning Forest (Watershed)



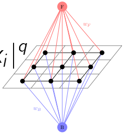
Video segmentation

Power watersheds



Unseeded segmentation

$$\bar{x} = \lim_{p \to \infty} \arg\min_{x} \sum_{e_{ij} \in E} w_{ij}^{p} |x_i - x_j|^q + \sum_{v_i} {w_{F_i}}^{p} |x_i - 1|^q + \sum_{v_i} {w_{B_i}}^{p} |x_i|^q$$



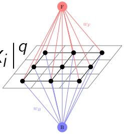
Image

Graph Cuts

Watershed

Unseeded segmentation

$$\bar{x} = \lim_{p\to\infty} \arg\min_{x} \sum_{e_{ij}\in E} w_{ij}^{p}|x_i - x_j|^q + \sum_{v_i} {w_{F_i}}^{p}|x_i - 1|^q + \sum_{v_i} {w_{B_i}}^{p}|x_i|^q$$



Image

Graph Cuts

Watershed

This is the first time that it is shown how to incorporate data unary terms into watershed computation.

Non-convex diffusion using power watersheds

- Anisotropic diffusion [Perona-Malik 1990]

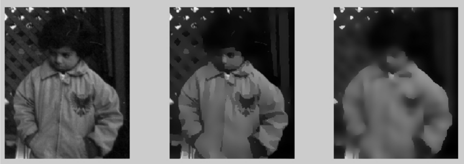


Image   100 iterations   200 iterations

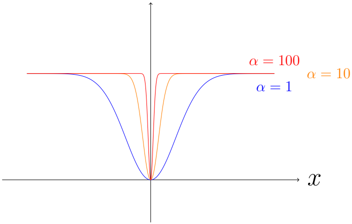
Goals of this work :

- perform anisotropic diffusion using an ℓ_0 norm to avoid the blurring effect
- optimize a non convex energy using Power Watershed [Couprie-Grady-Najman-Talbot, ICIP 2010]

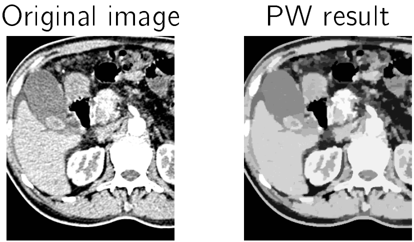
Anisotropic diffusion and ℓ_0 norm

$$x^* = \arg\min_x \underbrace{\sum_{e_{ij}\in E} \sigma(x_i - x_j)}_{\text{smoothness term}} + \lambda \underbrace{\sum_{v_i \in V} \sigma(x_i - f_i)}_{\text{data fidelity term}}$$

$$\sigma(x) = 1 - e^{-\alpha x^2}$$

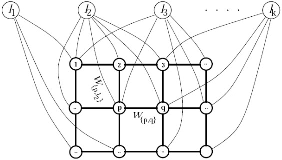


Leads to piecewise constant results



Stereovision using power watershed

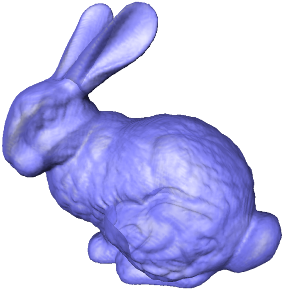
- Compute the disparity map from two aligned images



- Labels correspond to the disparities, weights to similarity coefficients between blocks

Surface reconstruction from a noisy set of dots

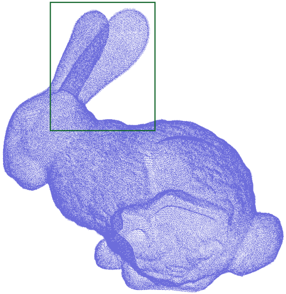
$\Rightarrow$



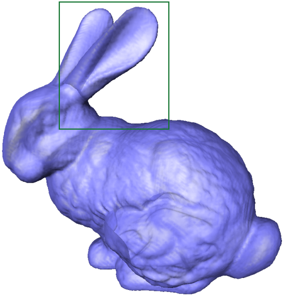
- Goal : given a noisy set of dots, find an explicit surface fitting the dots.

Joint work with Xavier Bresson

Surface reconstruction from a noisy set of dots



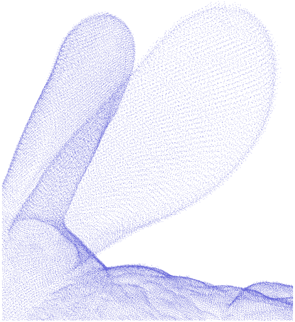
$\Rightarrow$



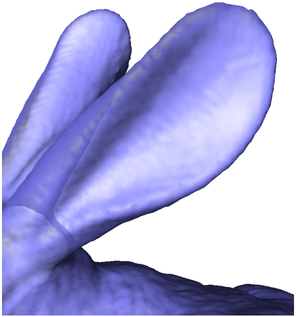
- Goal : given a noisy set of dots, find an explicit surface fitting the dots.

Joint work with Xavier Bresson

Surface reconstruction from a noisy set of dots



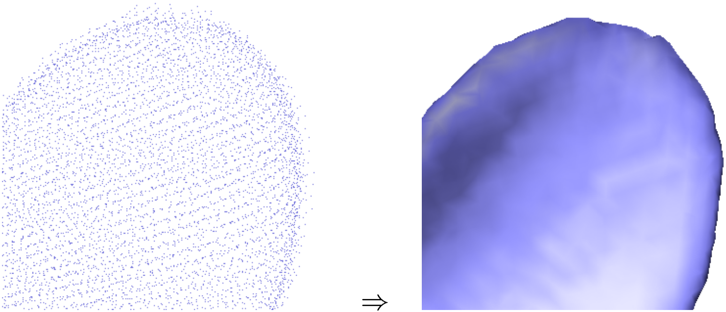
$\Rightarrow$



- Goal : given a noisy set of dots, find an explicit surface fitting the dots.

Joint work with Xavier Bresson

Surface reconstruction from a noisy set of dots



- Goal : given a noisy set of dots, find an explicit surface fitting the dots.

Joint work with Xavier Bresson

How to solve this problem

- Graph : 3D grid
- Here x represents the object indicator to recover.

$$\bar{x} = \lim_{p \to \infty} \arg\min_x \sum_{e_{ij} \in E} {w_{ij}}^p |x_i - x_j|^q$$

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

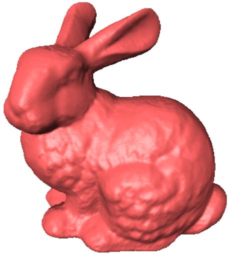
- weights : distance function from the set of dots to fit

Power watershed solution

Why PW are a good fit for this problem ?

numerous plateaus around the dots to fit $\rightarrow$ smooth isosurface is obtained

Comparisons



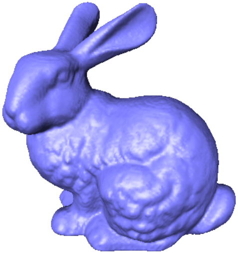
Total variation

Size of required seeds

Graph cuts

Size of required seeds

surface normals
estimation required

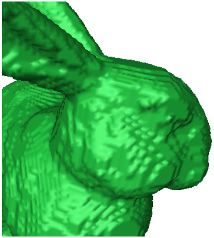


Power watershed

Size of required seeds

Comparisons

Total variation



Graph cuts

Power watershed

- Fast, accurate, globally optimal surface reconstruction from noisy set of dots
- Robust to markers placement
- No normal estimation information required
- No post-processing smoothing step

Outline

Outline

- **IV - Conclusion**

Conclusion

Reformulation of classical max flow method

- Block artifacts of classical max flow
- Convergence issues AT-CMF
- Filtering using Graph cuts expensive

Power watersheds answers several problems of standard methods

- Non unique solution
- Leaking effect
- Random Walker, Graph cuts : super-linear complexity

Conclusion

Reformulation of classical max flow method

- Block artifacts of classical max flow Not present with CCMF
- Convergence issues AT-CMF
- Filtering using Graph cuts expensive

Power watersheds answers several problems of standard methods

- Non unique solution
- Leaking effect
- Random Walker, Graph cuts : super-linear complexity

Conclusion

Reformulation of classical max flow method

- Block artifacts of classical max flow Not present with CCMF
- Convergence issues AT-CMF Convergence guaranteed
- Filtering using Graph cuts expensive

Power watersheds answers several problems of standard methods

- Non unique solution
- Leaking effect
- Random Walker, Graph cuts : super-linear complexity

Conclusion

Reformulation of classical max flow method

- Block artifacts of classical max flow Not present with CCMF
- Convergence issues AT-CMF Convergence guaranteed
- Filtering using Graph cuts expensive Extention of the CCMF framework leads to a flexible filtering framework on graphs

Power watersheds answers several problems of standard methods

- Non unique solution
- Leaking effect
- Random Walker, Graph cuts : super-linear complexity

Conclusion

Reformulation of classical max flow method

- Block artifacts of classical max flow Not present with CCMF
- Convergence issues AT-CMF Convergence guaranteed
- Filtering using Graph cuts expensive Extention of the CCMF framework leads to a flexible filtering framework on graphs

Power watersheds answers several problems of standard methods

- Non unique solution Unique solution
- Leaking effect
- Random Walker, Graph cuts : super-linear complexity

Conclusion

Reformulation of classical max flow method

- Block artifacts of classical max flow Not present with CCMF
- Convergence issues AT-CMF Convergence guaranteed
- Filtering using Graph cuts expensive Extention of the CCMF framework leads to a flexible filtering framework on graphs

Power watersheds answers several problems of standard methods

- Non unique solution Unique solution
- Leaking effect Reduction of the leaks
- Random Walker, Graph cuts : super-linear complexity

Conclusion

Reformulation of classical max flow method

- Block artifacts of classical max flow Not present with CCMF
- Convergence issues AT-CMF Convergence guaranteed
- Filtering using Graph cuts expensive Extention of the CCMF framework leads to a flexible filtering framework on graphs

Power watersheds answers several problems of standard methods

- Non unique solution Unique solution
- Leaking effect Reduction of the leaks
- Random Walker, Graph cuts : super-linear complexity Quasi-linear experimentally. Worst case : RW complexity.

Conclusion

Reformulation of classical max flow method

- Block artifacts of classical max flow Not present with CCMF
- Convergence issues AT-CMF Convergence guaranteed
- Filtering using Graph cuts expensive Extention of the CCMF framework leads to a flexible filtering framework on graphs

Power watersheds answers several problems of standard methods

- Non unique solution Unique solution
- Leaking effect Reduction of the leaks
- Random Walker, Graph cuts : super-linear complexity Quasi-linear experimentally. Worst case : RW complexity.
- More importantly : use of unary terms and multi labels opens the way to large field of applications

Conclusion

Continuous methods

Discrete calculus formulations

Optimization

Mathematical morphology

Next Challenge : Scene Parsing using New Global Energy Models

- Scene understanding of objects in video
- Need for efficient algorithm to process large amount of data
- Hierarchical CRF [Ladický-Russell-Kohli-Torr 2009] have shown good results.
- Study the possibility of applying this optimization approach for solving the problem using watersheds
- Advantages : speed, global optimality

Questions

Source code for segmentation available from:

http ://sourceforge.net/projects/powerwatershed/

References

Journals

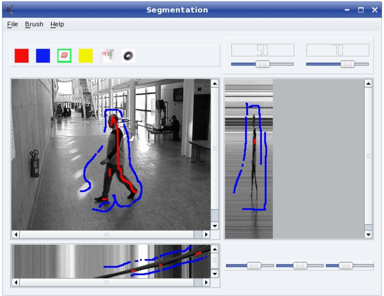
- C. Couprie, L. Grady, L. Najman, J.C. Pesquet, and H. Talbot : Constrained TV-based regularization on graphs. *Submitted, Oct. 2011*.
- C. Couprie, L. Grady, H. Talbot, and L. Najman : Combinatorial Continuous Max flows. In *SIAM journal on imaging sciences, 2011*.
- C. Couprie, L. Grady, L. Najman, and H. Talbot : Power Watersheds : A unifying graph-based optimization framework. In *IEEE Trans. on PAMI 2011*.

International conferences

- C. Couprie, H. Talbot, J.C. Pesquet, L. Najman, and L. Grady : Dual constrained tv-based regularization. In *Proc. of ICASSP, 2011*.
- C. Couprie, X. Bresson, L. Najman, H. Talbot and L. Grady : Surface reconstruction using Power watersheds. In *Proc. of ISMM 2011*.
- C. Couprie, L. Grady, L. Najman, and H. Talbot : Anisotropic diffusion using power watersheds. In *Proc. of ICIP 2010*.
- C. Couprie, L. Grady, L. Najman, and H. Talbot : Power watersheds : A new image segmentation framework extending graph cuts, random walker and optimal spanning forest. In *Proc. of ICCV 2009*.

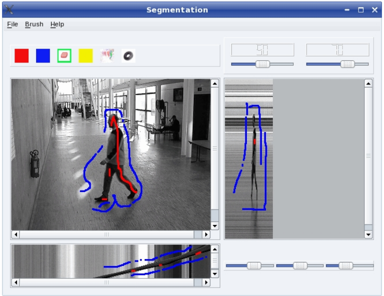
Video segmentation : real life situation

Graph cut segmentation



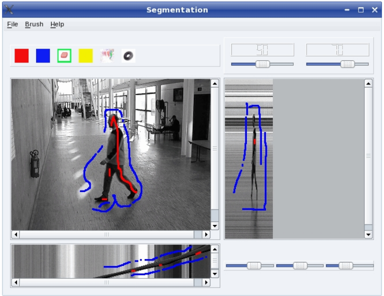
Video segmentation : real life situation

Prim's algorithm of Max Spanning Forest (Watershed)



Video segmentation : real life situation

Segmentation using Powerwatershed



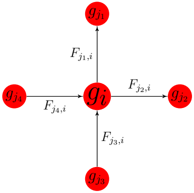
Chapter 3 : Dual constrained TV based formulation

$$\min_{x \in \mathbb{R}^N} \underbrace{\sup_{F \in C} \quad F^\top (Ax)}_{\text{regularization}} + \underbrace{\frac{1}{2\lambda} \|x - f\|_2^2}_{\text{data fidelity}}$$

- $C = \cap_{i=1}^{m-1} C_i, \quad C_i = \{F \in \mathbb{R}^M \mid \|\theta_i \cdot F\|_\alpha \leq g_i\}, \; \alpha \geq 1.$

Example adapted to image denoising

- $g_i \in \mathbb{R}^N$ weight on vertex i, inversely function of the gradient of f at node i.
- Flat area : weak gradient $\rightarrow$ strong g_i $\rightarrow$ strong $F_{i,j}$ $\rightarrow$ weak local variations of x.
- Contours : strong gradient $\rightarrow$ weak g_i $\rightarrow$ weak $F_{i,j}$ $\rightarrow$ large local variations of x allowed.



$$C_i = \{F \in \mathbb{R}^M \mid \sqrt{\sum_{j \in N_i} F_{j,i}^2} \leq g_i\}$$

Dual problem

- Fenchel-Rockafellar dual problem :

$$\min_{F \in \mathbb{R}^M} \sum_{i=1}^{s} f_i(F) + f_{s+1}(F)$$

Dual problem

- Fenchel-Rockafellar dual problem :

$$\min_{F \in \mathbb{R}^M} \sum_{i=1}^{s} \iota_{C_i}(F) + f_{s+1}(F)$$

where ι_C : indicator function of convex C (=0 in C, $+\infty$ outside), $f_{s+1} : F \mapsto \frac{1}{2} F^\top A \Gamma A^\top F - F^\top A \Gamma H^\top f$, and $\Gamma = (H^\top H)^{-1}$.

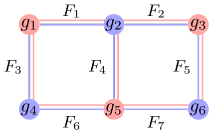
Dual problem

- Fenchel-Rockafellar dual problem :

$$\min_{F \in \mathbb{R}^M} \sum_{i=1}^{s} \iota_{C_i}(F) + f_{s+1}(F)$$

where ι_C : indicator function of convex C (=0 in C, $+\infty$ outside),
$f_{s+1} : F \mapsto \frac{1}{2} F^\top A \Gamma A^\top F - F^\top A \Gamma H^\top f$, and $\Gamma = (H^\top H)^{-1}$.

$$C = \{F \in \mathbb{R}^M \mid |A^\top| F^2 \leq g^2\}$$



$$C_1 = \{F \in \mathbb{R}^M | \; (F_1^2 + F_2^2 + F_4^2)^{\frac{1}{2}} \leq g_2^2, \; (F_3^2 + F_6^2)^{\frac{1}{2}} \leq g_4^2, \; (F_5^2 + F_7^2)^{\frac{1}{2}} \leq g_6^2\}$$

$$C_2 = \{F \in \mathbb{R}^M | \; (F_1^2 + F_3^2)^{\frac{1}{2}} \leq g_1^2, \; (F_2^2 + F_5^2)^{\frac{1}{2}} \leq g_3^2, \; (F_4^2 + F_6^2 + F_7^2)^{\frac{1}{2}} \leq g_5^2\}$$

Dual problem

- Fenchel-Rockafellar dual problem :

$$\min_{F \in \mathbb{R}^M} \sum_{i=1}^{s} \iota_{C_i}(F) + f_{s+1}(F)$$

where ι_C : indicator function of convex C (=0 in C, $+\infty$ outside), $f_{s+1} : F \mapsto \frac{1}{2} F^\top A \Gamma A^\top F - F^\top A \Gamma H^\top f$, and $\Gamma = (H^\top H)^{-1}$.

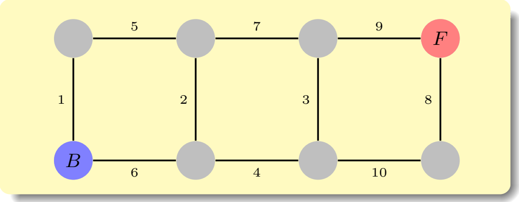
- The primal problem admits a unique solution $\widehat{x}$.
- If $\widehat{F}$ is a solution to the dual problem,

$$\widehat{x} = \Gamma \left(H^\top f - A^\top \widehat{F} \right).$$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \to \infty$ is an MSF-cut.

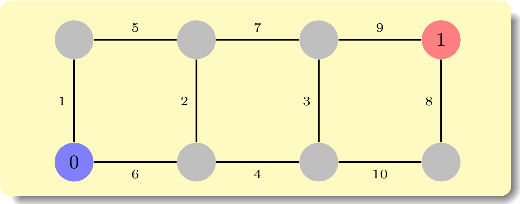


Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \to \infty$ is an MSF-cut.

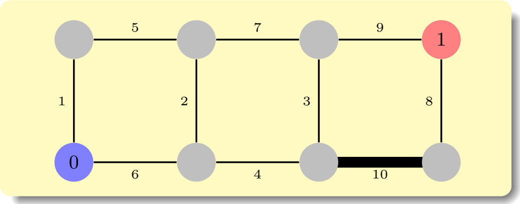


Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \to \infty$ is an MSF-cut.

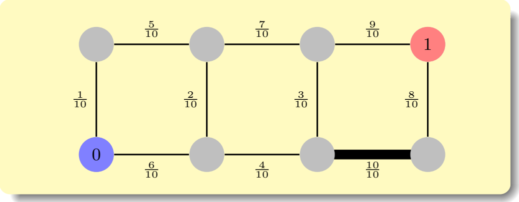


Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \to \infty$ is an MSF-cut.

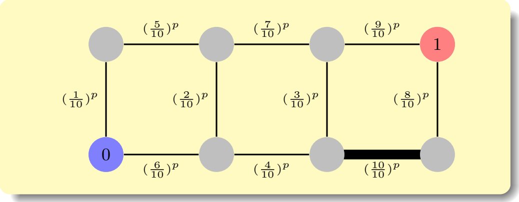


Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \to \infty$ is an MSF-cut.

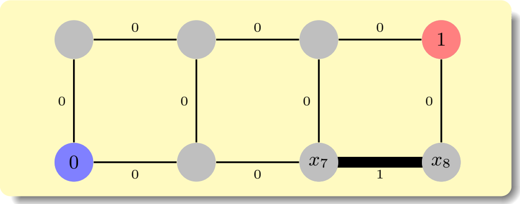


Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \to \infty$ is an MSF-cut.

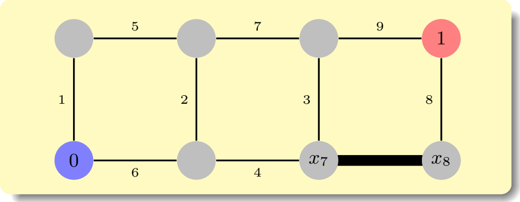


Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \rightarrow \infty$ is an MSF-cut.



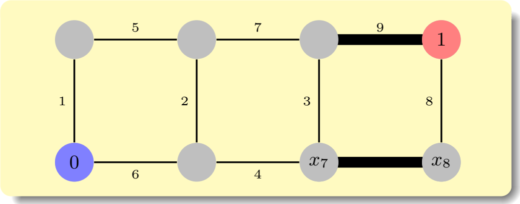
$x_7 = x_8$

Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \to \infty$ is an MSF-cut.



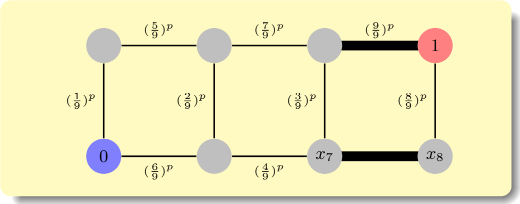
$x_7 = x_8$

Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \to \infty$ is an MSF-cut.



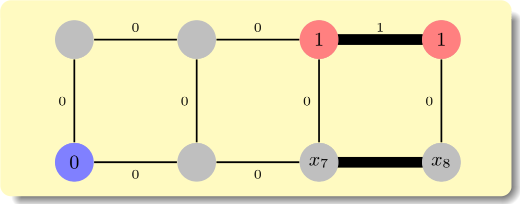
$x_7 = x_8$

Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \to \infty$ is an MSF-cut.



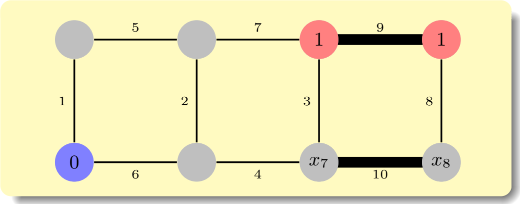
$x_7 = x_8$

Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \to \infty$ is an MSF-cut.



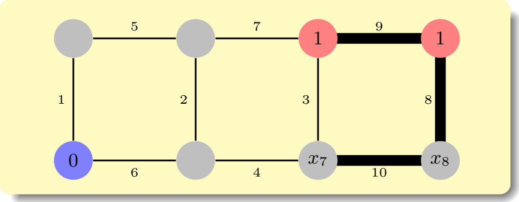
$x_7 = x_8$

Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \rightarrow \infty$ is an MSF-cut.



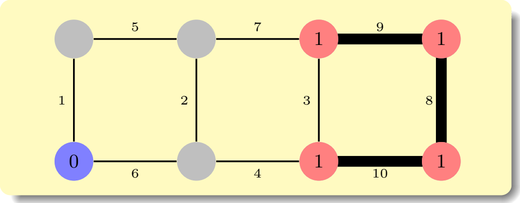
$x_7 = x_8$

Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \rightarrow \infty$ is an MSF-cut.

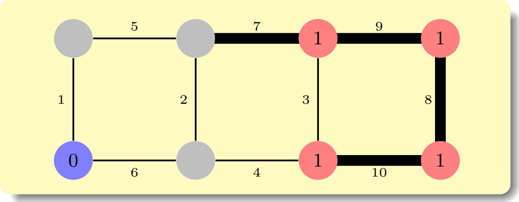


Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \to \infty$ is an MSF-cut.

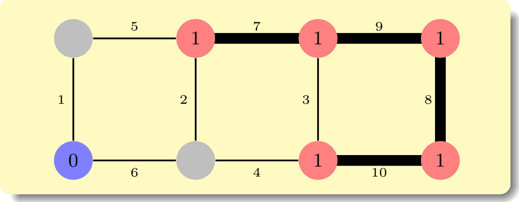


Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \to \infty$ is an MSF-cut.

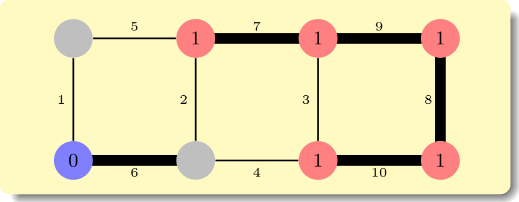


Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \to \infty$ is an MSF-cut.

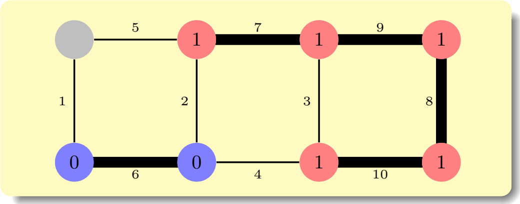


Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \to \infty$ is an MSF-cut.

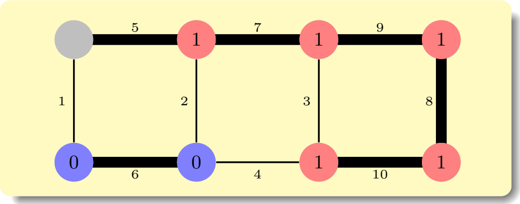


Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \to \infty$ is an MSF-cut.

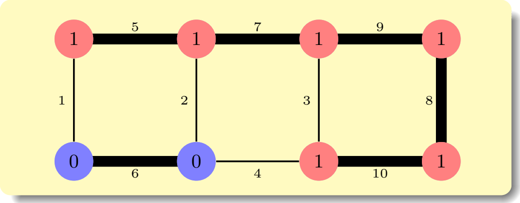


Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chaper 4 : Energy optimization and MSF cut

Theorem

If the weights are all different, any cut thresholding the optimal solution x minimizing $E_{p,q}$ when $q \geq 1$ and $p \to \infty$ is an MSF-cut.

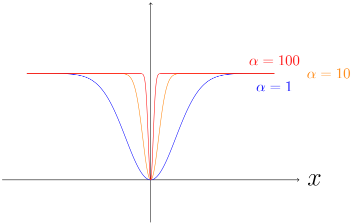


Recall the energy function : $E_{p,q} = \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \mathcal{D}(x)$

Chapter 5 : Nonconvex optimization using PWsheds

$$\min_x \underbrace{\sum_{e_{ij}\in E} \sigma(x_i - x_j)}_{\text{smoothness term}} + \lambda \underbrace{\sum_{v_i \in V} \sigma(x_i - f_i)}_{\text{data fidelity term}}$$

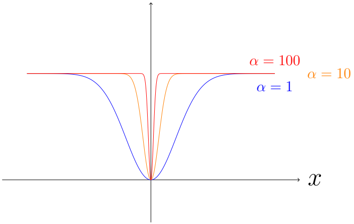
$$\sigma(x) = 1 - \mathrm{e}^{-\alpha x^2}$$



Chapter 5 : Nonconvex optimization using PWsheds

$$\min_x \underbrace{\sum_{e_{ij}\in E} \sigma(x_i - x_j)}_{\text{smoothness term}} + \lambda \underbrace{\sum_{v_i \in V} \sigma(x_i - f_i)}_{\text{data fidelity term}}$$

$\sigma(x) = 1 - e^{-\alpha x^2}$

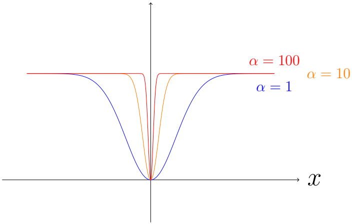


- High gradient $x_i - x_j \Rightarrow \sigma = 1$

Chapter 5 : Nonconvex optimization using PWsheds

$$\min_x \underbrace{\sum_{e_{ij}\in E} \sigma(x_i - x_j)}_{\text{smoothness term}} + \underbrace{\lambda \sum_{v_i \in V} \sigma(x_i - f_i)}_{\text{data fidelity term}}$$

$$\sigma(x) = 1 - \mathrm{e}^{-\alpha x^2}$$

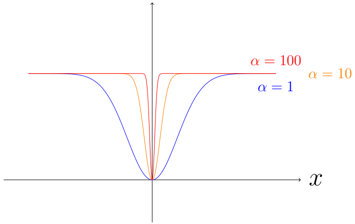


- High gradient $x_i - x_j \Rightarrow \sigma = 1$
- No gradient $\Rightarrow \sigma = 0$

Chapter 5 : Nonconvex optimization using PWsheds

$$\min_x \underbrace{\sum_{e_{ij}\in E} \sigma(x_i - x_j)}_{\text{smoothness term}} + \underbrace{\lambda \sum_{v_i\in V} \sigma(x_i - f_i)}_{\text{data fidelity term}}$$

$\sigma(x) = 1 - \mathrm{e}^{-\alpha x^2}$

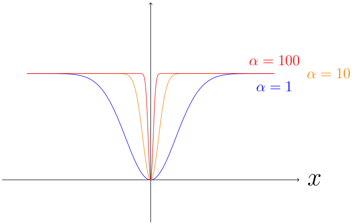


- High gradient $x_i - x_j \Rightarrow \sigma = 1$
- No gradient $\Rightarrow \sigma = 0$
- Finite α, low gradient $\Rightarrow$ $0 < \sigma < 1$ Piecewise smooth result

Chapter 5 : Nonconvex optimization using PWsheds

$$\min_x \underbrace{\sum_{e_{ij} \in E} \sigma(x_i - x_j)}_{\text{smoothness term}} + \underbrace{\lambda \sum_{v_i \in V} \sigma(x_i - f_i)}_{\text{data fidelity term}}$$

$$\sigma(x) = 1 - \mathrm{e}^{-\alpha x^2}$$



- High gradient $x_i - x_j \Rightarrow \sigma = 1$
- No gradient $\Rightarrow \sigma = 0$
- Finite α, low gradient $\Rightarrow$ $0 < \sigma < 1$ Piecewise smooth result
- $\alpha \to \infty$, approximation of ℓ_0 norm low gradient $\Rightarrow \sigma = 1$ Piecewise constant result

Chapter 5 : Nonconvex optimization using PWsheds

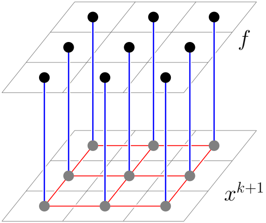
$$\min_x \underbrace{\sum_{e_{ij}\in E} \sigma(x_i - x_j)}_{\text{smoothness term}} + \underbrace{\lambda \sum_{v_i \in V} \sigma(x_i - f_i)}_{\text{data fidelity term}}$$

- Nonconvex energy
- Set the gradient of this energy to zero
- Fixed point iteration scheme with energy at step k :

Chapter 5 : Nonconvex optimization using PWsheds

$$\min_x \underbrace{\sum_{e_{ij}\in E} \sigma(x_i - x_j)}_{\text{smoothness term}} + \underbrace{\lambda \sum_{v_i \in V} \sigma(x_i - f_i)}_{\text{data fidelity term}}$$

- Nonconvex energy
- Set the gradient of this energy to zero
- Fixed point iteration scheme with energy at step k :

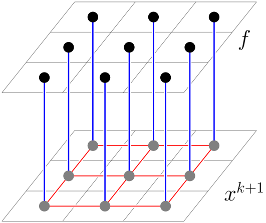


$$E_{k+1} = \sum_{e_{ij}\in E} e^{-\alpha(x_i^k - x_j^k)^2}(x_i^{k+1} - x_j^{k+1})^2 + \lambda \sum_{v_i\in V} e^{-\alpha(x_i^k - f_i)^2}(x_i^{k+1} - f_i)^2$$

Chapter 5 : Nonconvex optimization using PWsheds

$$\min_x \underbrace{\sum_{e_{ij}\in E} \sigma(x_i - x_j)}_{\text{smoothness term}} + \lambda \underbrace{\sum_{v_i\in V} \sigma(x_i - f_i)}_{\text{data fidelity term}}$$

- Nonconvex energy
- Set the gradient of this energy to zero
- Fixed point iteration scheme with energy at step k :



$$E_{k+1} = \sum_{e_{ij}\in E} \left(e^{-(x_i^k - x_j^k)^2}\right)^{\alpha} (x_i^{k+1} - x_j^{k+1})^2 + \lambda \sum_{v_i\in V} \left(e^{-(x_i^k - f_i)^2}\right)^{\alpha} (x_i^{k+1} - f_i)^2$$