OPPA European Social Fund
Prague & EU: We invest in your future.

Image as Markov Random Field and Applications[1]

Tomáš Svoboda, svoboda@cmp.felk.cvut.cz
Czech Technical University in Prague, Center for Machine Perception
http://cmp.felk.cvut.cz

Talk Outline **Last update:** December 2, 2010

- intro, applications
- MRF, labeling . . .
- how it can be computed at all?
- Applications in segmentation: GraphCut, GrabCut, demos

[1] Please note that the lecture will be accompanied be several sketches and derivations on the blackboard and few live-interactive demos in Matlab

About this very lecture

few notes before we start

- MRF is a complicated topic
- this lecture is introductory
- some simplifications in order not to lose the whole picture
- most important references provided
- many accessible explanations on the web (wikipedia . . .)

Markov Random Field – MRF

From wikipedia[2]:

A Markov random field, Markov network or undirected graphical model is a graphical model in which a set of random variables have a Markov property described by an undirected graph.

More formal definition, which we follow, can be found in the first chapter[3] of the book [4].

Let think about **images**:

- image intensities are the random variables
- the values depend only on their immediate spatial neighborhood which is the Markov property
- images are organized in a regular grid which can be seen as an undirected graph

[2] http://en.wikipedia.org/wiki/Markov_random_field

[3] freely available at: http://www.nlpr.ia.ac.cn/users/szli/MRF_Book/MRF_Book.html

Labeling for image analysis

Many image analysis and interpretation problems can be posed as labeling problems.

- Assign a label to image pixel (or to features in general).
- Image intensity can be considered as a label (think about pallete images).

Sites

$\mathcal{S}$ index a discrete set of m sites.

$$\mathcal{S} = \{1, \ldots, m\}$$

Site could be:

- individual pixel
- image region
- corner point, line segment, surface patch . . .

Labels and sites

A label is an event that may happen to a site.

Set of labels $\mathcal{L}$.

We will discuss disrete labels:

$$\mathcal{L} = \{l_1, \ldots, l_M\}$$

shortly

$$\mathcal{L} = \{1, \ldots, M\}$$

What can be a label?

- intensity value
- object label
- in edge detection binary flag $\mathcal{L} = \{\text{edge}, \text{nonedge}\}$
- ...

Ordering of labels

Some labels can be ordered some not.

Ordered labels can be used to measure distance between labels.

The Labeling Problem

Assigning a label from the label set $\mathcal{L}$ to each of the sites $\mathcal{S}$.

Example: edge detection in an image

Assign a label f_i from the set $\mathcal{L} = \{\text{edge}, \text{nonedge}\}$ to site $i \in \mathcal{S}$ where the elements in $\mathcal{S}$ index the image pixels. The set

$$f = \{f_1, \ldots, f_m\}$$

is called a labeling.

Unique labels

When each site is assigned a unique label, labeling can be seen as mapping from $\mathcal{S}$ to $\mathcal{L}$.

$$f : \mathcal{S} \longrightarrow \mathcal{L}$$

A labeling is also called a coloring in mathematical programming.

How many possible labelings?

Assuming all m sites have the same label set $\mathcal{L}$

$$\mathbb{F} = \underbrace{\mathcal{L} \times \mathcal{L} \times \ldots \times \mathcal{L}}_{m \text{ times}} = \mathcal{L}^m$$

Imagine the image restoration problem [3].

The the m is the number of pixels in the image and $\mathcal{L}$ equals to number of intensity levels.

Many, many possible labelings. Usually only few are good.

Labeling problems in image analysis

- image restoration
- region segmentation
- edge detection
- object detection and recognition
- stereo
- . . .

Labeling with contextual analysis

In images, site neighborhood matters.

A probability $P(f_i)$ does not depend only on the site but also on the labeling around. Mathematically speaking we must consider conditional probability

$$P(f_i|\{f_{i'}\})$$

where $\{f_{i'}\}$ denotes the set of other labels.

no context:

$$P(f) = \prod_{i \in \mathcal{S}} P(f_i)$$

Markov Random Field - MRF

$$P(f_i|f_{\mathcal{S}-\{i\}}) = P(f_i|f_{\mathcal{N}_i})$$

where $f_{\mathcal{N}_i}$ stands for the labels at the sites neighboring i.

MRF a Gibbs Random Fields

How to specify an MRF: in terms of conditional probabilities $P(f_i|f_{\mathcal{N}_i})$ or joint probability $P(f)$?

Hammersley–Clifford theorem about equivalence between MRF and Gibbs distribution.

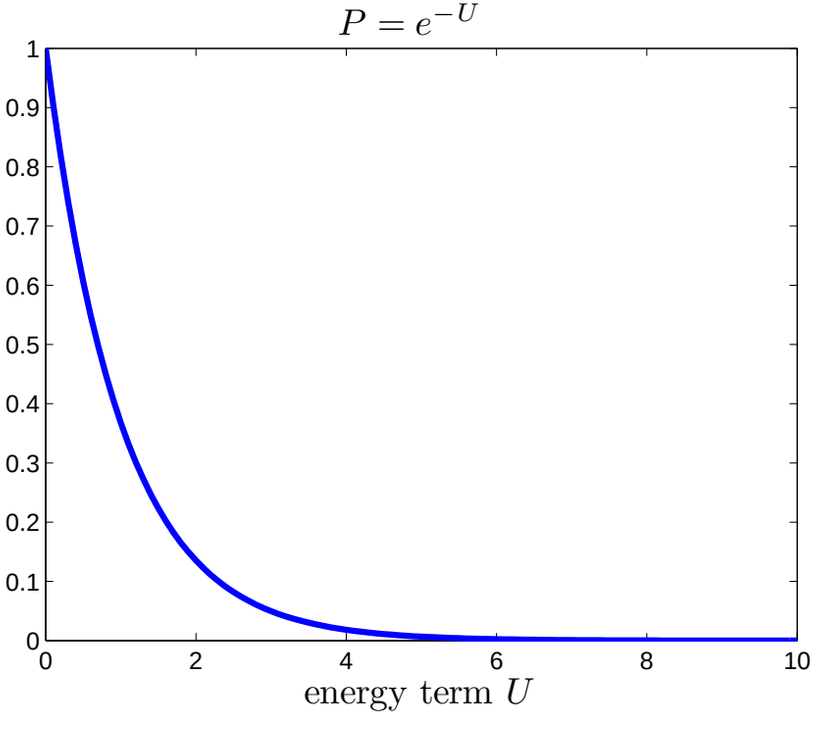
A set of random variables F is said to be a Gibbs Random Fields on $\mathcal{S}$ with respect to $\mathcal{N}$ iff its configurations obey a Gibbs distribution

$$P(f) = Z^{-1} \times e^{-\frac{1}{T}U(f)}$$

where

$$Z = \sum_{f \in \mathbb{F}} e^{-\frac{1}{T}U(f)}$$

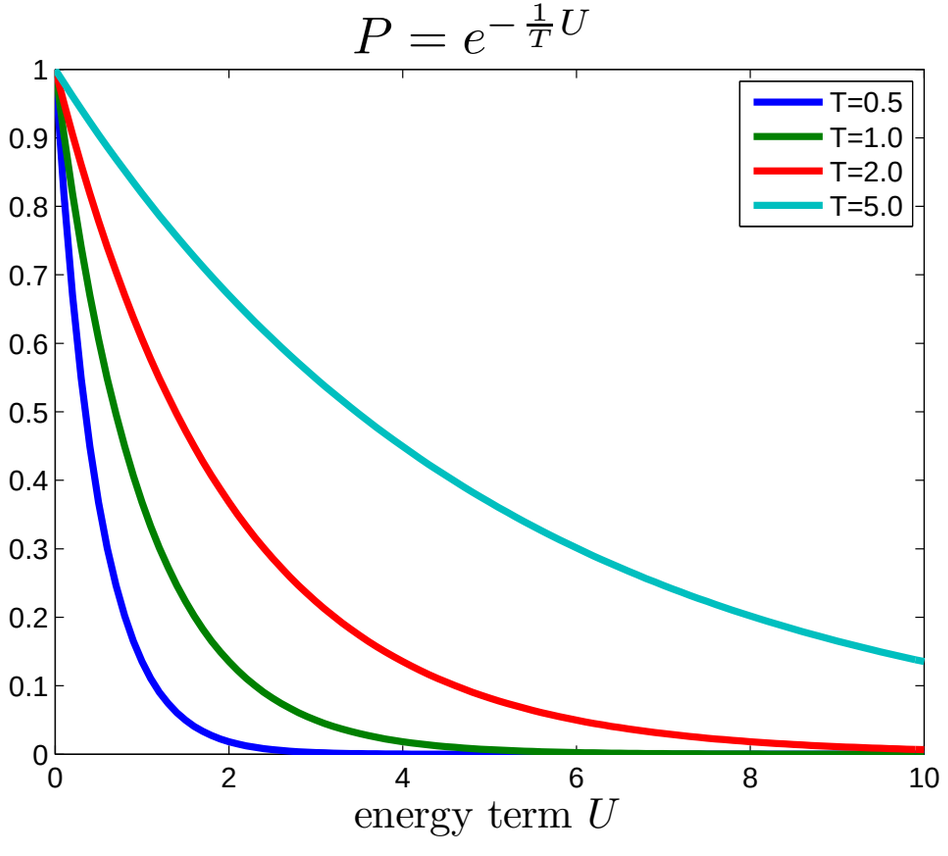
is a normalizing constant.



Gibbs distribution

$$P(f) = Z^{-1} \times e^{-\frac{1}{T}U(f)}$$

- T is tempertature, $T = 1$ unless stated otherwise
- $U(f)$ is the energy function



Energy function

$$U(f) = \sum_{c \in \mathcal{C}} V_c(f)$$

is a sum of clique potentials $V_c(f)$ over all possible cliques $\mathcal{C}$.

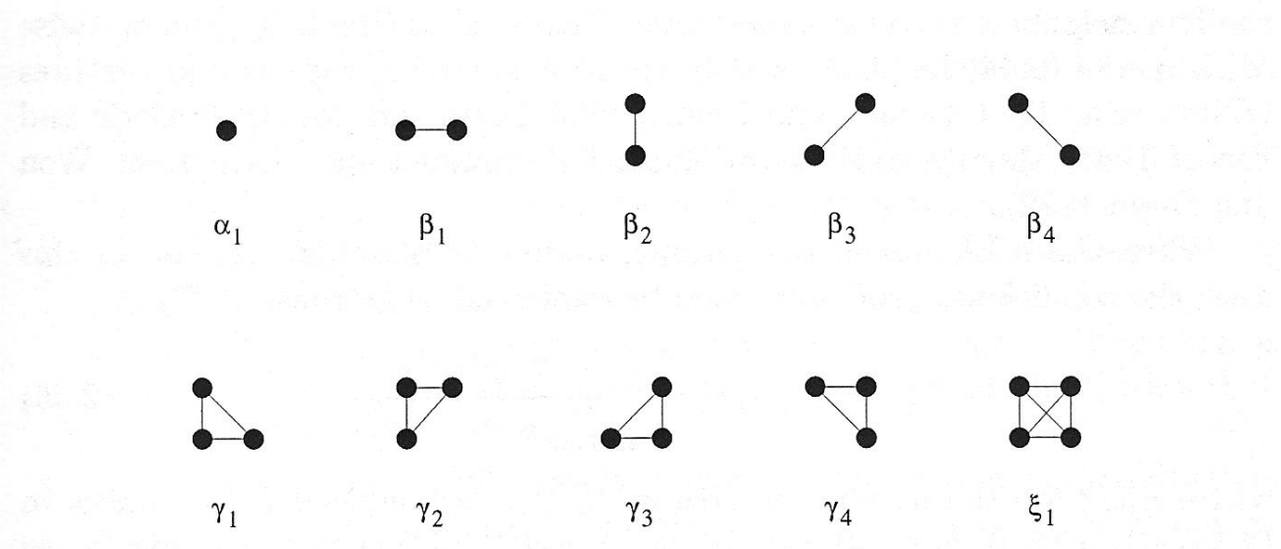


Figure 2.3: Clique types and associated potential parameters for the second-order neighborhood system. Sites are shown as dots and neighboring relationships as joining lines.

4

[4] Illustration from the book [4]

Simple cliques, Auto–models

Contextual constraints on two labels

$$U(f) = \sum_{i \in \mathcal{S}} V_1(f_i) + \sum_{i \in \mathcal{S}} \sum_{i' \in \mathcal{N}_i} V_2(f_i, f_{i'})$$

This can be interpreted as

$$U(f) = U_{data}(f) + U_{smooth}(f)$$

Neighborhood and cliques on a regular lattices

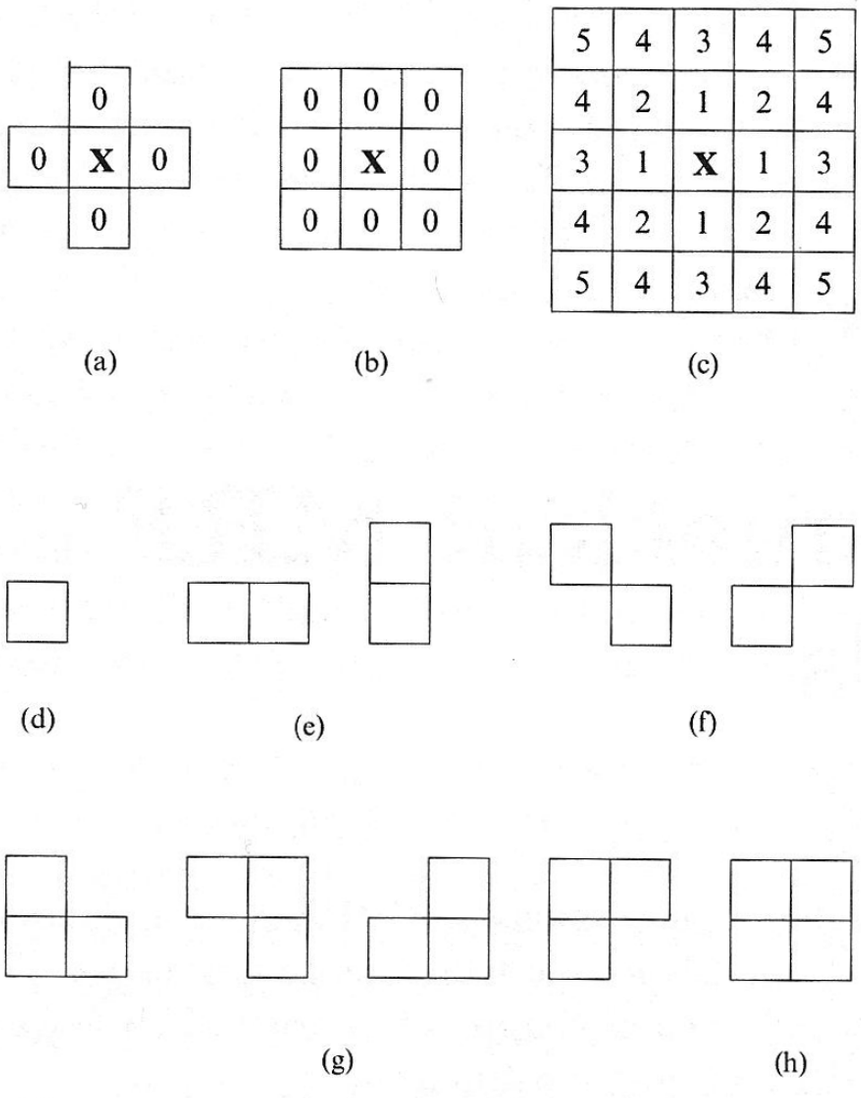


Figure 2.1: Neighborhood and cliques on a lattice of regular sites. [5]

[5] Illustration from the book [4]

Energy minimization

What is to be computed? Gibbs distribution:

$$P(f) = Z^{-1} \times e^{-\frac{1}{T}U(f)}$$

Remind that f is the desired labeling

$$f : \mathcal{S} \longrightarrow \mathcal{L}$$

In MAP formulation we seek the most probable labeling $P(f)$.

The best labeling minimizes energy $U(f)$

It is a combinatorial problem. The next explanation follows mainly [2].

Energy – data term and smoothness term

$$U(f) = U_{data}(f) + U_{smooth}(f)$$

The data term measures how well a label f_p fits the particular pixel (site) p. Globally,

$$U_{data}(f) = \sum_{p \in \mathcal{P}} D_p(f_p)$$

Example: in Image restoration $D_p(f_p) = (f_p - I_p)^2$, where I_p is the observed intensity and f_p is the label (assigned intensity).

Smoothness term

Expresses the context. Setting a proper smoothness term is much more tricky.

- smooth but not everywhere (think about object boundary)
- discontinuity preserving

Interacting pixels

We consider the energy

$$U(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{\{p,q\} \in \mathcal{N}} V_{p,q}(f_p, f_q)$$

where $\mathcal{N}$ is the set of interacting pixels, typically adjacent pixels. D_p is assumed to be nonnegative.

Interaction of adjacent pixels may have long range impact!

$V_{p,q}$ is called interaction penalty.

(reasonable) interaction penalties

labels $\alpha, \beta, \gamma \in \mathcal{L}$

$$V(\alpha,\beta) = 0 \quad \Leftrightarrow \quad \alpha = \beta\,, \tag{1}$$

$$V(\alpha,\beta) \quad = \quad V(\beta,\alpha) \geq 0\,, \tag{2}$$

$$V(\alpha,\beta) \quad \leq \quad V(\alpha,\gamma) + V(\gamma,\beta) \tag{3}$$

Penalty is metric if all hold and semimetric if only (1,2) is satisfied.

Examples of discontinutity preserving penalties

- truncated quadratic $V(\alpha,\beta) = \min(K, (\alpha-\beta)^2)$
- truncated absolute distance $V(\alpha,\beta) = \min(K, |\alpha-\beta|)$
- Potts model $V(\alpha,\beta) = KT(\alpha \neq \beta)$, where $T() = 1$ if argument is true, otherwise 0.

towards the best labeling (minimum energy)

Catch: finding global minimum is NP complete even for the simple Potts model.

$\longrightarrow$ local minimum is sought.

Problem

If the solution is poor

- poor choice of energy funtion
- local minimum is far from the global one

Local minimum

a labeling f is a local minimum of the energy U if

$$U(f) \leq U(f')$$

for any f' near to f. In case of discrete labeling near means withing single move of f.

Many local minimization method use standard moves, where only one pixel (site) may change label at a time.

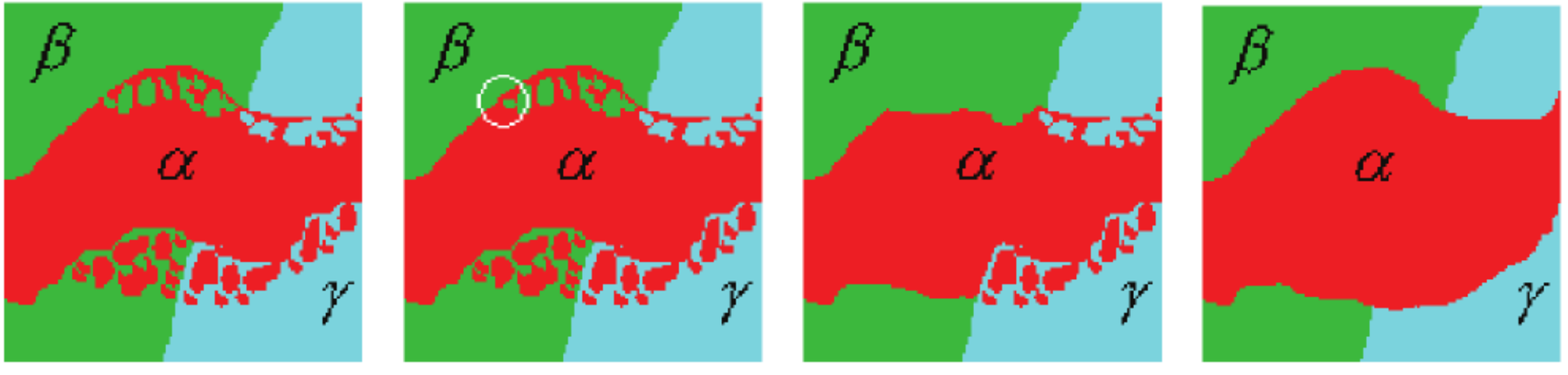
Example: greedy optimization

for each pixel, the label which gives the largest descrease of the energy is chosen.

Fast approximate energy minimization via graph cuts

We only sketch the main ideas from the seminal work [2][6]

Allow more than just one label change at a time



[7]

α-β swap and α-expansion.

[6] Freely available implementation. Many difficult problems in computer vision were solved by using this method and implementation.

[7] image from [2]

Algorithm

1. Start with an arbitrary labeling f
2. Set success := 0
3. For each pair of labels $\{\alpha, \beta\} \subset \mathcal{L}$
   3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α-β swap of f
   3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return f

---

1. Start with an arbitrary labeling f
2. Set success := 0
3. For each label $\alpha \in \mathcal{L}$
   3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α-expansion of f
   3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return f

Algorithm

1. Start with an arbitrary labeling f
2. Set success := 0
3. For each pair of labels $\{\alpha, \beta\} \subset \mathcal{L}$ iteration
   - 3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α-β swap of f
   - 3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return f

---

1. Start with an arbitrary labeling f
2. Set success := 0
3. For each label $\alpha \in \mathcal{L}$
   - 3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α-expansion of f
   - 3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return f

Algorithm

1. Start with an arbitrary labeling f — cycle
2. Set success := 0
3. For each pair of labels $\{\alpha, \beta\} \subset \mathcal{L}$ — iteration
   - 3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α-β swap of f
   - 3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return f

---

1. Start with an arbitrary labeling f
2. Set success := 0
3. For each label $\alpha \in \mathcal{L}$
   - 3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α-expansion of f
   - 3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return f

α-β swap

1. Start with an arbitrary labeling f — cycle
2. Set success := 0
3. For each pair of labels $\{\alpha, \beta\} \subset \mathcal{L}$ — iteration
   - 3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α-β swap of f
   - 3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return f

- How many iterations in each cycle?
- A cycle is successful if a strictly better labeling is found in any iteration.
- Cycling stops after first unsuccessful cycle.

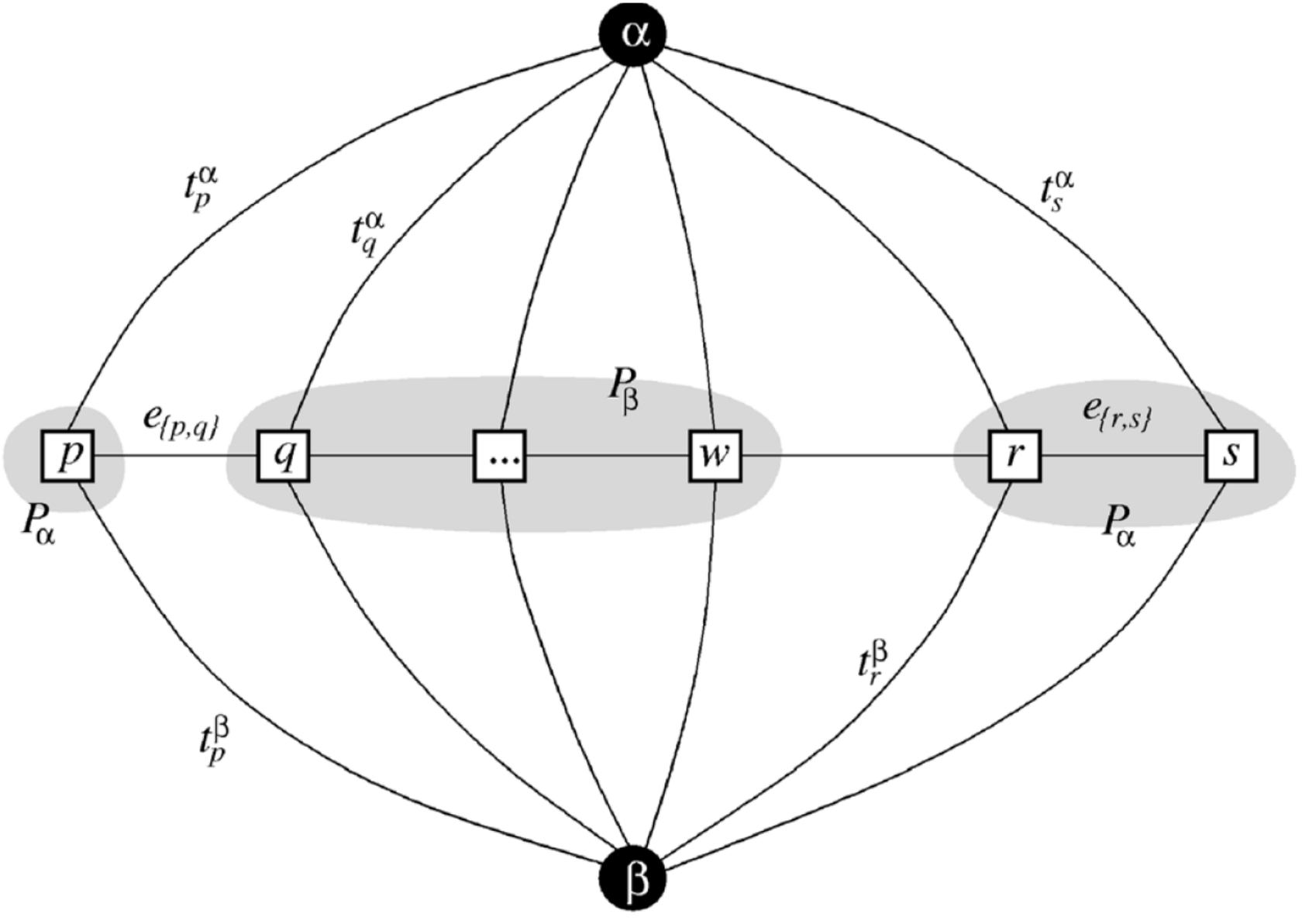
α-β swap

1. Start with an arbitrary labeling f — cycle
2. Set success := 0
3. For each pair of labels $\{\alpha, \beta\} \subset \mathcal{L}$ — iteration
   - 3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α-β swap of f
   - 3.2. If $E(\hat{f}) < E(f)$, set f := $\hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return f

The best α-β swap is found by composing a graph and finding min-cut.

- min-cut is equivalent to max-flow (Ford-Fulkerson theorem)
- Max-flow between terminals is a standard problem in Combinatorial Optimization.
- Algorithms with low-order polynomial complexities exist.

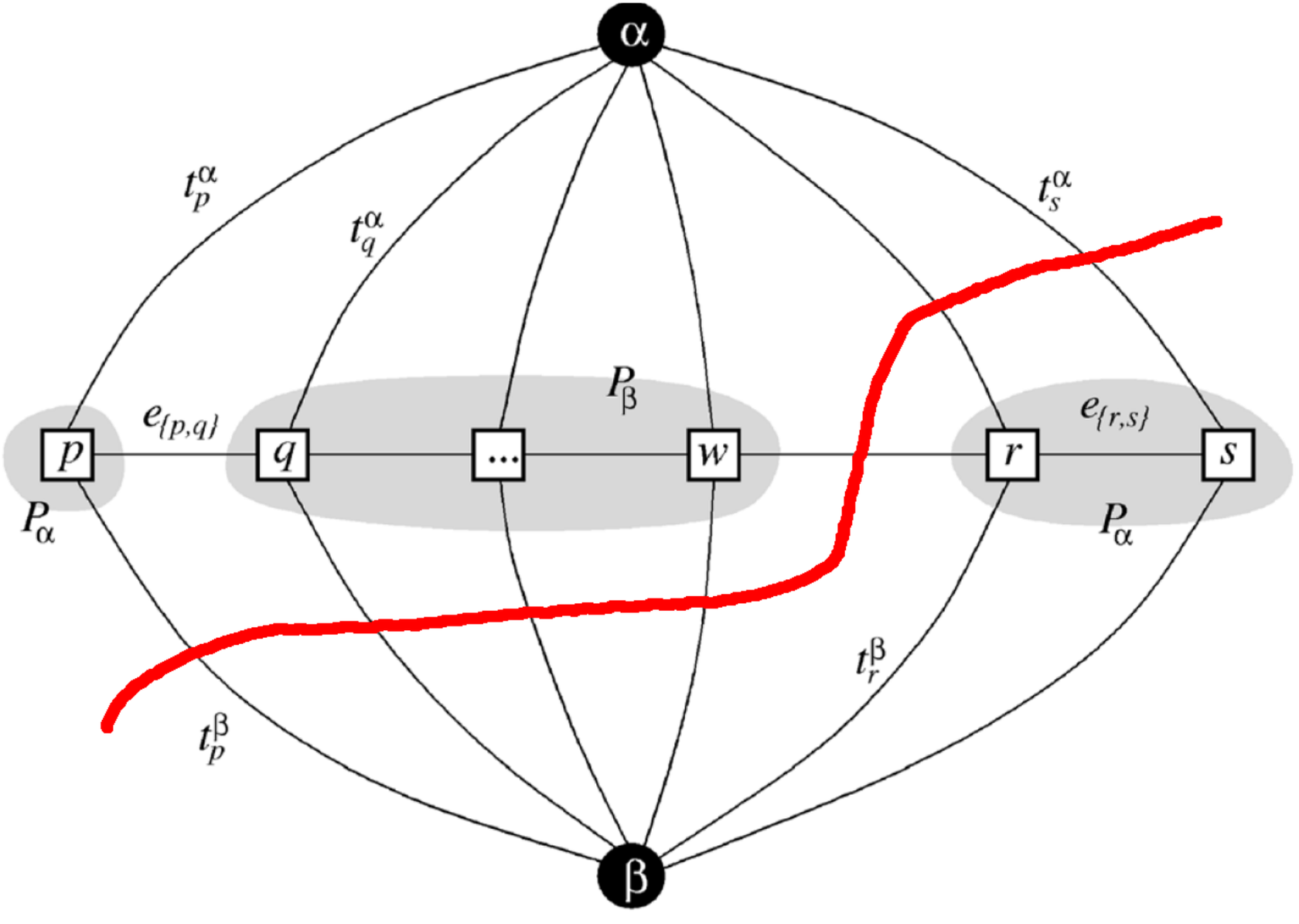
Finding optimal α-β swap



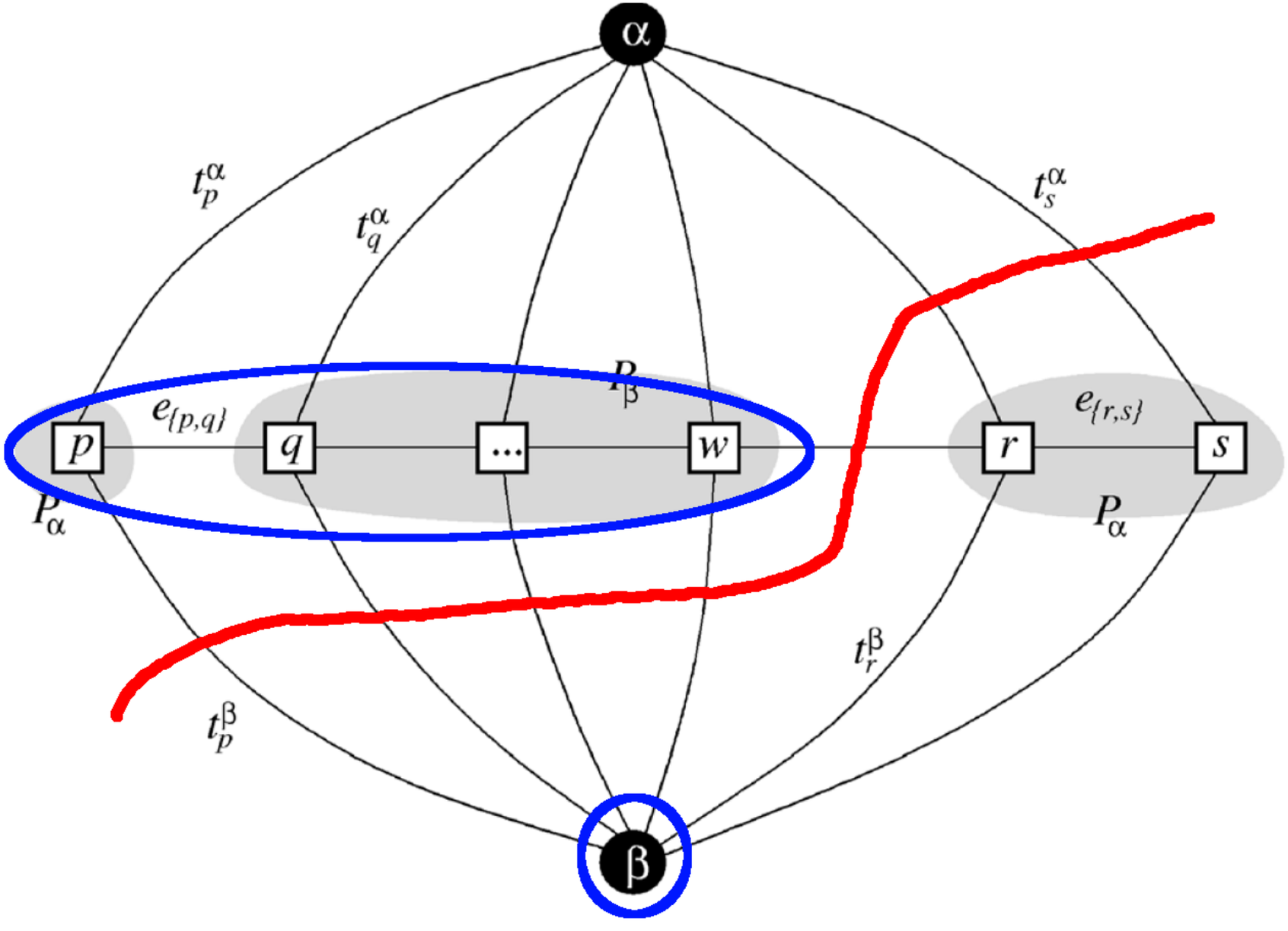
8

[8] illustration from [2]

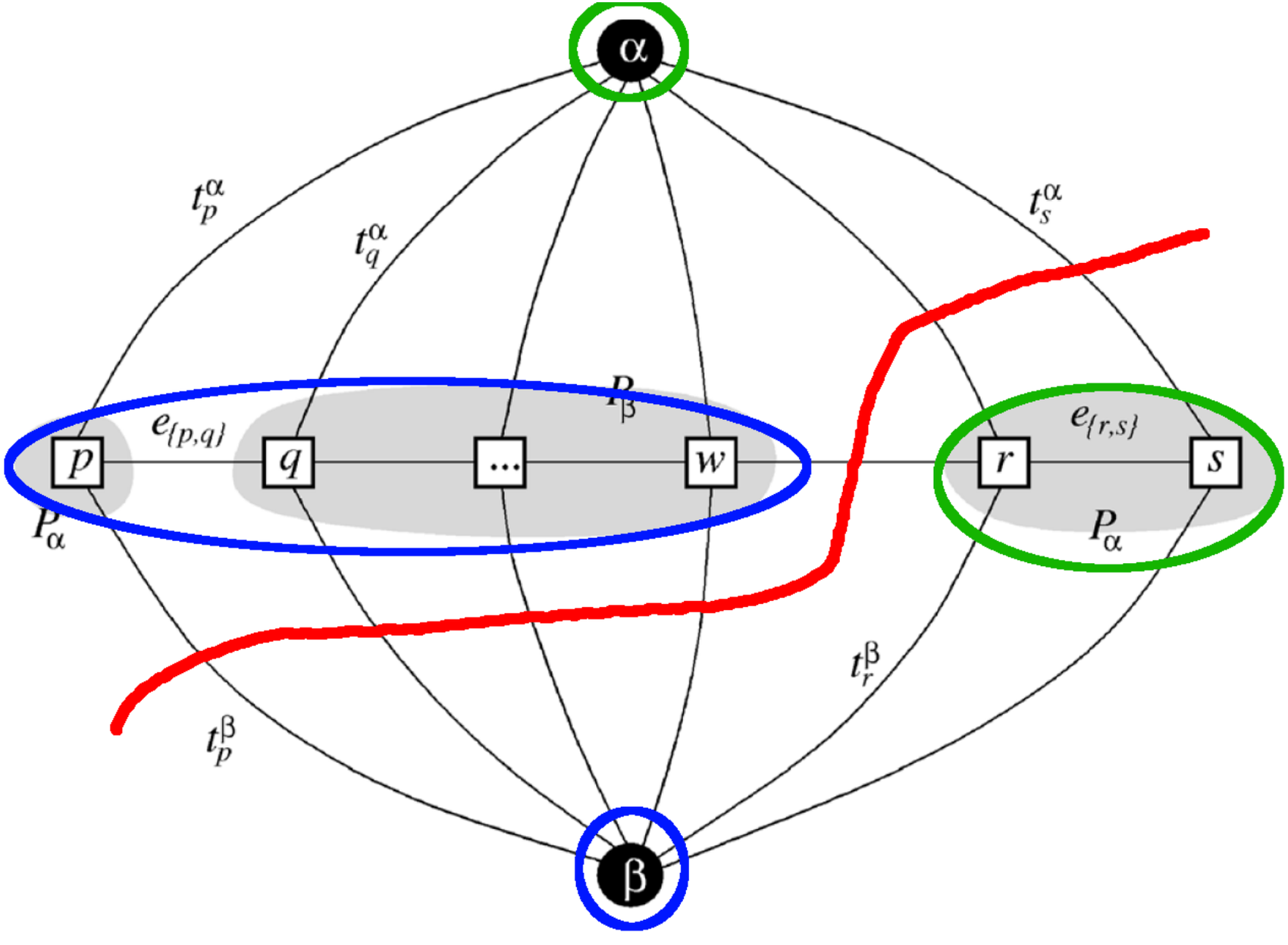
Finding optimal α-β swap, min-cut



Finding optimal α-β swap, re-labeling



Finding optimal α-β swap, re-labeling



MRF in image segmentation

Pátek 4.6.

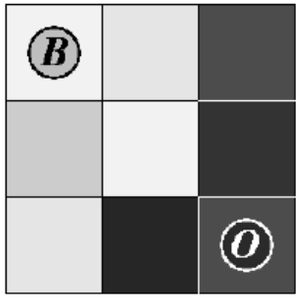
Segmentation with seeds – Interactive GraphCuts

Idea: denote few pixels that trully belongs to object or background and than refine (grow) by using soft constraints.

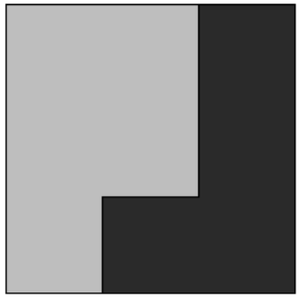


- data term
- boundary penalties and/or pixel interactions
- how to find the optimal boundary between object and background

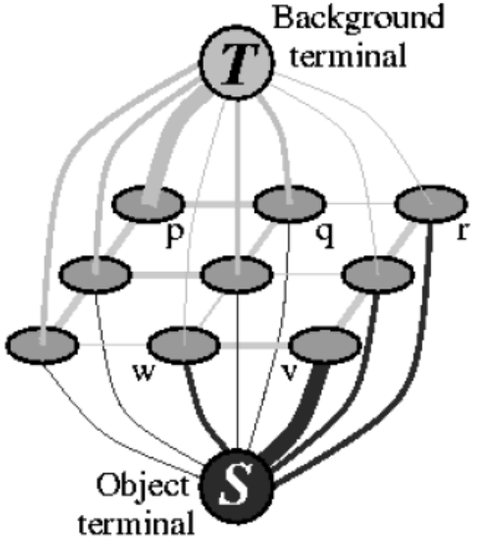
Segmentation with seeds – graph



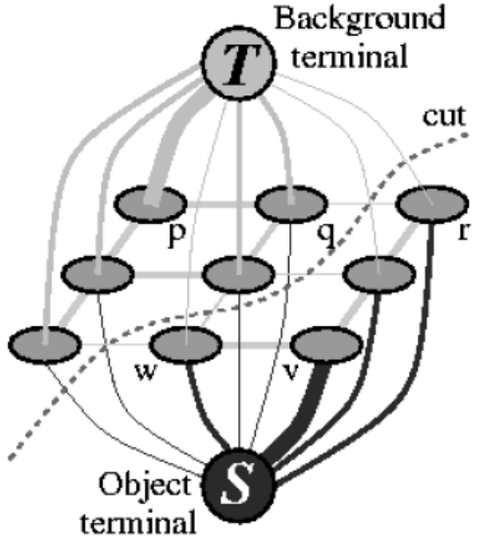
(a) Image with seeds.



(d) Segmentation results.



(b) Graph.



(c) Cut.

[9]illustration from [1]

Edges, cuts, segmentation

Explained on the blackboard.

See [1] for details.

What data term?

$$U_{data}(\text{obj}, \text{bck}) = \sum_{p \in \text{obj}} D(\text{obj}) + \sum_{p \in \text{bck}} D(\text{bck})$$

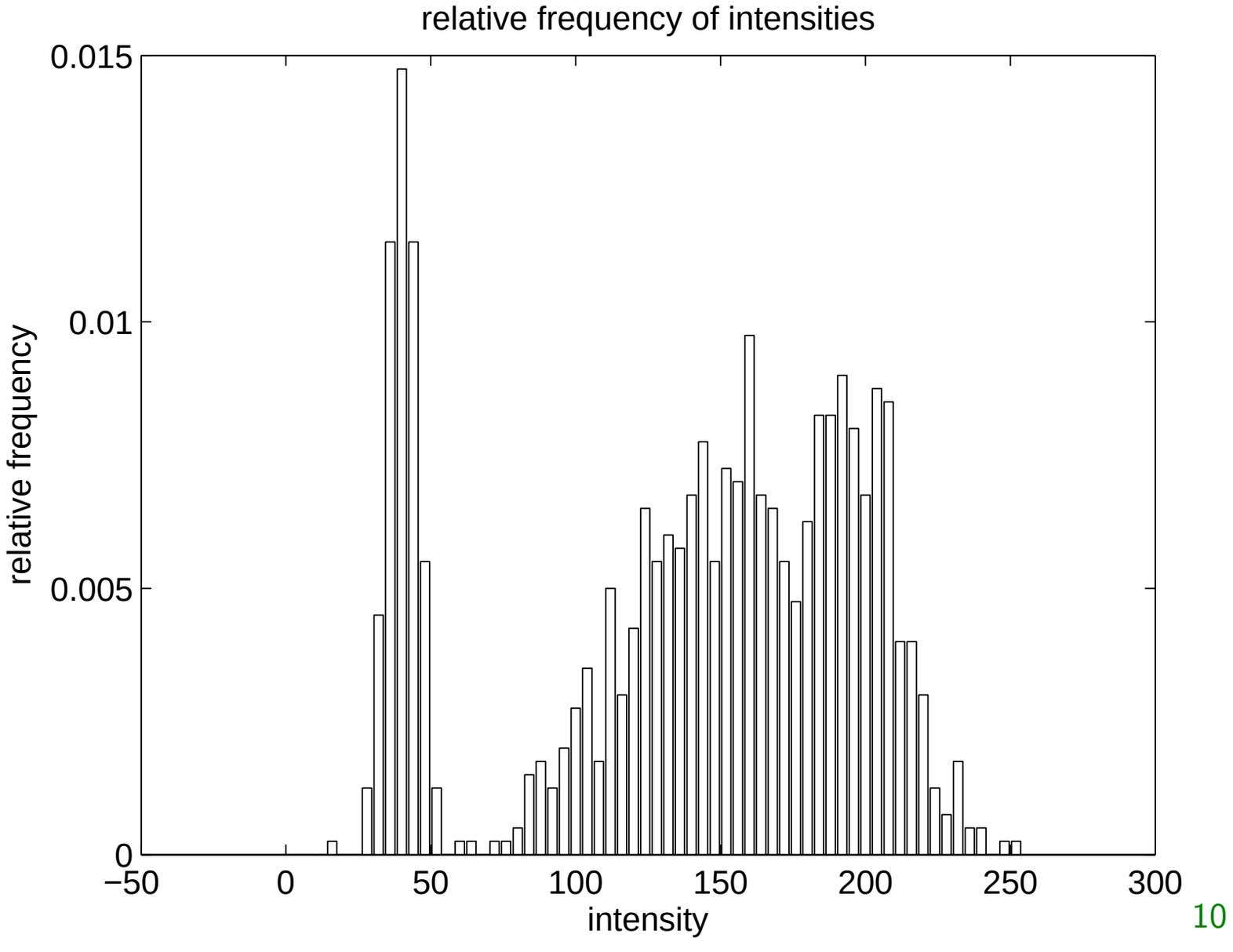
The better match between pixel and “obj” or “bck” model the lower energy (penalty).

How to model?

for simplicity, you may think about the opposite case, the better match the higher value

- background, foreground (object) pixels
- intensity or color distributions
- histograms
- parametric models: GMM – Gaussian Mixture Model

Image intensities - 1D GMM



[10]

[10] Demo codes from [6]

Image intensities - 1D GMM

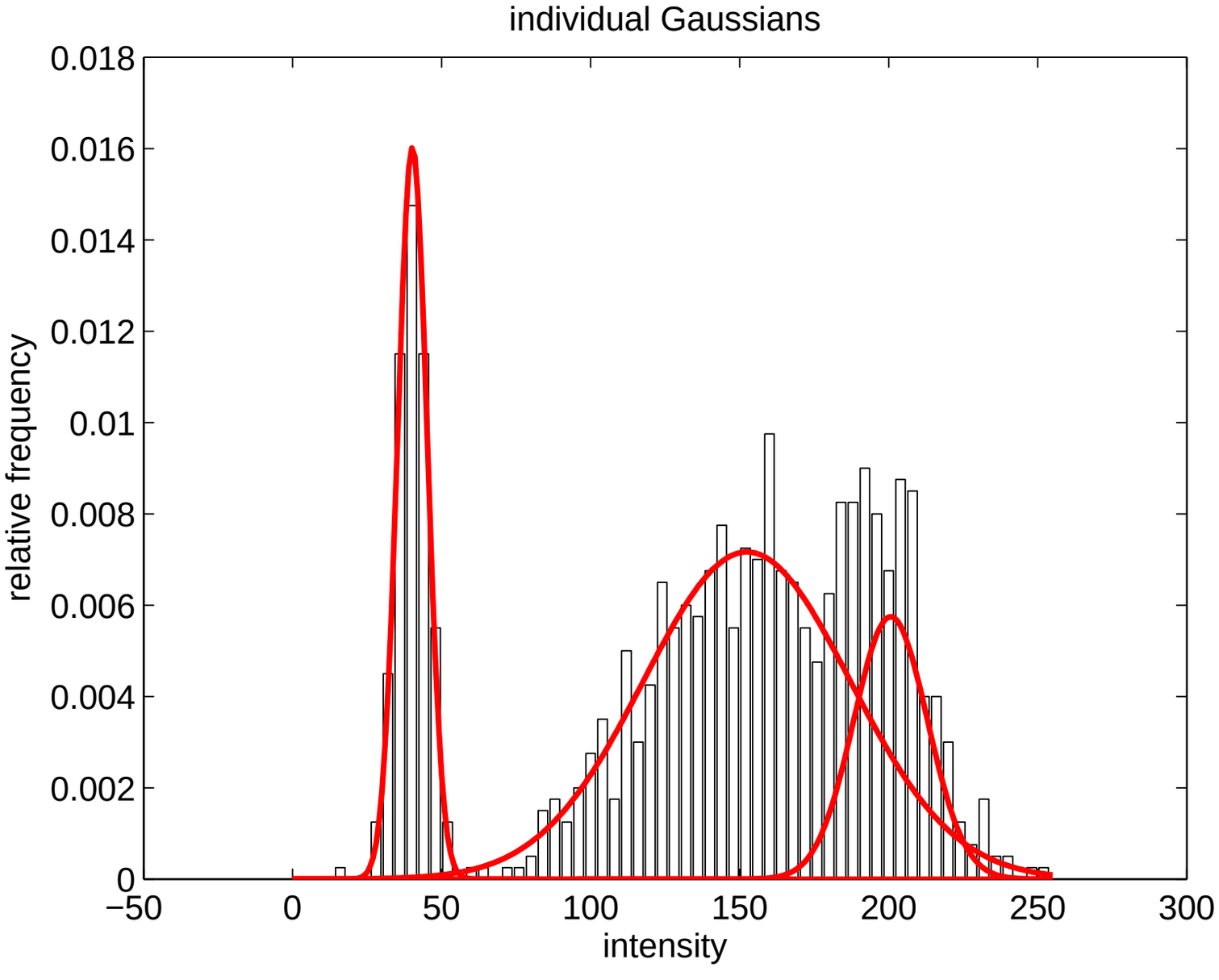
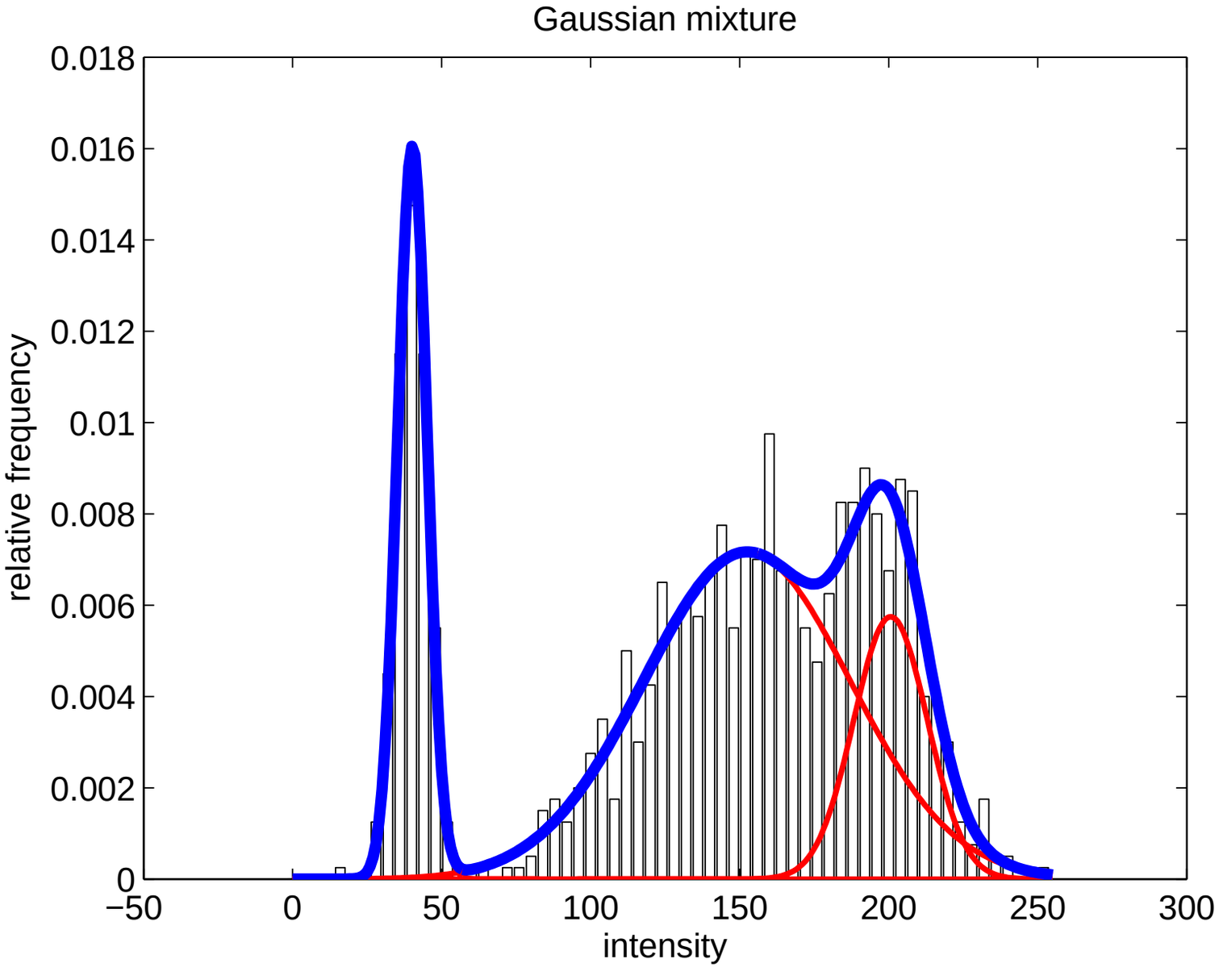
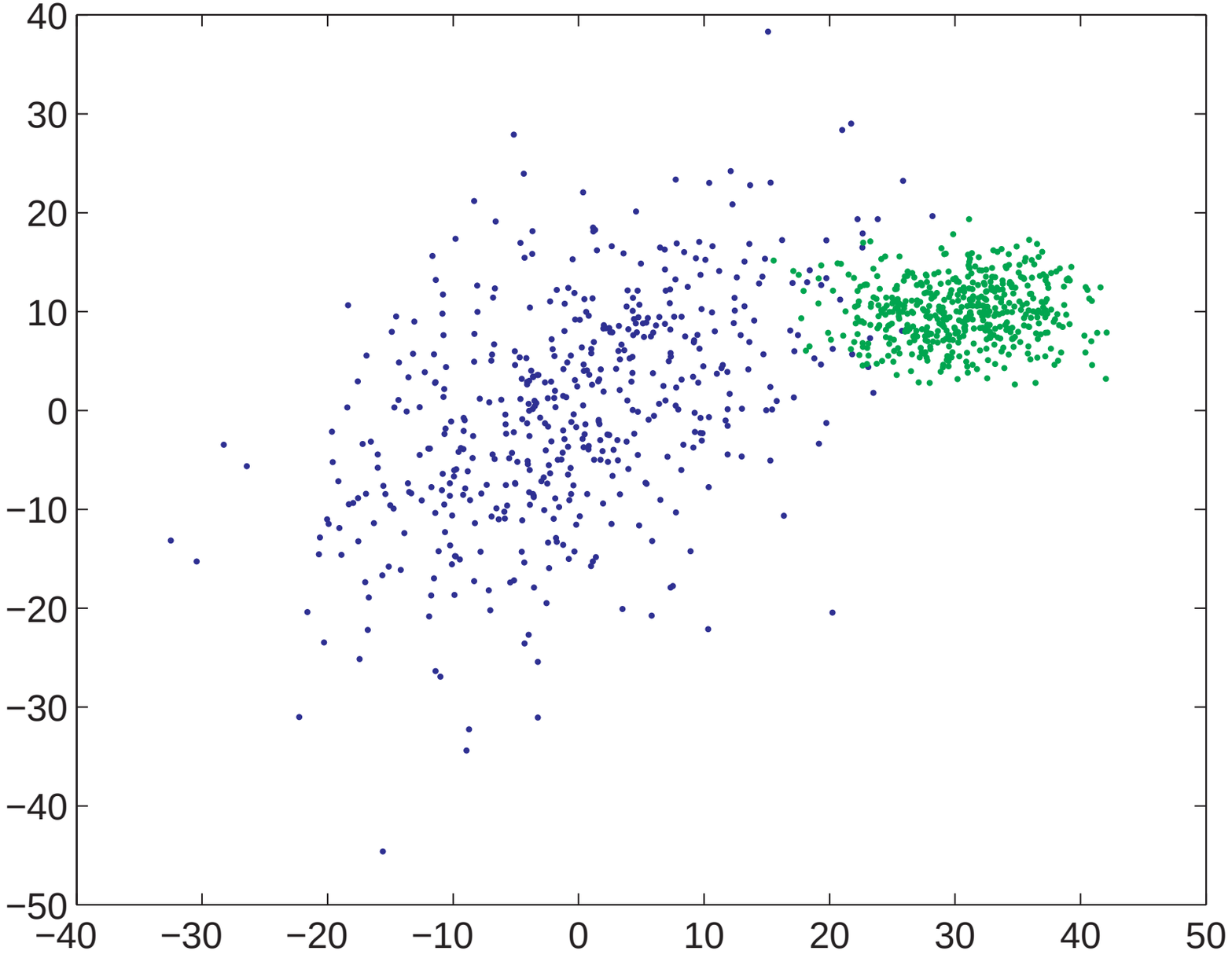


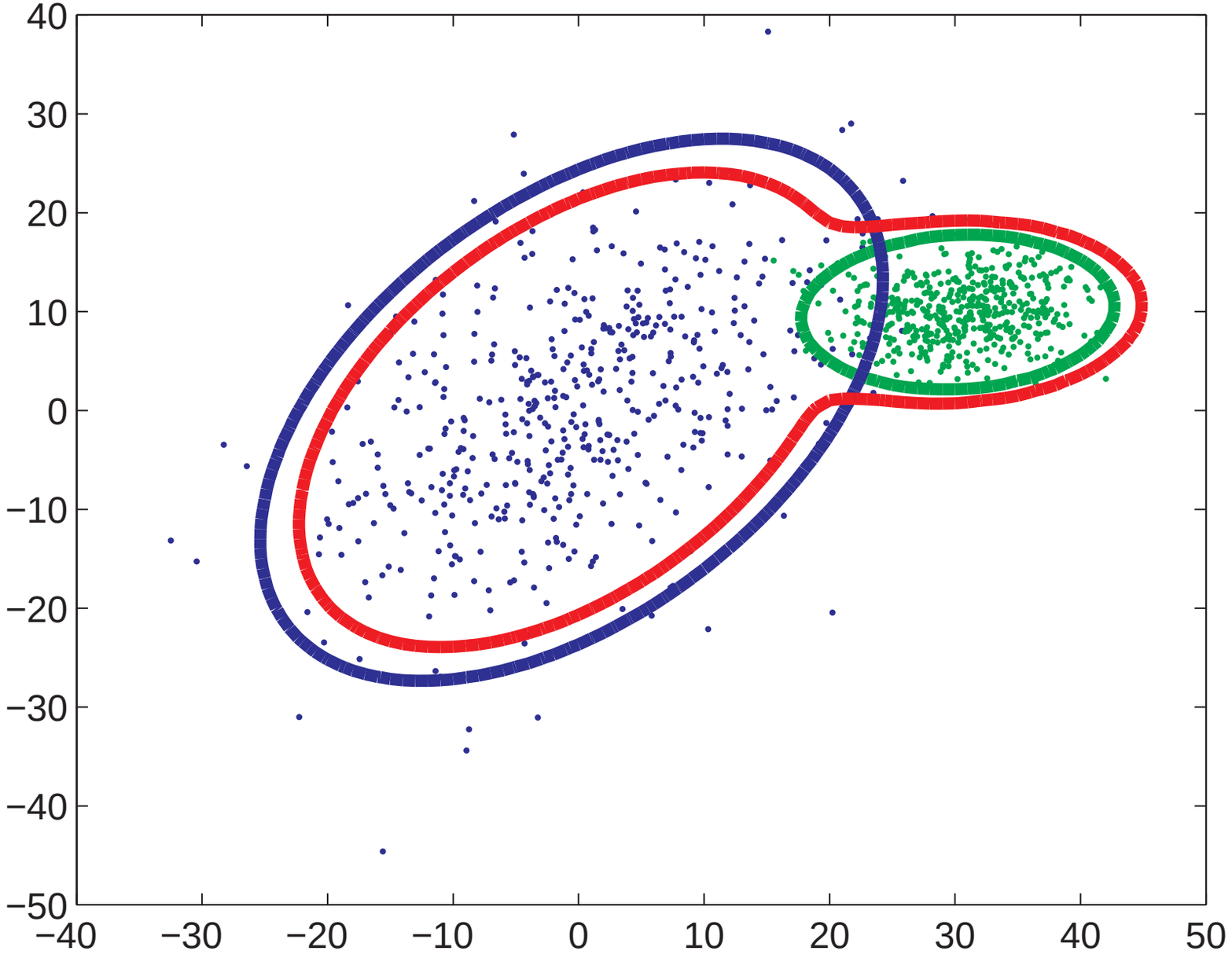
Image intensities - 1D GMM



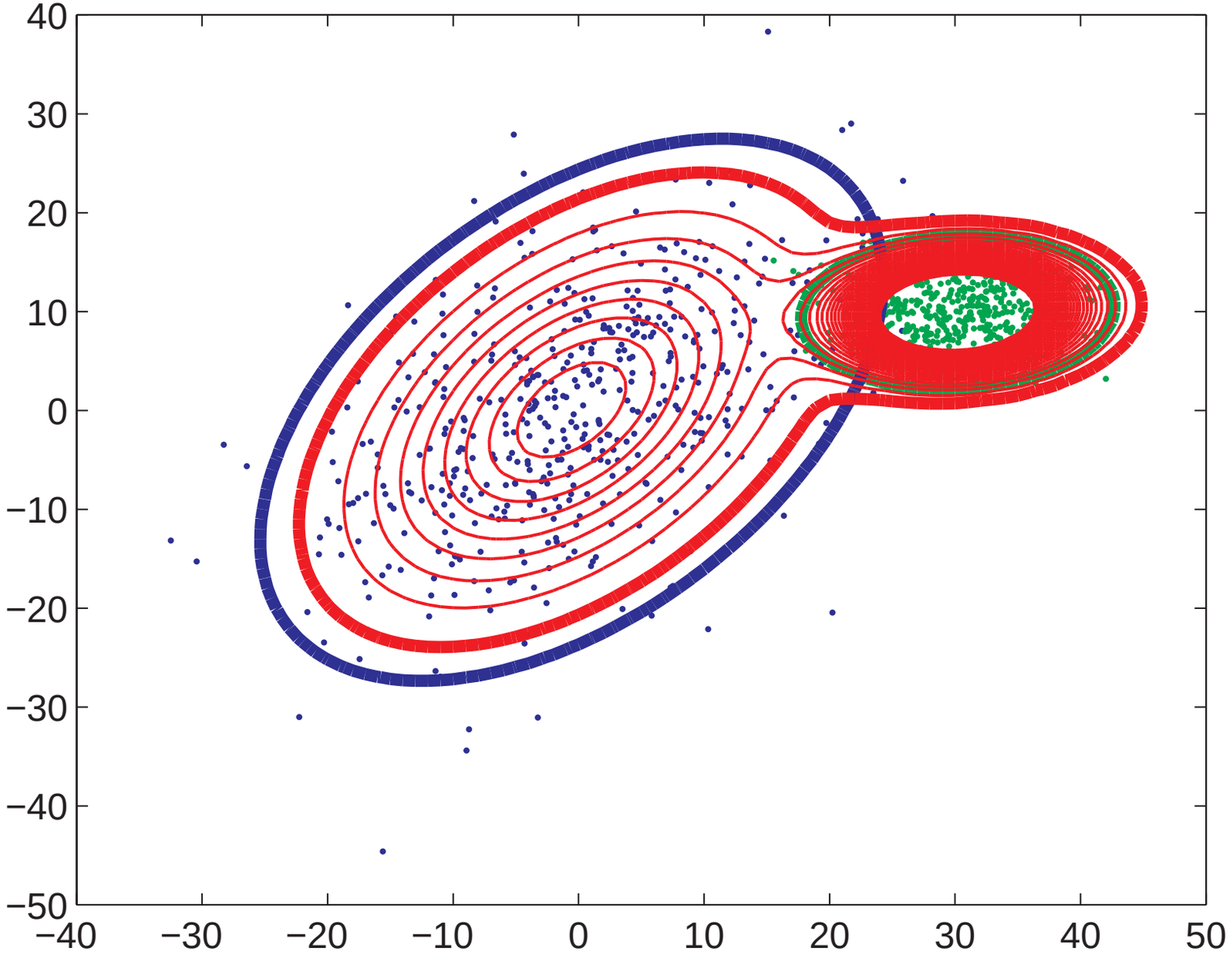
2D GMM



2D GMM



2D GMM



Data term by GMM – math summary

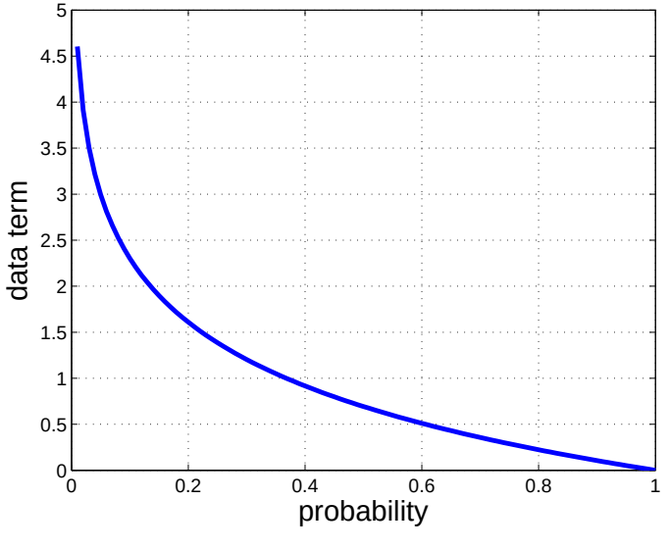
$$p(\mathbf{x}|f_i) = \sum_{k=1}^{K} w_k^{f_i} \frac{1}{(2\pi)^{\frac{d}{2}} \left|\Sigma_k^{f_i}\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_k^{f_i})^T \Sigma_k^{f_i\,-1}(\mathbf{x}-\mu_k^{f_i})\right),$$

where d is the dimension.

- K number of Gaussians. User defined.
- for each label $\mathcal{L} = \{\mathrm{obj}, \mathrm{bck}\}$ different w_k, μ_k, Σ_k estimated from the data (seeds)
- $\mathbf{x}$ pixel value, can be intensity, color vector, . . .

Data term

$$\begin{aligned} D(\mathrm{obj}) &= -\ln p(\mathbf{x}|\mathrm{obj}) \\ D(\mathrm{bck}) &= -\ln p(\mathbf{x}|\mathrm{bck}) \end{aligned}$$



Results data term only

See the live demo[11]

[11] Demo codes courtesy of V. Franc and A. Shekhovtsov

What boundary (interaction) term?

The Potts model: $V(p,q) = \lambda T(p \neq q)$, where $T() = 1$ if argument is true, otherwise 0.

Effect of λ, see the live demo.

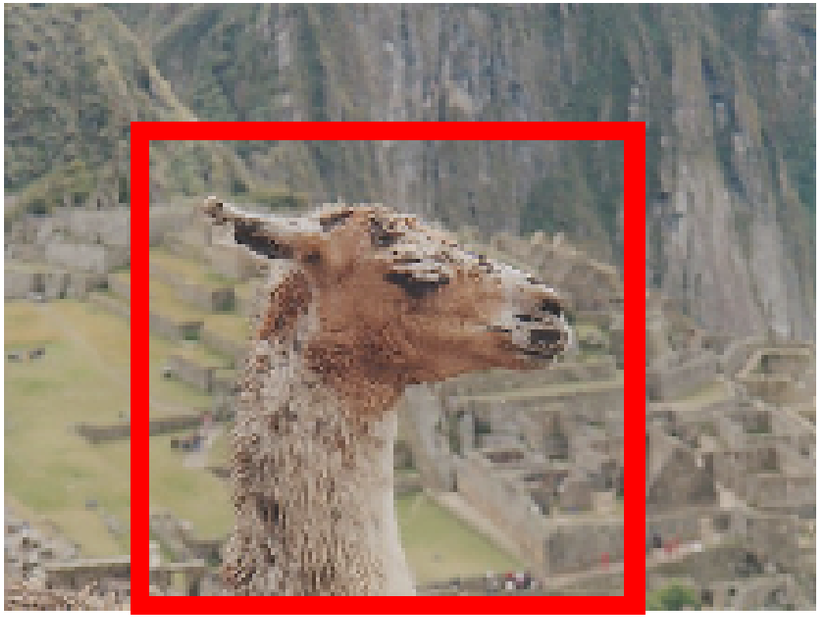
Results for data and interaction terms

GrabCut – going beyond the seeds

The main idea: Iterate the graphcut and refine the data terms in each iteration. Stop if the energy (penalty) does not decrease. [5]

The practical motivation

Further reduce the user interaction.



[12]

[12] images from [5]

GrabCut – algorithm

Init: Specify background pixels T_B. $T_F = 0$; $T_U = \bar{T}_B$

Iterative minimization:

1. assign GMM components to pixels in T_U
2. learn GMM parameters from data
3. segment by using min-cut (graphcut algorithm)
4. repeat from step 1 until convergence

Optional: edit some pixels and repeat the min-cut.

References

[1] Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In Proceedings of International Conference on Computer Vision (ICCV), 2001.

[2] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(11):1222–1239, November 2001.

[3] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6(6):721–741, 1984.

[4] Stan Z. Li. Markov Random Field Modeling in Image Analysis. Springer, 2009.

[5] Carsten Rother, Vladimir Kolomogorov, and Andrew Blake. GrabCut: Interactive foreground extraction isong iterated graph cuts. ACM Transactions on Graphics (SIGGRAPH'04), 2004.

[6] Tomáš Svoboda, Jan Kybic, and Václav Hlaváč. Image Processing, Analysis and Machine Vision. A MATLAB Companion. Thomson, 2007. Accompanying www site `http://visionbook.felk.cvut.cz`.

End

OPPA European Social Fund
Prague & EU: We invest in your future.