Power Watersheds and Contrast Invariance¹

Sravan Danda[∗]

[∗]Computer Science and Information Systems, BITS-Pilani, Goa, India

March 31, 2021

 $^{\rm 1}$ Joint work with Aditya Challa (Indian Institute of Science, Bangalore), B S Daya Sagar (Indian Statistical Institute, Bangalore Centre), and Laurent Najman (ESIEE, France)**KORK ERKER ADAM ADA**

Outline

1 [Invariances in Image Data](#page-2-0)

2 [Exploiting Contrast Invariance using Power Watersheds](#page-7-0)

- **3** [Selected Existing Applications](#page-15-0)
	- **[Fast Random Walker](#page-15-0)**
	- **[Fast Isoperimetric Segmentation](#page-21-0)**
	- **[Mutex Watershed: Power Watershed Approximation to](#page-26-0)** [Multi-Cut](#page-26-0)

KORKARYKERKER POLO

4 [Summary](#page-37-0)

Translation and Rotation Invariances

Translation of object does not affect the image class:

Convolutional Neural Nets are designed to utilize this inductive bias with shared parameters!

Image Source ²

² On Translation Invariance in CNNs: Convolutional Layers can Exploit [Abs](#page-1-0)ol[ute](#page-3-0) [S](#page-1-0)<u>pa</u>ti[al](#page-3-0) [L](#page-1-0)[oc](#page-2-0)[at](#page-6-0)[io](#page-7-0)[n,](#page-1-0) [C](#page-2-0)[V](#page-6-0)[PR](#page-7-0) [20](#page-0-0)<u>2</u>0

Contrast Invariance

Increasing/Decreasing contrast should not affect the object boundaries

Image Source ³

3 Alpert et al. Image segmentation by probabilistic bottom-up aggregation a[nd c](#page-4-0)[ue](#page-2-0) [int](#page-3-0)[eg](#page-4-0)[ra](#page-1-0)[ti](#page-2-0)[on](#page-6-0)[.](#page-7-0)[I](#page-7-0)[n](#page-1-0) [P](#page-2-0)[ro](#page-6-0)[ce](#page-7-0)[edin](#page-0-0)[gs](#page-37-0) of the IEEE Conference on Computer Vision and Pattern Recognition, June [200](#page-2-0)7 \rightarrow (\oplus \rightarrow (\oplus \rightarrow (\oplus \rightarrow 2990

Edge-Weighted Graph Models for Images

$$
\mathcal{G} = (V, E, W) : 4
$$
-adjacency edge-weighted graph

- \blacksquare V:- Pixels
- 2 $W : E \to \mathbb{R}$:- Dissimilarity/Similarity between adjacent pixels⁴

K ロ ▶ K 個 ▶ K 할 ▶ K 할 ▶ 이 할 → 9 Q Q →

⁴ depending on application at hand

Contrast Invariance ⁶

$$
\mathcal{G}' = (V, E, W') \leftrightarrow \mathcal{G} = (V, E, W): \text{ Same segmentation results}
$$

where $W' = T \circ W$ with $T' > 0^5$

 \Rightarrow Segmentation should depend on relative ordering of edge weights alone and not the actual weights!

⁵ T ⁰ *>* 1 denotes increase in contrast, 0 *<* T ⁰ *<* 1 denotes decrease in contrast

⁶ this is contrast defined using graph edge-based image gradients and h[ence](#page-4-0)i[s sl](#page-6-0)[ig](#page-4-0)[htly](#page-5-0) [di](#page-6-0)[ffe](#page-1-0)[re](#page-2-0)[n](#page-6-0)[t f](#page-7-0)[ro](#page-1-0)[m](#page-2-0) [c](#page-6-0)[las](#page-7-0)[sic](#page-0-0) notion that uses border pixels as gradients

Contrast Invariance: Illustration

Left: Image with two objects

Middle: Edge-weighted graph with weights denoting dissimilarity

Right: Doubled edge weights \approx increase in contrast

KORK EXTERNE PROVIDE

What is Power Watershed Framework?

Cost minimization problems on finite graphs⁷

Minimize

$$
Q(\mathbf{x}) = \sum_{e_{ij} \in E} f(w_{ij}) Q_{ij}(x_i, x_j)
$$
 (1)

- **x**: target labels of the nodes
- w_{ii} : weight of edge e_{ii}
- Q_{ii} : real-valued smooth function in two variables
- f : increasing function

^{7&}lt;br>⁷ also works for a more general cost form but combination of pairwise c[osts](#page-6-0) [has](#page-8-0) [so](#page-6-0)[me](#page-7-0) [sp](#page-8-0)[ec](#page-6-0)ial [p](#page-14-0)[ro](#page-15-0)[p](#page-6-0)[er](#page-7-0)[tie](#page-14-0)[s](#page-15-0) 2990

Power Watershed Optimization Framework ⁸

Recast an optimization problem into Nested minimization problems

$$
\mathbf{x}^{(\rho)}\rightarrow\mathbf{x}^{*}\;\,\left(?\right)
$$

where

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

and

$$
Q^{(p)}(\mathbf{x}) = \sum_{e_{ij} \in E} f(w_{ij})^p Q_{ij}(x_i, x_j)
$$
 (2)

⁸ Laurent Najman. Extending the Power Watershed framework thankst[o Γ-](#page-7-0)c[onv](#page-9-0)[er](#page-7-0)[genc](#page-8-0)[e.](#page-9-0) [S](#page-6-0)[I](#page-7-0)[AM](#page-14-0) [Jo](#page-6-0)[ur](#page-7-0)[n](#page-14-0)[al](#page-15-0) [on](#page-0-0) Imaging Sciences, 10(4):2275–2292, November 2017.

Why is Power Watershed Framework Useful?

1 **Contrast Invariance**:

lim_{p→∞}x^(*p*) invariant to relative ordering of edge weights.

2 **Empirically Similar Results**:

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

3 **Empirically Faster Computation**:

The answer depends on a substructure of the image graph: Union of Maximum/Minimum Spanning Trees (Similarity/Dissimilarity) of the image graph

What is a UMaxST/UMinST?

Left: Image Similarity Graph

Right: UMaxST: Induced subgraph with edges of all MaxSTs

KORK EXTERNE PROVIDE

Computing the Power Watershed Solution

Rearrange
\n
$$
Q^{(p)}(\mathbf{x}) = \sum_{e_{ij} \in E} f(w_{ij})^p Q_{ij}(x_i, x_j)
$$
\n
$$
(3)
$$

l

$$
Q^{(p)}(\mathbf{x}) = \sum_{i=1}^{I} f(w_i)^p Q_i(\mathbf{x})
$$
\n(4)

KID KØD KED KED E 1990

such that $f(w_i) < f(w_i)$ if $i < j$ and *l*: number of distinct weights

Computing the Power Watershed Solution

Intuition: Higher weights dominate the cost in the limiting case

- **1** Set $i = l$ and M_i is the entire solution space.
- **2** while $i > 1$

■ Compute the set of minimizers M_{i-1} = arg min_{x∈M}. $Q_i(\mathbf{x})$

KORKARYKERKER POLO

3 Return arbitrary $x \in M_1$.

Intuition Behind the Algorithm

$$
Q^{(p)}(x_1, x_2) = w_1^p \left[(x_1 - 1)^2 + x_2^2 \right] + w_2^p \left[(x_1 - x_2)^2 \right] \tag{5}
$$

where $w_1 = w$ and $w_2 = 2w$

Direct computation:

$$
\hat{x}_1^{(\rho)} = \frac{2^{\rho}}{2^{\rho+1}+1}
$$
\n
$$
\hat{x}_2^{(\rho)} = \frac{2^{\rho}+1}{2^{\rho+1}+1}
$$
\n(6)

 $\lim_{p\to\infty} \hat{\mathbf{x}}^p = (\frac{1}{2},\frac{1}{2})$ $\frac{1}{2}$

Intuition Behind the Algorithm

At first pass

$$
\frac{Q^{(p)}(x_1,x_2)}{w_2^p} = \frac{((x_1-1)^2+x_2^2)}{2^p} + (x_1-x_2)^2 \tag{8}
$$

KORKARYKERKER POLO

subspace $M_2 = \{ (x_1, x_2) \in \mathbb{R}^2 \mid x_1 = x_2 \}.$

At second pass:

 $M_1 = (\frac{1}{2}, \frac{1}{2})$ $\frac{1}{2}$ [Power Watersheds and Contrast Invariance](#page-0-0) [Selected Existing Applications](#page-15-0)

[Fast Random Walker](#page-15-0)

Random Walker ⁹

Left: Image with two labels

Middle: Probability that node 1 is labelled blue

Right: Probability that node 1 is labelled red

 Leo Grady. Random walks for image segmentation. IEEE PAMI, 28(1[1\):1](#page-14-0)7[68–](#page-16-0)[17](#page-14-0)[83,](#page-15-0) [20](#page-16-0)[06](#page-14-0) $\leftarrow \equiv$ ÷, $2Q$

[Selected Existing Applications](#page-15-0)

[Fast Random Walker](#page-15-0)

Random Walker as a Cost Minimization

$$
RWCost(\mathbf{x}) = \frac{1}{2} \sum_{e_{ij} \in E} w_{ij} (x_i - x_j)^2,
$$
\n(9)

\nsubject to $\mathbf{x}_{seed} = \mathbf{f}_{seed}$,

\n(10)

K ロ ▶ K @ ▶ K 할 ▶ K 할 ▶ | 할 | © 9 Q @

[Selected Existing Applications](#page-15-0)

[Fast Random Walker](#page-15-0)

Random Walker: Matrix Solver

$$
L = \begin{pmatrix} L_{seed} & B \\ B^T & L_U \end{pmatrix} \tag{11}
$$

$$
RWCost(\mathbf{x}) = \frac{1}{2} (\mathbf{x}_{seed}^T L_{seed} \mathbf{x}_{seed} + 2 \mathbf{x}_U^T B^T \mathbf{x}_{seed} + \mathbf{x}_U^T L_U \mathbf{x}_U), \quad (12)
$$

Solution satisfies:

$$
L_U \mathbf{x}_U = -B^T \mathbf{x}_{seed} \tag{13}
$$

$$
L_U X = -B^T S \tag{14}
$$

K ロ ▶ K @ ▶ K 할 ▶ K 할 ▶ 이 할 → 9 Q @

[Selected Existing Applications](#page-15-0)

[Fast Random Walker](#page-15-0)

Power Watershed-based Random Walker¹⁰

$$
RWCost^{(p)}(\mathbf{x}) = \frac{1}{2} \sum_{e_{ij} \in E} w_{ij}^{p} (x_i - x_j)^2,
$$
\nsubject to $\mathbf{x}_{seed} = \mathbf{f}_{seed}$,

\n(16)

 10 Camille Couprie, Leo Grady, Laurent Najman, and Hugues Talbot. Power watershed: [A u](#page-14-0)[ni](#page-15-0)[fy](#page-20-0)[in](#page-21-0)[g](#page-14-0) [gr](#page-15-0)[ap](#page-36-0)[h-](#page-37-0)[bas](#page-0-0)[ed](#page-37-0) optimization framework. IEEE Trans. Pattern Anal. Mach. Intell., 33(7):13[84–1](#page-17-0)3[99,](#page-19-0) [2](#page-17-0)[011.](#page-18-0) ▶ ﴿ \geq 》 (\geq 》 2990

[Selected Existing Applications](#page-15-0)

L[Fast Random Walker](#page-15-0)

Power Watershed-based Random Walker: A Nested Random Walker

Left: Image with two labels

Right: First Pass of PW

←ロ ▶ ィ伊 ▶ ィミ ▶ ィヨ ▶ B

L[Selected Existing Applications](#page-15-0)

[Fast Random Walker](#page-15-0)

Power Watershed-based Random Walker: A Nested Random Walker

Left: Second Pass of PW

Right: Label of 6 obtained by solving a RW on a small subgraph

[Selected Existing Applications](#page-15-0)

[Fast Isoperimetric Segmentation](#page-21-0)

Isoperimetric Segmentation

$$
IsoCost(A) = \frac{W(A,\bar{A})}{min\{|A|,|\bar{A}|\}},
$$
\n(17)

Avoids small cuts!

K ロ ▶ K 個 ▶ K 할 ▶ K 할 ▶ 이 할 → 9 Q Q →

[Selected Existing Applications](#page-15-0)

[Fast Isoperimetric Segmentation](#page-21-0)

Isoperimetric Partitioning: NP-Hard Problem

$$
IsoCost(A) = \frac{\mathbf{x}^T L \mathbf{x}}{\min\{\mathbf{x}^T \mathbf{1}, (\mathbf{1} - \mathbf{x})^T \mathbf{1}\}},
$$
(18)

where

L: unnormalized graph Laplacian,

$$
x_i = \begin{cases} 1 & \text{if } i \in A \\ 0 & i \in \bar{A} \end{cases}
$$
 (19)

L [Selected Existing Applications](#page-15-0)

[Fast Isoperimetric Segmentation](#page-21-0)

Continuous Relaxation Solution to Isoperimetric Partitioning

Set $x_r = 0$ and

Minimize
$$
\frac{\mathbf{x}_{-r}^T L_{-r} \mathbf{x}_{-r}}{\min\{\mathbf{x}_{-r}^T \mathbf{1}, (\mathbf{1} - \mathbf{x}_{-r})^T \mathbf{1}\}}, \text{subject to each } x_i \in [0, 1]
$$
\n(20)

Lagrange Multipliers \Rightarrow Enough to solve

$$
L_{-r} \mathbf{x}_{-r} = \mathbf{1} \tag{21}
$$

KORKARYKERKER POLO

L[Selected Existing Applications](#page-15-0)

[Fast Isoperimetric Segmentation](#page-21-0)

Fast Isoperimetric Partitioning

Maximum spanning tree solver ¹¹

$$
L_{-r}^{MaxST} \mathbf{x}_{-r} = \mathbf{1}
$$
 (22)

Power watershed solution ¹²

$$
L_{-r}^{UMaxST} \mathbf{x}_{-r} = \mathbf{1}
$$
 (23)

÷.

 Ω

¹¹ Leo Grady. Fast, quality, segmentation of large volumes - isoperimetric distance trees. In Computer Vision -ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part III, pages 449–462, 2006.

¹² Sravan Danda, Aditya Challa, BS Daya Sagar, and Laurent Najman. [Revis](#page-23-0)i[ting](#page-25-0) [t](#page-23-0)[he is](#page-24-0)[o](#page-25-0)[pe](#page-20-0)[ri](#page-21-0)[me](#page-25-0)[tr](#page-26-0)[ic](#page-14-0) [gr](#page-15-0)[a](#page-36-0)[ph](#page-37-0)
titioning problem. IEEE Access. 7:50636–50649. 2019. partitioning problem. IEEE Access, 7:50636–50649, 2019.

L[Selected Existing Applications](#page-15-0)

[Fast Isoperimetric Segmentation](#page-21-0)

Comparison of MaxST and UMaxST Solutions ¹⁴

Left: Matrix Solver on a MaxST vs Original Image graph. Monotonous \Rightarrow consistent solutions 13

Middle: UMaxST vs Original Image Graph

Right: Boxplot of inversions

13Each color is the solution map comparison of an image.

14 Sravan Danda, Aditya Challa, BS Daya Sagar, and Laurent Najman. [Revis](#page-24-0)i[ting](#page-26-0) [t](#page-24-0)[he is](#page-25-0)[o](#page-26-0)[pe](#page-20-0)[ri](#page-21-0)[me](#page-25-0)[tr](#page-26-0)[ic](#page-14-0) [gr](#page-15-0)[a](#page-36-0)[ph](#page-37-0)
titioning problem. IEEE Access, 7:50636-50649, 2019. partitioning problem. IEEE Access, 7:50636–50649, 2019.

B

 QQ

[Selected Existing Applications](#page-15-0)

[Mutex Watershed: Power Watershed Approximation to Multi-Cut](#page-26-0)

Mutex Watershed: Graph-Cut for Simultaneous Similarity and Dissimilarity¹⁵

Left: An image graph capturing similarities and dissimilarities with varied levels of confidence

Right: An ambiguous partitioning

15Steffen Wolf et al. The mutex watershed and its objective: Efficient, [para](#page-25-0)m[ete](#page-27-0)[r-f](#page-25-0)[ree](#page-26-0) [gr](#page-27-0)[ap](#page-25-0)[h](#page-26-0) [pa](#page-36-0)[rt](#page-37-0)[iti](#page-14-0)[on](#page-15-0)[in](#page-36-0)[g.](#page-37-0) [IE](#page-0-0)[EE](#page-37-0) PAMI, 2020.

L [Selected Existing Applications](#page-15-0)

[Mutex Watershed: Power Watershed Approximation to Multi-Cut](#page-26-0)

Mutex Watershed: Greedy Approximation to Multi-Cut ¹⁶

Subject to the consistency constraint, minimizing cut edges is same as minimizing negative sum of leftover edges:

Minimize
$$
Q(\mathbf{a}) = -\sum_{e \in E} a_e w_e
$$

subject to $\mathbf{a} \in \{0, 1\}^{|E|}, C_1(A) = \emptyset$ with $A = \{e \in E | a_e = 1\}$ (24)

NP-hard!

 16 Steffen Wolf et al. The mutex watershed and its objective: Efficient, parameter-free graph partitioning[.](#page-26-0) IEEE PAMI, 2020.

L[Selected Existing Applications](#page-15-0)

L[Mutex Watershed: Power Watershed Approximation to Multi-Cut](#page-26-0)

Mutex Watershed: Greedy Approximation to Multi-Cut¹⁷

Left: Adding edges greedily w.r.t. the confidence subject to cycle constraint

Right: Final partitioning removing the dissimilar edges

¹⁷ Steffen Wolf et al. The mutex watershed and its objective: Efficient, [para](#page-27-0)m[ete](#page-29-0)[r-f](#page-27-0)[ree](#page-28-0) [gr](#page-29-0)[ap](#page-25-0)[h](#page-26-0) [pa](#page-36-0)[rt](#page-37-0)[iti](#page-14-0)[on](#page-15-0)[in](#page-36-0)[g.](#page-37-0) [IE](#page-0-0)[EE](#page-37-0)
MI, 2020. $\iff x \in \mathbb{R}^n$ PAMI, 2020.

[Selected Existing Applications](#page-15-0)

[Mutex Watershed: Power Watershed Approximation to Multi-Cut](#page-26-0)

Mutex Watershed: Power Watershed Approximation to Multi-Cut

Minimize
$$
Q^{(p)}(\mathbf{a}) = -\sum_{e \in E} a_e w_e^p
$$

subject to $\mathbf{a} \in \{0, 1\}^{|E|}, C_1(A) = \emptyset$ with $A = \{e \in E | a_e = 1\}$
(25)

K ロ ▶ K 個 ▶ K 할 ▶ K 할 ▶ 이 할 → 9 Q Q →

[Selected Existing Applications](#page-15-0)

[Mutex Watershed: Power Watershed Approximation to Multi-Cut](#page-26-0)

Mutex Watershed: Power Watershed Approximation to Multi-Cut

K ロ ▶ K 個 ▶ K 할 ▶ K 할 ▶ 이 할 → 9 Q Q →

$$
E_r = \{e \in E | w_e = w_r\} \text{ for each } 1 \leq r \leq l
$$

 $w_1 > \cdots > w_1$ are distinct weights

[Selected Existing Applications](#page-15-0)

[Mutex Watershed: Power Watershed Approximation to Multi-Cut](#page-26-0)

Mutex Watershed: Power Watershed Approximation to Multi-Cut

Level 1: No longer NP-Hard

Minimize
$$
-\sum_{e \in E_l} a_e
$$

subject to $\mathbf{a} \in \{0, 1\}^{|E_l|}$, $C_1(A) = \emptyset$ with $A = \{e \in E_l | a_e = 1\}$ (26)

KORKARYKERKER POLO

 A_l : Solution space

L[Selected Existing Applications](#page-15-0)

[Mutex Watershed: Power Watershed Approximation to Multi-Cut](#page-26-0)

Mutex Watershed: Power Watershed Approximation to Multi-Cut

Level 2:

Minimize
$$
-\sum_{e \in E_{l-1}} a_e
$$

subject to $\mathbf{a} \in \{0, 1\}^{|E_{l-1}|}$, $C_1(A) = \emptyset$ with $A = A_l \cup \{e \in E_{l-1} | a_e = 1\}$ (27)

KORK ERKER ADAM ADA

 A_{l-1} : Solution space

Continue until edges at all *I* levels are exhausted

[Selected Existing Applications](#page-15-0)

[Mutex Watershed: Power Watershed Approximation to Multi-Cut](#page-26-0)

A Glimpse of Other Applications: Fast Ratio Cut

Fast Spectral Clustering¹⁸

¹⁸Aditya Challa, Sravan Danda, B S Daya Sagar, and Laurent Najman. [Pow](#page-32-0)e[r sp](#page-34-0)[ec](#page-32-0)[tral](#page-33-0) [cl](#page-34-0)[us](#page-25-0)[te](#page-26-0)[ri](#page-36-0)[ng](#page-37-0)[.](#page-14-0) [Jo](#page-15-0)[ur](#page-36-0)[na](#page-37-0)[l of](#page-0-0) Mathematical Imaging and Vision, 62(9):1195-1213, 2020. $2Q$

[Selected Existing Applications](#page-15-0)

[Mutex Watershed: Power Watershed Approximation to Multi-Cut](#page-26-0)

A Glimpse of Other Applications: Fast Approximation to Shortest Path Filters

Shortest path pairwise weights

KORKARYKERKER POLO

 $\Theta_i(i) = 3$

[Selected Existing Applications](#page-15-0)

[Mutex Watershed: Power Watershed Approximation to Multi-Cut](#page-26-0)

A Glimpse of Other Applications: Fast Approximation to Shortest Path Filters

Tree Filter ¹⁹ as a PW approximation to Shortest Path Weighted Average Filters ²⁰

Gradient image

 QQ

¹⁹Bao et al. Tree filtering: Efficient structure-preserving smoothing with a minimum spanning tree. IEEE TIP, 23(2):555–569, 2014.

 20 Sravan Danda, Aditva Challa, B S Daya Sagar, and Laurent Najman. Some theoretical links between short[est](#page-37-0) path filters and minimum spanning tree filters. Journal of Mathematical Ima[ging](#page-34-0) [and](#page-36-0) [V](#page-34-0)[isio](#page-35-0)[n,](#page-36-0) [6](#page-25-0)[1\(](#page-26-0)[6\)](#page-36-0)[:7](#page-37-0)[45](#page-14-0)[-7](#page-15-0)[6](#page-36-0)[2,](#page-37-0) [2019](#page-0-0)

L[Selected Existing Applications](#page-15-0)

[Mutex Watershed: Power Watershed Approximation to Multi-Cut](#page-26-0)

A Glimpse of Other Applications: Fast Approximation to Shortest Path Filters

Pairwise weights on UMinST act as PW approximation

KORKARYKERKER POLO

 $t_i(j) = 9$

 L_{Summary} L_{Summary} L_{Summary}

Summary and Perspectives

- **1** Scalability of image segmentation algorithms based on cost optimization on graphs
- 2 Pairwise cost \Rightarrow Enough to work on UMaxST/UMinST
- 3 Can be used at test phase for end-to-end learned counterparts of these algorithms 21

 \equiv Ω

²¹end-to-end random walker for example: Lorenzo Cerrone, Alexander Zeilmann, and Fred A Hamprecht. End-to-end learned random walker for seeded image segmentation. In Proce[edin](#page-36-0)g[s of](#page-37-0) [th](#page-36-0)[e IEE](#page-37-0)[E](#page-36-0) [Confe](#page-37-0)[re](#page-36-0)[nce on](#page-37-0)
Computer Vision and Pattern Recognition, pages 12559–12568, 2019. Computer Vision and Pattern Recognition, pages 12559–12568, 2019.