Some Applications of the Power Watershed Framework to Image Segmentation and Image Filtering

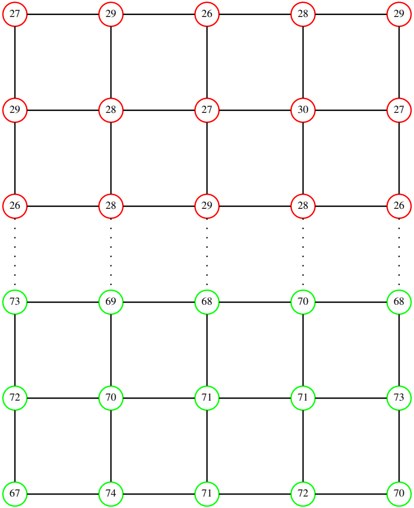
Sravan Danda*

Supervisor: B.S.Daya Sagar*
Joint Supervisor: Laurent Najman†
*SSIU, Indian Statistical Institute, Bangalore Centre
†LIGM, Université Paris Est, ESIEE, France

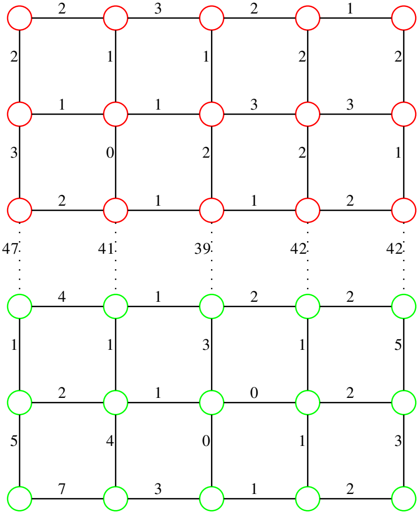
October 11, 2019

Overview

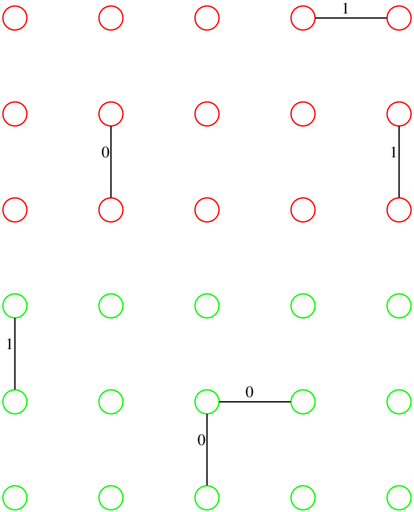
A synthetic gray-scale image



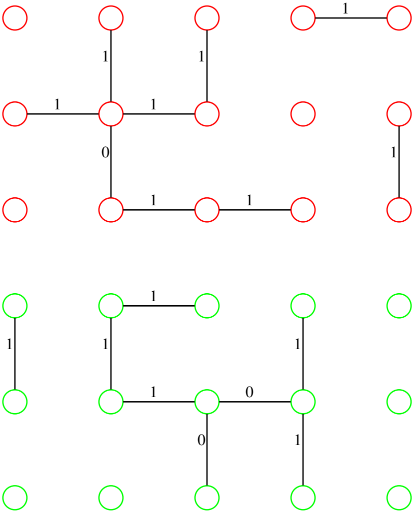
Gradient Image



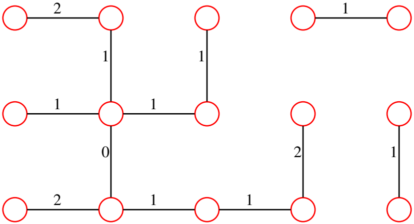
Flooding Simulation

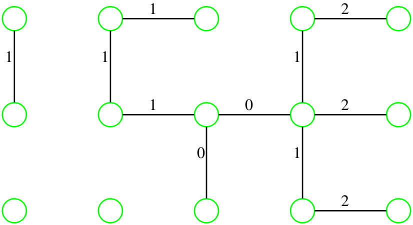


Flooding Simulation

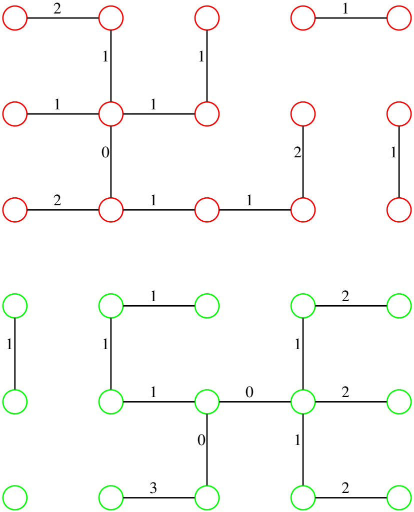


Flooding Simulation

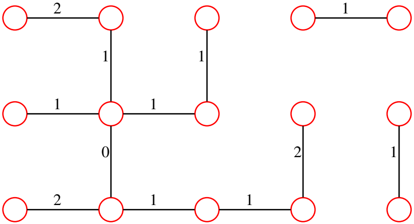


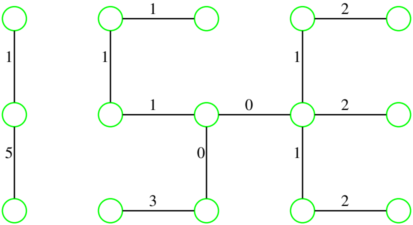


Flooding Simulation



Flooding Simulation





Watershed as a Graph-cut

Watershed Cut:

- Minimum Spanning Forest Cut w.r.t. Minima [1]

[1] Watershed cuts: Minimum spanning forests and the drop of water principle, J Cousty, G Bertrand, L Najman, M Couprie, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (8), 1362-1374

Watershed as a Limit of Total Variation Minimizers

Power Watershed [2]

- $\lim_{p\to\infty} \mathbf{x}^{(p)}$ where

$$\mathbf{x}^{(p)} = \arg\min_{\mathbf{x}}(Q^{(p)}(\mathbf{x}))$$

$$Q^{(p)}(\mathbf{x}) = \sum_{e_{ij}\in E} w_{ij}^{p}|x_i - x_j|^2 + \sum_{i\in \text{Seed}} w_i^{p}|x_i - f_i|^2$$

[2] Power watershed: A unifying graph-based optimization framework, C Couprie, L Grady, L Najman, H Talbot, IEEE transactions on pattern analysis and machine intelligence 33 (7), 1384-1399

Power Watershed: Fast Watershed Cut

Power Watershed: Seeded Image Segmentation [3]

[3] Power watershed: A unifying graph-based optimization framework, C Couprie, L Grady, L Najman, H Talbot, IEEE transactions on pattern analysis and machine intelligence 33 (7), 1384-1399

Power Watershed Framework

Let $0 < \lambda_1 < \lambda_2 < \cdots < \lambda_k$

$$Q(\mathbf{x}) = \sum_i \lambda_i Q_i(\mathbf{x})$$

Power Watershed Framework

Let $0 < \lambda_1 < \lambda_2 < \cdots < \lambda_k$

$$Q^{(p)}(\mathbf{x}) = \sum_i \lambda_i^p Q_i(\mathbf{x})$$

Power Watershed Framework

Let $0 < \lambda_1 < \lambda_2 < \cdots < \lambda_k$

$$Q^{(p)}(\mathbf{x}) = \sum_i \lambda_i^p Q_i(\mathbf{x})$$

$$\mathbf{x}^{(p)} = \arg\min_{\mathbf{x}} Q^{(p)}(\mathbf{x})$$

Power Watershed Framework

Let $0 < \lambda_1 < \lambda_2 < \cdots < \lambda_k$

$$Q^{(p)}(\mathbf{x}) = \sum_i \lambda_i^p Q_i(\mathbf{x})$$

$$\mathbf{x}^{(p)} = \arg\min_{\mathbf{x}} Q^{(p)}(\mathbf{x})$$

$$\mathbf{x}^{(p)} \to \mathbf{x}^* \quad (?)$$

Power Watershed - Generic Algorithm

Algorithm 1 Generic Algorithm to compute limit of minimizers [4]

Input: Function $Q^{(p)}(\mathbf{x}) = \sum_{i=1}^{k} \lambda_i^p Q_i(\mathbf{x})$, where $\lambda_k > \lambda_{k-1} > \cdots > \lambda_1 > 0$.

Output: $\mathbf{x}^*$:

1: $M_k = \arg\min Q_k(\mathbf{x})$ where $\mathbf{x} \in C$
2: **for** i from $k-1$ to 1 **do**
3: Compute $M_i = \arg\min Q_i(\mathbf{x})$ where $\mathbf{x} \in M_{i+1}$
4: **end for**

[4] Extending the Power Watershed Framework Thanks to Γ-Convergence,L Najman, SIAM Journal on Imaging Sciences 10 (4), 2275-2292

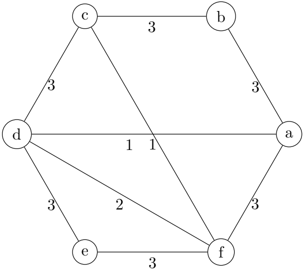
Power Watershed Framework - Why?

- Relates Watershed-Cuts with Random Walker and Shortest Path Segmentation

Power Watershed Framework - Why?

- Relates Watershed-Cuts with Random Walker and Shortest Path Segmentation
- Results in a faster Watershed-Cut algorithm

Laplacian Matrix

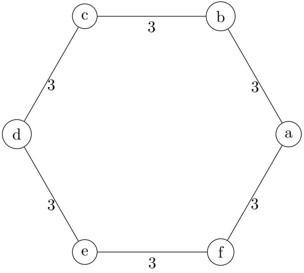


$$
\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array}
\begin{array}{c}
\begin{array}{cccccc} a & b & c & d & e & f \end{array} \\
\begin{pmatrix}
7 & -3 & 0 & -1 & 0 & -3 \\
-3 & 6 & -3 & 0 & 0 & 0 \\
0 & -3 & 7 & -3 & 0 & -1 \\
-1 & 0 & -3 & 9 & -3 & -2 \\
0 & 0 & 0 & -3 & 6 & -3 \\
-3 & 0 & -1 & -2 & -3 & 9
\end{pmatrix}
\end{array}
$$

Fast Spectral Clustering using PW Framework

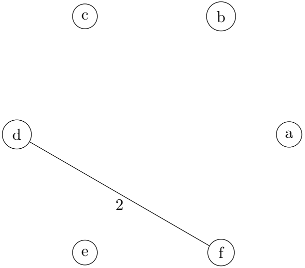
$$\begin{aligned} \underset{H \in \mathbb{R}^{n \times m}}{\text{minimize}} \quad & Tr(H^t L H) \\ \text{subject to} \quad & H^t H = I \end{aligned} \tag{1}$$

Laplacian Matrix: Decomposition



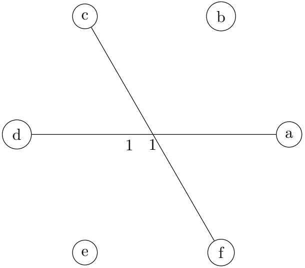
$$
\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array}
\begin{array}{c}
\begin{array}{cccccc} a & b & c & d & e & f \end{array} \\
\begin{pmatrix}
6 & -3 & 0 & 0 & 0 & -3 \\
-3 & 6 & -3 & 0 & 0 & 0 \\
0 & -3 & 6 & -3 & 0 & 0 \\
0 & 0 & -3 & 6 & -3 & 0 \\
0 & 0 & 0 & -3 & 6 & -3 \\
-3 & 0 & 0 & 0 & -3 & 6
\end{pmatrix}
\end{array}
$$

Laplacian Matrix: Decomposition



	a	b	c	d	e	f
a	0	0	0	0	0	0
b	0	0	0	0	0	0
c	0	0	0	0	0	0
d	0	0	0	2	0	−2
e	0	0	0	0	0	0
f	0	0	0	−2	0	2

Laplacian Matrix: Decomposition



$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Fast Spectral Clustering using PW Framework

$$
\begin{aligned}
\underset{H \in \mathbb{R}^{n \times m}}{\text{minimize}} \quad & \sum_i w_i^p Tr(H^t L_i H) \\
\text{subject to} \quad & H^t H = I
\end{aligned}
\tag{2}
$$

Scalability of Spectral Clustering Algorithms

- Traditional Spectral Clustering: $\mathcal{O}(n^{\frac{3}{2}})$

where n are non-zero entries in L

Scalability of Spectral Clustering Algorithms

- Traditional Spectral Clustering: $\mathcal{O}(n^{\frac{3}{2}})$
- Power Spectral Clustering: $\mathcal{O}(nlogn)$

where n are non-zero entries in L

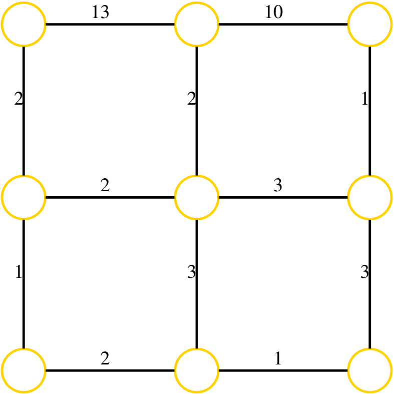
PW Framework for other Image Segmentation Algorithms?

- Can we obtain faster algorithms for other image segmentation methods?

PW Framework for other Image Segmentation Algorithms?

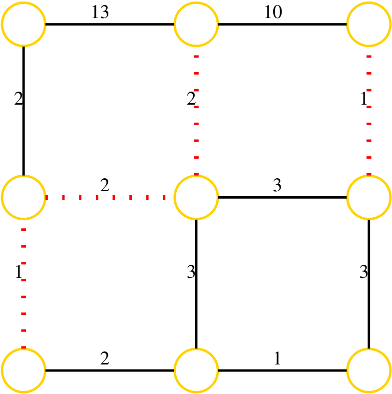
- Can we obtain faster algorithms for other image segmentation methods?
- Yes!

Image: Similarity Graph



$$w_{ij} = 100 \exp(-\frac{||i-j||}{\sigma})$$

Isoperimetric Partitioning



Isoperimetric Partitioning

$$\text{Isoperimetric Cost}(A) = \frac{W(A, \overline{A})}{\min\{|A|, n - |A|\}}$$

Isoperimetric Partitioning

$$x_i = \begin{cases} 0 & \text{if } v_i \in A \\ 1 & \text{if } v_i \in \overline{A} \end{cases}$$

Isoperimetric Partitioning

$$x_i = \begin{cases} 0 & \text{if } v_i \in A \\ 1 & \text{if } v_i \in \overline{A} \end{cases}$$

$$\begin{aligned} \mathbf{x}^t L \mathbf{x} &= \mathbf{x}^t D \mathbf{x} - \mathbf{x}^t W \mathbf{x} \\ &= \sum_{i,j} x_i d_{ij} x_j - \sum_{i,j} x_i w_{ij} x_j \\ &= \sum_{i,j} w_{ij} x_i^2 - \sum_{i,j} x_i w_{ij} x_j \\ &= \sum_{i,j} w_{ij} (x_i^2 - 2x_i x_j + x_j^2) \\ &= \sum_{i,j} w_{ij} (x_i - x_j)^2 \end{aligned}$$

Isoperimetric Partitioning

$$x_i = \begin{cases} 0 & \text{if } v_i \in A \\ 1 & \text{if } v_i \in \overline{A} \end{cases}$$

$$\mathbf{x}^t L \mathbf{x} = W(A, \overline{A})$$

Isoperimetric Partitioning

$$x_i = \begin{cases} 0 & \text{if } v_i \in A \\ 1 & \text{if } v_i \in \overline{A} \end{cases}$$

$$|A| = \mathbf{x}^t \mathbf{1}$$

$$n - |A| = (\mathbf{1} - \mathbf{x})^t \mathbf{1}$$

Isoperimetric Partitioning

Discrete Formulation:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \frac{\mathbf{x}^t L \mathbf{x}}{\min\{\mathbf{x}^t \mathbf{1}, (\mathbf{1}-\mathbf{x})^t \mathbf{1}\}} \\ \text{subject to} \quad & x_i \in \{0,1\} \;\; \forall i \end{aligned} \tag{3}$$

Isoperimetric Partitioning

Continuous Relaxation:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \frac{\mathbf{x}^t L \mathbf{x}}{\min\{\mathbf{x}^t \mathbf{1}, (\mathbf{1}-\mathbf{x})^t \mathbf{1}\}} \\ \text{subject to} \quad & x_i \in [0,1] \; \forall i \end{aligned} \tag{3}$$

Select best threshold!

Isoperimetric Partitioning

Seed Constraint $x_j = 0$

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \frac{\mathbf{x}_{-j}^t L_{-j} \mathbf{x}_{-j}}{\min\{\mathbf{x}_{-j}^t \mathbf{1}, (\mathbf{1} - \mathbf{x}_{-j})^t \mathbf{1}\}} \\ \text{subject to} \quad & x_i \in [0,1] \text{ for } i \neq j \end{aligned} \tag{3}$$

Select best threshold!

Isoperimetric Partitioning

Lagrange Multipliers [5]

$$L_{-j}\mathbf{x}_{-j} = \mathbf{1} \tag{3}$$

[5] Isoperimetric graph partitioning for image segmentation, L Grady, EL Schwartz, IEEE Transactions on Pattern Analysis and Machine Intelligence, 469-475

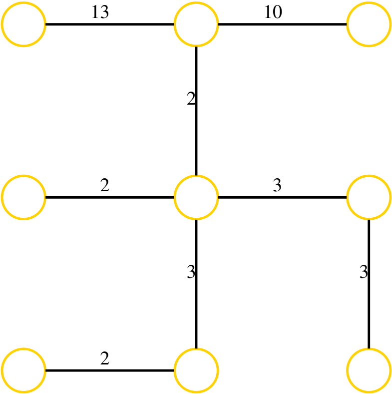
Fast Isoperimetric Partitioning

Solve [6]

$$L_{-j}^{MaxST}\mathbf{x}_{-j} = \mathbf{1} \tag{4}$$

[6] Fast, quality, segmentation of large volumes - isoperimetric distance trees, L Grady, European Conference on Computer Vision, 449-462

Fast Isoperimetric Partitioning



Fast Isoperimetric Partitioning: Computational Cost

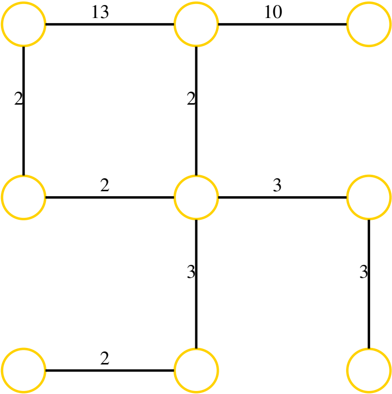
$\mathcal{O}(n)$: where n are non-zero entries in L

Why does the MST heuristic work?

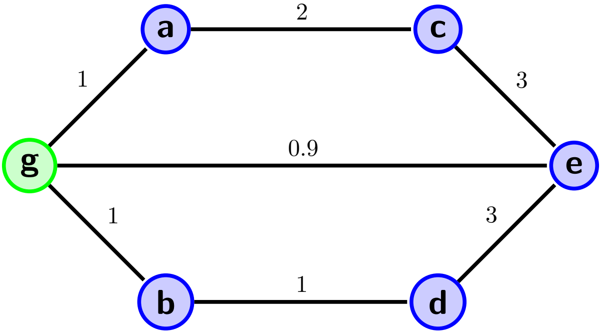
Power Watershed Framework $\Rightarrow$ Enough to solve on UMaxST! [7]

[7] Revisiting the Isoperimetric Graph Partitioning Problem, **S Danda**, A Challa, BD Sagar, L Najman, available at https://hal.archives-ouvertes.fr/hal-01810249

Fast Isoperimetric Partitioning Using PW

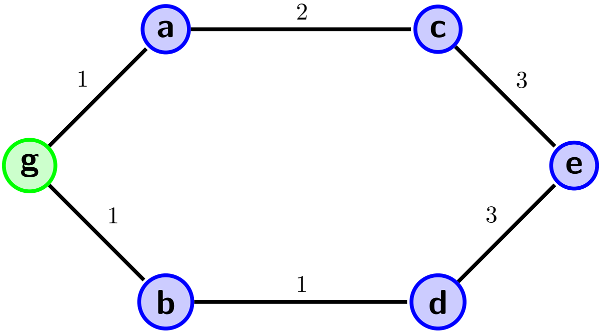


Are Solutions on MST and UMST the same?



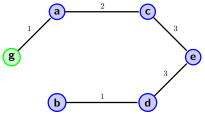
Original Graph

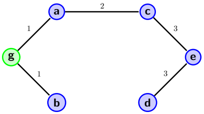
Are Solutions on MST and UMST the same?

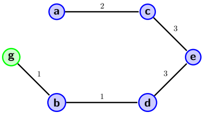


UMST Graph

Are Solutions on MST and UMST the same?







All Possible MSTs

Solving Linear System on Graph, UMST and an arbitrary MST are different!

Node	Original	UMST	MST_1	MST_2	MST_3
g	0.00	0.00	0.00	0.00	0.00
a	1.69	2.68	5.00	4.00	11.16
b	1.54	2.32	9.66	1.00	5.00
c	2.04	3.52	7.00	5.50	10.66
d	2.09	3.64	8.66	6.50	9.00
e	1.94	3.74	8.00	6.16	10.00

How different are Solutions on MST and UMST?

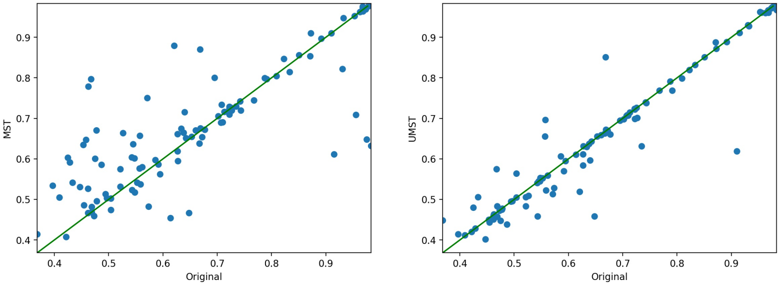
Lemma

Let T_{umst} and T_{mst} denote the operators on UMST and MST respectively, as defined above. Then there exists two positive constants K_1 and K_2 such that

$$K_1 \sum_{i=1}^{k} (u_i - m_i)^2 w_i^2 \leq \| T_{umst} - T_{mst} \| \leq K_2 \sum_{i=1}^{k} (u_i - m_i)^2 w_i^2. \quad (5)$$

Results in Practice

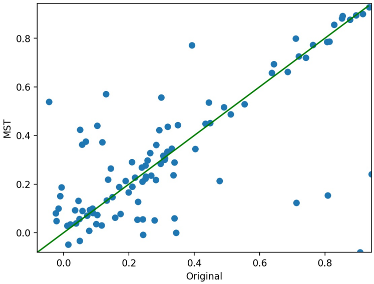
F-Score

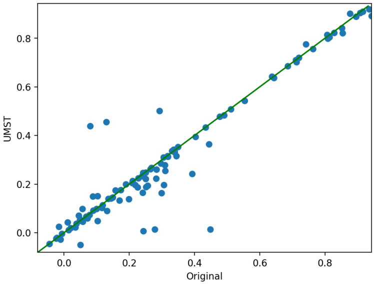


Comparison of MaxST and UMaxST as a sufficient statistic!

Results in Practice

Adjusted Rand Index

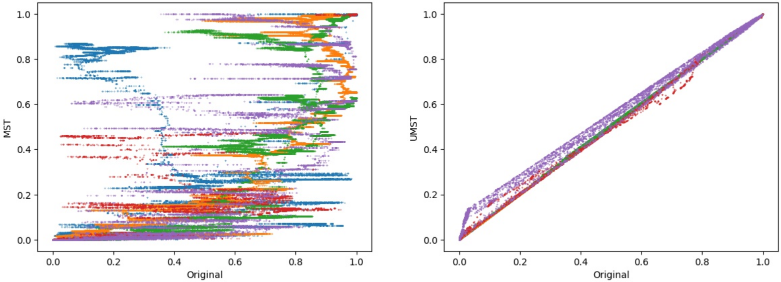




Comparison of MaxST and UMaxST as a sufficient statistic!

Results in Practice

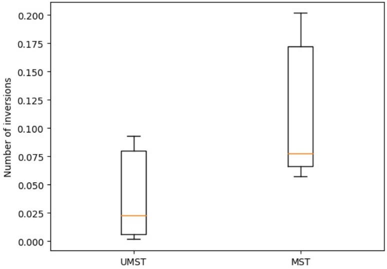
Scatter plot of Normalized values of solutions



Strictly increasing plot implies perfectly consistent solutions!

Results in Practice

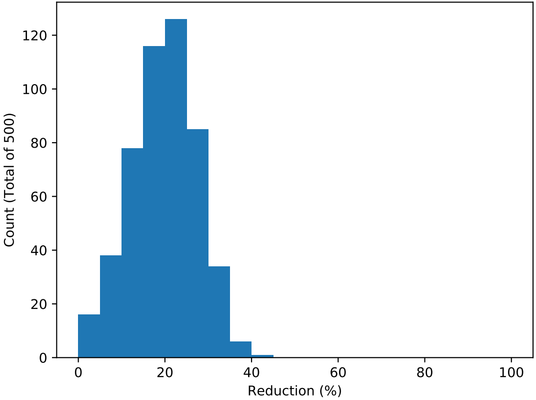
Inversions



Comparison of MaxST and UMaxST as a sufficient statistic!

Results in Practice

Data Reduction



Contributions

1. Detailed Analysis of the relaxed Cheeger Cut problem
2.
3.
4.

Contributions

1. Detailed Analysis of the relaxed Cheeger Cut problem
2. Establish using PW framework that considering UMST graph acts as a sufficient statistic
3.
4.

Contributions

1. Detailed Analysis of the relaxed Cheeger Cut problem
2. Establish using PW framework that considering UMST graph acts as a sufficient statistic
3. Establish bounds between UMST and MST based implementations
4.

Contributions

1. Detailed Analysis of the relaxed Cheeger Cut problem
2. Establish using PW framework that considering UMST graph acts as a sufficient statistic
3. Establish bounds between UMST and MST based implementations
4. Empirically establish that UMST based reduction is robust compared to MST based implementation

Mutex Watershed: [8] The Setup

- $\mathcal{G} = (V, E, W)$
-
-

[8] Steffen Wolf, Constantin Pape, Alberto Bailoni, Nasim Rahaman, Anna Kreshuk, Ullrich Kothe, and Fred A. Hamprecht. The mutex watershed: Efficient, parameter-free image partitioning. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part 4, volume 11208 of Lecture Notes in Computer Science, pages 571–587. Springer, 2018

Mutex Watershed: [8] The Setup

- $\mathcal{G} = (V, E, W)$
- $f : E \to \{-1, +1\}$
-

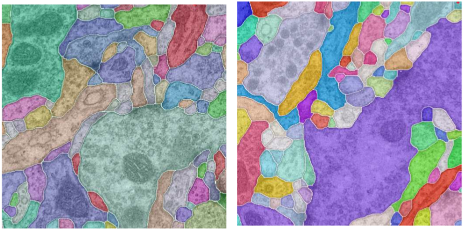
[8] Steffen Wolf, Constantin Pape, Alberto Bailoni, Nasim Rahaman, Anna Kreshuk, Ullrich Kothe, and Fred A. Hamprecht. The mutex watershed: Efficient, parameter-free image partitioning. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part 4, volume 11208 of Lecture Notes in Computer Science, pages 571–587. Springer, 2018

Mutex Watershed: [8] The Setup

- $\mathcal{G} = (V, E, W)$
- $f : E \rightarrow \{-1, +1\}$
- $W : E \rightarrow \mathbb{R}^+$

[8] Steffen Wolf, Constantin Pape, Alberto Bailoni, Nasim Rahaman, Anna Kreshuk, Ullrich Kothe, and Fred A. Hamprecht. The mutex watershed: Efficient, parameter-free image partitioning. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part 4, volume 11208 of Lecture Notes in Computer Science, pages 571–587. Springer, 2018

Mutex Watershed: State-of-the-art on ISBI 2012

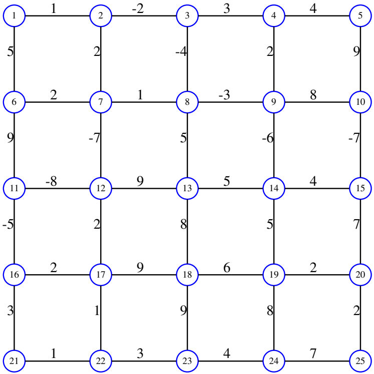


Mutex Watershed: Algorithm

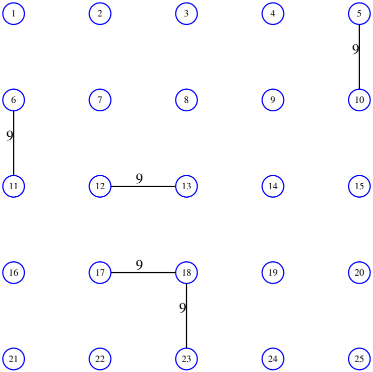
Algorithm 2 Mutex Watershed

Initialize $A = \emptyset$.
for each edge e in descending order of $W(e)$ **do**
  if $A \cup e$ does not violate the mutex condition **then**
    $A \leftarrow A \cup e$
  end if
end for
return Subgraph induced by $\{e \in A | f(e) = +1\}$

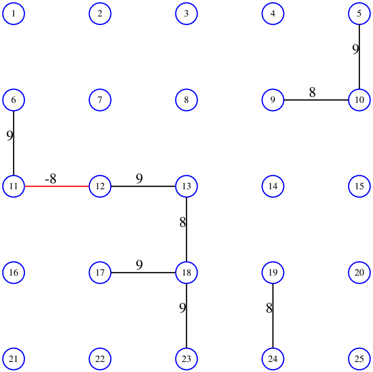
Mutex Watershed: An Example



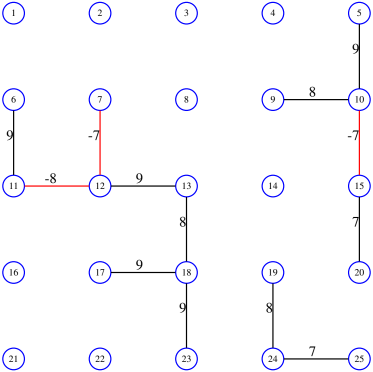
Mutex Watershed: Walk-Through



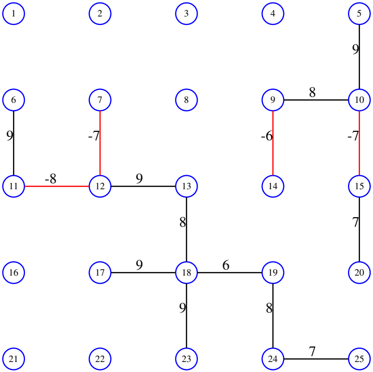
Mutex Watershed: Walk-Through



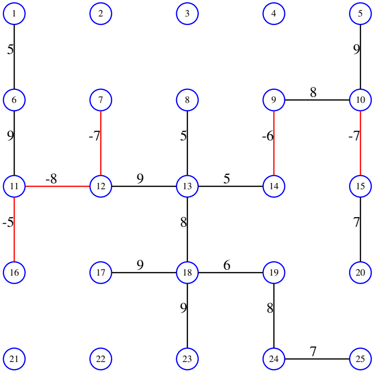
Mutex Watershed: Walk-Through



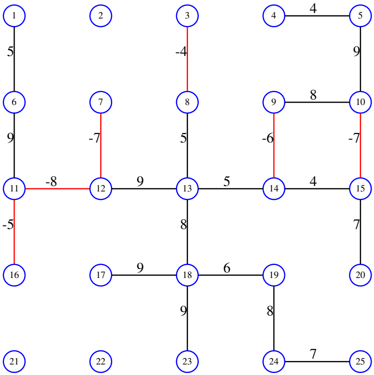
Mutex Watershed: Walk-Through



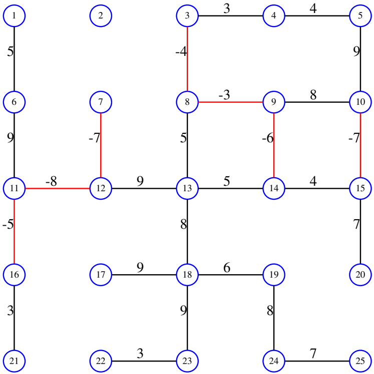
Mutex Watershed: Walk-Through



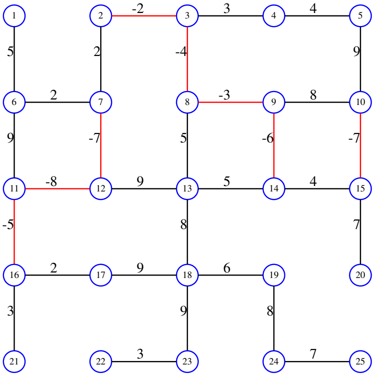
Mutex Watershed: Walk-Through



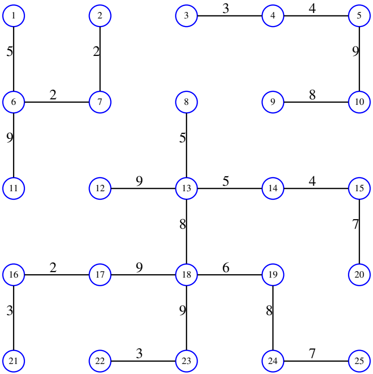
Mutex Watershed: Walk-Through



Mutex Watershed: Walk-Through



Mutex Watershed: Segments



Multi-Cut Graph Partitioning

$$
\begin{aligned}
Q(a) = \min_{a \in \{0,1\}^{|E|}} \quad & -\sum_{e \in E} a_e w_e \\
s.t \quad & \mathcal{C}_1(A) = \emptyset \text{ with } A = \{e \in E | a_e = 1\}
\end{aligned} \tag{6}
$$

NP-Hard!

Mutex Watershed: PW Limit of Multi-Cut

$$
\begin{aligned}
Q^{(p)}(a) = \min_{a \in \{0,1\}^{|E|}} \quad & -\sum_{e \in E} a_e w_e^p \\
s.t \quad & \mathcal{C}_1(A) = \emptyset \text{ with } A = \{e \in E | a_e = 1\}
\end{aligned} \tag{7}
$$

Mutex Watershed: PW Limit of Multi-Cut

$$\mathcal{G}_k = (V, E_k, W|_{E_k})$$

$$\min_{a \in \{0,1\}^{|E_k|}} \quad -\sum_{e \in E_k} a_e$$
$$s.t \quad \mathcal{C}_1(A) = \emptyset \text{ with } A = \{e \in E_k | a_e = 1\} \tag{8}$$

denote the solution space by A_k.

Mutex Watershed: PW Limit of Multi-Cut

$$\mathcal{G}_{k-1} = (V, E_{k-1}, W|_{E_{k-1}})$$

$$\begin{aligned} \min_{a \in \{0,1\}^{|E_{k-1}|}} \quad & -\sum_{e \in E_{k-1}} a_e \\ s.t \quad & \mathcal{C}_1(A) = \emptyset \text{ with } A = A_k \cup \{e \in E_{k-1} | a_e = 1\} \end{aligned} \tag{9}$$

denote the solution space by A_{k-1}.

Mutex Watershed: PW Limit of Multi-Cut

Repeat until all edges are processed.

Mutex Watershed: PW Limit of Multi-Cut

- Sub-problems can be handled with a ‘Union-Find’ data structure.

Mutex Watershed: PW Limit of Multi-Cut

- Sub-problems can be handled with a ‘Union-Find’ data structure.
- Sub-problems are tractable!

Contributions

- Mutex Watershed is PW limit of Multi-cut partitioning

PW Framework for Image Filtering?

- Can we relate image filtering tools?

PW Framework for Image Filtering?

- Can we relate image filtering tools?
- Yes!

Shortest Path Filter

$$SPF_i = \sum_{j \in V} g_i(j) I_j \ ,$$

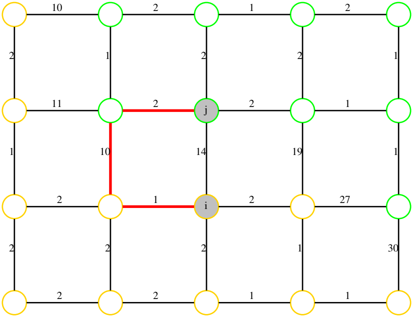
Shortest Path Filter

$$SPF_i = \sum_{j \in V} g_i(j) I_j \ ,$$

where

$$g_i(j) = \frac{exp\left(-\frac{\Theta(i,j)}{\sigma}\right)}{\sum_{k \in V} exp\left(-\frac{\Theta(i,k)}{\sigma}\right)}$$

Shortest Path Filter



$\Theta_i(j) = 3$

Tree Filter [9]

$$\mathrm{TF}_i = \sum_j t_i(j) I_j$$

[9] Linchao Bao, Yibing Song, Qingxiong Yang, Hao Yuan, and Gang Wang. Tree filtering: Efficient structure-preserving smoothing with a minimum spanning tree. IEEE TIP, 23(2): 555-569, 2014.

Tree Filter [9]

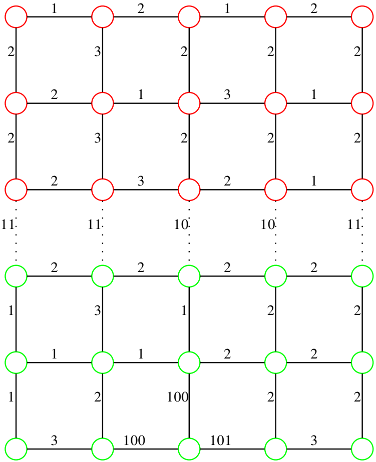
$$\mathrm{TF}_i = \sum_j t_i(j) I_j$$

where

$$t_i(j) = \frac{\exp(-\frac{D(i,j)}{\sigma})}{\sum_q \exp(-\frac{D(i,q)}{\sigma})}$$

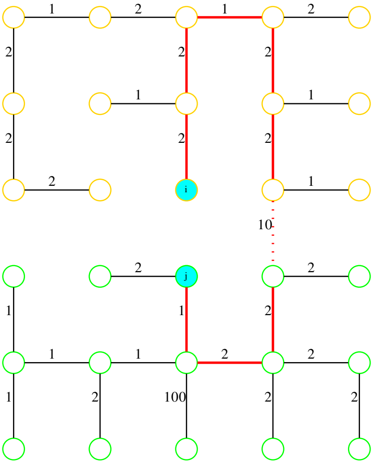
[9] Linchao Bao, Yibing Song, Qingxiong Yang, Hao Yuan, and Gang Wang. Tree filtering: Efficient structure-preserving smoothing with a minimum spanning tree. IEEE TIP, 23(2): 555-569, 2014.

Tree Filter on a Synthetic Graph



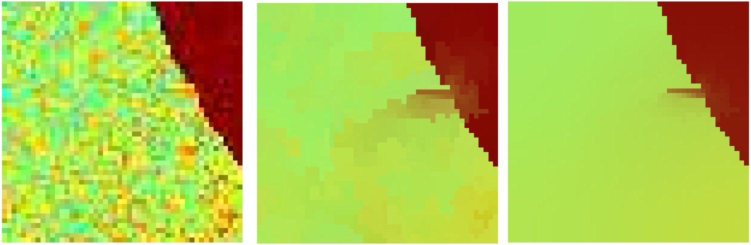
Gradient image

Tree Filter on a Synthetic Graph



$t_i(j) = 9$

Tree Filter on a Synthetic Image



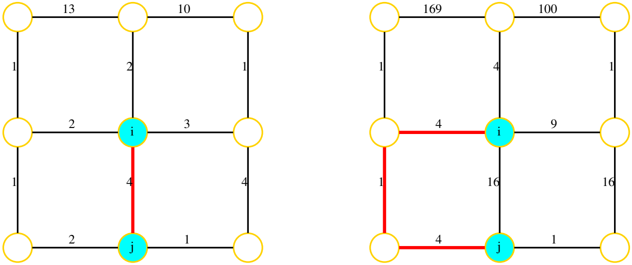
L to R: Noisy Image, TF, TF + BF

Can the Tree Filter be explained?

Power Watershed Framework $\Rightarrow$ Tree Filter is an approximate limit of Shortest Path Filters [10]

[10] *Some Theoretical Links between Shortest Path Filters and Minimum Spanning Tree Filters*, **S Danda**, A Challa, BD Sagar, L Najman, Journal of Mathematical Imaging and Vision, January 2019

Limit of Shortest Path Filters



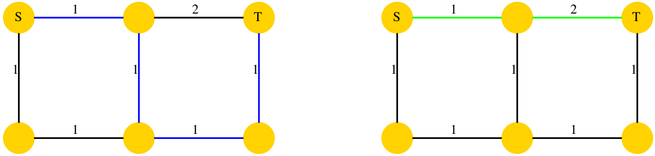
L to R: Image Graph, Image Graph with edge weights raised to power 2

Limit of Shortest Path Filters: Characterization

Lemma

Let $\mathcal{G} = (V, E, W)$. For every pair of pixels i and j in V, there exists $p_0 \geq 1$ such that, a path $P(i,j)$ is a shortest path between i and j in $\mathcal{G}^{(p)}$ for all $p \geq p_0$ if and only if $P(i,j)$ is a smallest path w.r.t. reverse dictionary order between i and j in $\mathcal{G}$. Further, p_0 is independent of i and j.

Reverse Dictionary Order: Illustration

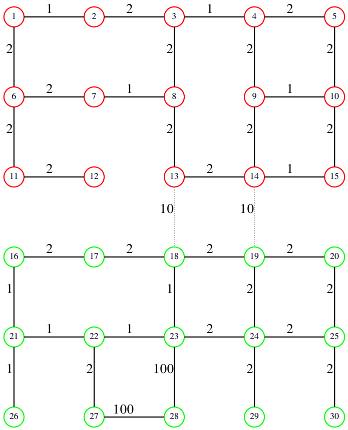


Limit of Shortest Path Filters: Characterization

Lemma

Every smallest path w.r.t. reverse dictionary order between any two arbitrary nodes in $\mathcal{G} = (V, E, W)$ lies on an MST of $\mathcal{G}$ and hence on the UMST of $\mathcal{G}$.

Limit of Shortest Path Filters: UMST Filter

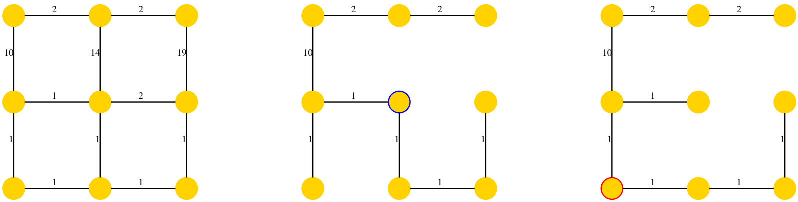


UMST Filter: Characterization

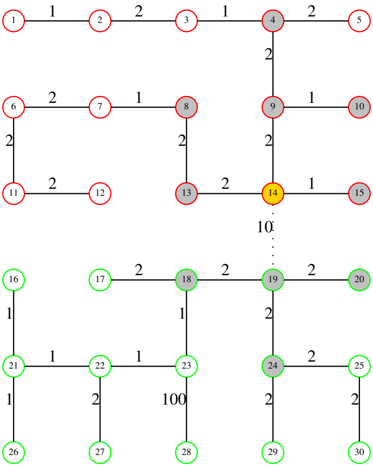
Lemma

For every pixel i in the image I, there exists a spanning tree T_i (termed as adaptive spanning tree), such that T_i contains a smallest path with respect to reverse dictionary ordering between pixels i and any other pixel j in I.

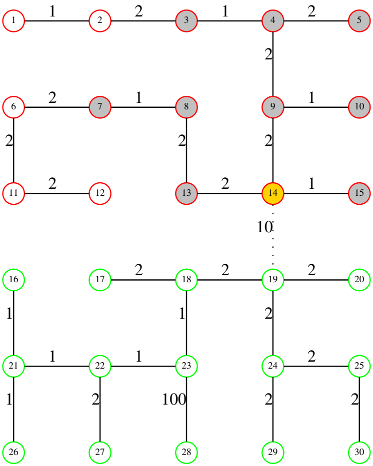
UMST Filter: Adaptive Spanning Trees



UMST Filter: Depth-Based Approximation



UMST Filter: Order-Based Approximation



Results in Practice

Salt and Pepper Noise

L to R: House Image, Bilateral Filter, Tree Filter, Our Approximation to Limit of SPFs

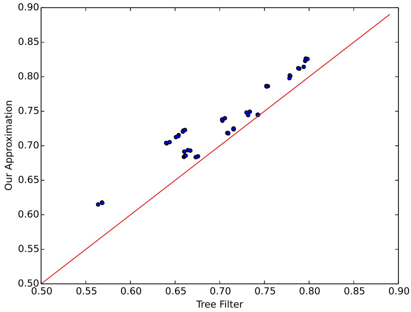
Results in Practice

Structural Similarity Indices

	Mean SSIM on Salt Pepper Noise		
	BF	**TF**	**UMSTF**
House	0.69	0.80	0.83
Barbara	0.72	0.66	0.72
Lena	0.69	0.75	0.79
Pepper	0.62	0.74	0.74
Mean	**0.68**	**0.74**	**0.77**

Results in Practice

SSIM: Tree Filter vs UMST Filter [11]



[11] Some theoretical links between shortest path filters and minimum spanning tree filters, **S Danda**, A Challa, BSD Sagar, L Najman, available at https://hal.archives-ouvertes.fr/hal-01617799v5

Contributions

1. Establish UMST filter as a limit of shortest path filters
2.
3.

Contributions

1. Establish UMST filter as a limit of shortest path filters
2. Tree filter as an approximation to UMST filter
3.

Contributions

1. Establish UMST filter as a limit of shortest path filters
2. Tree filter as an approximation to UMST filter
3. Implement Depth-based and Order-based approximations of UMST filter

Perspectives

1. Adaptive Spanning Trees can be processed in parallel!
2.

Perspectives

1. Adaptive Spanning Trees can be processed in parallel!
2. Can we learn edge-aware features using the adaptive spanning trees?

Perspectives

1. Can we speed-up other tree-based algorithms such as scale-set analysis?
2.
3.
4.

Perspectives

1. Can we speed-up other tree-based algorithms such as scale-set analysis?
2. Total Variation $\leftrightarrow$ Cheeger Cut. Application to TV minimization!
3.
4.

Perspectives

1. Can we speed-up other tree-based algorithms such as scale-set analysis?
2. Total Variation $\leftrightarrow$ Cheeger Cut. Application to TV minimization!
3. Understanding working principle behind PW framework?
4.

Perspectives

1. Can we speed-up other tree-based algorithms such as scale-set analysis?
2. Total Variation $\leftrightarrow$ Cheeger Cut. Application to TV minimization!
3. Understanding working principle behind PW framework?
4. PW implies UMST is a sufficient statistic for image segmentation and filtering. Can we obtain sufficient statistics for graph-modelled data in general?

I would like to thank my advisors, Prof. B S Daya Sagar and Prof. Laurent Najman all their support, anonymous reviewers of my articles and anonymous examiners and CCSD faculty for their suggestions, and Indian Statistical Institute for providing me fellowship to pursue this research.