Graph, cuts and PDE minimization for image processing

Xue-Cheng Tai, University of Bergen, Norway

Collaborations with:
Egil Bae, Jing Yuan.

January 20, 2012

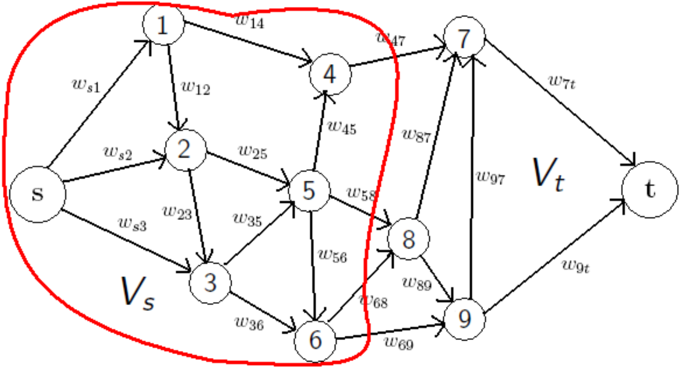
Research activities

- Started from 2000
- Noise removal
- Segmentation
- Registration
- Inpainting
- Diffusion Tensor image analysis
- Surfaces and edges detection, construction and analysis
- PET reconstruction–numerical algorithms
- ...

Landmarks

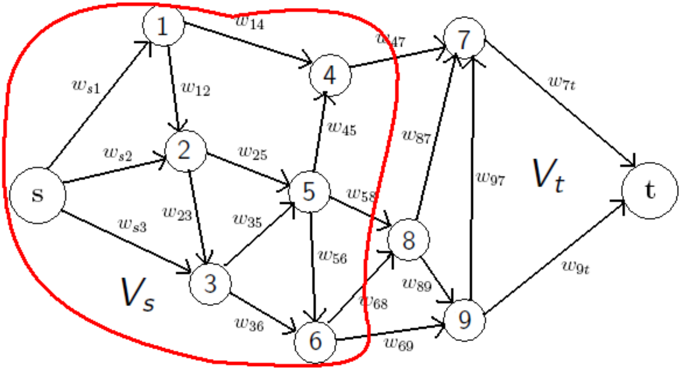
- AOS scheme – one popular scheme for nonlinear filtering (proposed in Lu-Tai-Neittaanmaki (1991, 1992), popularized by Weickert-etal-1998).
- MG mesh independent convergence proof (Tai- Numer Math. 2003)
- Higher order nonlinear filters (Lysaker-Lundervold-Tai IEEE TIP 2003)
- PCLSM (Lie-Lysaker-Tai (IEEE TIP, 2006, MATH COMP 2006))
- Global minimization of nonliear noncovex models (with Bae, Yuan, CVPR2008, SSVM2009, IJCV2009, ECCV2010, EMMCVPR2009, BMCV2010, cam-11-83).

Max-Flow / Min-Cut



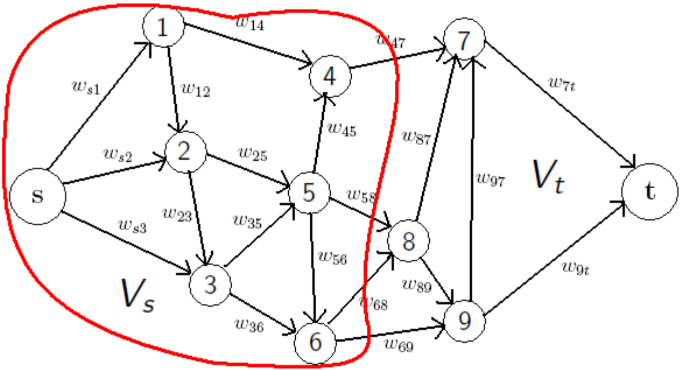
(V_s, V_t) is a cut, w_{ij} = cost of cutting edge(i, j)
cost of cut $c(V_s, V_t) = \sum_{i \in V_s, j \in V_t} w_{ij}$

Max-Flow / Min-Cut



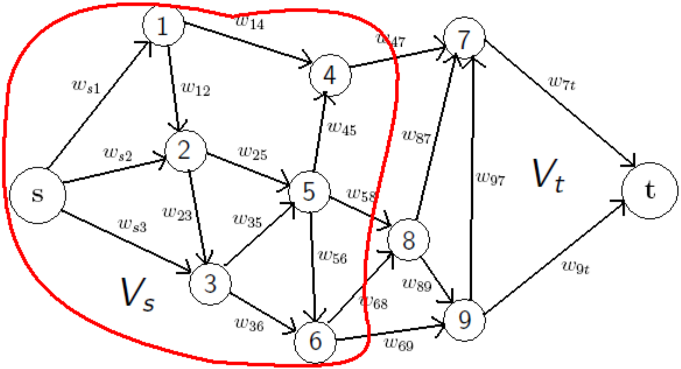
(V_s, V_t) is a cut, w_{ij} = cost of cutting edge(i, j)
cost of cut $c(V_s, V_t) = \sum_{i \in V_s, j \in V_t} w_{ij}$
Min-cut: find cut of minimum cost,

Max-Flow / Min-Cut



(V_s, V_t) is a cut, w_{ij} = cost of cutting edge(i, j)
cost of cut $c(V_s, V_t) = \sum_{i \in V_s, j \in V_t} w_{ij}$
Min-cut: find cut of minimum cost,
Max-Flow: Find the maximum amount of flow from s to t.

Max-Flow / Min-Cut

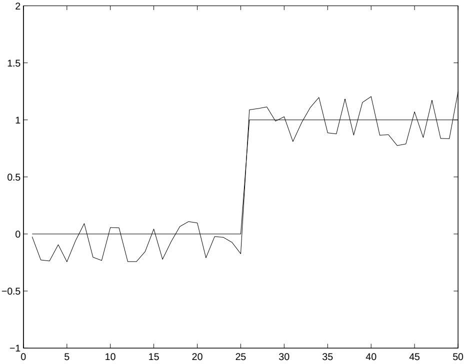


(V_s, V_t) is a cut, w_{ij} = cost of cutting edge(i,j)
cost of cut $c(V_s, V_t) = \sum_{i \in V_s, j \in V_t} w_{ij}$
Min-cut: find cut of minimum cost,
Max-Flow: Find the maximum amount of flow from s to t.

Max-flow = min-cut.

Graph-cut for image segmentation

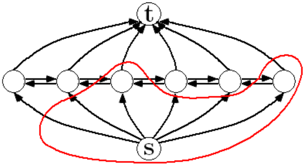
A simple 1d signal $I(x)$:



Graph-cut for images: Boykov-Kolmogorov (2001).

Graph-cut for image segmentation

The graph:



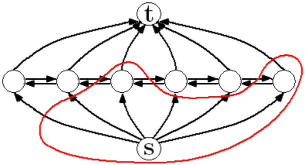
1	1	2	1	1	2

costs:

$$w_{s,p} = |I(p) - c_1|^2, w_{t,p} = |I(p) - c_2|^2, c_1 = 0, c_2 = 1.$$

Graph-cut for image segmentation

The graph:



1	1	2	1	1	2

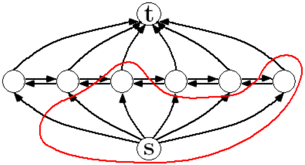
costs:

$$w_{s,p} = |I(p) - c_1|^2, w_{t,p} = |I(p) - c_2|^2, c_1 = 0, c_2 = 1.$$

What happens if $w_{p,q} = 0$??

Graph-cut for image segmentation

The graph:



1	1	2	1	1	2

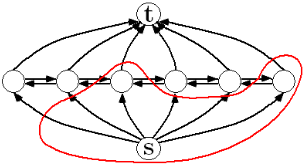
costs:

$$w_{s,p} = |I(p) - c_1|^2, w_{t,p} = |I(p) - c_2|^2, c_1 = 0, c_2 = 1.$$

What happens if $w_{p,q} = 0$?? It is the simple threshold.

Graph-cut for image segmentation

The graph:



1	1	2	1	1	2

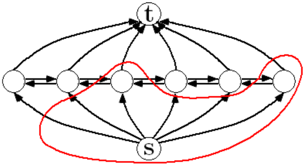
costs:

$$w_{s,p} = |I(p) - c_1|^2, w_{t,p} = |I(p) - c_2|^2, c_1 = 0, c_2 = 1.$$

What happens if $w_{p,q} = 0$?? It is the simple threshold.
What happens if $w_{p,q} = \alpha > 0$??

Graph-cut for image segmentation

The graph:



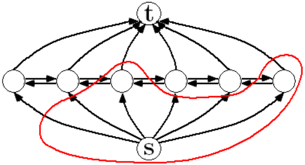
1	1	2	1	1	2

costs:

$$w_{s,p} = |I(p) - c_1|^2, w_{t,p} = |I(p) - c_2|^2, c_1 = 0, c_2 = 1.$$

What happens if $w_{p,q} = 0$?? It is the simple threshold.
What happens if $w_{p,q} = \alpha > 0$?? It is not the simple threshold.

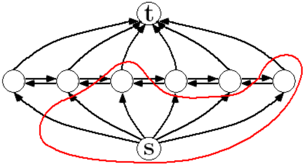
Relation with k-mean ($\alpha = 0$)



1	1	2	1	1	2

- Given c_1 and c_2.

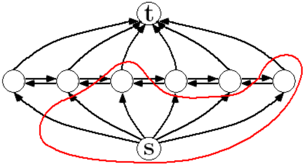
Relation with k-mean ($\alpha = 0$)



1	1	2	1	1	2

- Given c_1 and c_2.
- use cut (threshold) to get Ω_1 and Ω_2.

Relation with k-mean ($\alpha = 0$)

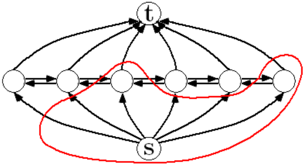


1	1	2	1	1	2

- Given c_1 and c_2.
- use cut (threshold) to get Ω_1 and Ω_2.
- update

$$c_i = \frac{\int_{\Omega_i} I(x)}{Area(\Omega_i)}, i = 1, 2.$$

Relation with k-mean ($\alpha = 0$)



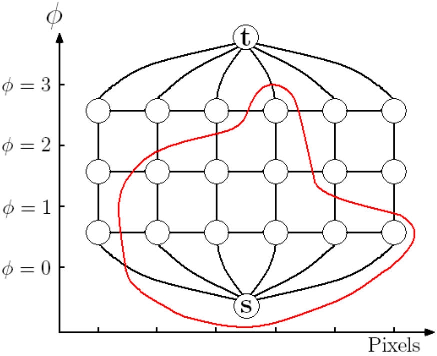
1	1	2	1	1	2

- Given c_1 and c_2.
- use cut (threshold) to get Ω_1 and Ω_2.
- update

$$c_i = \frac{\int_{\Omega_i} I(x)}{Area(\Omega_i)}, i = 1, 2.$$

- go to the next iteration.

Multiphases



0	2	2	3	1	1

Costs: $|I(p) - c_0|^2, |I(p) - c_1|^2, |I(p) - c_2|^2, |I(p) - c_3|^2$.

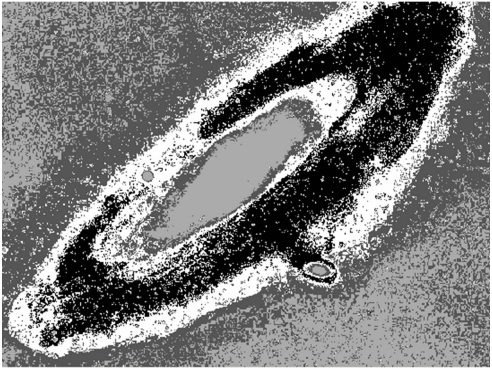
Ref: Ishikawa (PAMI 2003), Bae-Tai (SSVM 2009).

k-mean model ($\alpha = 0$)

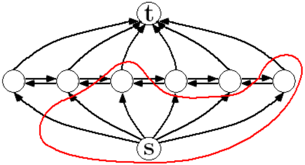
k-mean algorithm is an alternating minimization procedure for:

$$\min_{c_i,\Omega_i} \sum_i \int_{\Omega_i} (I(x) - c_i)^2.$$

Problems with k-mean model ($\alpha = 0$)



Regularized Graph-cut: $\alpha \neq 0$



1	1	2	1	1	2

costs:

$$w_{s,p} = |I(p) - c_1|^2, w_{t,p} = |I(p) - c_2|^2, w_{p,q} = \alpha.$$

The corresponding minimization problem:

$$\min_{u(p)\in\{1,2\}} \sum_{p\in\Omega_1} |I(p) - c_1|^2 + \sum_{p\in\Omega_2} |I(p) - c_2|^2 + \alpha \sum_{p} \sum_{q\in N(p)} |u(p) - u(q)|.$$

Discrete vs continuous

Discrete minimization:

$$\min_{u(p)\in\{1,2\}} \sum_{p\in\Omega_1} |I(p)-c_1|^2+\sum_{p\in\Omega_2} |I(p)-c_2|^2+\alpha\sum_{p}\sum_{q\in N(p)} |u(p)-u(q)|.$$

Continuous minimization:

$$\min_{u(x)\in\{1,2\}} \int_{\Omega_1} |I(x)-c_1|^2 + \int_{\Omega_2} |I(x)-c_2|^2 + \alpha \int_{\Omega} |Du|.$$

$$\min_{u(x)\in\{1,2\}} \int_{\Omega} |I(x)-c_1|^2(2-u) + \int_{\Omega} |I(x)-c_2|^2(u-1) + \alpha \int_{\Omega} |Du|.$$

Discrete vs continuous

Allowed: Discrete minimization:

$$\min_{u(p)\in\{0,1\}} \sum_{p\in\Omega_1} |I(p)-c_1|^2+\sum_{p\in\Omega_2} |I(p)-c_2|^2+\alpha\sum_{p}\sum_{q\in N(p)} |u(p)-u(q)|^2.$$

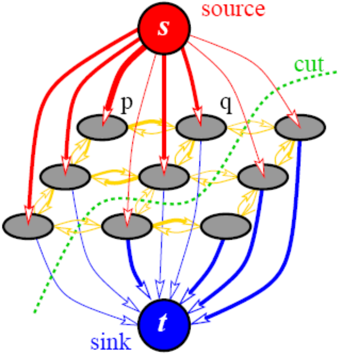
Not allowed Continuous minimization:

$$\min_{u(x)\in\{0,1\}} \int_{\Omega_1} |I(x)-c_1|^2 + \int_{\Omega_2} |I(x)-c_2|^2 + \alpha \int_{\Omega} |Du|^2.$$

$$\min_{u(x)\in\{0,1\}} \int_{\Omega} |I(x)-c_1|^2)(1-u) + \int_{\Omega} |I(x)-c_2|^2 u + \alpha \int_{\Omega} |Du|^2.$$

Higher dimensional problems

A graph for 2D images:



Continuous cut - 2-phase

Max-flow/min-cut in the continuous setting.

Max-Flow / Min-Cut (graph cut)

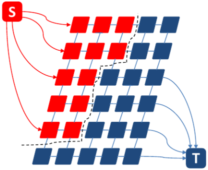


Figure: Graph used for discrete binary labeling

- Can solve binary problems of the form

$$\min_{\lambda \in \{0,1\}} \alpha \sum_{v \in \mathcal{V}} \sum_{u \in \mathcal{N}_v^k} w_{vu} |\lambda_v - \lambda_u| + \sum_{v \in \mathcal{V}} \lambda_v f_1(v) + (1 - \lambda_v) f_2(v)$$

- Connection between cut and λ

$$\lambda_v = 1, \text{ iff } v \in V_s$$

$$\lambda_v = 0, \text{ iff } v \in V_t$$

Max-Flow / Min-Cut (graph cut)

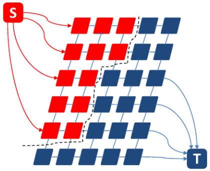


Figure: Graph used for discrete binary labeling

Max-flow formulation

$$\max_{p_s} \sum_{v\in\mathcal{V}\backslash\{s,t\}} p_s(v)$$

subject to

$$\begin{aligned} q(v,u) &\leq w_{vu}, & \forall (v,u) \in \mathcal{V}\times\mathcal{V} \\ 0 \leq p_s(v) &\leq f_1(v), & \forall v \in \mathcal{V}\backslash\{s,t\}; \\ 0 \leq p_t(v) &\leq f_2(v), & \forall v \in \mathcal{V}\backslash\{s,t\}; \\ \Big(\sum_{u\in N(v)} q(v,u)\Big) - p_s(v) + p_t(v) &= 0, & \forall v \in \mathcal{V}\backslash\{s,t\};. \end{aligned}$$

Continuous Max-Flow and Min-Cut

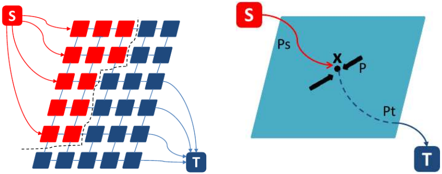


Figure: Graph used for binary labeling: Discrete (left) vs. Continuous (right)

Continuous max-flow formulation

$$\sup_{p_s, p_t, q} \int_\Omega p_s(x)\, dx$$

subject to

$$\begin{aligned} |q(x)| = \sqrt{q_1^2(x) + q_2^2(x)} \ &\leq \ g(x), \quad && \forall x \in \Omega; \\ p_s(x) \ &\leq \ f_1(x), \quad && \forall x \in \Omega; \\ p_t(x) \ &\leq \ f_2(x), \quad && \forall x \in \Omega; \\ \operatorname{div} q(x) - p_s(x) + p_t(x) \ &= \ 0, \quad && \text{a.e. } x \in \Omega. \end{aligned}$$

Continuous Max-Flow and Min-Cut

Lagrange multiplier λ for flow conservation condition

$$\operatorname{div} q(x) - p_s(x) + p_t(x) \ = \ 0, \quad \text{a.e. } x \in \Omega.$$

yields primal-dual formulation

$$\sup_{p_s, p_t, q} \inf_{\lambda} \int_{\Omega} p_s + \lambda \big(\operatorname{div} q - p_s + p_t \big) \, dx$$

$$\text{s.t.} \quad p_s(x) \ \le \ f_1(x) \, , \quad p_t(x) \le f_2(x) \, , \quad |q(x)| \ \le \ g(x) \, .$$

Optimizing for flows p_s, p_t, q results in Chan et. al. (2006) relaxation

$$\min_{\lambda} \ \int_{\Omega} \big(1 - \lambda(x)\big) f_1(x) + \lambda(x) f_2(x) \, dx + g(x) \, |\nabla \lambda| \ dx \, .$$

subject to $\lambda(x) \in [0, 1], \ \forall x \in \Omega$

Continuous Max-Flow: Remarks

- Combinatorial problem is minimizing a energy functional. Not using the decent (gradient) info of the energy.
- Continuous max-flow/min-cut is a convex minimization problem. A lot of choices, can use decent (gradient) info.
- Continuous cut: may not be a cut during iterations. Will be a cut at convergence or by truncation.
- Not all combinatorial problem can be convexified.
- Guaranteed global minimize for: nonlinear, noncovex models, independent of initial values.

Continuous Max-Flow and Min-Cut

Multiplier-Based Maximal-Flow Algorithm
Augmented lagrangian functional (Glowinski & Le Tallec, 1989)

$$L_c(p_s, p_t, q, \lambda) \; := \; \int_\Omega p_s \, dx + \lambda\big(\operatorname{div} q - p_s + p_t \big) - \frac{c}{2} | \operatorname{div} q - p_s + p_t |^2 \, dx.$$

minmax subject to:
$p_s(x) \; \leq \; f_1(x) \, , \quad p_t(x) \leq f_2(x) \, , \quad |q(x)| \; \leq \; g(x)$
ADMM algorithm: For k=1,... until convergence, solve

$$q^{k+1} \; := \arg \max_{\|q\|_\infty \leq \alpha} \; L_c(p_s^k, p_t^k, q, \lambda^k)$$

$$p_s^{k+1} \; := \arg \max_{p_s(x) \leq f_1(x)} \; L_c(p_s, p_t^k, q^{k+1}, \lambda^k)$$

$$p_t^{k+1} \; := \arg \max_{p_t(x) \leq f_2(x)} \; L_c(p_s^{k+1}, p_t, q^{k+1}, \lambda^k)$$

$$\lambda^{k+1} \; = \; \lambda^k - c \, (\operatorname{div} q^{k+1} - p_s^{k+1} + p_t^{k+1})$$

Continuous Max-Flow and Min-Cut

Other algorithms for solving relaxed problem

- Bresson et. al.
  - fix μ^k and solve ROF problem

$$\lambda^{k+1} := \arg\min_{\lambda} \left\{ \alpha \int_{\Omega} |\nabla \lambda(x)| \, dx + \frac{1}{2\theta} \|\lambda(x) - \mu^k(x)\|^2 \right\}$$

  - fix λ^{k+1} and solve

$$\mu^{k+1} := \arg \min_{\mu \in [0,1]} \left\{ \frac{1}{2\theta} \|\mu(x) - \lambda^{k+1}\|^2 + \int_{\Omega} \mu(x) \big(f_1(x) - f_2(x) \big) \, dx \right\}$$

- Goldstein-Osher: Split Bregman / augmented lagrangian

Convergence

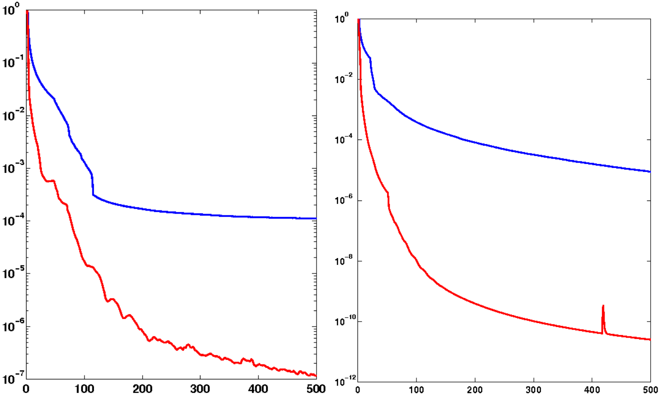
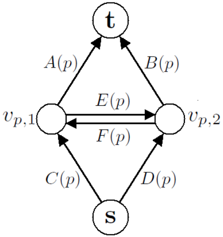
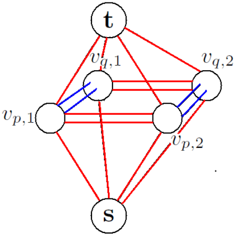


Figure: Red line: max-flow algorithm. Blue line: Splitting algorithm (Bresson et. al. 2007)

CV Graph construction (known c_i)



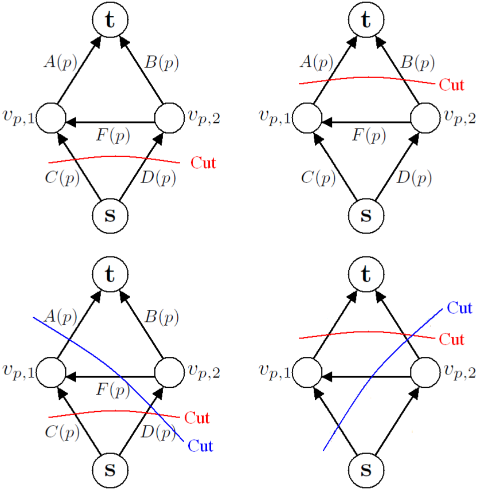
One pixel



two pixels

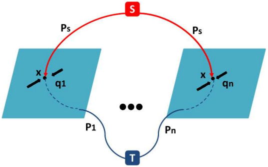
- Associate two vertices to each grid point ($v_{p,1}$ and $v_{p,2}$)
- For any cut (V_s, V_t)
  - If $v_{p,i} \in V_s$ then $\phi^i = 1$ for $i = 1, 2$
  - If $v_{p,i} \in V_t$ then $\phi^i = 0$ for $i = 1, 2$
- Figure left: graph corresponding to one grid point p
- Figure right: graph corresponding to two grid points p and q
  - Red: Data edges, constituting $E^{data}(\phi_1, \phi_2)$
  - Blue: Regularization edges with weight w_{pq}

Cuts for the CV-graph(Bae-Tai, EMMCVPR2009)



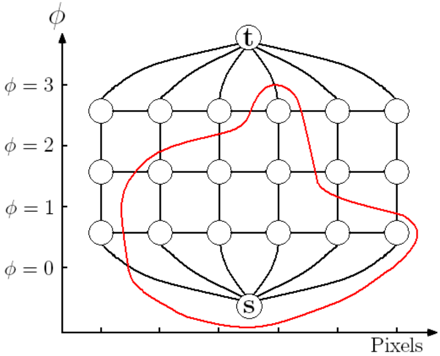
Multiple Phases: Convex Relaxed Potts Model (CR-PM) –Yuan-Bae-Tai (ECCV'10)

Continuous Max-Flow Model (CMF-PM)



1. n copies Ω_i, $i = 1, \dots, n$, of Ω;
2. For $\forall x \in \Omega$, the same source flow $p_s(x)$ from the source s to x at Ω_i, $i = 1, \dots, n$, simultaneously;
3. For $\forall x \in \Omega$, the sink flow $p_i(x)$ from x at Ω_i, $i = 1, \dots, n$, of Ω to the sink t. $p_i(x)$, $i = 1, \dots, n$, may be different one by one;
4. The spatial flow $q_i(x)$, $i = 1, \dots, n$ defined within each Ω_i.

Multiphases



0	2	2	3	1	1

Costs: $|I(p) - c_0|^2, |I(p) - c_1|^2, |I(p) - c_2|^2, |I(p) - c_3|^2$.

Ref: Ishikawa (PAMI 2003), Bae-Tai (SSVM 2009).

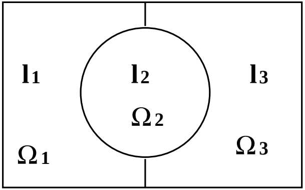
Fast Continuous Max-Flow Approaches to Medical Image Segmentation

Jing Yuan and Xue-Cheng Tai

Math. Institute, Univ. Bergen

Jan. 17, 2012

Minimal Image Partition



(a) 3 partitions

Minimal Image Partition: partitions with the minimal total perimeter:

$$\min_{\{\Omega_i\}_{i=1}^n} \sum_{i=1}^n \int_{\Omega_i} \rho(l_i, x)\, dx + \alpha \sum_{i=1}^n |\partial \Omega_i|$$

subject to n *'disjoint'* regions

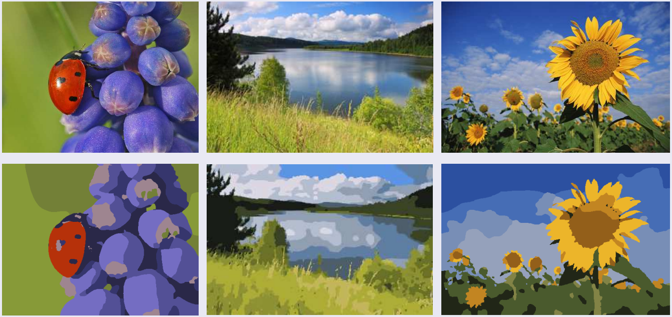
$$\cup_{i=1}^n \Omega_i \, = \, \Omega \, , \; \Omega_k \cap \Omega_l \, = \, \emptyset \, , \; \forall k \neq l \, .$$

Minimal Image Partition

Applications: Image Segmentation

$$\rho(l_i, x) = -\log P(l_i, x), \quad i = 1, \dots, n$$

Related works (Convex Relax.): *Pocks et al CVPR'09, Lellmann et al SSVM'09+ECCV'10, Bae-Yuan-Tai (CAM'09), Yuan-Bae-Tai(ECCV'10)* etc.

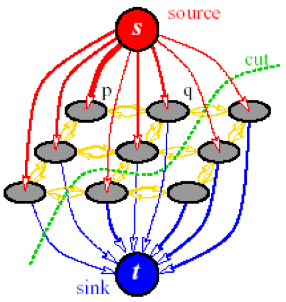


Old and new approaches

Discrete Settings: Graph and MRF

Graph $G = (V, E)$, V: nodes, E: edges.
Formulated by MRF:

$$\mathcal{E} = \sum_{p \in V} \mathcal{F}(l_p) + \sum_{(p,q) \in E} \mathcal{V}(l_p, l_q) \, .$$



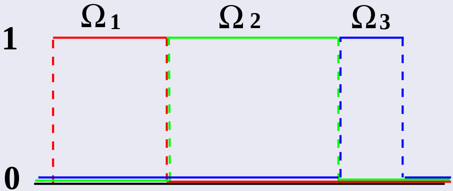
Algorithm: Graph-Cuts (*Boykov PAMI'01, Kolmogorov PAMI'04*), **Roof Duality Relaxation/QPBO** (*Hammer Math. Prog.'84, Kolmogorov PAMI'07, Rother CVPR'07*) etc.

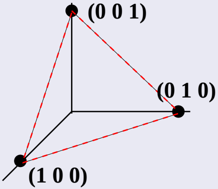
$$\text{Pairwise Prior: } \mathcal{V}(l_p, l_q) \; := \begin{cases} \alpha \, , & \text{when } l_p \neq l_q \\ 0 \, , & \text{when } l_p = l_q \end{cases} \, .$$

Pros and Cons: Efficient with an optim. bound, but suffers from metric errors and hard to be parallelized (a)!

Old and new approaches

Continuous Settings: Snakes, Level Sets, PCLSM





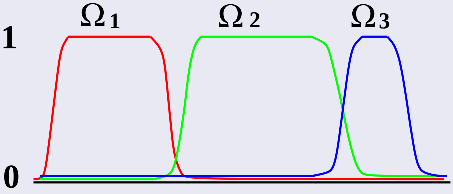
$$\min_{u(x)} \quad \sum_{i=1}^{n} \int_\Omega u_i(x)\,\rho(l_i,x)\,dx \;+\; \alpha \sum_{i=1}^{n} \int_\Omega |\nabla u_i|\;dx$$

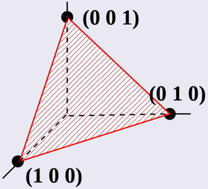
$$\text{s.t.} \quad \sum_{i=1}^{n} u_i(x) \;=\; 1\,, \quad {\color{red} u_i(x) \in \{0,1\}}\,; \quad \forall x \in \Omega$$

Algorithm: Snakes (Kass-et-al'95), Level sets (Osher & Fedkiw'02), Level sets (Chan & Vese, IJCV'02), Variant level sets (Tai et al, IEEE Img. Proc.'06) etc.

Why Convex Relaxation?

Continuous Settings: Convex Relaxation Model



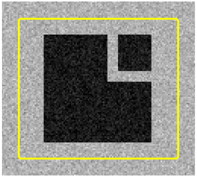


$$\min_{u(x)\in\triangle_+} \sum_{i=1}^{n} \int_\Omega u_i(x)\,\rho(l_i,x)\,dx \;+\; \alpha \sum_{i=1}^{n} \int_\Omega |\nabla u_i|\;dx$$

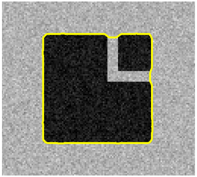
$$\text{where } \triangle_+ \colon \sum_{i=1}^{n} u_i(x) \;=\; 1\,, \quad u_i(x) \in [0,1]\,; \quad \forall x \in \Omega\,.$$

Keynote works: *Nikolova-Esedoglu-Chan SIAM App. Math'06*, *Chambolle-Cremers-Pock Report'08*, *Lellmann et al Report'08*, *Bae-Yuan-Tai CAM'09-75*.

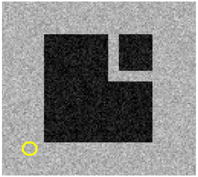
Why Convex Relaxation?



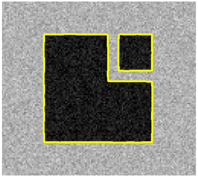
(a) Chan+Vese TIM'02



(b) Chan+Vese TIM'02



(c) Bresson et al IJCV'07



(d) Bresson et al IJCV'07

Why convex?

Level Sets

- Nonconvex, converges to a local optimum;
- Slow;
- Quite dependent on initial settings!

Convex Relaxation

- Convex, converges to the global optimum;
- Fast;
- Not dependent on initial settings!

Two Phases: Continuous Min-Cut Model (CMC)

Given two data costs $C_s(x)$ and $C_t(x)$, Nikolova et al proved

$$\min_{u(x) \in \{0,1\}} \int_\Omega \big\{ \big(1 - u(x)\big) C_s(x) + u(x) C_t(x) \big\} \, dx + \alpha \int_\Omega |\nabla u| \, dx$$

can be solved exactly and globally by **its convex relaxation**:

$$\min_{u(x) \in [0,1]} \int_\Omega \big\{ \big(1 - u(x)\big) C_s(x) + u(x) C_t(x) \big\} \, dx + \alpha \int_\Omega |\nabla u| \, dx \, ,$$

the so-called **Continuous Min-Cut Model (CMC)**!

The same result was also proved by *Yuan-Bae-Tai(CAM'10-61)* through its dual flow-maximization model.

Two Phases: Continuous Min-Cut Model (CMC)

Continuous Max-Flow Method by *Yuan-Bae-Tai(CVPR'10)*

Fact: For an image graph, fast min-cut algorithms are based on max-flows!

Continuous Max-Flow Model (CMF):

$$\max_{p_s, p_t, p} \int_\Omega p_s(x)\, dx \quad \textbf{s.t.}$$

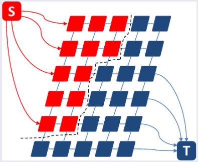
$$\text{Flow capacities: } |p(x)| \leq C(x) ;$$

$$p_s(x) \leq C_s(x) ;$$

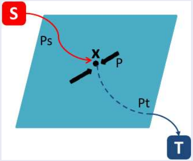
$$p_t(x) \leq C_t(x) ;$$

$$\text{Flow conservation: } \operatorname{div} p(x) - p_s(x) + p_t(x) = 0 .$$

Duality: CMF $\Longleftrightarrow$ CMC !



(a) Graph



(b) Continuous Ω

Two Phases: Continuous Min-Cut Model (CMC)

Continuous Max-Flow Algorithm (ALM based)

Define the augmented Lagrangian function:

$$L_c(p_s, p_t, p, u) = \int_\Omega p_s(x)\,dx + \langle u, \operatorname{div} p - p_s + p_t \rangle - \frac{c}{2} \| \operatorname{div} p - p_s + p_t \|^2$$

each k-th iteration explores the following steps

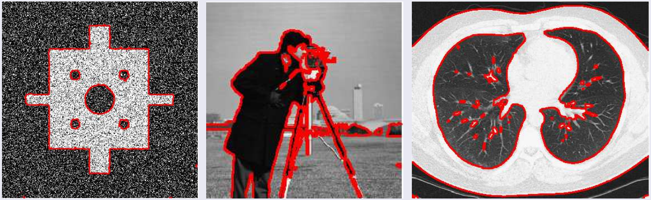
1. **Maximize over** $p^{k+1} := \arg\max_{|p| \leq C} L_c(p_s^k, p_t^k, p, u^k)$;
2. **Maximize over** $p_s^{k+1} := \arg\max_{p_s \leq C_s} L_c(p_s, p_t^k, p^{k+1}, u^k)$;
3. **Maximize over** $p_t^{k+1} := \arg\max_{p_t \leq C_t} L_c(p_s^{k+1}, p_t, p^{k+1}, u^k)$;
4. **Update** u by $u^{k+1} = u^k - c\,(\operatorname{div} p^{k+1} - p_s^{k+1} + p_t^{k+1})$.

Note: Flow-maximization steps $+$ one labeling update!

Two Phases: Continuous Min-Cut Model (CMC)

Validation Experiments

By Windows 7 (Intel Xeon X5460 CPU, RAM 32 Gb), GPU (Nvidia Quadro FX 4800)

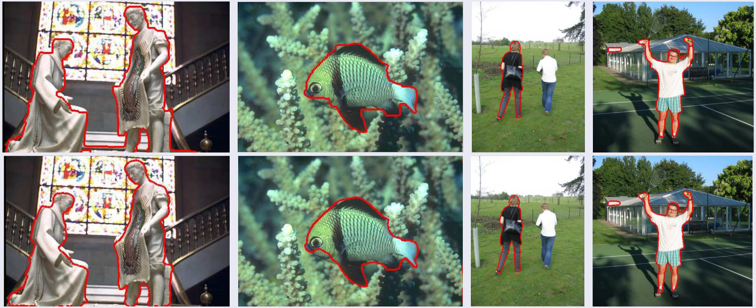


B-K Graph-Cut	173.0 msec	39.2 msec	117.2 msec
CMF (CPU)/(GPU)	170/33 msec	58.1/12.2 msec	106/25 msec

Two Phases: Continuous Min-Cut Model (CMC)

More Experiments

Experiments over the Grabcut data set (50 images), only show 4 images.



1st Row: by Graph Cut. **2nd Row:** CMC results. Avg. time by graph-cut: 580 msec, avg. time by cont. max-flow (GPU): 103 msec!

Multiple Phases: Convex Relaxed Potts Model (CR-PM)

Convex Relaxed Potts Model (CR-PM) – Yuan-Bae-Tai (ECCV'10)

$$\textbf{CR-PM} \quad \min_{u_i(x)\in[0,1]} \sum_{i=1}^{n} \Big\{ \int_\Omega u_i(x)\rho(l_i,x)\,dx \, + \, \alpha \int_\Omega |\nabla u_i| \, dx \Big\}$$

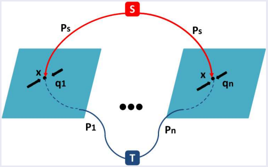
$$\text{s.t.} \quad \sum_{i=1}^{n} u_i(x) \, = \, 1 \, , \; \forall x \in \Omega$$

Note: No global discrete optimum or rounding scheme is guaranteed, different to CMC!

Analysis on the optimality bound can be found in *Chambolle et al (Report'08), Bae et al (CAM'09-75), Lellmann et al (EMMCVPR'11)*.

Multiple Phases: Convex Relaxed Potts Model (CR-PM) –Yuan-Bae-Tai (ECCV’10)

Continuous Max-Flow Model (CMF-PM)



1. n copies Ω_i, $i = 1, \ldots, n$, of Ω;
2. For $\forall x \in \Omega$, the same source flow $p_s(x)$ from the source s to x at Ω_i, $i = 1, \ldots, n$, simultaneously;
3. For $\forall x \in \Omega$, the sink flow $p_i(x)$ from x at Ω_i, $i = 1, \ldots, n$, of Ω to the sink t. $p_i(x)$, $i = 1, \ldots, n$, may be different one by one;
4. The spatial flow $q_i(x)$, $i = 1, \ldots, n$ defined within each Ω_i.

Multiple Phases: Convex Relaxed Potts Model (CR-PM)

Continuous Max-Flow Model (CMF-PM)

$$\max_{p_s, p, q} \int_\Omega p_s(x)\, dx$$

$$\text{s.t.} \quad \underbrace{|q_i(x)| \leq \alpha\,, \quad p_i(x) \leq \rho(l_i, x)\,; \quad i = 1 \ldots n\,;}_{\textbf{Flow Capacities}}$$

$$\underbrace{\operatorname{div} q_i(x) - p_s(x) + p_i(x) = 0\,, \quad i = 1 \ldots n\,.}_{\textbf{Flow Conservation}}$$

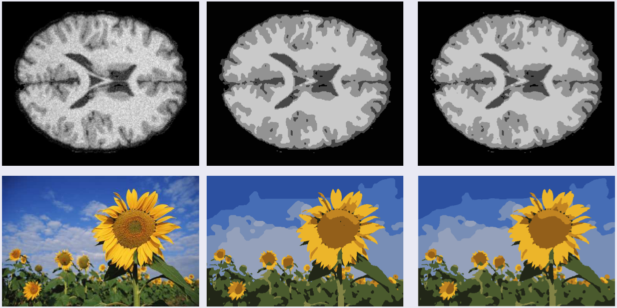
Duality: CMF-PM $\Longleftrightarrow$ CR-PM *Yuan et al (ECCV'10)*

Continuous Max-Flow Algorithm is designed based on ALM, which outperforms the state of art methods by a factor 4! It can be easily implemented by parallelism over GPU.

Continuous Max-Flow Algorithm (CMF-PM)

Experiments

Given image f and l_i, $\rho(l_i, x) = |f - l_i|^2$ and performs piecewise constant Mumford-Shah model:

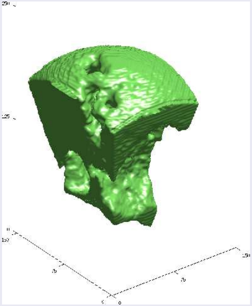


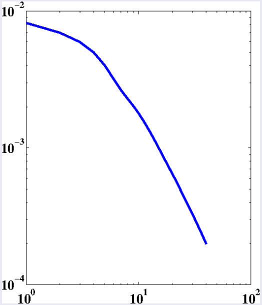
Given f Graph-cut Proposed method

The proposed max-flow approach outperforms the state of art methods (Zach et al's ECCV08, Lellmann et al's SSVM09) by the factor 4!

Medical Imaging Applications

3D / 4D Ultrasound Cardiac Segmentation

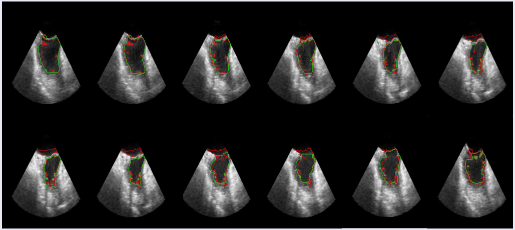


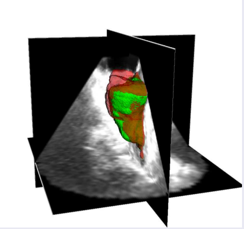


Simple Case: The size of the given 3D image volume is $150 \times 150 \times 250$ pixels, error bound for convergence is 2×10^{-4}. The fast continuous max-flow algorithm (GPU: NVidia Tesla C1060) takes 38 iterations to converge, around 0.280 sec!

Medical Imaging Applications

3D / 4D Ultrasound Cardiac Segmentation

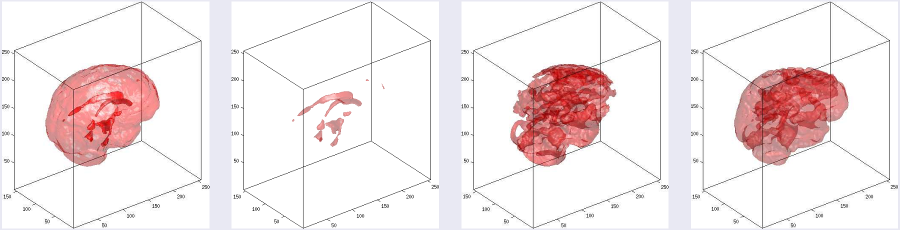




4D Experiments: (Rajchl, M. et al, SPIE'12) Fisher-Tippett distribution is taken as the image model. **Left Fig.** shows segmentation results over a single cardiac cycle. **Right Fig.** shows both gold standard and segmented contours are surface rendered and visualized over time. Resulting error on volumes (over 10^6 voxels) is $2.15 \pm 0.6mm$ and dice coefficient is 0.82. **The avarage optimization time (over NVidia GX 580) is under 0.1 sec!**

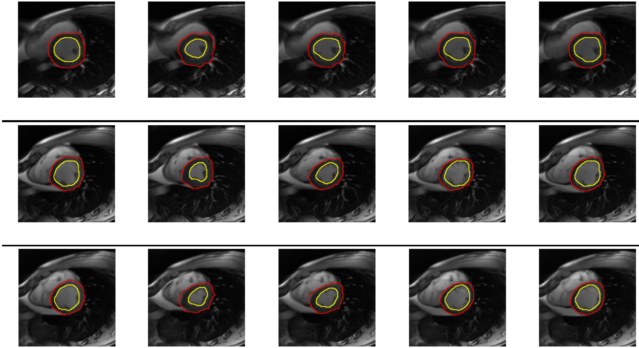
Medical Imaging Applications

Multi-Region 3D MRI Brain Segmentation



3D Brain Segmentation: 3D MRI volume is given as 256 × 156 × 256 voxels. The gaussian distribution is taken as image model. 4 brain layers are computed by the continuous max-flow algorithm. **The optimization takes** 0.63 **sec. (GPU: NVidia Tesla C1060)!**

Medical Imaging Applications: LV Segmentation with Geometrical Constraints



Subject 1, basal
frames 142, 150
153, 155, and 160

Subject 1, mid-cavity
frames 82, 87
90, 92, and 100

Subject 1, apical
frames 62, 69
70, 74, and 80

Figure: Results for one subject at different slice levels: Basal (1st row); Mid-cavity(2nd row); Apical(3th row).

Medical Imaging Applications: LV Segmentation with Geometrical Constraints

Methods	Dice Metric		Reliab.		*RMSE*		*time*
	Cavity	Myocardium	Cav.	Myo.	Cav.	Myo.	(sec.)
CMFM	0.92 ± 0.07	0.80 ± 0.10	1	0.70	1.35	2.05	1.79
LSM	0.88 ± 0.09	0.81 ± 0.10	0.89	0.75	2.46	1.89	15.96

Table: *DM*, Reliability of *DM*, the average *RMSE* (in pixels), and average time load to process a frame over 20 subjects (2280 images) for the proposed method (*CMFM*) and *LSM* (Ismail et al's TMI'09); Statistics of the *DM* expressed as mean ± standard deviation; Algorithms have been run over 2 *GHZ CPU*.

Convex approach with both l_i and Ω_i

$$\min_{\Omega_i, c_i} \sum_{i=1}^{n} \int_{\Omega_i} |I(x) - l_i|^\beta dx + \alpha \sum_{i=1}^{n} \int_{\Omega_i} ds.$$

We only know 3 approaches:

- Strandmark-Kahl-Overgaard (ICCV, 2009).
  (i) Two-phase; (ii) convex; (iii) Exhaust all possible pairing values.
- Brown-Chan-Bresson (cam-report 2010, IJCV, 2011).
  (i) Use lifting technique; (ii) In increase the dimension by one for each new l_i.
- Bae-T. (cam-11-83, 2011).
  (i) for multiphases; (ii) no increase of dimension; (iii) Quantized images (L values for l_is).
  Essential ideas: add more constraints to

$$u_i \in \{0, 1\}, \sum u_i = 1.$$

Source codes are available via Matlab website:

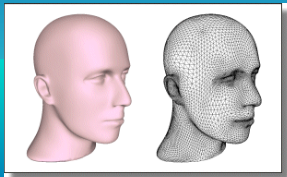
http://www.mathworks.com/matlabcentral/fileexchange/34

http://www.mathworks.com/matlabcentral/fileexchange/34

Applications of max-flow/min-cuts

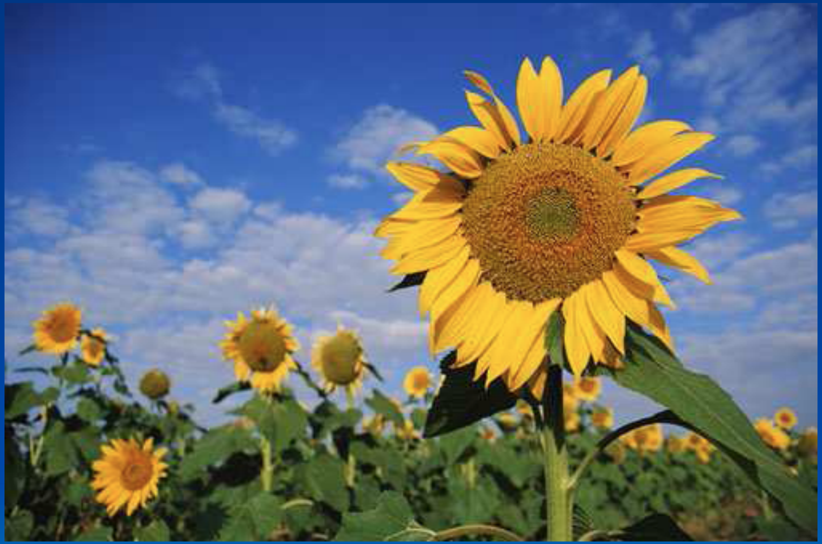
Department of Mathematics

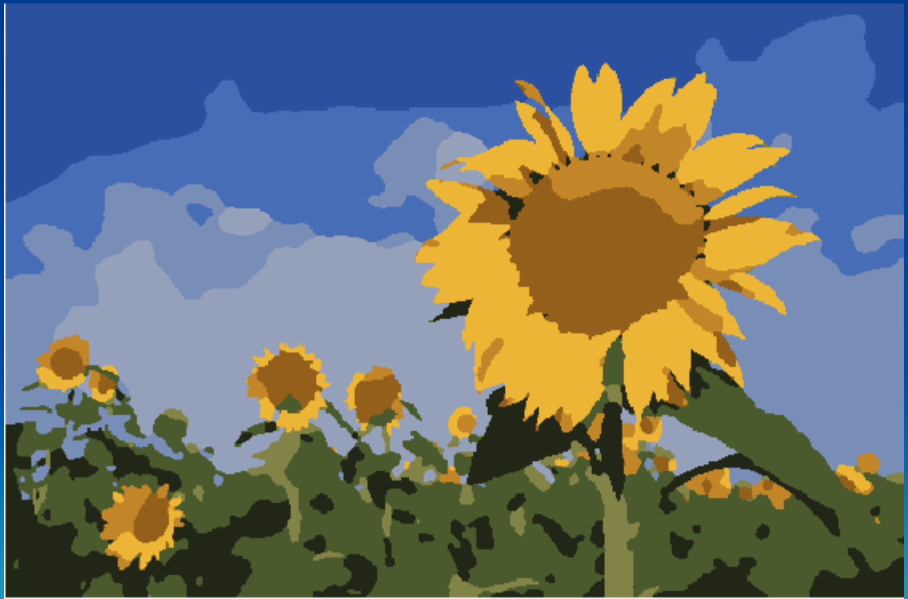
University of Bergen



Segmentation = restoration

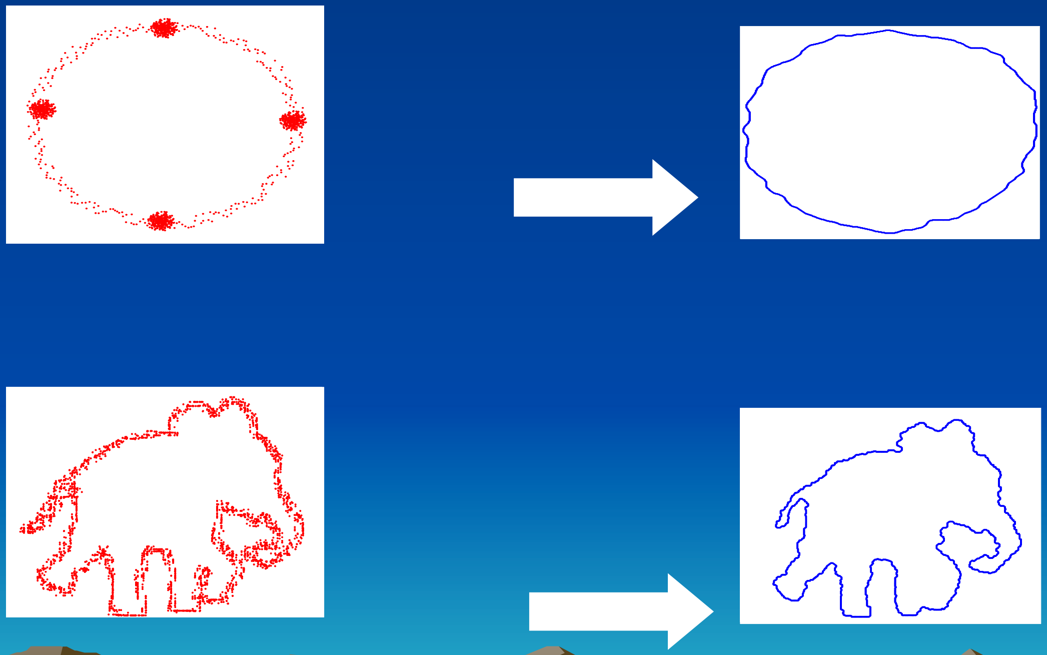
- Input image 32 bits, output 3 bits





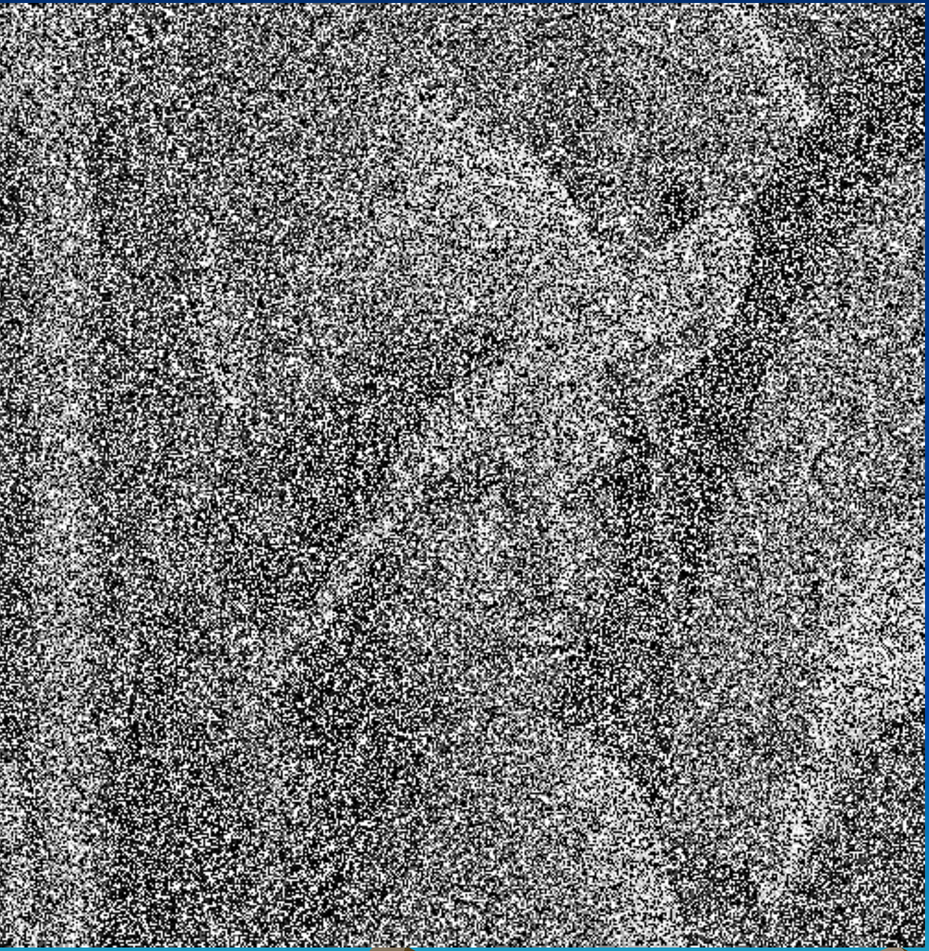
Curve construction

- Variational curve reconstruction based on graph-cuts
  - from points to curve
  - curve smoothing based on min-cut

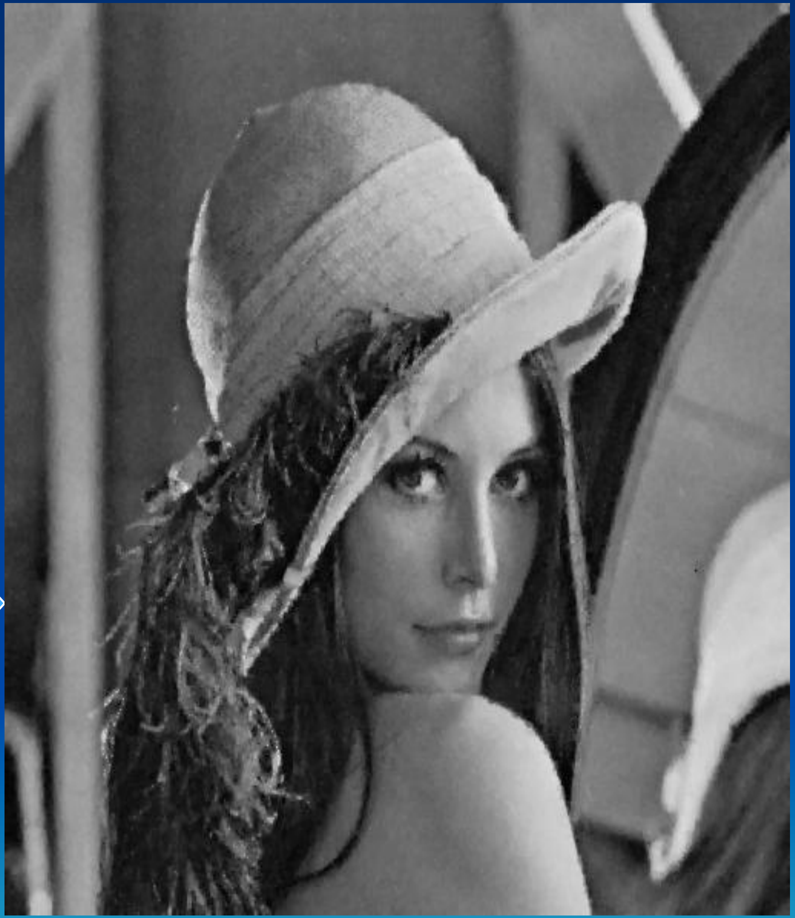


Restoration and Compression

- Image Restoration



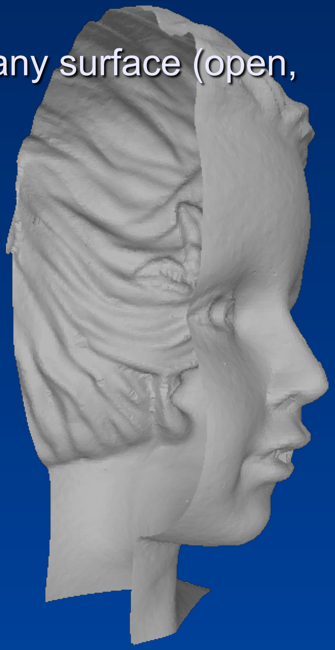
Input Image

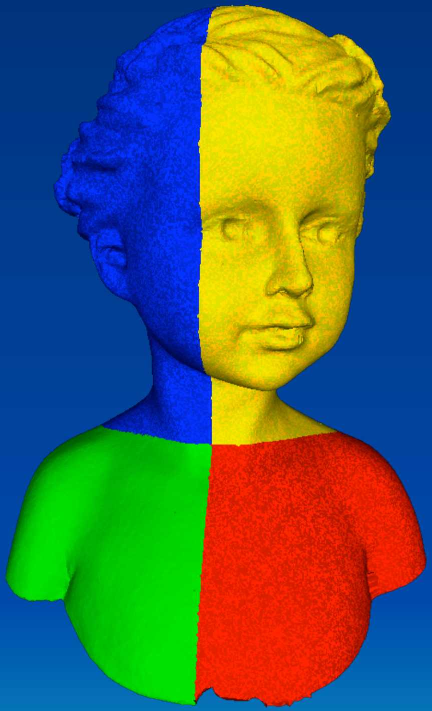


restored Image

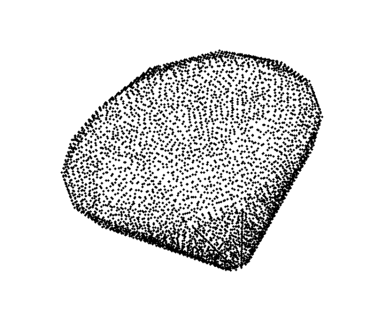
Surface re-construction:

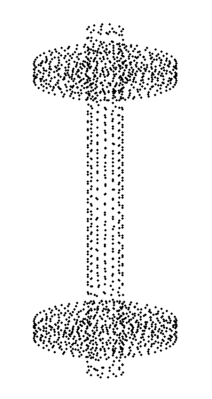
it is based on domain decomposition using graph cut. Can construct and process any surface (open, non-oriented) with very complicated structures.

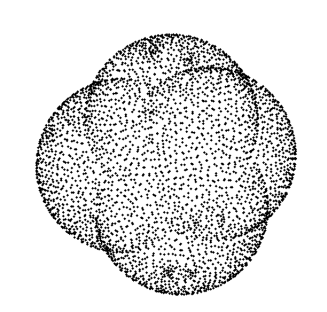


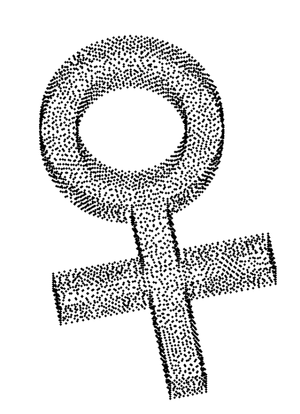


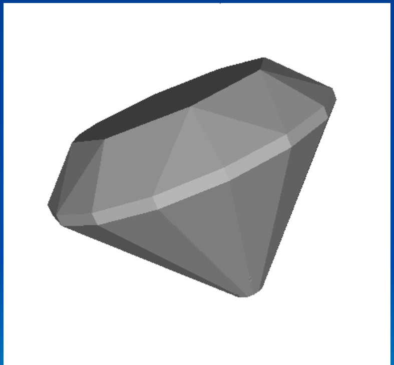
Surfaces with details and features

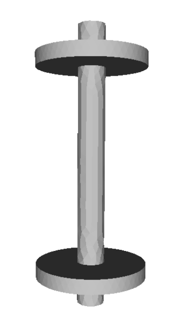


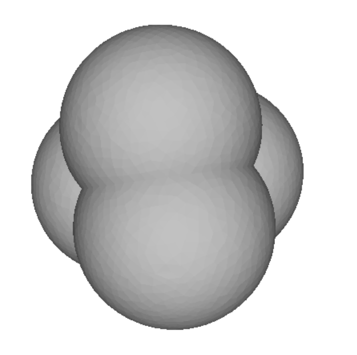


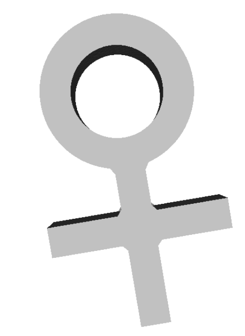






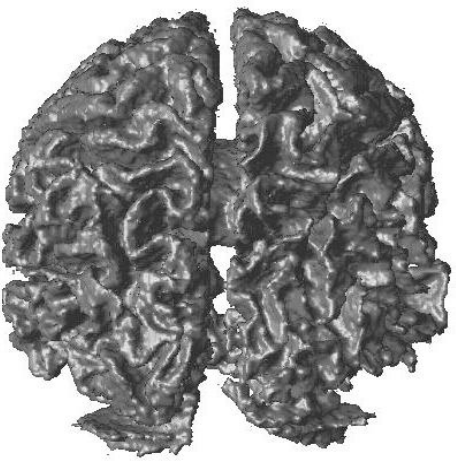


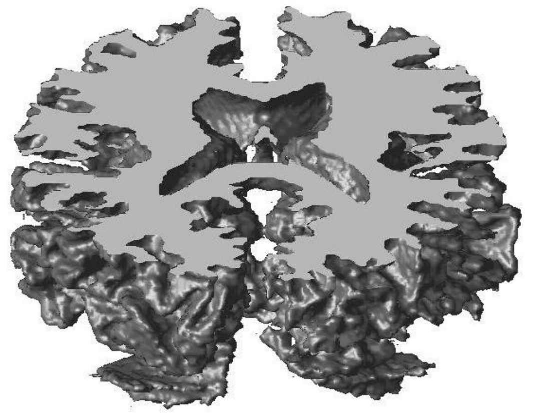




MRI brain image analysis

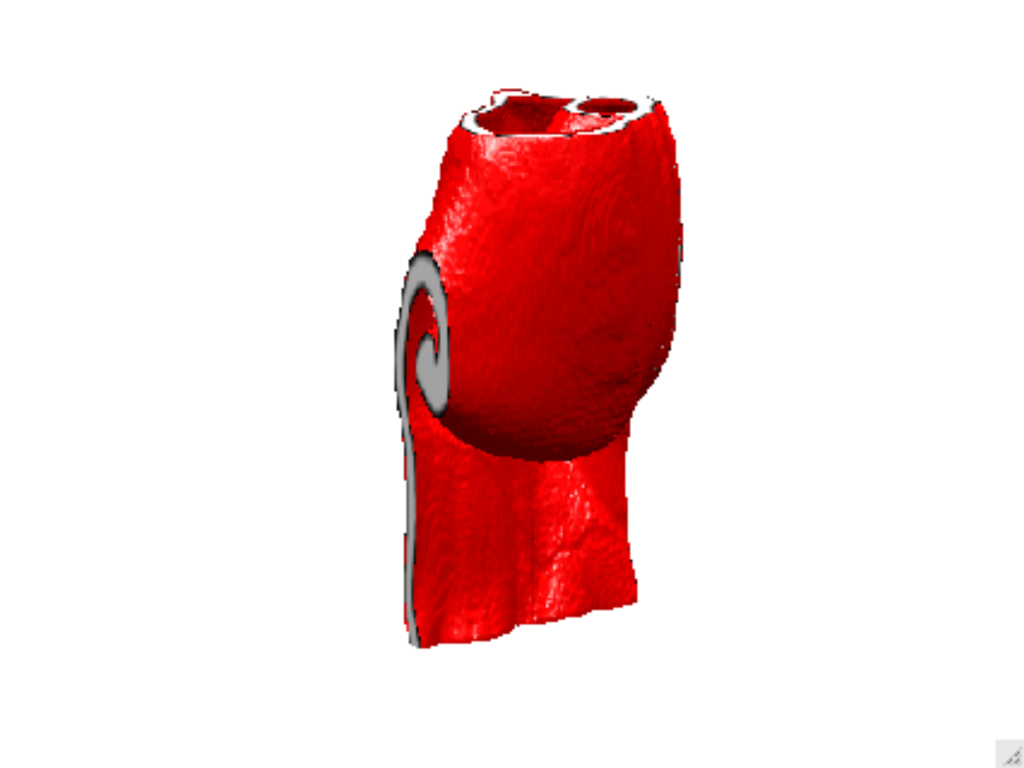
- Surface reconstruction and processing on 3D MRI medical image
  - New method: piecewise constant level-set method (PCLSM)
  - New Method: Graph-cut for multiphase surface reconstruction





Research Progress: Surface construction and object identification

The left is a CT scan of a real object. It is difficult to see the shape of the object.



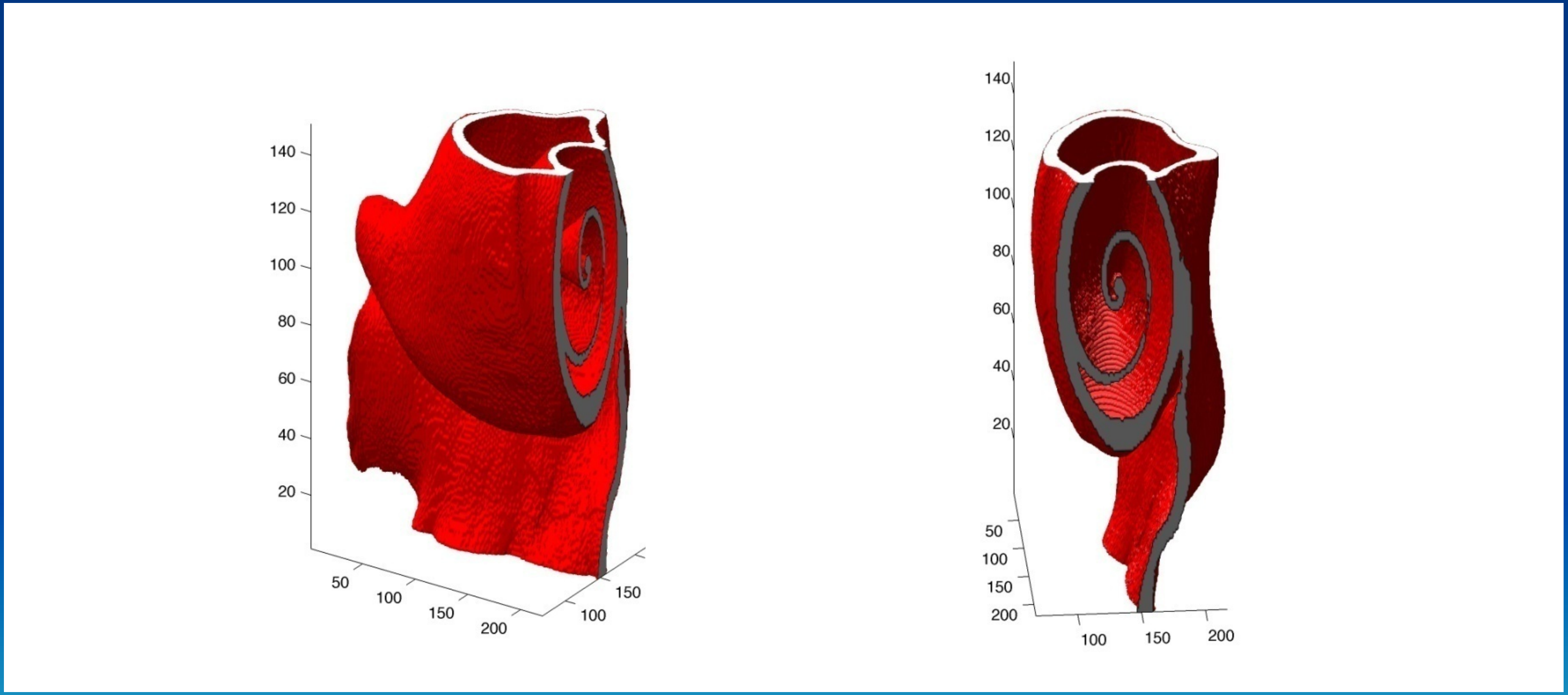
The right is the surface for the CT data constructed from methods developed in this project.

Applications:

1) Airport security control
2) Customs control
3) Industrial and medical CT
4) E-commerce

CT data analysis

- Industrial CT applications
  - New method: fast 3D objects reconstruction



Texture analysis and segmentation

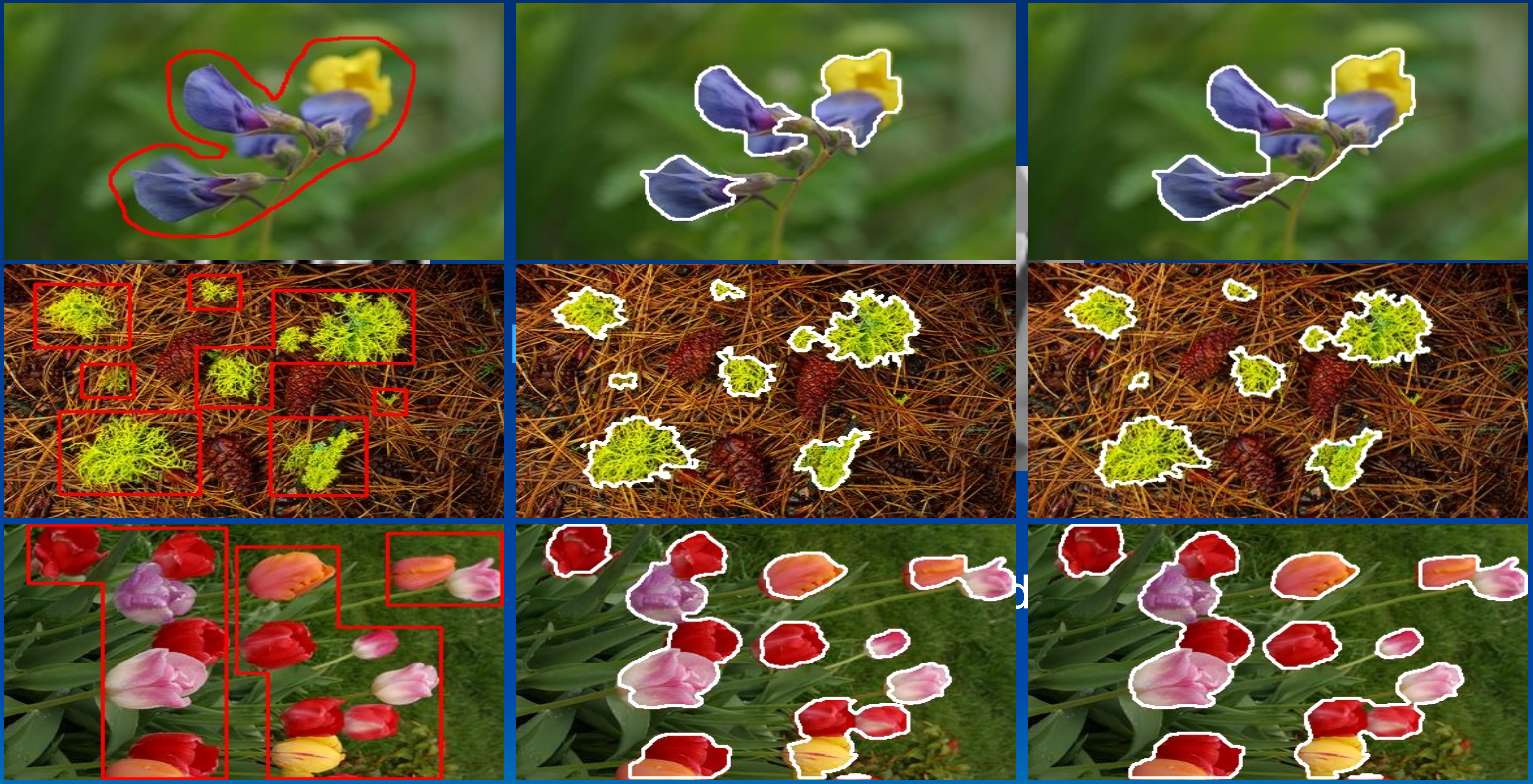


Fig. 9 The comparison between the proposed method based on MPC-GAC by k-means learning and the GrabCut method based on MAP-MRF by GMM estimating. Columns (from left to right): the original image and the initial curve, the results by the proposed method, the result by the GrabCut method.

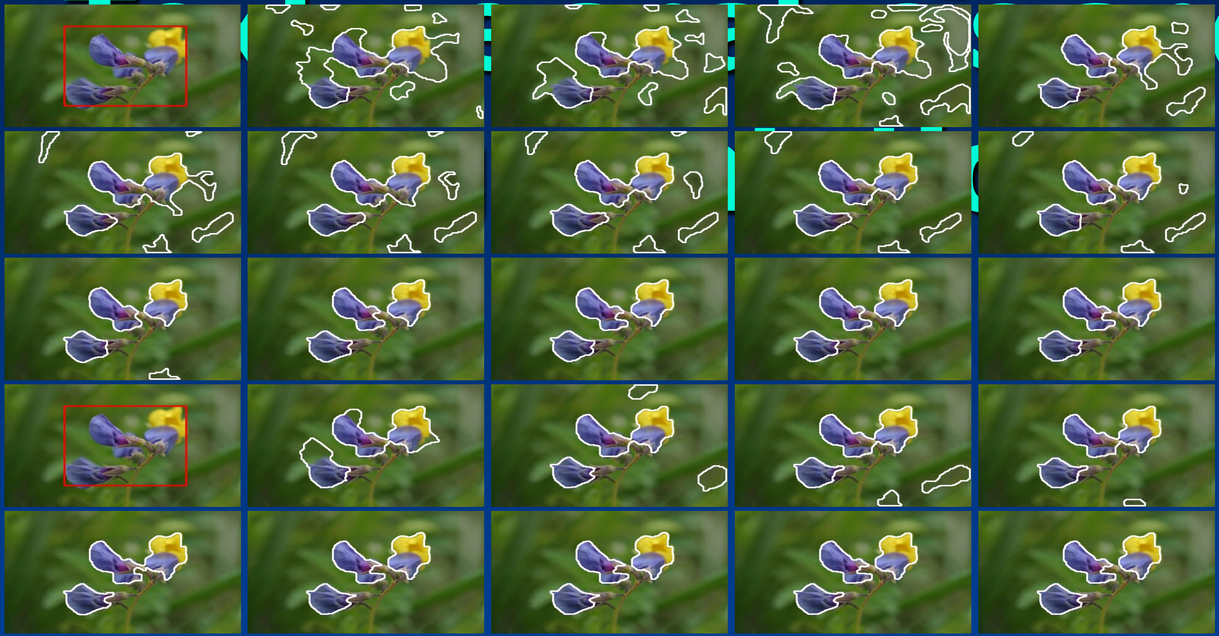


Fig. 11 The iterative process of the image with the blue and yellow flowers. Row 1 to 3 is the iterative process by MPC model and row 5 and 5 is the iterative process by MPC-GAC model.

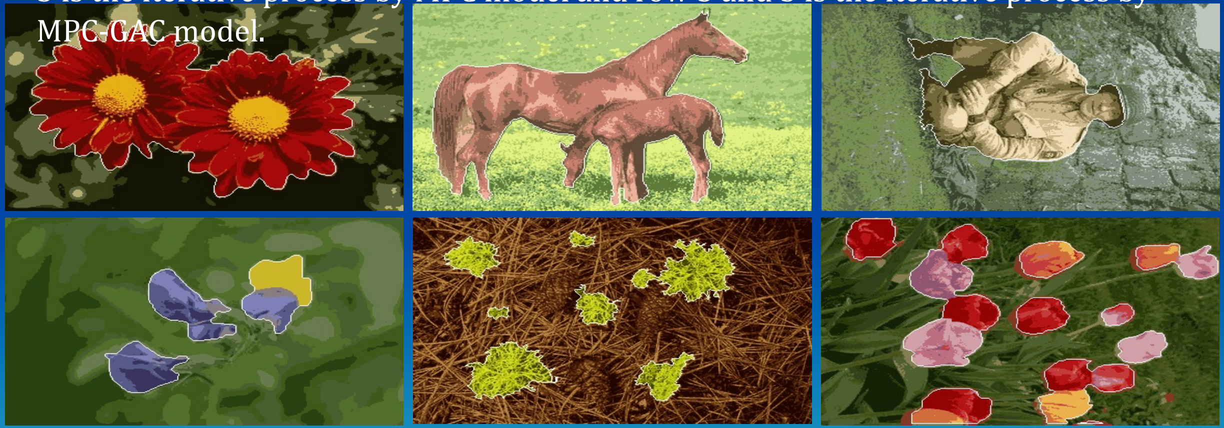
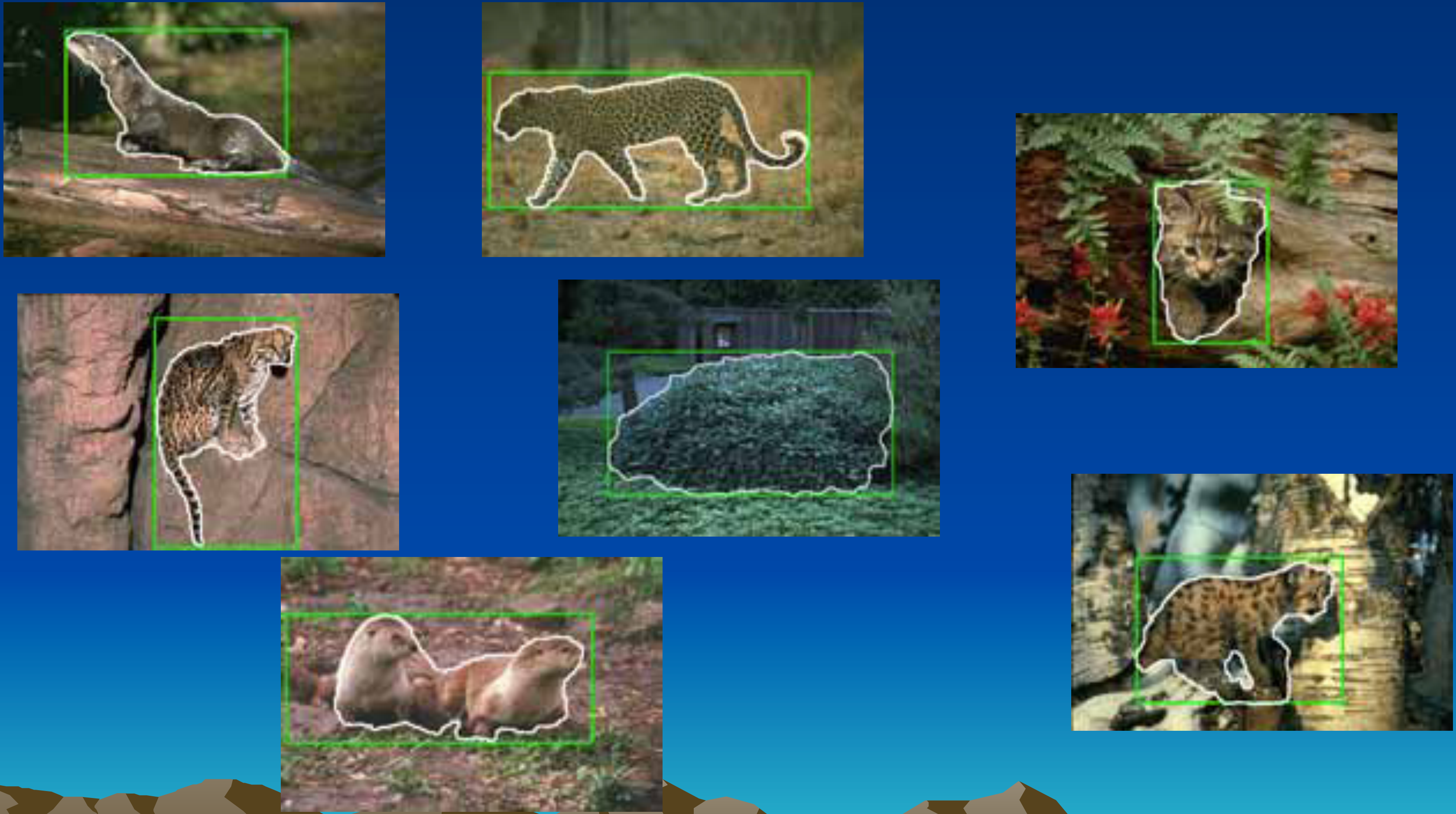
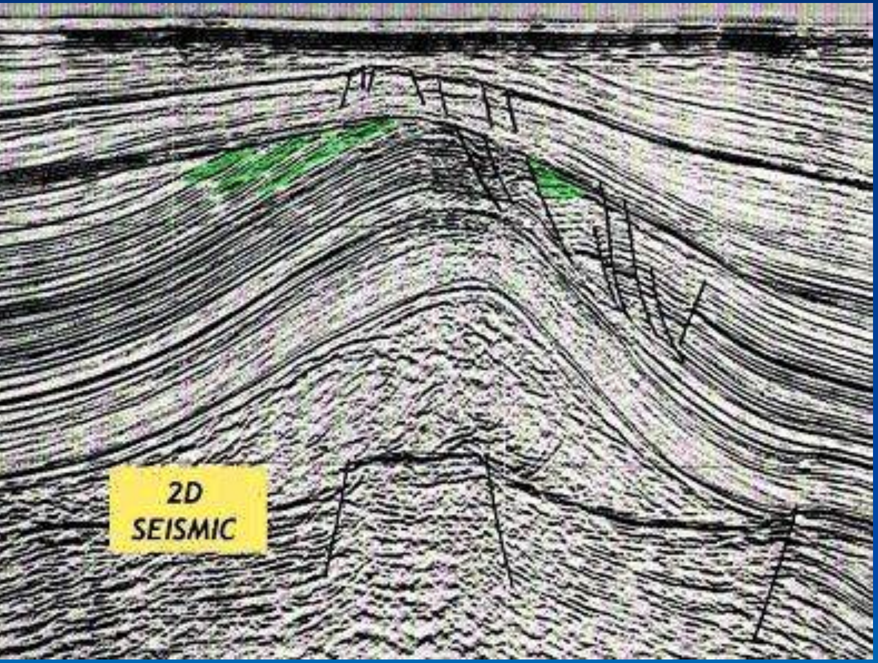


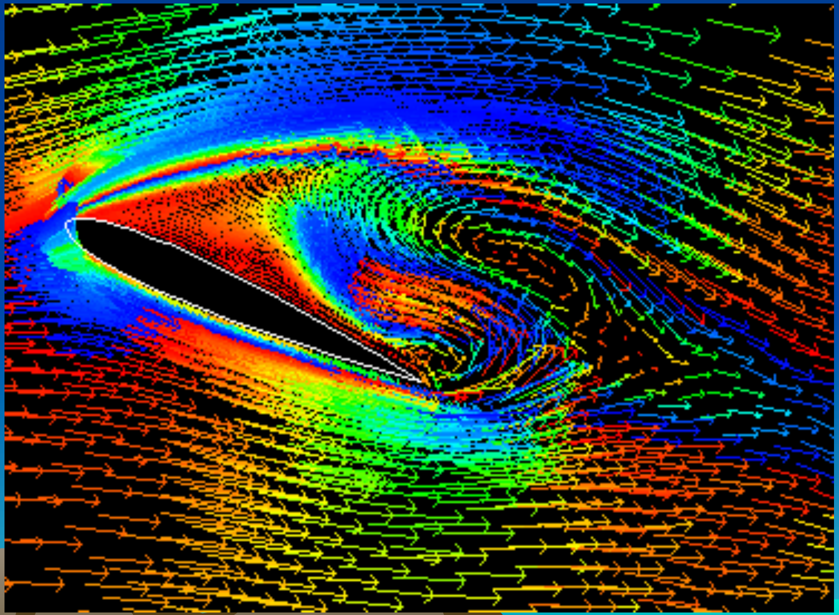
Fig. 12 The multiple piecewise constant approximation of six images after convergence by formula (34), n_1=n_2=6.

Texture extraction: real cases



Otherwise Applications: seismic,vector (multi-chanel) images





Questions? Comments?

- Contact:
  - Tai Xue-Cheng (Tai@math.uib.no)
- Homepage: http://www.math.uib.no/~tai
- All the results can be downloaded from my webpage and publications.

Noname manuscript No.
(will be inserted by the editor)

Multi-Class Transductive Learning based on ℓ^1 Relaxations of Cheeger Cut and Mumford-Shah-Potts Model

Xavier Bresson · Xue-Cheng Tai · Tony F. Chan · Arthur Szlam

Received: date / Accepted: date

Abstract Recent advances in ℓ^1 optimization for imaging problems provide promising tools to solve the fundamental high-dimensional data classification in machine learning. In this paper, we extend the main result of [26], which introduced an exact ℓ^1 relaxation of the Cheeger ratio cut problem for unsupervised data classification. The proposed extension deals with the multi-class transductive learning problem, which consists in learning several classes with a set of labels for each class. Learning several classes (i.e. more than two classes) simultaneously is generally a challenging problem, but the proposed method builds on strong results introduced in imaging to overcome the multi-class issue. Besides, the proposed multi-class transductive learning algorithms also benefit from recent fast ℓ^1 solvers, specifically designed for the total variation norm, which plays a central role in our approach. Finally, experiments demonstrate that the proposed ℓ^1 relaxation algorithms are more accurate and robust than standard ℓ^2 relaxation methods s.a. spectral clustering, particularly when considering a very small number of labels for each class to be classified. For instance, the mean error of classification for the benchmark MNIST dataset of 60,000 data in $\mathbb{R}^{784}$ using the proposed ℓ^1 relaxation of the multi-class Cheeger cut is 2.4% when only one label is considered for each class, while the error of classification for the ℓ^2 relaxation method of spectral clustering is 24.7%.

1 Introduction

Partitioning data into sensible groups is a fundamental problem in machine learning and science in general. One of the most popular approaches is to find the best (balanced) cut of a graph representing data, the such as the normalized cut of Shi and Malik [24] or the Cheeger ratio cut [9]. However, solving balanced/ratio cut problems is NP-hard, which has lead people to compute approximate solutions. The most well-known approach to approximate the solution of a ratio cut is the spectral clustering method, which is based on a ℓ^2 relaxation of the original ratio cut. This ℓ^2 relaxation reduces to solving a generalized system of eigenvectors for the graph Laplacian, then selects the 2nd smallest eigenvector and finally partitions into two groups by thresholding (this requires testing multiple thresholds). Different normalizations of the graph Laplacian lead to different spectral clustering methods. These methods often provide good solutions but can fail on somewhat benign problems; for example see the two-moons example in Figure 1. In this case, the relaxation leading to the spectral clustering methods is too weak. A stronger relaxation was introduced by Bühler and Hein in [7]. They described the p-spectral clustering method, which considers the ℓ^p relaxation of the Cheeger ratio cut, instead of the ℓ^2 relaxation. They showed that the relaxed solution

Xavier Bresson
Department of Computer Science
City University of Hong Kong
E-mail: xbresson@cityu.edu.hk

Xue-Cheng Tai
Department of Mathematics
University of Bergen
E-mail: xue-cheng.tai@uib.no

Tony F. Chan
Department of Mathematics and Computer Science
Hong Kong University of Science and Technology
E-mail: tonyfchan@ust.hk

Arthur Szlam
Department of Mathematics
The City College of New York
E-mail: aszlam@courant.nyu.edu

of the p-spectral clustering problem tends asymptotically to the solution of the Cheeger cut problem when $p \to 1$. In [10,26] (also see [25]), it was proved that the relaxation for $p = 1$ is actually exact, i.e. the solution of the ℓ^1 relaxation problem provides an exact solution of the Cheeger cut problem. Unfortunately, there is no algorithm that guarantees to find global minimizers of the ℓ^1 relaxation problem (we recall that the problem is NP-hard). However, the experiments in [7,26] showed that good results can be obtained with these stronger relaxations; the works [15,3,16] have further strengthened the case for ℓ^1 relaxation methods and related ideas, and have charted a new and promising research direction for improving spectral clustering methods.

In this work, we propose to extend [26]. In particular, we are interested in extending to the challenging multi-class ratio cut problem, and adding label information to obtain a transductive problem. Standard approaches for the unsupervised learning problem usually proceed by recursive two-class clustering. In this paper, we will use results recently introduced in imaging science to solve the multi-class learning problem. The papers [28,19,20,8,6,1] have proposed tight approximations of the solution of the multi-phase image segmentation problem based on ℓ^1 relaxation techniques. The main contribution of this paper is to develop efficient multi-class algorithms for the transductive learning problem. We will introduce two multi-class algorithms based on the ℓ^1 relaxation of the Cheeger cut and the piecewise constant Mumford/Shah or Potts models [22,23]. Experiments show that these new multi-class transductive learning algorithms improve the classification results compared to spectral clustering algorithms, particularly in the case of a very few numbers of labels.

2 Unsupervised data classification with ℓ^1 relaxation of the Cheeger cut

2.1 The model

In this section, we recall the main result of [26] and proposed a modified and improved version of the algorithm introduced there. Let $G = (V, E)$ be a graph where V is the set of nodes and E is the set of edges weighted by a function W_{ij}, $\forall (ij) \in E$. A classical method for clustering is to consider the Cheeger minimization problem [9]:

$$\min_{\Omega \subset V} \frac{\text{Cut}(\Omega, \Omega^c)}{\min(|\Omega|, |\Omega^c|)} \tag{1}$$

which partitions the set V of points into two sets Ω and Ω^c (the complementary set of Ω in V). The cut is defined as $\text{Cut}(\Omega, \Omega^c) := \sum_{i \in \Omega, j \in \Omega^c} w_{ij}$ and $|.|$ provides the number of points in a given set. The Cheeger problem is NP-hard. However, it was shown in [10], and by the authors of this paper using a different argument in [26], that there exists an exact continuous relaxation of (1) as follows. Let us consider the minimization problem w.r.t. a function $u : V \to [0, 1]$:

$$\min_{u \in [0,1]} \frac{||Du||_1}{||u - m(u)||_1} \tag{2}$$

where $||Du||_1 := \sum_{ij} W_{ij} |u_i - u_j|$ is the graph-based total variation of the function u, $m(u)$ is the median of u, and $||u - m(u)||_1 = \sum_i |u_i - m(u)|$. If a global minimizer $u^\star$ of (2) can be computed, then it can be shown that this minimizer would be the indicator of a set $\Omega^\star$ (i.e. $u^\star = 1_{\Omega^\star}$) corresponding to a solution of the NP-hard problem (1). But there is no algorithm that guarantees to compute global minimizers of (2) as the problem is non-convex. However, experiments show that the proposed minimization algorithm in [26], which we will review below, produces good approximations of the solution.

Recent advances in ℓ^1 optimization offer powerful tools to design a fast and accurate algorithm to solve the minimization problem (2). First, observe that minimizing (2) is equivalent to:

$$\min_{u \in [0,1]} \frac{||Du||_1}{||u||_1} \quad \text{s.t.} \quad m(u) = 0, \tag{3}$$

Indeed, the energy is not changed if a constant is added to u. So it is possible to restrict the minimization problem to functions u with zero median. Then, the ratio minimization problem (3) can be solved using the method of Dinkelbach [11] (also used in imaging problems s.a. [18,17]) which introduces the minimax problem:

$$\min_{u \in [0,1]} \max_{\lambda \in \mathbb{R}} ||Du||_1 - \lambda ||u||_1 \quad \text{s.t.} \quad m(u) = 0. \tag{4}$$

Then, we consider a standard two-step iterative algorithm:

(i) Fix λ, compute the solution of the constrained minimization problem:

$$u^{n+1} = \underset{u \in [0,1]}{\text{argmin}} \; ||Du||_1 - \lambda^n ||u||_1 \quad \text{s.t.} \quad m(u) = 0 \tag{5}$$

(ii) Fix u, compute the solution of the maximization problem:

$$\lambda^{n+1} = \underset{\lambda \in \mathbb{R}}{\text{argmax}} \; ||Du^{n+1}||_1 - \lambda ||u^{n+1}||_1 \tag{6}$$

For the minimization problem (5), observe that the constraint zero median is not linear, but it can be replaced by the approximate linear constraint $\sum_i u_i < |V|/2$. Indeed, suppose that $u_i \in \{0, 1\}$ then the median of u is zero if $\sum_i u_i < \sum_i (1 - u_i)$ which yields to $\sum_i u_i < |V|/2$. We will consider the notation $\overrightarrow{1}.u := \sum_i u_i$ in the rest of the paper.

In order to deal efficiently with the non-differentiability of the ℓ^1 norm in (6), a splitting approach associated with an augmented Lagrangian method and the Alternating Direction Method of Multipliers [13] can be used along the same lines as [14,4]. Hence, we consider the constrained minimization problem:

$$\min_{u,v\in[0,1],d} \; ||d||_1 - \lambda||v||_1$$

$$\text{s.t.} \;\; d = Du, \;\; v = u, \;\; \overrightarrow{1}.v < |V|/2. \qquad (7)$$

whose linear constraints can be enforced with an augmented Lagrangian method as:

$$\begin{cases} (u^{n+1}, v^{n+1}, d^{n+1}) = \text{argmin}_{u,v\in[0,1],d} \; ||d||_1 - \lambda||v||_1 \\ +\alpha_d.(d - Du) + \frac{r_d}{2}|d - Du|^2 \\ +\alpha_v.(v - u) + \frac{r_v}{2}(v - u)^2 + \alpha_m.(\overrightarrow{1}.v - |V|/2) \\ \alpha_d^{n+1} = \alpha_d^n + r_d.(d^{n+1} - Du^{n+1}) \\ \alpha_v^{n+1} = \alpha_v^n + r_v.(v^{n+1} - u^{n+1}) \\ \alpha_m^{n+1} = \max(0, \alpha_m^n + r_m.(\overrightarrow{1}.v^{n+1} - |V|/2)) \end{cases} \qquad (8)$$

Three sub-minimizations need to be solved. The minimization problem w.r.t. u:

$$\min_u \; \frac{r_d}{2}\Big|Du - (d + \frac{\alpha_d}{r_d})\Big|^2 + \frac{r_v}{2}\Big(u - (v + \frac{\alpha_v}{r_v})\Big)^2$$

whose solution $u^\star$ is given by a Poisson problem:

$$(r_v + r_d D^T D)u = r_d D^T\Big(d + \frac{\alpha_d}{r_d}\Big) + r_v\Big(v + \frac{\alpha_v}{r_v}\Big) \qquad (9)$$

The solution of (9) can be estimated by a few steps of conjugate gradient descent as D is extremely sparse. The minimization problem w.r.t. v:

$$\min_{v\in[0,1]} -\lambda||v||_1 + \frac{r_v}{2}\Big(v - (u - \frac{\alpha_v}{r_v})\Big)^2 + \alpha_m.(\overrightarrow{1}.v - |V|/2)$$

has an analytical solution given by unshrinkage [26] and truncated into $[0,1]$:

$$v^\star = \Pi_{[0,1]}\Big(f_v + \frac{\lambda}{r_v}\frac{f_v}{|f_v|}\Big), \;\; \text{with} \;\; f_v := u - \frac{\alpha_v}{r_v} - \frac{\alpha_m}{r_v} \qquad (10)$$

To avoid the constant trivial solution, we also apply the "renormalization" step: $v^\star \leftarrow \frac{v^\star - \min(v^\star)}{\max(v^\star) - \min(v^\star)}$. The minimization problem w.r.t. d:

$$\min_d \; ||d||_1 + \frac{r_d}{2}\Big|d - (Du - \frac{\alpha_d}{r_d})\Big|^2$$

has also an analytical solution given by shrinkage [12]:

$$d^\star = \max\Big(|f_d| - \frac{1}{r_d}, 0\Big)\frac{f_d}{|f_d|}, \;\; \text{with} \;\; f_d := Du - \frac{\alpha_d}{r_d} \qquad (11)$$

For the maximization problem (6), the solution is as follows:

$$\lambda^{n+1} = \frac{||Du^{n+1}||_1}{||u^{n+1}||_1} \qquad (12)$$

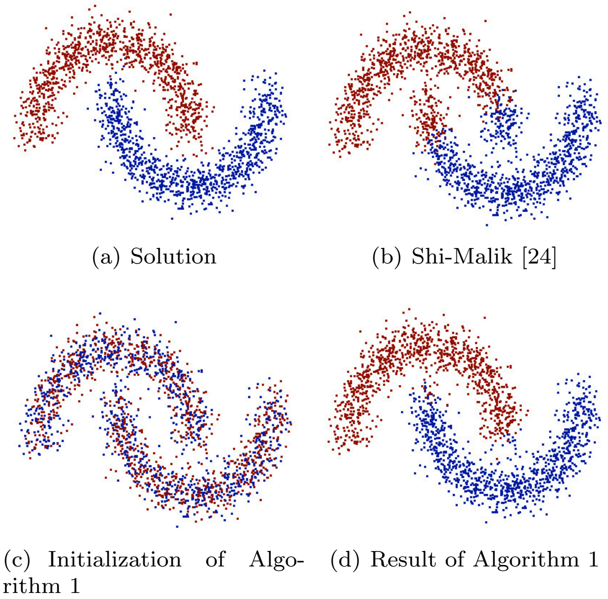
We will consider a steepest gradient descent method instead of (12) to get a smoother evolution of λ^{n+1}:

$$\lambda^{n+1} = \lambda^n - \delta_\lambda.\Big(\lambda^n - \frac{||Du^{n+1}||_1}{||u^{n+1}||_1}\Big). \qquad (13)$$

To summarize the algorithm introduced in this section, we write down the pseudo-code Algorithm 1.

Algorithm 1 Unsupervised learning with ℓ^1 relaxation of the Cheeger cut

```
$u^{n=0}$ given by random initialization
while outer loop not converged do
  $\alpha_d^{q=0}, \alpha_v^{q=0}, \alpha_m^{q=0} \leftarrow 0$
  while inner loop not converged do
    $u^{n+1,q+1}$ given by (9)
    $v^{n+1,q+1}$ given by (10)
    $d^{n+1,q+1}$ given by (11)
    $\alpha_v^{q+1}$ given by (8)
    $\alpha_d^{q+1}$ given by (8)
    $\alpha_m^{q+1}$ given by (8)
  end while
  $\lambda^{n+1}$ given by (13)
end while
```



(a) Solution

(b) Shi-Malik [24]

(c) Initialization of Algorithm 1

(d) Result of Algorithm 1

Fig. 1 Unsupervised classification of the two-moon dataset. Each moon has $1,000$ data points in $\mathbb{R}^{100}$. Figure (b) is the result given by the spectral clustering method of Shi and Malik [24]. It fails to produce the correct result as the ℓ^2 relaxation is too weak. Figure (d) is the result of the ℓ^1 relaxation algorithm and Figure (c) is the random initialization. The proposed algorithm succeeds to compute the correct result. This also shows that the solution of the ℓ^1 relaxation is tighter than the solution of the ℓ^2 relaxation. (Note: it is a color figure.)

2.2 Experiments

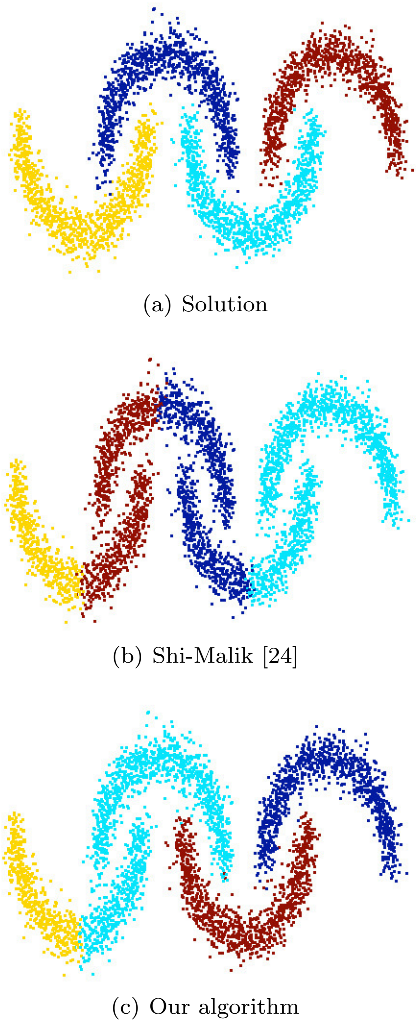
In this section, we demonstrate results using the unsupervised classification algorithm 1. Figure 1 presents the well-known two-moon dataset [7]. Each moon has $1,000$ data points in $\mathbb{R}^{100}$. This example shows that the solution of the ℓ^1 relaxation is tighter than the solution of the ℓ^2 relaxation (see caption for more details). In Table 1, we compare quantitatively our algorithm with the spectral clustering method of Shi and Malik [24] and the related method of Hein and Bühler in [15], which is available at http://www.ml.uni-saarland.de/code/ one SpectralClustering/oneSpectralClustering.html ([16] is not yet available for comparison). Our method and [15] outperform the spectral clustering method, and our method also does slightly better than [15].

	% misclassification
Algorithm 1	**1.53**
Hein and Bühler [15]	1.61
Spectral clustering [24]	1.75

Table 1 Unsupervised learning for the two-moon dataset. We have made 100 experiments and computed the mean percentage of misclassification. Note that for each experiment, the initialization were chosen randomly and the same random initialization was used for Algorithm 1 and [15].

In Figure 2, we apply the standard recursive two-class partitioning approach to deal with more than two classes. Figure 2(b) shows the result by spectral clustering and Figure 2(c) presents the result with our algorithm (see caption for more details).

On the right hand side of Figure 3, we display a projection of the MNIST benchmark dataset, available at http://yann. lecun.com/exdb/mnist/, to 3 dimensions via PCA. This data set consists of 70,000 28 × 28 images of handwritten digits, 0 through 9, usually broken into a 60000 point training set and a 10000 point test set; thus the data is presented as 70000 points in $\mathbb{R}^{784}$). The data was preprocessed by projecting onto 50 principal components. Table 2 compares quantitatively our algorithm with the spectral clustering method of Shi and Malik [24] and the related method of Hein and Bühler in [15]. Our method and [15] outperform the spectral clustering method, and our method also does slightly better than [15].



(a) Solution

(b) Shi-Malik [24]

(c) Our algorithm

Fig. 2 Unsupervised classification for the four-moon dataset. The standard recursive two-class partitioning approach is applied. Figure (b) shows the result by spectral clustering [24] and Figure (c) presents the result with Algorithm 1. Although our algorithm produces a better result than spectral clustering, it still fails to compute the solution. When more than two classes are considered then the quality of the results given by the recursive algorithm actually strongly depends on the choice of the initialization. In fact, for most initializations, the standard recursive two-class partitioning approach will not be able to give the solution. (Note: it is a color figure.)

3 Transductive data classification with ℓ^1 relaxation of the multi-class Cheeger cut

In this section, we extend the unsupervised two-phase Cheeger learning algorithm of Section 2 to a transductive multi-class Cheeger learning algorithm. The most natural extension of (1) to K classes is as follows:

$$\min_{\Omega_1,\dots,\Omega_K} \sum_{k=1}^{K} \frac{\mathrm{Cut}(\Omega_k,\Omega_k^c)}{\min(|\Omega_k|,|\Omega_k^c|)}$$

s.t. $\cup_{k=1}^{K} \Omega_k = V$ and $\Omega_i \cap \Omega_j = \emptyset \ \forall i \neq j$

The previous minimization problem is equivalent to the following problem:

$$\min_{\{u_k\}_{k=1}^K \in \{0,1\}} \sum_{k=1}^{K} \frac{||Du_k||_1}{||u_k - m(u_k)||_1}$$
$$\text{s.t.} \sum_{k=1}^{K} u_k(i) = 1 \ \forall i \in V. \tag{14}$$

The set of minimization used in the above minimization problem is not convex because binary functions do not make a convex set. Thus we consider the following

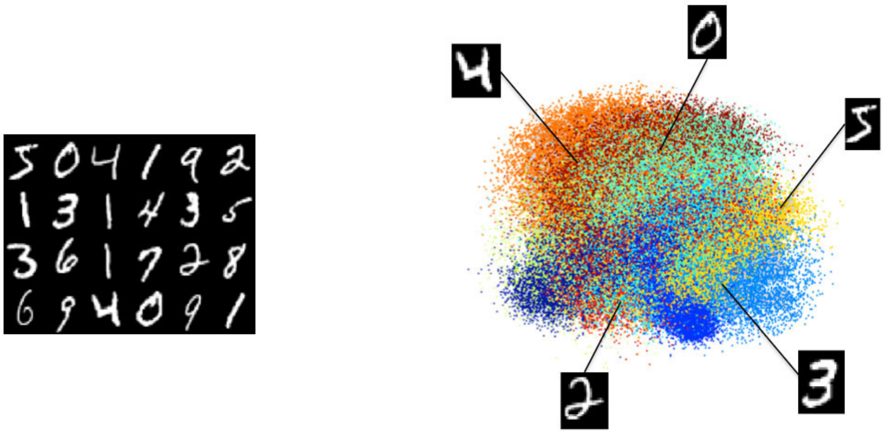


Fig. 3 Projection into a 3D space (via PCA) of the MNIST benchmark dataset. This data set consists of 60,000 28 × 28 images and 10,000 training images (each image is a data point in $\mathbb{R}^{784}$) of handwritten digits, 0 through 9. (Note: it is a color figure.)

	% misclassification
Algorithm 1	**11.69**
Hein and Bühler [15]	11.70
Spectral clustering [24]	29.88

Table 2 Unsupervised learning for the MNIST dataset. This table compares quantitatively Algorithm 1 with the spectral clustering method of Shi and Malik [24] and the related method of Hein and Bühler in [15].

relaxation:

$$\min_{\{u_k\}_{k=1}^K\in[0,1]} \sum_{k=1}^K \frac{||Du_k||_1}{||u_k - m(u_k)||_1}$$
$$\text{s.t.} \quad \sum_{k=1}^K u_k(i) = 1 \ \forall i \in V. \tag{15}$$

In Section 2, we recall that the continuous ℓ^1 relaxation of the two-phase Cheeger minimization problem is exact, meaning that the (continuous) solution of (2) provides a (discrete) solution of the original Cheeger problem (1). We do not know if the ℓ^1 relaxation is still exact when multiple classes are considered, i.e. if the (continuous) solution of (15) provides a (discrete) solution of the original multi-class Cheeger problem (14). For the multi-class Cheeger-based learning problem considered in this paper, experiments show that the solutions $\{u_k\}_{k=1}^K$ are close to binary functions, but there is no theoretical guarantee of this observation.

As the transductive learning problem is considered here then a (small) set l_k of labels is given for each class Ω_k (i.e. $l_k \subset \Omega_k$) and the following minimization problem is thus considered:

$$\min_{\Omega_1,\dots,\Omega_K} \sum_{k=1}^K \frac{\text{Cut}(\Omega_k,\Omega_k^c)}{\min(|\Omega_k|,|\Omega_k^c|)} \quad \text{s.t.}$$
$$\cup_{k=1}^K \Omega_k = V \text{ and } \Omega_i \cap \Omega_j = \emptyset \ \forall i \neq j \text{ and given } \{l_k\}_{k=1}^K$$

which is equivalent to:

$$\min_{\{u_k\}_{k=1}^K\in\{0,1\}} \sum_{k=1}^K \frac{||Du_k||_1}{||u_k - m(u_k)||_1} \quad \text{s.t.}$$
$$\sum_{k=1}^K u_k(i) = 1 \ \forall i \in V \text{ and } u_k(i) = \begin{cases} 1 & \text{if } i \in l_p \text{ and } k = p \\ 0 & \text{if } i \in l_p \text{ and } k \neq p \end{cases}$$

and which is relaxed to:

$$\min_{\{u_k\}_{k=1}^K\in[0,1]} \sum_{k=1}^K \frac{||Du_k||_1}{||u_k - m(u_k)||_1} \quad \text{s.t.}$$
$$\sum_{k=1}^K u_k(i) = 1 \ \forall i \in V \text{ and } u_k(i) = \begin{cases} 1 & \text{if } i \in l_p \text{ and } k = p \\ 0 & \text{if } i \in l_p \text{ and } k \neq p \end{cases}$$

We now extend the two-phase algorithm designed in Section 2 to the multi-phase case:

$$\min_{\{u_k\}_{k=1}^K\in[0,1]} \ \max_{\{\lambda_k\}_{k=1}^K\in\mathbb{R}} \sum_{k=1}^K ||Du_k||_1 - \lambda_k ||u_k||_1 \quad \text{s.t.}$$
$$m(u_k) = 0, \ \sum_{k=1}^K u_k(i) = 1 \ \forall i \in V,$$
$$\text{and } u_k(i) = \begin{cases} 1 & \text{if } i \in l_p \text{ and } k = p \\ 0 & \text{if } i \in l_p \text{ and } k \neq p \end{cases}$$

The median constraint is relaxed to $\overrightarrow{1}.u_k < |V|/K$. We again consider a standard two-step iterative algorithm: (i) Fix λ_k, compute the solution for the K minimization problems:

$$u_k^{n+1} = \underset{u_k \in [0,1]}{\text{argmin}} \ ||Du_k||_1 - \lambda_k^n ||u_k||_1 \text{ s.t.}$$

$$m(u_k) = 0, \ \sum_{k=1}^{K} u_k(i) = 1 \ \forall i \in V,$$

$$\text{and } u_k(i) = \begin{cases} 1 & \text{if } i \in l_p \text{ and } k = p \\ 0 & \text{if } i \in l_p \text{ and } k \neq p \end{cases}$$

(ii) Fix u_k, compute the solution of the K maximization problems:

$$\lambda_k^{n+1} = \underset{\lambda \in \mathbb{R}}{\text{argmax}} \ ||Du_k^{n+1}||_1 - \lambda ||u_k^{n+1}||_1 \qquad (16)$$

The minimization problems (16) are solved as follows:

$$\begin{cases} (u_k^{n+1}, v_k^{n+1}, d_k^{n+1}) = \text{argmin}_{u_k, v_k \in [0,1], d_k} \ ||d_k||_1 \\ -\lambda ||v_k||_1 + \alpha_{dk}.(d_k - Du_k) + \frac{r_d}{2}|d_k - Du_k|^2 \\ +\alpha_{vk}.(v_k - u_k) + \frac{r_v}{2}(v_k - u_k)^2 \\ +\alpha_{mk}.(\overrightarrow{1}.v_k - |V|/K) \\ \ \text{s.t. } \sum_{k=1}^{K} v_k = 1 \ \text{ and} \\ v_k(i) = \begin{cases} 1 & \text{if } i \in l_p \text{ and } k = p \\ 0 & \text{if } i \in l_p \text{ and } k \neq p \end{cases} \\ {\alpha_d}_k^{n+1} = \alpha_d^n + r_d.(d_k^{n+1} - Du_k^{n+1}) \\ {\alpha_v}_k^{n+1} = \alpha_v^n + r_v.(v_k^{n+1} - u_k^{n+1}) \\ {\alpha_m}_k^{n+1} = \max(0, {\alpha_m}_k^n + r_m.(\overrightarrow{1}.v_k^{n+1} - |V|/K)) \end{cases} \qquad (17)$$

The solution of the minimization problems w.r.t. u_k, v_k, d_k is the same as the solution given in the previous section. Finally, the projection onto the convex simplex set $\sum_{k=1}^{K} v_k = 1$ is given by [21,28]. Observe that the final solution $\{u_k^\star\}_{k=1}^K$ of (16) is not guaranteed to be binary. Hence, a conversion step is required to make $\{u_k^\star\}_{k=1}^K$ binary. The most natural conversion is as follows:

$$\hat{u}_k^\star(i) = \begin{cases} 1 & \text{if } k = \arg\max_{p \in \{1,\dots,K\}} u_p^\star(i) \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in V \ (18)$$

where $\{\hat{u}_k^\star\}_{k=1}^K$ are binary functions satisfying $\sum_{k=1}^K \hat{u}_k^\star = 1$.

To summarize the algorithm introduced in this section, we write down the pseudo-code Algorithm 2.

Algorithm 2 Transductive learning with ℓ^1 relaxation of the multi-class Cheeger cut

```
$u_k^{n=0}$ given by a few steps of heat diffusion of the indicator functions of labels
while outer loop not converged do
    ${\alpha_d}_k^{q=0}, {\alpha_v}_k^{q=0}, {\alpha_m}_k^{q=0} \leftarrow 0$
    while inner loop not converged do
        $u_k^{n+1,q+1}$ given by (9)
        $v_k^{n+1,q+1}$ given by (10) + simplex projection [21,28] + labels given by (17)
        $d_k^{n+1,q+1}$ given by (11)
        ${\alpha_d}_k^{q+1}$ given by (17)
        ${\alpha_v}_k$ given by (17)
        ${\alpha_m}_k^{q=0}$ given by (17)
    end while
    $\lambda_k^{n+1}$ given by (13)
end while
```

4 Transductive data classification with ℓ^1 relaxation of the multi-class Mumford-Shah-Potts model

In this section, we develop an alternative to the multi-class Cheeger transductive classification algorithm introduced in the previous section. A successful multiphase segmentation algorithm in imaging is the multiphase piecewise constant Mumford-Shah method [22] (continuous setting) or the Potts method [23] (discrete setting). These methods are well suited to solve the image segmentation problem and the idea in this section is to extend them to the transductive learning problem. Note that the piecewise constant Mumford-Shah/Potts models have been first implemented with the level set method [30,27] and the graph cut method [5]. However, these methods are either too slow, not optimal, not accurate enough or the memory allocation can be important. Recent advances in ℓ^1 optimization algorithms provide efficient tool to solve the piecewise constant Mumford-Shah/Potts models [28,19,20, 8,6,1]. These recent improvements will be used to develop an efficient algorithm for the transductive Potts model:

$$\min_{\Omega_1,\dots,\Omega_K} \ \sum_{k=1}^{K} \underbrace{\text{Cut}(\Omega_k, \Omega_k^c)}_{\simeq \text{Per}(\Omega_k)} \ \text{ s.t.}$$

$$\cup_{k=1}^K \Omega_k = V \text{ and } \Omega_i \cap \Omega_j = \emptyset \ \forall i \neq j \text{ and given } \{l_k\}_{k=1}^K,$$

where Per stands for perimeter. The previous minimization problem is equivalent to the following problem:

$$\min_{\{u_k\}_{k=1}^K \in \{0,1\}} \ \sum_{k=1}^{K} ||Du_k||_1$$

$$\text{s.t. } \ \sum_{k=1}^{K} u_k(i) = 1 \ \forall i \in V,$$

$$\text{and } u_k(i) = \begin{cases} 1 & \text{if } i \in l_p \text{ and } k = p \\ 0 & \text{if } i \in l_p \text{ and } k \neq p \end{cases}$$

The set of minimization used in the above minimization problem is not convex because binary functions do not make a convex set. Thus we consider the following relaxation:

$$\min_{\{u_k\}_{k=1}^K \in [0,1]} \ \sum_{k=1}^{K} ||Du_k||_1$$

$$\text{s.t. } \ \sum_{k=1}^{K} u_k(i) = 1 \ \forall i \in V,$$

$$\text{and } u_k(i) = \begin{cases} 1 & \text{if } i \in l_p \text{ and } k = p \\ 0 & \text{if } i \in l_p \text{ and } k \neq p \end{cases}$$

The previous minimization problem is solved as:

$$\begin{cases} (u_k^{n+1}, v_k^{n+1}, d_k^{n+1}) = \text{argmin}_{u_k, v_k \in [0,1], d_k} \ ||d_k||_1 \\ +\alpha_{d k} \cdot (d_k - Du_k) + \frac{r_d}{2}|d_k - Du_k|^2 \\ +\alpha_{v k} \cdot (v_k - u_k) + \frac{r_v}{2}(v_k - u_k)^2 \\ \ \text{s.t.} \ \sum_{k=1}^K v_k = 1 \ \text{ and } v_k(i) = \begin{cases} 1 & \text{if } i \in l_p \text{ and } k = p \\ 0 & \text{if } i \in l_p \text{ and } k \neq p \end{cases} \\ \alpha_{d\,k}^{\,n+1} = \alpha_d^n + r_d.(d_k^{n+1} - Du_k^{n+1}) \\ \alpha_{v\,k}^{\,n+1} = \alpha_v^n + r_v.(v_k^{n+1} - u_k^{n+1}) \end{cases}$$

The solution of the minimization problems w.r.t. u_k, d_k is the same as the solution given in Section 2. The minimization w.r.t. v_k is simply given by:

$$v_k^\star = \Pi_{[0,1]}(f_{v k}) \ \text{ with } \ f_{v k} := u_k - \frac{\alpha_{v k}}{r_v} \tag{19}$$

and project onto the convex simplex set $\sum_{k=1}^K v_k = 1$ using [21,28]. Observe that the final solution $\{u_k^\star\}_{k=1}^K$ of (16) is not guaranteed to be binary. Hence, a conversion step is required to make $\{u_k^\star\}_{k=1}^K$ binary. Like in the previous section, the binary conversion is as follows:

$$\hat{u}_k^\star(i) = \begin{cases} 1 & \text{if } k = \arg\max_{p \in \{1,\ldots,K\}} u_p^\star(i) \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in V \tag{20}$$

where $\{\hat{u}_k^\star\}_{k=1}^K$ satisfy $\sum_{k=1}^K \hat{u}_k^\star = 1$.

To summarize the algorithm introduced in this section, we write down the pseudo-code Algorithm 3.

Algorithm 3 Transductive learning with ℓ^1 relaxation of multi-class Mumford-Shah-Potts model

```
$u_k^{n=0}$ given by a few steps of heat diffusion of the indicator functions of labels
$\alpha_{d\,k}^{\,n=0}, \alpha_{v\,k}^{\,n=0}, \alpha_{m\,k}^{\,n=0} \leftarrow 0$
while outer loop not converged do
    $u_k^{n+1}$ given by (9)
    $v_k^{n+1}$ given by (19) + simplex projection [21,28] + labels given by (17)
    $d_k^{n+1}$ given by (11)
    $\alpha_{d\,k}^{\,n+1}$ given by (17)
    $\alpha_{v\,k}^{\,n+1}$ given by (17)
end while
```

5 Experiments

In this section, we show classification results using the transductive algorithms developed in sections 3 and 4. We will work on the four moons and MNIST datasets described above. For both data sets, we build the weights matrix using the self-tuning construction of [29]. We use ten nearest neighbors, and the tenth neighbor determines the local scale. The universal scaling parameter is set to 1. For Algorithm 2, we set $r_d = 10$, $r_v = 100$, $r_m = 6K/N$, where N is the number of data points and K is the number of classes, and $\delta_\lambda = 0.4$. For Algorithm 3, we set $r_d = 10$ and $r_v = 100$. We choose the labeled points randomly, and fix a number of labeled points to draw from each class; we repeat each experiment 10 times.

In the Tables 3 and 4 below we compare Algorithm 2 and Algorithm 3 with a spectral transductive learning method from [2], which uses linear least squares on the eigenvectors of the normalized Laplacian to estimate the classes. That is, given the weight matrix W as before, we set $\mathcal{L} = I - S^{-1/2}WS^{-1/2}$, where S is the diagonal matrix with the row sums on the diagonal, that is, $S_{ii} = \sum_j W_{ij}$. We compute the $l+1$ lowest eigenvalue eigenvectors $\phi_0, ..., \phi_l$ of $\mathcal{L}$, and form the $N \times l$ matrix $\Phi = [\phi_1....\phi_l]$; note that as usual we have omitted the density vector ϕ_0. Each row of Φ corresponds to a data point. Next we form the matrix Φ_{lab} by extracting the rows of Φ corresponding to the labeled data points. Let L denote the number of classes, and p be the number of labeled data points. Given the $p \times L$ binary label matrix Y , we compute

$$A = \left(\Phi_{\text{lab}}^T \Phi_{\text{lab}}\right)^{-1} \Phi_{\text{lab}}^T Y.$$

To compute the class labels of the unlabeled points, we set $R = \Phi A$, and let

$$y_j = \underset{i}{\text{argmax}}\, R_{ji}.$$

In both experiments, we see that the ℓ^1 relaxations outperform the ℓ^2 relaxation method when there are few labeled examples; and the Cheeger cut outperforms the Potts for very few labeled examples.

6 Conclusion

The paper introduces new ℓ^1 relaxation methods for the multi-class transductive learning problem. These relaxation methods are inspired from recent advances in imaging science which offer fast, accurate and robust ℓ^1 optimization tools which allow to go beyond standard ℓ^2 relaxation methods, i.e. spectral clustering methods. Experiments demonstrate that the ℓ^1 relaxations of the multi-class Cheeger cut and the Mumford-Shah-Potts outperform the spectral clustering method, and even more significantly when a very small number of labels is considered.

Acknowledgment

Xavier Bresson is supported by the Hong Kong RGC under Grant GRF110311.

# labels per class	1	3	6	10	25	50	100	200
Algorithm 2 (Cheeger)	32.38%	**2.08%**	**0.45%**	**0.46%**	**0.43%**	**0.42%**	0.42%	0.36%
Algorithm 3 (Mumford-Shah-Potts)	**21.59%**	9.48%	3.03%	0.5%	0.44%	0.43%	**0.39%**	**0.34%**
Spectral clustering [2]	38.99%	11.35%	1.62%	0.63%	0.45%	0.45%	**0.39%**	0.37%

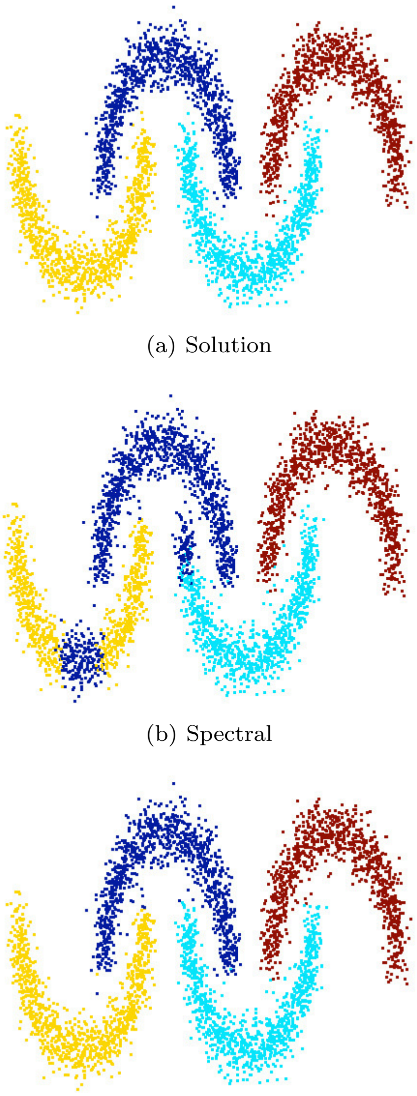
Table 3 Transductive learning for the four-moon dataset. This table compares the proposed ℓ^1 relaxations of the multi-class Cheeger cut (Algorithm 2) and the Mumford-Shah-Potts (Algorithm 3) with the spectral method of [2] (by selecting the number l of eigenvectors which minimizes the error). We have tested different numbers of labels (first row of the table) and for each column we have made 10 experiments and computed the mean percentage of misclassification. For each experiment, the labeled points were chosen randomly and the same labeled points were used for the multi-class Cheeger cut model, the Mumford-Shah-Potts model and the spectral method. The ℓ^1 relaxations of the multi-class Cheeger cut and the Mumford-Shah-Potts outperform the spectral method in all cases, significantly so when a very small number of points are labeled. We also observe that the ℓ^1 relaxation of the Cheeger cut seems to do a better job than the ℓ^1 relaxation of the Mumford-Shah-Potts for a very small number of labels, i.e. 3-50, and inversely when the number of labels is larger than 50.

# labels per class	1	5	10	25	50	100	250	All 10,000 labels
Algorithm 2 (Cheeger)	**2.43%**	**2.45%**	2.45%	2.42%	2.41%	2.38%	2.35%	1.99%
Algorithm 3 (Mumford-Shah-Potts)	14.32%	2.47%	**2.38%**	**2.40%**	**2.33%**	**2.30%**	**2.26%**	**1.74%**
Spectral clustering [2]	24.78%	8.08%	4.48%	3.11%	2.82%	2.47%	2.44%	2.32%

Table 4 Transductive classification for the MNIST dataset. This table compares the proposed ℓ^1 relaxations of the multi-class Cheeger cut (Algorithm 2) and the Mumford-Shah-Potts (Algorithm 3) with the spectral method of [2] (by selecting the number l of eigenvectors which minimizes the error). We have tested different numbers of labels (first row of the table) and for each column we have made 10 experiments and computed the mean percentage of misclassification. For each experiment, the labeled points were chosen randomly and the same labeled points were used for the multi-class Cheeger cut model, the Mumford-Shah-Potts model and the spectral method. The ℓ^1 relaxations of the multi-class Cheeger cut and the Mumford-Shah-Potts outperform the spectral clustering method in all cases and significantly so when a very small number of points are labeled. We also observe that the ℓ^1 relaxation of the Cheeger cut seems to do a better job than the ℓ^1 relaxation of the Mumford-Shah-Potts for a very small number of labels, i.e. 1-5, and inversely when the number of labels is larger than 5.

References

1. E. Bae, J. Yuan, and X.-C. Tai. Global Minimization for Continuous Multiphase Partitioning Problems Using a Dual Approach. *International Journal of Computer Vision*, 92(1):112–129, 2009.
2. M. Belkin. *Problems of learning on manifolds*. PhD thesis, University of Chicago, 2003.
3. A. Bertozzi and A. Flenner. Diffuse Interface Models on Graphs for Classification of High Dimensional Data. *UCLA CAM report 11-27*, 2011.
4. J.M. Bioucas-Dias and M.A. Figueiredo. A New TwIST: Two-Step Iterative Shrinkage/Thresholding Algorithms for Image Restoration. *IEEE Transactions on Image Processing*, 16(12):2992–3004, 2007.
5. Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.
6. E.S. Brown, T.F. Chan, and X. Bresson. A Convex Relaxation Method for a Class of Vector-valued Minimization Problems with Applications to Mumford-Shah Segmentation. *UCLA CAM Report 10-43*, 2010.
7. T. Bühler and M. Hein. Spectral Clustering Based on the Graph p-Laplacian. In *International Conference on Machine Learning*, pages 81–88, 2009.
8. A. Chambolle, D. Cremers, and T. Pock. A Convex Approach for Computing Minimal Partitions. *Technical report TR-2008-05, Dept. of Computer Science, University of Bonn, Bonn*, 2008.
9. J. Cheeger. A Lower Bound for the Smallest Eigenvalue of the Laplacian. *Problems in Analysis*, pages 195–199, 1970.
10. F. R. K. Chung. *Spectral graph theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. Published for the Conference Board of the Mathematical Sciences, Washington, DC, 1997.
11. W. Dinkelbach. On Nonlinear Fractional Programming. *Management Science*, 13:492–498, 1967.
12. D. Donoho. De-Noising by Soft-Thresholding. *IEEE Transactions on Information Theory*, 41(33):613–627, 1995.
13. R. Glowinski and P. Le Tallec. *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*. SIAM, 1989.
14. T. Goldstein and S. Osher. The Split Bregman Method for L1-Regularized Problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.
15. M. Hein and T. Bühler. An Inverse Power Method for Nonlinear Eigenproblems with Applications in 1-Spectral Clustering and Sparse PCA. In *In Advances in Neural Information Processing Systems (NIPS)*, pages 847–855, 2010.
16. M. Hein and S. Setzer. Beyond Spectral Clustering - Tight Relaxations of Balanced Graph Cuts. In *In Advances in Neural Information Processing Systems (NIPS)*, 2011.
17. K. Kolev and D. Cremers. Continuous Ratio Optimization via Convex Relaxation with Applications to Multiview 3D Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
18. V. Kolmogorov, Y. Boykov, and C. Rother. Applications of Parametric Maxflow in Computer Vision. In *International Conference on Computer Vision*, pages 1–8, 2007.
19. J. Lellmann, J. Kappes, J. Yuan, F. Becker, and C. Schnörr. Convex Multi-Class Image Labeling by Simplex-Constrained Total Variation. In *International*



(a) Solution

(b) Spectral

(c) Algorithm 2

Fig. 4 Transductive classification of the four-moon dataset. The objective is to classify the four moons using 3 labels for each moon. Figure (b) presents the result with the spectral method (ℓ^2 relaxation) and Figure (c) shows the result with the ℓ^1 relaxation of the multi-class Cheeger cut (Algorithm 2). The ℓ^1 relaxation produces a better classification result than the ℓ^2 relaxation. (Note: it is a color figure.)

Conference on Scale Space and Variational Methods in Computer Vision, pages 150–162, 2009.

20. J. Lellmann and C. Schnörr. Continuous Multiclass Labeling Approaches and Algorithms. *Univ. of Heidelberg, Tech. Rep.*, 2010.
21. C. Michelot. A Finite Algorithm for Finding the Projection of a Point onto the Canonical Simplex of Rn. *Journal of Optimization Theory and Applications*, 50(1):195–200, 1986.
22. D. Mumford and J. Shah. Optimal Approximations of Piecewise Smooth Functions and Associated Variational Problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.
23. R.B. Potts and C. Domb. Some generalized order-disorder transformations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 48:106–109, 1952.
24. J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(8):888–905, 2000.
25. G. Strang. Maximal Flow Through A Domain. *Mathematical Programming*, 26:123–143, 1983.
26. A. Szlam and X. Bresson. Total variation and cheeger cuts. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1039–1046, 2010.
27. L.A. Vese and T.F. Chan. A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model. *International Journal of Computer Vision*, 50(3):271–293, 2002.
28. C. Zach, D. Gallup, J.M. Frahm, and M. Niethammer. Fast Global Labeling for Real-Time Stereo using Multiple Plane Sweeps. In *Vision, Modeling, and Visualization*, pages 243–252, 2008.
29. L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17 (NIPS 2004)*, 2004.
30. H.K. Zhao, T.F. Chan, B. Merriman, and S. Osher. A Variational Level Set Approach to Multiphase Motion. *Journal of Computational Physics*, 127:179–195, 1996.