Combinatorial Optimization in Computer Vision (IN2245)

Frank R. Schmidt
Csaba Domokos

Winter Semester 2015/2016

11. Parametric Maximum Flow 2 / 27

Parametric Energies 3 / 27

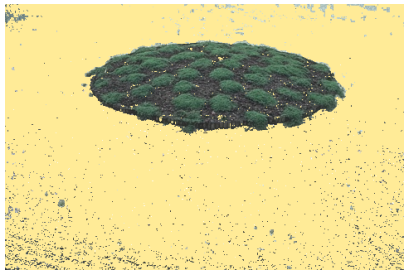
Image Segmentation

Image Segmentation minimizes a combination of data and length terms.

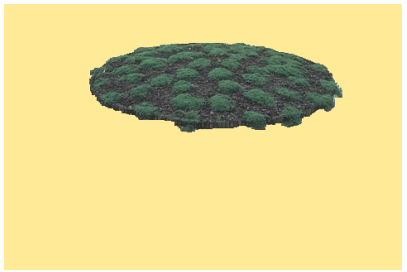
$$E(x;\lambda) = \sum_{i\in\Omega} f_i x_i + \lambda \cdot \sum_{i\in\Omega} \sum_{j\in\mathcal{N}(i)} f_{ij} x_i \bar{x}_j$$

The quality of the segmentation depends on the weight λ.

It is therefore important to choose λ carefully.



$\lambda = 0$



$\lambda = 1$

$\lambda = \infty$

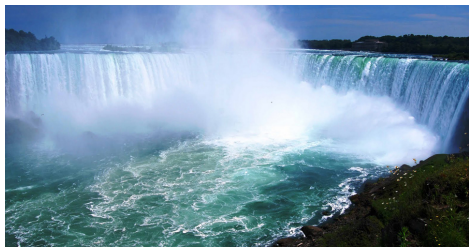
Discretization

For small λ we obtain more details, but also more noise.
For large λ we obtain smoother results, while **over-smoothing** the boundary.

Unfortunately, the choice of λ does not only depend on the observed object, but also on the **image resolution**.

Downsampling an image by a factor of k reduces the **area by a factor of k^2** while reducing the **length by a factor of k**.

Thus, λ has to be reduced by a factor of k to get the same segmentation.



high resolution

low resolution

Dynamic Length Term

We want to compute the best segmentation for every possible length term:

$$\begin{aligned} E_{\text{para}} \colon [0;\infty] &\to \mathbb{R} \\ \lambda &\mapsto \min_{x \in \mathbb{B}^n} E(x;\lambda) \end{aligned}$$

If we can compute this energy efficiently, we can select a $\lambda \geqslant 0$ such that

$$x^*(\lambda) \in \operatorname*{argmin}_{x \in \mathbb{B}^n} E(x;\lambda)$$

has certain properties like **number of connected components**, **minimizing a different energy** E', etc.

E_{para} can help us to transform the discrete domain $\mathbb{B}^n$ of segmentations into the continuous domain $[0;\infty]$. This transformation is not invertible!

Parametric Energies

Given two different pseudo-Boolean energies $E_0, E_1 \colon \mathbb{B}^n \to \mathbb{R}$, we denote by $E(x;\lambda)$ and $E_{\text{para}}(\lambda)$ the energies

$$\begin{aligned} E \colon \mathbb{B}^n \times I &\to \mathbb{R} \\ (x;\lambda) &\mapsto E_0(x) + \lambda \cdot E_1(x) \end{aligned} \qquad \begin{aligned} E_{\text{para}} \colon I &\to \mathbb{R} \\ \lambda &\mapsto \min_{x \in \mathbb{B}^n} E(x;\lambda) \end{aligned}$$

$I \subset \mathbb{R} + \{-\infty, \infty\}$ describes the domain of the parameter λ. In the following, we assume that I is chosen in a way that $E(\cdot, \lambda)$ is submodular for all $\lambda \in I$.

In particular we have for sumodular E_0

- $I \subset \mathbb{R}_0^+ + \{\infty\}$ if E_1 is not supermodular
- $I \subset \mathbb{R}_0^- + \{-\infty\}$ if E_1 is not submodular
- $I \subset \mathbb{R} + \{-\infty, \infty\}$ if E_1 is modular

Concavity

Given a binary segmentation $x \in \mathbb{B}^n$, the energy $E(x; \lambda)$ is linear in λ.

Hence, E_{para} is the minimum of finite linear functions.
In other words $E_{\text{para}}(\lambda)$ is a **piecewise linear, concave function**:

$$E_{\text{para}}(\lambda) = \min_{i=0,\dots,B} E(x_i; \lambda)$$

We assume that the x_i are ordered in a way that for each $i < B$, there is an λ_i such that

$$E(x_i; \lambda_i) = E(x_{i+1}; \lambda_i).$$

We call these λ_i the **breakpoints** of E_{para}.

E_{para} is defined by its B breakpoints $\lambda_0, \dots, \lambda_{B-1}$.

Boundedness of the λ-domain

Since we only have finite breakpoints λ_i, it is enough to study E_{para} on the bounded subdomain

$$I_0 := I \cap [\lambda_0; \lambda_{B-1}]$$

Computing all breakpoints could be done by binary search.

To this end we need to know $x_\infty := x_B \in \operatorname{argmin}_{x \in \mathbb{B}^n} E(x; \lambda)$ for $\lambda > \lambda_{B-1}$.

In other words

$$x_\infty \in \operatorname*{argmin}_{x \in \mathbb{B}^n} E(x; \infty) \subset \operatorname*{argmin}_{x \in \mathbb{B}^n} E_1(x)$$

Finding λ_∞

After determining x_∞, we want to find its breakpoint $\lambda_\infty := \lambda_{B-1} \in I$.
We assume $I = [0; \infty]$.

1. Set $\lambda^* = 0$.
2. Find $x^* \in \operatorname{argmin}_{x \in \mathbb{B}^n} E_0(x) = \operatorname{argmin}_{x \in \mathbb{B}^n} E(x; \lambda^*)$.
3. Find $\hat{\lambda}$ such that $E(x^*; \hat{\lambda}) = E(x_\infty, \hat{\lambda})$, *i.e.*,

$$\hat{\lambda} = \frac{E_0(x_\infty) - E_0(x^*)}{E_1(x^*) - E_1(x_\infty)} \geqslant \lambda^*$$

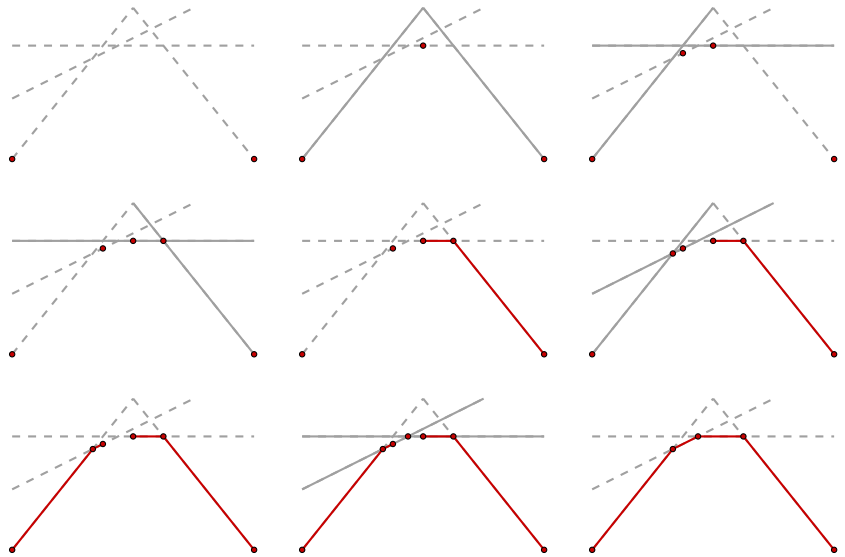
4. Find $\hat{x} \in \operatorname{argmin}_{x \in \mathbb{B}^n} E(x; \hat{\lambda})$.
5. If $E(\hat{x}; \hat{\lambda}) < E(x_\infty; \hat{\lambda})$ set $(x^*, \lambda^*) := (\hat{x}, \hat{\lambda})$ and go to Step 3
6. Return $\lambda_\infty = \hat{\lambda}$.

Parametric Maximum Flow

After finding $(\lambda_0, x_0), (\lambda_\infty, x_\infty)$, we have to find the remaining breakpoints.

1. Set $\mathcal{L} = \{(\{\lambda_0\}, x_0), (\{\lambda_\infty\}, x_\infty)\}$.
2. Let $(I_i, x_i), (I_j, x_j) \in \mathcal{L}$ adjacent with $\lambda_i := \max I_i < \min I_j =: \lambda_j$.
3. Find λ that satisfies $E(x_i; \lambda) = E(x_j, \lambda)$.
4. If $\lambda = \lambda_i$ set $I_j := I_j \cup [\lambda_i; \lambda_j]$. Go to Step 9.
5. If $\lambda = \lambda_j$ set $I_i := I_i \cup [\lambda_i; \lambda_j]$. Go to Step 9.
6. Find $\hat{x} \in \operatorname{argmin}_{x \in \mathbb{B}^n} E(x, \lambda)$.
7. If $\hat{x} \in \{x_i, x_j\}$ set $I_i := I_i \cup [\lambda_i; \lambda]$ and $I_j := I_j \cup [\lambda, \lambda_j]$. Go to Step 9.
8. Set $\mathcal{L} := \mathcal{L} + \{(\{\lambda\}, \hat{x})\}$.
9. If $\bigcup_{I \in \mathcal{L}} I \neq [\lambda_0, \lambda_\infty]$ go to Step 2.
10. Return $\mathcal{L}$.

Example



Example of Ratio Energy

The motivation for the parametric maximum flow was the need to be independent of the image discretization. It helps to compute several solutions, which can then be post-processed.

Nonetheless, this is a rather heuristic approach. It would be better to formulate an energy that itself does not depend on the discretization.

One example is the ratio of **weighted length** and **weighted area**.

$$\begin{aligned} R\colon \mathbb{B}^n &\to \mathbb{R} \\ x &\mapsto \frac{\text{length}(x)}{\text{area}(x)} = \frac{\sum_{i\in\Omega}\sum_{j\in\mathcal{N}(i)} f_{ij} x_i \bar{x}_j}{\sum_{i\in\Omega} f_i x_i} \end{aligned}$$

Discretization Independence

If we downsample an image I by a factor of k, the ratio energy R is also scaled down by a factor of k. Nonetheless, the $\operatorname{argmin} R$ is not changed at all.

In that sense, we have a very different situation than with the parametrized length term. Instead of scaling different parts of the energy in different ways, the whole energy is scaled down.

This makes ratio energies very useful in pratice. Nonetheless, it would be difficult to write this energy in a multilinear way. Potentially, we can end up with exponential many cliques.

Instead we want to find the smallest $r \in \mathbb{R}$ such that $E(\cdot; r)$ possesses a root $x_r \in \mathbb{B}^n$, *i.e.*, $E(x_r; r) = 0$ with

$$\begin{aligned} E\colon \mathbb{B}^n \times \mathbb{R} &\to \mathbb{R} \\ (x; r) &\mapsto \operatorname{length}(x) - r \cdot \operatorname{area}(x) \end{aligned}$$

General Ratio Energies

Given two pseudo-Boolean energies $P\colon \mathbb{B}^n \to \mathbb{R}$ and $Q\colon \mathbb{B}^n \to \mathbb{R}_0^+$, we define its **ratio energy** $R\colon \mathbb{B}^n \to \mathbb{R} + \{-\infty, \infty\}$ as

$$R(x) = \begin{cases} \frac{P(x)}{Q(x)} & \text{if } Q(x) \neq 0 \\ +\infty & \text{if } Q(x) = 0 \text{ and } P(x) > 0 \\ -\infty & \text{if } Q(x) = 0 \text{ and } P(x) < 0 \end{cases}$$

In order to minimize R, we will use the energy

$$\begin{aligned} E\colon \mathbb{B}^n \times \mathbb{R} &\to \mathbb{R} \\ (x; r) &\mapsto P(x) - r \cdot Q(x) \end{aligned}$$

E is submodular if

1. P is submodular, Q is submodular and $r \leqslant 0$ or
2. P is submodular, Q is supermodular and $r \geqslant 0$ or
3. P is submodular, Q is modular

Ratio Optimization

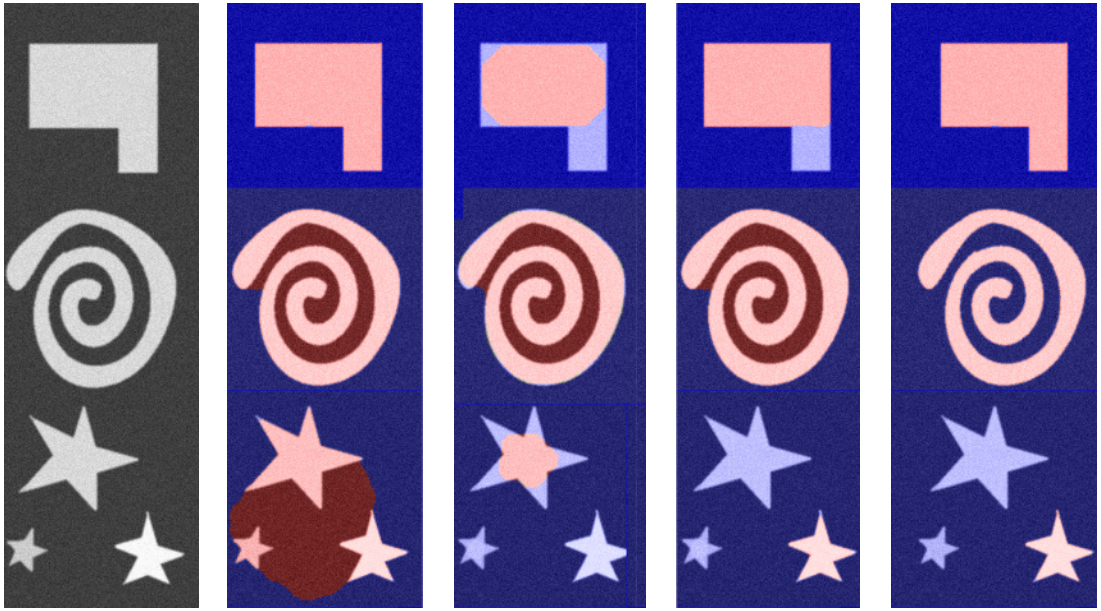
To minimize R, we will minimize $E(\cdot; r)$ for different $r \in \mathbb{R}$ until we are unable to decrease r. The obtained r represents the minimal ratio and $x_r \in \operatorname{argmin} E(\cdot; r)$ its segmentation.

1. Pick $\hat{x} \in \mathbb{B}^n$ with $Q(\hat{x}) > 0$ and $r := R(\hat{x})$.
2. Let $x^* \in \operatorname{argmin}_{x \in \mathbb{B}^n} E(x; r)$.
3. If $Q(x^*) = 0$ and $P(x^*) < 0$ go to Step 6.
4. Set $r^* := R(x^*)$. If $r = r^*$ go to Step 6.
5. Set $(\hat{x}, r) := (x^*, r^*)$. Go to Step 2.
6. Return x^* as segmentation of minimal ratio.

This method solves the ratio problem for the following situations

1. P is submodular, Q is submodular and $P(\hat{x}) < 0$ in Step 1.
2. P is submodular, Q is supermodular and $P(x) \geqslant 0$ for all $x \in \mathbb{B}^n$.
3. P is submodular, Q is modular

Results



Images | $\frac{\text{len}_g(S)}{\text{area}(S)}$ | $\frac{\text{area}_f(S)}{\text{len}(S)}$ | $\frac{\text{area}_f(S)}{\text{len}_g(S)}$ | $\frac{\text{flux}(S)}{\text{len}(S)}$

$$\text{flux}(S) = \int_{\partial S} \langle N(s), \nabla I(s) \rangle \, \text{ds} = \int_S \Delta I(x) \text{dx}$$

Length Ratio

If we want to minimize the ratio

$$R(x) = \frac{\mathrm{len}_1(x)}{\mathrm{len}_2(x)}$$

of two length terms len_1 and len_2, we cannot use the parametric maximum flow framework. This is because $\mathrm{len}_1(x) \geqslant 0$ for all segmentations $x \in \mathbb{B}^n$.

Jermyn and Ishikawa showed that this problem can nonetheless be optimized with a different approach that only considers closed paths $p \in \mathbb{B}^{\mathcal{E}}$ of a digraph $G = (V, \mathcal{E})$ with two length weights $c_1, c_2 \colon \mathcal{E} \to \mathbb{R}^+$.

The method only works for ratios of length terms. We cannot replace len_1 or len_2 by arbitrary quadratic submodular energies.

In particular, we cannot combine data terms with length terms.

Method of Jermyn-Ishikawa

To minimize $R(p) = \frac{\text{len}_1(p)}{\text{len}_2(p)}$, we will not minimize

$$E(p; r) = \text{len}_1(p) - r\,\text{len}_2(p).$$

Instead, we will look for $p \in \mathbb{B}^{\mathcal{E}}$ such that $E(p; r) \leqslant 0$.

1. Pick closed path $p \in \mathbb{B}^{\mathcal{E}}$ and $r := R(p)$.
2. Find closed path $p^* \in \mathbb{B}^{\mathcal{E}}$ such that $E(p^*; r) < 0$.
3. If p^* does not exist, go to Step 5
4. Set $(p, r) := (p^*, R(p^*))$. Go to Step 2.
5. Return p as closed path of minimal length ratio.

This method solves the ratio problem, if there is an efficient way to compute the negative cycle in Step 2.

Reminder: Bellman-Ford Algorithm

Given the current ratio r, we define the following weighting function $c\colon \mathcal{E} \to \mathbb{R}$ on the edges: $c(i,j) = c_1(i,j) - r \cdot c_2(i,j)$.

Assuming that there exists no cycle of negative length, the Bellman-Ford algrithm can compute the shortest path between a source node s and all other nodes $v \in V$ in the graph. The worst case running time is $\mathcal{O}(|V| \cdot |E|)$.

Applying the same method to a graph with a negative cycle p will never terminate. Instead every path tries to pass through p as often as possible in order to reduce the distance even further.

Running Bellman-Ford for $\mathcal{O}(|V| \cdot |E|)$ time either provides us with a shortest path or not. In the first case there is no negative cycle in the graph. In the second case, the negative cycle will appear in the *degenerated Dijkstra tree* that the Bellman-Ford algorithm maintains.

Shape Prior Image Segmentation

Using Bayes' rule, we can write any image segmentation problem as a probability maximization problem

$$P(x|I) = \frac{P(I|x) \cdot P(x)}{P(I)}$$

The *likelihood* $P(I|x)$ tells us how well a segmentation $x \in \mathbb{B}^n$ fits to the observed image I.

The *prior* $P(x)$ tells us how likely a certain segmentation $x \in \mathbb{B}^n$ is. It does not depend on the observation. The MRF model for example favors small contours.

If we expect a certain contour, we can also encode this knowledge into our optimization framework. We call this prior **shape prior** if it is invariant with respect to translation, rotation and scaling.

Modelling Shape Prior

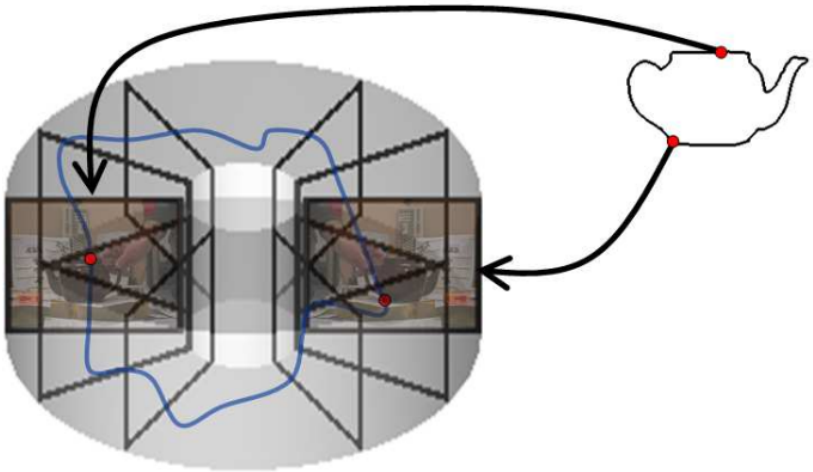
A closed contour p in a digraph $G = (V, \mathcal{E})$ with $V \subset \mathbb{R}^2$ connects vertices in a certain order. If we want additionally that p looks similar to a certain closed contour C, we have to find for each vertex of p a corresponding vertex of C.

Therefore, we have two information for every point on p, the position in V and the position in C. In other words, we are looking for a shortest path in the 3D space $C \times V$.

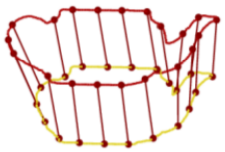
shape model

image

Optimizing Shape Prior



Product Space



Output

Literature *

Parametric Maximum Flow

- Gallo, Grigoriadis, Tarjan, *A fast parametric Maximum Flow Algorithm and Applications*, 1989, SIAM J. Computing, 18, 30–55.
- Kolmogorov, Boykov, Rother, *Applications of parametric Maxflow in Computer Vision*, 2007, IEEE ICCV.

Length Ratio

- Jermyn, Ishikawa, *Globally optimal regions and boundaries as minimum ratio weight cycles*, 2001, IEEE TPAMI 23(10), 1075–1088
- Schoenemann, Schmidt, Cremers, *Image Segmentation with Elastic Shape Priors via Global Geodesics in Product Spaces*, 2008, BMVC, 1–10.