Chapter 1

Image Segmentation

1.1 The Labeling Problem

Většina problémů strojového vidění se dá vyjádřit jako tzv. problém labelování, tj. přidělení labelu každému obrazovému elementu. To specify a labeling problem one has to define a set of sites and a set of labels first as in [Li, 2009]. Let $\boldsymbol{\mathcal{S}}$ index a discrete set of $\boldsymbol{m}$ sites

$$\boldsymbol{\mathcal{S}} = \{\mathbf{1}, \dots, \boldsymbol{m}\} \tag{1.1}$$

where $\mathbf{1}, \dots, \boldsymbol{m}$ are indices. In vision problems a site doesn't necessarily have to be an image pixel. A site can represent any kind of a spel[1] or an image feature such as a line, a corner point, edges, a surface patch, or a volume region. A set of sites may be categorized in terms of their regularity to spatially regular, e.g. a rectangular 2D image lattice, or spatially irregular, e.g. line segments and other similar image features extracted in more abstract level. In MRF models the sites are treated as unordered, i.e. a pixel $(\boldsymbol{i}, \boldsymbol{j})$ can be reindexed by a single number $\boldsymbol{k} \in \{\mathbf{1}, \dots, \boldsymbol{m}\}$. The interrelationship between two sites is characterized by a neighborhood system.

Let $\boldsymbol{\mathcal{L}}$ be a set of labels. In contrast to the definition of the set of sites the set of labels doesn't have to be only a discrete set. For example $\boldsymbol{L}$ can represent the dynamic range for an analog pixel intensity:

$$\boldsymbol{\mathcal{L}_c} = [\boldsymbol{x_l}, \boldsymbol{x_u}] \subset \mathbb{R} \tag{1.2}$$

where $\boldsymbol{x_l}$ and $\boldsymbol{x_u}$ represents the lower and the upper value of the dynamic range. In the case of image segmentation the set $\boldsymbol{L}$ is defined on a discrete space

$$\boldsymbol{\mathcal{L}_d} = \{\mathbf{1}, \dots, \boldsymbol{M}\} \tag{1.3}$$

in which $\boldsymbol{M}$ denotes the number of labels that corresponds to the number of image segments.

In many cases (image restoration etc.) the label set is ordered, which means that a distance measure can be defined to describe the qualitative difference between two labels, e.g. the Euclidean distance. In contrast the segmentation problem often allows only the relationship ”equal” and ”nonequal”.

---

[1] Spel (spatial element) can represent one pixel or an arbitrary image region.

Once both the set of sites and the set of labels are defined one can proceed to assigning a label to each of the sites. This is called the labeling problem. The labeling of the sites in S in terms of the labels in $\mathcal{L}$ is a set

$$f = \{f_1, \dots, f_m\}. \tag{1.4}$$

One can look at the labeling as a mapping from $\mathcal{S}$ to $\mathcal{L}$:

$$f : \mathcal{S} \to \mathcal{L}. \tag{1.5}$$

where each site is assigned a unique label. In the MRF theory, a labeling is often called a *configuration* of the field. A total number of possible configurations of a markov field over m sites with M possible labels is computed as M^m:

$$\mathbb{F} = \underbrace{\mathcal{L} \times \mathcal{L} \dots \times \mathcal{L}}_{m \text{ times}} = \mathcal{L}^m. \tag{1.6}$$

This number is astronomic even for a moderate number of labels and small images thus makes it impossible for optimizing by brute force.

In some cases possible labels can differ from site to site which gives the configuration space

$$\mathbb{F} = \mathcal{L}_1 \times \mathcal{L}_2 \dots \times \mathcal{L}_m = \mathcal{L}^m. \tag{1.7}$$

1.1.1 Types of Labeling Problems

In accordance with the work [Li, 2009], labeling problems can be classified w.r.t. the regularity of the sites and the continuity of the labels to the following four categories:

- LP1: Regular sites with continuous labels.
- LP2: Regular sites with discrete labels.
- LP3: Irregular sites with discrete labels.
- LP4: Irregular sites with continuous labels.

The first category form problems such as real (non binary) image restoration and smoothing. A typical representative of LP2 problems is image segmentation where the set of labels are given by number of image segments. Also edge detection belongs to this category. Object matching based on image features like corner points or line segments belongs to the category LP3 and a typical example of LP4 problem is pose estimation from a set of correspondences.

1.1.2 Involving the Contextual Constraints

The use of contextual information is ultimately necessary for proper image understanding as shown in [Pavlidis, 1986]. The first use of contextual information for solving an image analysis problem is published in [Chow, 1962].

From the probability point of view, the contextual constraints may be expressed either locally or globally. For locally expression conditional probabilities $P(f_i|\{f_{i'}\})$ takes place. Here $f_{i'}$ denotes the set of labels at the other sites $i \neq i'$. To express contextual constraints globally the joint probability $P(f)$ is used. Due to the more frequent cases where local information is more directly observed, a global inference is made based on those local properties.

In situations where no contextual constraints are made the labels are independent of one another:

$$P(f_i|\{f_{i'}\} = P(f_i),\ i \neq i'. \tag{1.8}$$

Therefore the joint probability is the simple product of local probabilities:

$$P(f) = \prod_{i \in S} P(f_i). \tag{1.9}$$

While this is advantageous for problem solving in most cases the contextual constraints should be involved to achieve better image understanding. The equation 1.8 don't hold anymore and making a global inference based on local information becomes a nontrivial task.

1.2 The Role of Optimization Principles

The exact solution to a vision problem hardly exist in many cases. This is due to various uncertainties in involved processes such as noise and other degradation factors. Using an optimization principle leads to a solution that optimizes a predefined objective function either explicitly or implicitly.

From optimization point of view three basic issues need to be figured out ([Li, 2009]): problem representation, formulation of objective function, and deriving an optimizing algorithm. To solve the first issue the site and label sets have to be defined. Probably the most critical part is formulation of objective function. It's a function that maps a solution to a real number and thus reflects the quality of that solution. The formulation also describes how various image features and contextual constraints are involved in the optimization procedure. The last issue describes the procedure for optimizing the objective function. A perfect algorithm should be able to find a local or even the global extrema of the objective function in an efficient way. The efficiency here reflects not only the computational time but also the size of searched solution space.

Assume without loss of generality that our goal is to minimize the objective function. Then the objective function is often called an energy function and it should be formulated so that a solution to a given problem corresponds to its minimum. Another role of the energy function is to guide the search through the solution space. From this point of view the minimum should be located in convex area. The problem that often arise is that a low-quality local minimum is found instead of the global one or a superior local one. To overcome the problem of finding a local minimum a stochastic method that can temporarily accept worse solution can be used. A typical member of this kind of methods is simulated annealing. Of course methods that are able to find global minimum are preferred but not always usable.

The formal approaches to optimization an energy function provide a convenient ways for the evaluation of different solutions. For improving the search efficiency by pruning of the solution space different heuristics can be used. Combining both of the approaches leads to a good strategy for solving a general optimization task.

An energy function E is defined by its form and parameters involved. The minimal solution is then expressed as follows:

$$f^* = \arg\min_f E(f|d, \theta) \tag{1.10}$$

where $\boldsymbol{f}$ denotes a solution, $\boldsymbol{d}$ are the observed data and $\boldsymbol{\theta}$ corresponds to the set o parameters. Different $\boldsymbol{f}$ and/or $\boldsymbol{d}$ defines a different energy function, which leads to a different optimal solution $\boldsymbol{f}^*$. The parameters $\boldsymbol{\theta}$ are either known a priori or are estimated from the data.

As already stated above, an energy function in the MRF theory is formulated using established criteria. Probably the most popular one is the Bayes criteria and the maximal a priori (MAP) principle that leads to the MAP-MRF framework described in more details in the next section.

Chapter 2

Markov Random Fields Theory

As time goes by optimization principles become more and more popular for solving image analysis problems. Finding an optimal solution to a vision task is in most real cases impossible due to different uncertainties. Hence using optimizing principles is a natural way for solving this issues. The solution can be sought not only explicitly but implicitly as well. For proper solving of a computer vision task contextual constraints have to be used. The context has a critical influence for understanding a visual information in images. From this point of view the solution of a problem can be divided to two subproblems:

1. objective function definition that involved contextual constraints
2. optimizing this objective to find an optimal solution.

Markov random field forms a branch of probability theory and provides tools suitable for the characterization of contextual constraints. A certain MRF model favors its own class of patterns and assigns it a higher probability. This is a classical probability approach used in many classifiers. The added value of MRF is the possibility of involving the contextual constraints. Together with decision and estimation theory MRF can be interpreted as a systematic way for developing algorithms and thus avoiding the use of ad hoc heuristics. One such an approach is the maximum a posteriori (MAP) concept that leads to MAP-MRF framework, which is one of the most used approaches. The MAP-MRF framework was introduced in [Geman and Geman, 1984].

The issue of involving needed contextual constraints while defining a objective function is solvable by MRF theory, which is both convenient and consistent. For this purpose conditional MRF distributions are used, which describe mutual influences between each other. From these conditional distributions the joint distribution need to be derived in most applications. But in case of MRF's this task turns out to be very difficult. A breakthrough in the area of practical use of MRF's represents the Hammersley-Clifford theorem stating the equivalence between MRF's and Gibbs distribution. It was established in [Hammersley and Clifford, 1971] and further extended in [Besag, 1974]. Using this theorem the joint distribution of MRF's can be replaced by Gibbs distribution, which is of a simple form.

Uzavrit to tak, aby plynule navazovala dalsi sekce

2.1 Spatial Relation of Sites

To specify contextual constraints between sites one has to define their spatial relation first.

něco přidat, možná smoothnes prior

2.1.1 Neighborhood System

The spatial relation between sites from the set $\mathcal{S}$ can be specified by a neighborhood system:

$$\mathcal{N} = \{\mathcal{N}_p | \forall p \in \mathcal{S}\} \tag{2.1}$$

where $\mathcal{N}_p$ denotes the set of all sites that neighboring with given site p. One property of neighborhood system is that a site is not neighboring to itself, i.e. $p \notin \mathcal{N}_p$. Another important property is the mutuality, i.e. if a site i is neighbor of another site $q \neq p$, than q is also neighbor of the site p: $p \in \mathcal{N}_q \Longleftrightarrow q \in \mathcal{N}_p$.

A typical definition of a neighborhood system is via a distance measure.

$$\mathcal{N}_p = \{q \in \mathcal{S} \mid [\mathrm{dist}(s_q, s_p)]^2 \leq r^2,\ q \neq p\} \tag{2.2}$$

where r denotes the radius and *dist* is an arbitrary distance measure. Defining the system this way is suitable for both regular and irregular set of sites.

In most cases the *Euclidean distance* is used, especially for the irregular sites. The Euclidean distance between pixels (p, q) and (r, s) in the 2D image is as follows:

$$D_E\left((p,q),(r,s)\right) = \sqrt{(p-r)^2 + (q-s)^2}. \tag{2.3}$$

The main advantage of Euclidean distance is its intuitive meaning. On the other hand, due to the square root its calculation is costly. In the case where sites are placed on a regular lattice another distances can be used to overcome the problems time efficiency. The *Manhattan distance* (or city block or D_4 distance) is described as a minimal number of elementary steps in the digital grid which are needed to move from one point to another [Sonka et al., 2007]:

$$D_4\left((p,q),(r,s)\right) = |p-r| + |q-s|. \tag{2.4}$$

Another representative that is often used in discrete cases is the *chessboard distance* (or D_8). In comparison with the Manhattan distance the diagonal moves are allowed here:

$$D_8\left((p,q),(r,s)\right) = \max\{|p-r|,\ |q-s|\}. \tag{2.5}$$

The most used types of neighborhood system is the first (von Neumann) and second (Moore) order system. Its 2D version are shown in figure 2.2.

2.1.2 Cliques in a Graph

For further use the term clique in a graph needs to be defined. A graph can be represented by a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$ between those nodes, i.e. $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$. Next let the edges be derived based on a neighborhood system $\mathcal{N}$. Then a subset $c \subset S$ is called a clique if every pair of sites in

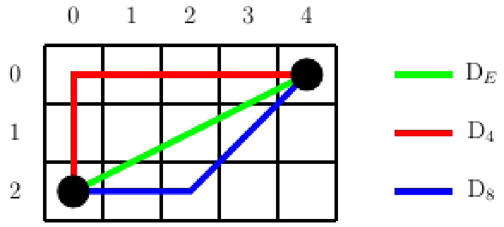


Figure 2.1: Different distance measures

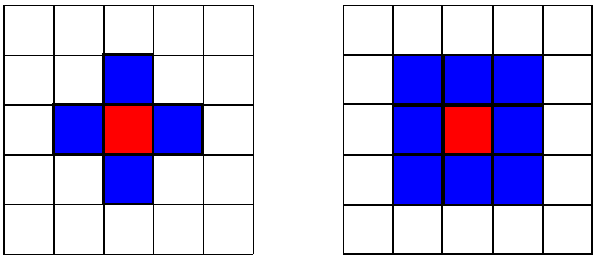

(a) (b)

Figure 2.2: Most common types of neighborhood systems are the first (a) and the second (b) order systems.

this subset are neighbors, i.e. $\forall \boldsymbol{s_p}, \boldsymbol{s_q} \in \boldsymbol{c}, \boldsymbol{p} \neq \boldsymbol{q} : \ \boldsymbol{s_p} \in \boldsymbol{\mathcal{N}_q}$. In other words for each pair of sites in a clique there exists an edge in $\boldsymbol{\mathcal{E}}$ between corresponding nodes: $\forall \boldsymbol{s_p}, \boldsymbol{s_q} \in \boldsymbol{c}, \boldsymbol{p} \neq \boldsymbol{q} : \ (\boldsymbol{s_p}, \boldsymbol{s_q}) \in \boldsymbol{\mathcal{E}}$. The cliques can be categorized w.r.t. the number of sites/nodes from which they consists. The cliques can be of the order of one - $\boldsymbol{c_1} = \{\boldsymbol{p}\}$, two - $\boldsymbol{c_2} = \{\boldsymbol{p}, \boldsymbol{q}\}$ and so on. Based on their order they're called singletons, doubletons etc. The set of all cliques of the first order is denoted $\boldsymbol{\mathcal{C}_1}$, of the second order $\boldsymbol{\mathcal{C}_2}$ and so on. The set of all cliques is then denoted $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{C}_1} \cup \boldsymbol{\mathcal{C}_2} \ldots \boldsymbol{\mathcal{C}_n}$ where $\boldsymbol{n}$ is the maximal clique order.

For the first order neighborhood system 2.2a there is the total number of three cliques coresponding to the singleton and the horizontal and vertical doubletons shown in 2.3a and 2.3b, respectively. In the case of the second order neighborhood system 2.2b there're also the diagonal doubletons 2.3c, the tripletons 2.3d and the quadrupletons 2.3e.

The total number of cliques grows rapidly with increasing order of the neighborhood system. To overcome computational expenses only the first or second order neighborhood system is used in most cases.

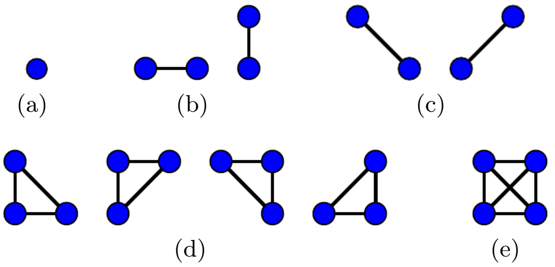


Figure 2.3: Cliques on a regular lattice: singleton (a), doubletons (b) and (c), tripletons (d) and a quadrupleton (e).

2.2 The MAP-MRF Framework

To describe the motivation for using a probability theory and MAP principle specifically, lets define a set of image features $\boldsymbol{d} = \{\overrightarrow{\boldsymbol{d}}_{\boldsymbol{s}} : \boldsymbol{s} \in \boldsymbol{\mathcal{S}}\}$ (a so called observation) and the a set of labels $\boldsymbol{f} = \{\boldsymbol{f_s} : \boldsymbol{s} \in \boldsymbol{\mathcal{S}}, \boldsymbol{f_s} \in \boldsymbol{\mathcal{L}}\}$ as earlier. For an image of size $\boldsymbol{N} \times \boldsymbol{M}$, there are the total number of $|\boldsymbol{\mathcal{L}}|^{\boldsymbol{NM}}$ possible labelings. That's an intractable set of possibilities even for a moderate size of images. To decide which of them is the right one the probabilistic approach can be used. This leads to a definition of a probability measure over the set of all possible labelings and choosing the most probable one. In other words one need to define a measure $\boldsymbol{P(f|d)}$ that corresponds to the probability of a lebeling $\boldsymbol{f}$ given observation $\boldsymbol{d}$. Then the goal is to find an optimal labeling $\hat{\boldsymbol{f}}$ that maximizes this probability $\boldsymbol{P(f|d)}$. This procedure is called the Maximum a Posterior (MAP) estimate:

$$\hat{\boldsymbol{f}}^{\mathrm{MAP}} = \arg\max_{\boldsymbol{f} \in \mathbb{F}} \boldsymbol{P(f|d)} \tag{2.6}$$

where $\mathbb{F}$ denotes a set of all possible labelings.

Sometimes this is called an inverse problem, where one tries to recover a large number of unknown or hidden variables - $\boldsymbol{f}$ in our case. This procedure is based on another set of variables called observation - $\boldsymbol{d}$ in our case. The observation corresponds to image features derived in advance. Both of the sets can be of the same nature, e.g. two images in restoration problems, or of completely different nature, e.g. an input image and a labeling in segmentation task. In some cases more than one observation need to be used for inference of the hidden variables, e.g. two input images in stereo vision.

For computing the posterior probability the Bayes rule is used:

$$\boldsymbol{P(f|d)} = \frac{\boldsymbol{p(d|f)P(f)}}{\boldsymbol{p(d)}}. \tag{2.7}$$

Here $\boldsymbol{P(f)}$ denotes the prior probability of labelings (often called simply prior) $\boldsymbol{f}$ and $\boldsymbol{p(d)}$ is the density function of $\boldsymbol{d}$. The last term $\boldsymbol{p(d|f)}$ is the conditional probability density function (p.d.f.) of the observation $\boldsymbol{d}$ for given labeling $\boldsymbol{f}$. This is called the likelihood function of $\boldsymbol{f}$ for fixed $\boldsymbol{d}$ also known as the evidence for the labeling $\boldsymbol{f}$ from the observation $\boldsymbol{d}$ and it describes how the labeling $\boldsymbol{f}$ fits given data $\boldsymbol{d}$. For given data $\boldsymbol{d}$ the density $\boldsymbol{p(d)}$ is constant thus simplifying the equation 2.7 to

$$\boldsymbol{P(f|d)} \propto \boldsymbol{p(d|f)P(f)}. \tag{2.8}$$

To solve this equation one need to define both of the terms $p(d|f)$ and $P(f)$ and to do that the MRF theory is used.

Validation of Modeling? MRFMIA str. 18

2.3 Markov Random Fields

To be consistent with previous definitions the MRF will be defined in terms of graph theory. Having this in mind the MRF can be defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \dots, N\}$ denotes the set of nodes. Each node is associated with a random variable F_p for $p = 1, 2, \dots, N$, each random variable can take a value $f_p \in \mathcal{L}$. To define the set of edges $\mathcal{E}$ the neighborhood system $\mathcal{N}$ must be known. The set of nodes to which the node p is adjacent defines its neighborhood $\mathcal{N}_p$, i.e. $q \in \mathcal{N}_p \iff (p, q) \in \mathcal{E}$. The set $\mathcal{N}_p$ is called the Markov blanket of node p.

A realization of the MRF f, i.e. an event where each node takes a value $f = \{f_1, f_2, \dots, f_m\}$, is called a configuration of that field. The probability that a random variable F_p takes the value f_p is denoted $P(F_p = f_p) = P(f_p)$ and the probability of a configuration f is denoted $P(F = f) = P(f)$. In the case of continuous set of possible labels $\mathcal{L}$ the probability density functions $p(F_p = f_p)$ and $p(F = f)$ are used instead.

If a random field (RF) is to be a Markov random field two following conditions must be satisfied:

1. positivity:
$$P(f) > 0, \ \forall f \in \mathbb{F} \tag{2.9}$$

2. Markovianity:
$$P(f_p|\{f_q\}_{j \in \mathcal{V} \setminus i}) = P(f_p|q \in \mathcal{N}_p) = P(f_p|f_{\mathcal{N}_p}) \tag{2.10}$$

The positivity guarantees that the joint probability $P(f)$ of any random field is uniquely determined by its local conditional probabilities as stated in [Besag, 1974]. The Markovianity describes a *knock-on effect* where explicit short-range linkages give rise to implied long-range correlations ([Blake et al., 2011]). This feature is the great attraction for using MRFs.

To specify a MRF two approaches can be used. The first one is based on conditional probabilities $P(f_i|f_{\mathcal{N}_i})$ and the second one is based on joint probability $P(f)$. The former approach is preferred still there were some problems using it. Fortunately, the already mentioned Hammersley-Clifford theorem defines the equivalence between MRFs and Gibbs distributions that provides mathematical tools for specifying the joint probability of a MRF.

2.4 Gibbs Random Fields

A set of random variables F forms a *Gibbs random field* (GRF) on the set of sites $\mathcal{S}$ w.r.t the neighobrhood system $\mathcal{N}$ if and only if its configurations obey a Gibbs distribution of the following form:

$$P(f) = \frac{1}{Z} \times e^{-\frac{1}{T}E(f)} \tag{2.11}$$

where T is a constant called temperature that controls the sharpness of the distribution, $E(f)$ denotes the energy function of a configuration f and Z is a normalizing coefficient often called the partition function. It ensures that that the distribution sums to 1 and takes the following form:

$$Z = \sum_{f \in \mathbb{F}} e^{-\frac{1}{T}E(f)} \tag{2.12}$$

The advantage of the Gibbs distribution is that the energy function can be easily expressed as a sum of *clique potentials* $V_c(f)$:

$$E(f) = \sum_{c \in \mathcal{C}} V_c(f) \tag{2.13}$$

where $\mathcal{C}$ is the set of all possible cliques in a graph.

The equation 2.13 can be rewritten in a form that couples cliques based on their order:

$$E(f) = \sum_{\{p\} \in \mathcal{C}_1} V_1(f_p) + \sum_{\{p,q\} \in \mathcal{C}_2} V_2(f_p, f_q) + \ldots + \sum_{\{p,q\ldots\} \in \mathcal{C}_n} V_n(f_p, f_q \ldots) \tag{2.14}$$

where n denotes the maximal clique order. If the first order neighborhood system is used only the cliques of order up to two are considered. This is an important special case that is very often used for solving vision problems. The equation 2.14 then simplifies to the following form:

$$E(f) = \sum_{\{p\} \in \mathcal{C}_1} V_1(f_p) + \sum_{\{p,q\} \in \mathcal{C}_2} V_2(f_p, f_q) = E_{\text{data}}(f) + E_{\text{smoothness}}(f) \tag{2.15}$$

The first term E_{data} of equation 2.15 is often called simply the data term and evaluates the fit of the labeling to the observation. The second term $E_{\text{smoothness}}$ of equation 2.15 is often called simply the smoothness term and encourages homogeneous regions. In literature the data term of site p is often denoted as $D_p(f_p)$ and the smoothness term between two sites p and q as $V_{p,q}(p,q)$. V_i with i been a numeric index denotes a clique potential of the clique of i-th order. Thus, the equation 2.15 can be rewritten as follows:

$$E(f) = E_{\text{data}}(f) + E_{\text{smoothness}}(f) = \sum_{p \in \mathcal{S}} D_p(f_p) + \sum_{\{p,q\} \in \mathcal{N},\, p,q \in \mathcal{S}} V_{p,q}(f_p, f_q) \tag{2.16}$$

This notation will be used further in this text.

One problem that must be solved to calculate a Gibbs distribution is evaluation of the partition function Z. As already stated, the set of all possible labelings $\mathbb{F}$ is intractable even for moderate number of random variables. Therefore one of approximation methods is often used to deal with this problem. priklad metody?

From the equation 2.11 is clear that more probable configurations are those with smaller energy, in other words, the lower the energy of a configuration the higher its probability. Therefore finding the most probable configuration of an GRF is another energy minimizing task.

2.5 Hammersley-Clifford Theorem

The main difference between a MRF and a GRF lies in their characterization. An MRF is characterized by the Markovianity, i.e. its local property, whereas an GRF is characterized by the Gibbs distribution, i.e. its global property. As already said, the motivation for using GRF is the possibility of defining the energy function in terms of clique potentials. Fortunately, the Hammersley-Clifford theorem stated in [Hammersley and Clifford, 1971] establishes the equivalence between MRFs and GRFs.

The mentioned theorem is as follows:

Theorem 1. *... given the neighbourhood structure of the model, for any set of sites within the lattice, their associated contribution of the Gibbs energy function should be non-zero, if and only if the sites form a clique. The contribution of each clique may be assigned arbitrarily.*

In other words, the set of random variables $\boldsymbol{F}$ is an MRF on the set of sites $\boldsymbol{\mathcal{S}}$ w.r.t. the neighborhood system $\boldsymbol{\mathcal{N}}$ if and only if $\boldsymbol{F}$ is a GRF on the same set of sites $\boldsymbol{\mathcal{S}}$ w.r.t. the same neighborhood system $\boldsymbol{\mathcal{N}}$. A proof of this theorem can be found for example in [Besag, 1974].

A MRF can have two important properties - homogenity and isotropy. A MRF is said to be *homogeneous* if the conditional probability $\boldsymbol{P(f_p|f_{\mathcal{N}_q})}$ is independent of the relative location of the node/site $\boldsymbol{p}$ in $\boldsymbol{\mathcal{S}}$. The homogenity is assumed in most MRF vision models making the computations more convenient. Moreover, to define an RF that is not homogeneous is much more difficult. A MRF is said to be *isotropic* if the clique potentials $\boldsymbol{V_c}$ are independent of the orientation of the clique $\boldsymbol{c}$. This property is met if undirected graphs are used for describing a vision problem, e.g. image segmentation in most cases.

2.6 Models of Markov Random Fields

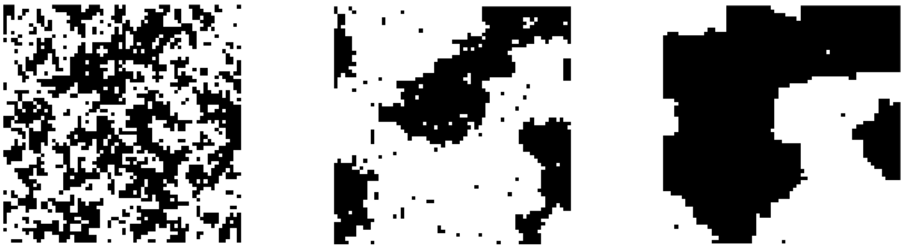
The Hammersley-Clifford theorem allows a very general basis for the specification of MRF joint distribution function that is based on clique potentials. One particular, important and very often used group of MRF models are called *auto-models* by [Besag, 1974]. The form of auto-models can be derived from the dual energy equation 2.15 when $\boldsymbol{D_p = f_p G_p(f_p)}$ and $\boldsymbol{V_{p,q} = \beta_{p,p} f_p f_q}$. The $\boldsymbol{G_p(f_p)}$ are unspecified functions and $\boldsymbol{\beta_{p,q}}$ denotes pre-defined model parameters which reflects the pair-site interactions. The energy function of auto-models is then characterized as follows:

$$\boldsymbol{E(x) = \sum_{\{p\}\in\mathcal{C}_1} f_p G_p(f_p) + \sum_{\{p,q\}\in\mathcal{C}_2} \beta_{p,q} f_p f_q} \tag{2.17}$$

The auto-models can be further classified w.r.t. the specifications of individual terms in energy function 2.17. Probably the most used are the *Ising* and the *Potts* model that are described in more details in following text.

2.6.1 Ising Model

One of the basic feature of our world is the smoothness. The world preserves smoothness not only in time but also in space assuming that physical properties in close neighborhood or in a short interval of time don't change abruptly. Transferring this assumption to the domain of image processing can be expressed as an assumption that site labels in close neighborhood should be the same. In other words, changing labels should be penalized in some way. And exactly this assumptions is encoded in Ising and Potts models.



(a) $\beta = 0.7$ (b) $\beta = 1.1$ (c) $\beta = 2$

Figure 2.4: Samples from Gibbs distribution of an Ising model for different values of the parameter β. Images taken from [Perez, 1998].

The Ising model is a binary model which means that each site can be assigned with one out of two labels, i.e. $\mathcal{L} = \{0, 1\}$ or sometimes $\mathcal{L} = \{-1, +1\}$. Another assumption is that only the first order neighborhood system 2.2a is used. The energy function can be then written as follows:

$$E(f) = \sum_{\{p\} \in \mathcal{C}_1} \alpha_p f_p + \sum_{\{p,q\} \in \mathcal{C}_2} \beta_{p,q} f_p f_q \tag{2.18}$$

where $\beta_{p,q}$ as called the interaction coefficients. In the case where homogenity is preserved the coefficients α_p and $\beta_{p,q}$ are independent on the position of site p, i.e. $\alpha_p = \alpha$ and $\beta_{p,q} = \beta$.

The Ising model allows to define the interaction potentials in the form of label discontinuities. The parameter β can be interpreted as a penalty for discontinuity and the interaction potentials can be defined as follows:

$$V_{p,q}(f_p, f_q) = \beta |f_p - f_q| = \beta \delta(f_p, f_q) \tag{2.19}$$

where $\delta(f_p, f_q)$ is the Kronecker delta:

$$\delta(f_p, f_q) = \begin{cases} 1 & \text{if } f_p \neq f_q \\ 0 & \text{if } f_p = f_q \end{cases} \tag{2.20}$$

This model prefers configurations with large-scale agreement between the labels of adjacent sites, i.e. it prefers few large areas in the image over many smaller ones. The interaction term can be interpreted also as a penalty for boundary - the total penalty of the interaction term is the total perimeter of boundaries separating individual regions times β. Having this in mind the interaction potential can be rewritten in the following form:

$$V_{p,q}(f_p, f_q) = \beta \cdot (\text{boundary length}) \tag{2.21}$$

thus making the longer boundaries less probable. The boundary length is given by Manhattan (D_4) distance.

The influence of the parameter β is shown in figure 2.4 where few samples from the Gibbs distribution that corresponds to the Ising model are shown.

2.6.2 Potts Model

The Ising model described in the previous section is usable in the case where there're only two possible labels, e.g. binary image segmentation. However, many vision task need to solve problems with more than two discrete labels. For this purposes the Ising model can be generalized to the Potts model, where there are $M > 2$ discrete labels in the label set $\mathcal{L} = \{1, 2, \ldots, M\}$. The interaction potentials can be defined as follows:

$$V_{p,q}(f_p, f_q) = \begin{cases} \beta_c & \text{if } f_p \neq f_q \\ -\beta_c & \text{if } f_p = f_q \end{cases} \tag{2.22}$$

The parameter β_c denotes the cost of an edge connecting nodes with different labels and depends on the type c of the clique. In the case of isotropic model the clique potentials are independent of the orientation, which means that $\beta_c = \beta, \ \forall c \in \mathcal{C}_2$. As β increases, regions become more homogeneous.

2.6.3 Other Models

Gaussian MRF, mozna FRAME

2.7 Optimization techniques

Currently, lot of methods for finding local minimum are available but to find a global minimum of a function is still a big challenge. Methods for global minimization can be based on finding the exact global minimum or an approximation. The use of the former methods usually yields to an exhaustive search and inefficient algorithms. Approximative methods are efficient in both the computational load and the size of search space but the result is still only an approximation that could be far away from the exact solution.

2.7.1 Simulated Annealing

The problem of global minimization is to overcome cases where the methods stuck in a local minimum. To solve this problem an approach that allows accepting of possible worse configurations, i.e. configurations with higher energy, can be used. Probably the best known representative of this kind of methods is simulated annealing independently introduces by [Černý, 1985] and [Kirkpatrick et al., 1983].

Simulated annealing (SA) is based on physical annealing process in which a physical substance is melted and then cooled down in search of a low energy configuration. It's an iterative procedure where in each step neighbours of the current state are examined. In classical methods the algorithm moves only to a state with lower configuration which makes it prone to getting stuck in local minima. The SA algorithm allows moving to states with higher energy with a probability proportional to current temperature T. As the temperature T goes down the probability of accepting such a worse state is smaller.

[Metropolis et al., 1953] simulated the annealing process using the Monte Carlo technique. Assume that a configuration $f \in \mathbb{F}$ can change to its neighbouring configuration $f' \in \mathbb{F}$. The probability of this change at the temperature T is denoted as $P(f \rightarrow f', T)$. To calculate this probability one can use the Metropolis criterion where the probability of acceptance of the new state is given by following equation:

$$P(f \rightarrow f') = \begin{cases} 1, & \text{for } E(f) < E(f') \\ e^{-(E(f)-E(f'))/T} & \text{for } E(f) \geq E(f') \end{cases} \tag{2.23}$$

In case where the new configuration has higher energy, the probability is inversely proportional to the temperature and difference between the energy of the two configurations. The higher the temperature and/or the difference the lower the energy. This means that accepting a less profitable configuration is more probable in the early stages of computation where the temperature is still high.

Initially, when the temperature is high, the probability distribution $P(f, T)$ is uniform, i.e. the probability doesn't reflect the goodness of the configurations. As the temperature approaches to zero, the probability distribution $P(f, T)$ is concentrated around the peaks of $P(f)$, i.e. the fittest configurations are preferred.

The procedure in which the temperature is decremented - so called *cooling schedule*, plays an important role in the speed of the convergence. In [Geman and Geman, 1984] the cooling schedule takes following form:

$$T^{\{t\}} = \frac{C}{\ln(1+t)} \tag{2.24}$$

where the parameter C is a constant. Another cooling schedule can be found for example in [Kirkpatrick et al., 1983].

2.7.2 Genetic Algorithms

Genetic algorithms are methods based on heuristic approaches introduced for example in [Holland, 1975] and [Goldberg, 1989]. They're inspired by the principle of natural evolution in the biological world where the rule of the survival of the fittest takes place. The natural selection is guided by the quality of individuals that ensures that better individuals will survive to the next generation.

Genetic algorithm is parallel in its nature - it stores several solutions that are called *individuals*, the set of all solutions in a given time is called a *population*. One individual is given by its *chromozome*, i.e. a set of *genes*. A value of a gene is often called an *allele*. In labeling process one gene can represent a certain site and its allele value correspond to one of the class labels. The fitness of the individual then corresponds to the energy of given labeling, which can be calculated in the sense of MRF. When most (typically over 90%) of individuals share the same value of one particular gene then we say that this gene have converged. When all the genes have converged then we say that the population have converged.

Recombination

To produce new individuals, i.e. possible solutions, a process called recombination is applied. The recombination could be seen as an unary or a binary operator. The latter one is called *crossover*. The input of the crossover are two individuals that are combined in a certain way to create two offsprings. The parents are chosen w.r.t. their fitness, i.e. the better the individual the higher the probability that it will be chosen, for more information see [Bäck, 1996]. This simulates the natural selection.

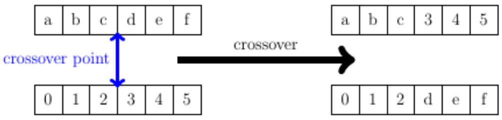


Figure 2.5: Illustration of the crossover operator based on a point.

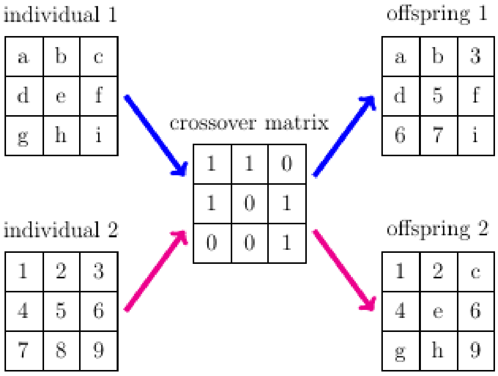


Figure 2.6: Illustration of the crossover operator based on a matrix.

With this in mind, the offsprings bear the information of a possible good solution. Furthermore, the combination of two such a individuals can lead to a new even better one.

The unary recombination process is the *mutation*. Its role is to slightly change one or more genes of an individual to produce different one. Thus the mutation has similar role as the temperature parameter in simulated annealing, i.e. to prevent of getting stuck in a local minimum.

A scheme of a standard genetic algorithm is shown in Algorithm 1.

The recombination processes are not always invoked. The probability of the crossover is somewhere between 0.6 and 1 [Li, 2009] and the probability of the mutation is typically 0.0001. To define where the crossover takes place several approaches exists. One such an approach is to randomly derive a crossover point. To this point all genes takes value from the first parent and after this point from the second parent as shown in fig. 2.5. In the case of images an approach based on random mask can be used. In this case a binary mask is derived that specifies from what parent a gene takes value as shown in fig. 2.6. The disadvantage of this approach is that it could break a homogeneous areas with no obvious progress. The mutation operator is shown in fig. 2.7.

To prevent situations where a good solution can be lost due to the recombination one can use a process called *elitism*. Here, a portion (typically 10% of the total number of individuals) are saved

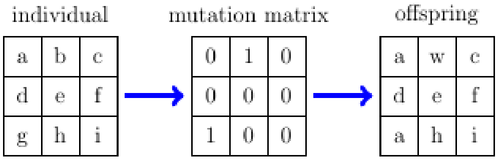


Figure 2.7: Illustration of the mutation operator.

```
begin
    generate initial population $\boldsymbol{P} = \{\boldsymbol{f}^1, \ldots, \boldsymbol{f}^n\}$
    repeat
        compute $\boldsymbol{E}(\boldsymbol{f})\ \forall \boldsymbol{f} \in \boldsymbol{P}$
        // creating new population
        $\boldsymbol{P}_{\text{new}} = \{\}$
        repeat
            $\boldsymbol{f}^a$, $\boldsymbol{f}^b$ = individuals_selection($\boldsymbol{P}$)
            // crossover
            if rand[0, 1] < $\tau_{crossover}$ then
                $\boldsymbol{f}^1$, $\boldsymbol{f}^2$ = crossover($\boldsymbol{f}^a$, $\boldsymbol{f}^b$)
            else
                $\boldsymbol{f}^1 = \boldsymbol{f}^a$
                $\boldsymbol{f}^2 = \boldsymbol{f}^b$
            end
            // mutation
            for $\boldsymbol{f}^i \in \{\boldsymbol{f}^1, \boldsymbol{f}^2\}$ do
                if rand[0, 1] < $\tau_{mutation}$ then
                    $\boldsymbol{f}^i$ = mutation($\boldsymbol{f}^i$)
                end
            end
            $\boldsymbol{P}_{\text{new}} = \boldsymbol{P}_{\text{new}} \cup \{\boldsymbol{f}^1, \boldsymbol{f}^2\}$
        until new population created
        $\boldsymbol{P} = \boldsymbol{P}_{\text{new}}$
    until termination condition satisfied
    return($\arg\min_{\boldsymbol{f} \in \boldsymbol{P}} \boldsymbol{E}(\boldsymbol{f})$)
end
```

Algorithm 1: A standard genetic algorithm

and then replaces the same portion of the worst individuals. One thing that should be highlighted is that having few bad individuals is not necessarily a disadvantage. These individuals bear a different information that could produce, in combination with a good individual, a solution that can outperform all others. Moreover, bad individuals are also helpful for getting of a local minimum and for searching a different part of the solution space.

Locust Swarm

Though the main principle remains the same, many different genetic algorithms exists that differs for example in the chromozome representation or in the way the recombination is done. A well known example of such a different approach are locust-based methods that were firstly introduced in [Kennedy and Eberhart, 1995] and further used for example in [Park and Song, 1998] and [Amelio and Pizzuti, 2012].

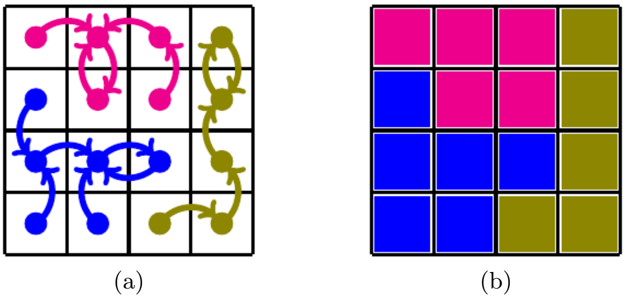

(a) (b)

Figure 2.8: In lucust swarm methods allele values represents links to neighbors (a). Given individual represents segmentation shown in (b).

IN this approach, the allele values don't represent class labels. Instead they represent another genes thus making links between each other. The more similar two neighboring genes are the higher the probability that a link will be created between them. One gene can have more than one allele value and the recombination is done by switching an combining those alleles. The final image segments are represented by individual connected components that are created as the algorithm evolves.

The inspiration of this algorithm comes from the behavior of a swarm. The swarm is formed by many individuals of the same species, i.e. bees or grasshoppers, where each individual follows only a few of his neighbors. The swarm behavior is then given by combination of the behavior of several smaller groups.

Comb Algorithm

A combination of local search methods and genetic algorithms yields to a hybrid approach often called a memetic algorithm, for more information see [Moscato, 1989] or [Li, 2009]. The main idea of this method is to preserve any good information found so far. Such a good information is represented by sharing the same value of a particular gene through different individuals. This means that if the allele value of a gene $\boldsymbol{i}$ in an individual $\boldsymbol{a}$ is the same as the allele value of the same gene $\boldsymbol{i}$ in a different individual $\boldsymbol{b}$, i.e. $\boldsymbol{f_i^a} = \boldsymbol{f_i^b}$, then it is probable that this information is a good one and should be saved in both offsprings $\boldsymbol{f_i^1}$, $\boldsymbol{f_i^2}$. Else an allele value to be inherited in an offspring is randomly chosen from the values of the parents.

The evolutionary part of this algorithm lays in the combination of two different local minima that were already found. This crossover like procedure, sometimes called *comb initialization*, produces two new starting points for the subsequent local search method that yields to possible new solutions. If the energy of a new solution is lower than any of the solutions found so far, than it replaces the worst one.

The pseudocode of the comb initialization is shown in Algorithm 2 and the pseudocode of the whole Comb algorithm is shown in Algorithm 3.

```
begin
    for $i \in \mathcal{S}$ do
        // uniform crossover
        if $f_i^a == f_i^b$ then
            $f_i^1 = f_i^2 = f_i^a$
        else if $\mathrm{rand}[0,1] < 0.5$ then
            $f_i^1 = f_i^a$
            $f_i^2 = f_i^b$
        else
            $f_i^1 = f_i^b$
            $f_i^2 = f_i^a$
        end
        // mutation
        if $\mathrm{rand}[0,1] < \tau_{mutation}$ then
            $f_i^1 = \mathrm{rand}(\mathcal{L})$
        end
        if $\mathrm{rand}[0,1] < \tau$ then
            $f_i^2 = \mathrm{rand}(\mathcal{L})$
        end
    end
end
```

Algorithm 2: Comb initialization

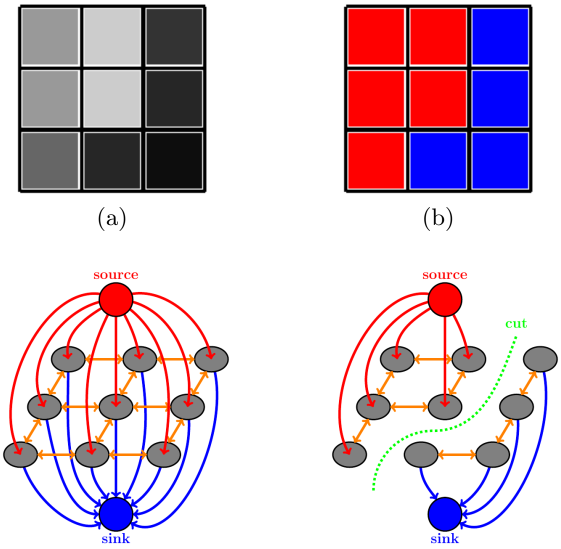
```
begin
    initialize $F = \{f^1, \ldots, f^N\}$
    repeat
        $f^a$, $f^b$ = random_selection($F$)
        $f^1$, $f^2$ = Comb_initialization($f^a$, $f^b$)
        $f^{1*}$ = steepest_descent($f^1$)
        $f^{2*}$ = steepest_descent($f^2$)
        $F$ = update($F$, $f^{1*}$, $f^{2*}$)
    until termination condition satisfied
    return($\arg\min_{f \in F} E(f)$)
end
```

Algorithm 3: Comb algorithm

2.7.3 Min Cut / Max Flow

The main goal of this method is to find a minimum cut in a graph that represents an image. Finding such a minimum cut is not a trivial task. Fortunately, the minimum cut problem is equivalent to the maximum flow problem as stated in the Ford-Fulkerson theorem that was introduced in [Ford and Fulkerson, 1956]. Finding a maximum flow belongs to the classical optimizing problems.

The first step is to obtain a weighted graph $\mathbf{G}(\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}})$ from the image, where $\boldsymbol{\mathcal{V}}$ represents set of nodes and $\boldsymbol{\mathcal{E}}$ represents set of edges. Each node corresponds to an image site, i.e. $\boldsymbol{\mathcal{V}} = \{\boldsymbol{p} | \boldsymbol{p} \in \boldsymbol{\mathcal{S}}\}$ where $\boldsymbol{\mathcal{S}}$ is traditionally a set of all image sites. Each edge between these nodes corresponds to a neighboring relationship between these nodes, i.e. $\boldsymbol{\mathcal{E}} = \{\boldsymbol{e_{p,q}} | \boldsymbol{p}, \boldsymbol{q} \in \boldsymbol{\mathcal{V}}, \boldsymbol{p} \in \boldsymbol{\mathcal{N}_q}\}$. These edges are often called *n-links* (neighborhood links). The graph contains two additional nodes, the source $\boldsymbol{s}$ and the sink $\boldsymbol{t}$, respectively. The former one represents an object while the latter one the background. These special nodes are called terminals. Each node from the set $\boldsymbol{\mathcal{V}}$ is connected to the terminals with a link called *t-link* (terminal link). An illustration of a graph constructed this way is shown in figure 2.9(c).



(a) (b)

(c) (d)

Figure 2.9: Illustration of image segmentation by graph cut. The input image 2.9(a) and its corresponding graph 2.9(c) is segmented by a cut shown in 2.9(d) which yields the final segmentation 2.9(b).

We can imagine the graph like a pipeline system (in the basic version of Graph cut method). Each pipe has its own capacity given by the weight of the corresponding edge. Segmentation consists in gradual filling this pipeline with a medium through the source and the leakage of this medium through the sink. Some of the pipes become saturated in this process, which means that through this pipe flows the maximum value of the medium given by its capacity. After a certain amount of the medium was sent to the system, there can occur a state in which there isn't a path from the source to the sink without any saturated edge. In this case the segmentation ends and the object is separated from the background by the set of the saturated edges. This set of saturated edges represents a cut of the graph.

In other words, a cut on a graph is a set of edges $\mathcal{C} \subset \mathcal{E}$ such that the terminal nodes are separated in the induced graph $\mathcal{G}(\mathcal{C}) = \{\mathcal{V}, \mathcal{E} - \mathcal{C}\}$. Moreover, there isn't a subset $\mathcal{C}'$ of $\mathcal{C}$ that will separate the terminals in $\mathcal{G}(\mathcal{C})$. The cost of a cut is denoted as $|\mathcal{C}|$ and corresponds to the sum of all edges in this cut. This value is equal to the value of maximal flow that can be sent to the graph as said in the already mentioned Ford-Fulkerson theorem ([Ford and Fulkerson, 1956]). The figure 2.9 illustrates the process of image segmentation, where the input image 2.9(a) and its corresponding graph 2.9(c) is segmented by a cut shown in 2.9(d) which yields the final segmentation 2.9(b).

The energy that is minimized by this process takes the typical form given by equation 2.15. One common example how to define the data term (see equation 2.25) and the smoothness term (see equations 2.26 and 2.27) that yields neither to the Ising nor the Potts model is formulated in [?]:

$$V_{p,q}(f_p, f_q) = \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\frac{1}{||p,q||}\right) \tag{2.25}$$

$$D_p(p|obj) = -\ln\left(P(I_p|Obj)\right) \tag{2.26}$$

$$D_p(p|bgd) = -\ln\left(P(I_p|Bgd)\right) \tag{2.27}$$

where p, q are two different image sites that are $||p,q||$ far apart and have intensities I_p and I_p, σ can be interpreted as expected intensity variation within the object and/or background. The probabilities $P(I_p|Obj)$ and $P(I_p|Bgd)$ describe how the intensity value I_p fits to object or background, respectively. Following these equations one can assign weights to individual edges as shown in following table:

edge	weight	for
(p,q)	$V_{p,q}(f_p, f_q)$	$(p,q) \in \mathcal{N}$
(s,p)	$\lambda D_p(p\|bgd)$	$p \in \mathcal{S}, p \notin (O \cup B)$
	K	$p \in O$
	0	$p \in B$
(p,t)	$\lambda D_p(p\|obj)$	$p \in \mathcal{S}, p \notin (O \cup B)$
	0	$p \in O$
	K	$p \in B$

Table 2.1: Edge weights for graph cut image segmentation. Constant K is big enough so that the given edge gets never saturated.

There're two basic algorithms for finding the maximum flow in a graph. The first one is so called *push-relabel* algorithm introduced by [Goldberg and Tarjan, 1988], the second one is based on *augmenting paths* and was firstly presented in [Ford and Fulkerson, 1956] and [Ford and Fulkerson, 1962]. A relatively new approach which outperforms the former ones is formulated in [Boykov and Kolmogorov, 2001]. This algorithm is based on two competitive trees that "grow" in image domain and try to occupy image sites.

2.7.4 Graph Cut With Large Moves

In paper [Boykov et al., 2001] the problems of finding a global optimum are discussed. A labeling $\boldsymbol{f}$ is said to be a local minimum of the energy $\boldsymbol{E}$ if

$$\boldsymbol{E}(\boldsymbol{f}) \leq \boldsymbol{E}(\boldsymbol{f}') \text{ for any } \boldsymbol{f}' \text{ "near to" } \boldsymbol{f}. \tag{2.28}$$

The nearness of two labelings is defined in the sense of a single optimization move. Many optimization algorithms can change label only of one site at a time - these moves are called *standard moves* in [Boykov et al., 2001]. This often yields to poor results as shown in figure 2.10. In contrary, algorithms described in this section use large moves that allows changing labels of arbitrarily large set of image sites simultaneously. Thanks to that the algorithms are very efficient in approximating the energy minimization.

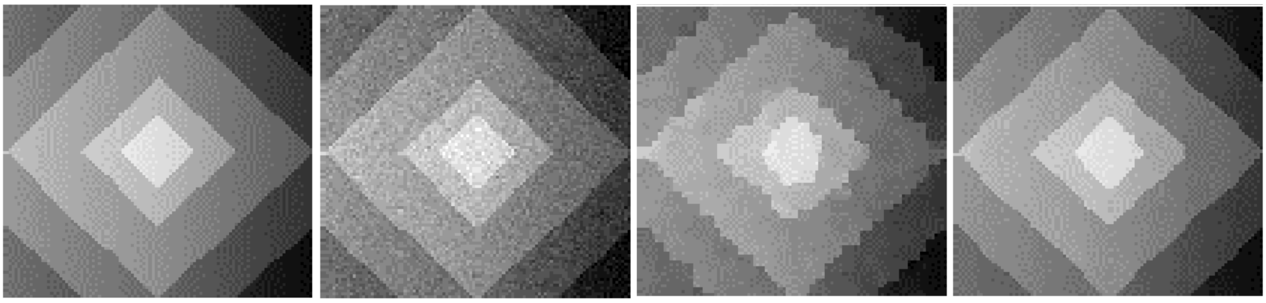

(a) (b) (c) (d)

Figure 2.10: Comparison of image restoration. The input image is shown in (a) and its noisy observation in (b). Using standard moves yields to low quality solution (c). The solution (d) is obtained using expansion moves. Images were taken from [Boykov et al., 2001].

Expansion Moves

The alpha expansion algorithm is based on successive labeling all $\boldsymbol{\alpha}$ and non-$\boldsymbol{\alpha}$ sites with graph cuts. In each iteration an optimal expansion move is determined that expands the $\boldsymbol{\alpha}$ label in an optimal way. The algorithm then continues with another label. The pseudocode of $\boldsymbol{\alpha}$ expansion algorithm is shown in alg. 4.

In each iteration a graph $\boldsymbol{\mathcal{G}_\alpha}$ is constructed that reflects current labeling. The set of vertices contains not only image sites $\boldsymbol{p} \in \boldsymbol{\mathcal{S}}$ but two terminals $\boldsymbol{\alpha}$ and $\bar{\boldsymbol{\alpha}}$ as well. In addition, an *auxiliary* node is created between each pair of neighboring sites that have different label. Each site is connected to the both terminals with links $\boldsymbol{t_p^\alpha}$ and $\boldsymbol{t_p^{\bar{\alpha}}}$. Between each pair of neighboring sites $\{\boldsymbol{p}, \boldsymbol{q}\} \in \boldsymbol{\mathcal{N}}$ with different labels $\boldsymbol{f_p} \neq \boldsymbol{f_q}$ a triplet of edges $\boldsymbol{\mathcal{E}_{\{p,q\}}} = \{\boldsymbol{e_{\{p,a\}}}, \boldsymbol{e_{\{a,q\}}}, \boldsymbol{t_a^{\bar{\alpha}}}\}$ is created, where $\boldsymbol{a}$ is the

```
Start with an arbitrary labeling $f$
begin
    repeat
        for each label $\alpha \in \mathcal{L}$ do
            $\hat{f} = \arg\min E(f')$ among $f'$ within one $\alpha$ expansion move of $f$
            if $E(f') < E(f)$ then
                $f = f'$
                improved = True
            else
                improved = False
            end
        end
    until not improved
end
```

Algorithm 4: Alpha expansion

corresponding auxiliary node. An example of a graph constructed this way is shown in fig. 2.11 and weights of individual edges are shown in table 2.2.

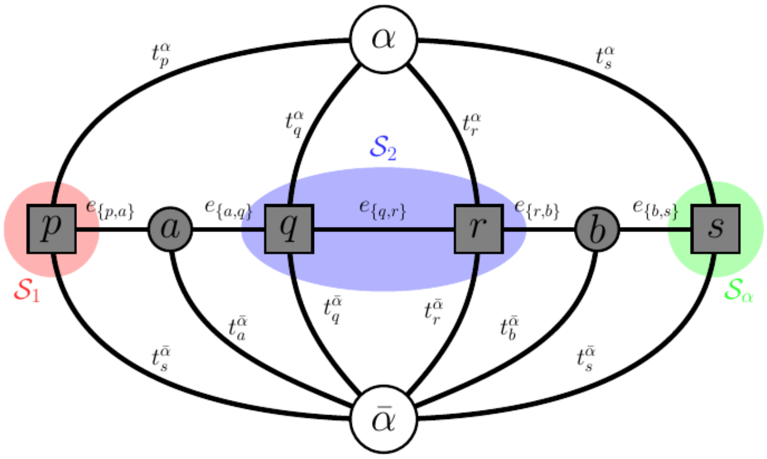


Figure 2.11: An example of $\mathcal{G}_\alpha$ with sites p, q, r and s, terminals α and $\bar{\alpha}$ and two auxiliary nodes a and b. Three different partitions are shown in this graph: $\mathcal{S}_1 = \{p\}$, $\mathcal{S}_2 = \{q, r\}$ and $\mathcal{S}_\alpha = \{s\}$.

To use this algorithm the interaction term $V(\alpha, \beta)$ have to be a metric. $V(\alpha, \beta)$ is called a metric if it satisfies the following three conditions:

$$V(\alpha,\beta) = 0 \quad \Leftrightarrow \quad \alpha = \beta \tag{2.29}$$

$$V(\alpha,\beta) \quad = \quad V(\beta,\alpha) \tag{2.30}$$

$$V(\alpha,\beta) \quad \leq \quad V(\alpha,\gamma) + V_2(\gamma,\beta) \tag{2.31}$$

for any labels $\alpha, \beta, \gamma \in \mathcal{L}$. The use of expansion moves in case where the interaction term is a metric brings a very strong optimality property. This property says that a local minimum found by

edge	weight	for
$t_p^{\bar{\alpha}}$	∞	$p \in \mathcal{S}_\alpha$
$t_p^{\bar{\alpha}}$	$D_p(f_p)$	$p \notin \mathcal{S}_\alpha$
t_p^{α}	$D_p(\alpha)$	$p \in \mathcal{S}$
$e_{\{p,a\}}$ $e_{\{a,q\}}$ $t_a^{\bar{\alpha}}$	$V_{p,a}(f_p, \alpha)$ $V_{a,q}(\alpha, f_q)$ $V_{p,q}(f_p, f_q)$	$\{p,q\} \in \mathcal{N},\ f_p \neq f_q$
$e_{\{p,q\}}$	$V_{p,q}(f_p, \alpha)$	$\{p,q\} \in \mathcal{N},\ f_p = f_q$

Table 2.2: Edge weights for a $\mathcal{G}_\alpha$ used in α expansion algorithm.

the expansion moves is within a known factor c of the global minimum - described in [Boykov et al., 2001] as following theorem:

Theorem 2. *Let* $\hat{f}$ *be a local minimum when the expansion moves are allowed and* f^* *be the globally optimal solution. Then* $E(\hat{f}) \leq 2cE(f^*)$.

The factor c is defined in the next equation:

$$c = \frac{\max_{\alpha \neq \beta \in \mathcal{L}} V_2(\alpha, \beta)}{\min_{\alpha \neq \beta \in \mathcal{L}} V_2(\alpha, \beta)} \tag{2.32}$$

If the interaction term is dependent on a neighboring pairs (p, q), then the previous equation is changed to following:

$$c = \max_{\{p,q\} \in \mathcal{N}} \left(\frac{\max_{\alpha \neq \beta \in \mathcal{L}} V_2(\alpha, \beta)}{\min_{\alpha \neq \beta \in \mathcal{L}} V_2(\alpha, \beta)} \right) \tag{2.33}$$

In a special case when the interaction term is given by the Potts model (already described in the section 2.6.2). In this case, discontinuities between any pair of labels are penalized equally, thus the factor c is given by the equation 2.32. Moreover, as shown in [Boykov et al., 2001] the factor $c = 1$ and thus the local minimun is within a factor of two of the global minimum.

Swap Moves

The alpha-beta swap algorithm is based on successive labeling all α and β sites with graph cuts. In each iteration an optimal swap move is determined that change labels between segments α and β. This is done successively for each pair of labels $\{\alpha, \beta\} \in \mathcal{L}$. The pseudocode of α-β swap algorithm is shown in alg. 5.

Similarly as in the expansion algorithm, a graph $\mathcal{G}_{\alpha\beta} = \{\mathcal{V}_{\alpha\beta}, \mathcal{E}_{\alpha\beta}\}$ is constructed in each iteration with its dynamically determined by the current labeling. Let the sites with label α and β be denoted $\mathcal{S}_\alpha$ and $\mathcal{S}_\beta$, respectively. Next let the set of sites with label α or β be denoted

```
Start with an arbitrary labeling $f$
begin
    repeat
        for each pair $\{\alpha, \beta\}$ : $\alpha, \beta \in \mathcal{L}$ do
            $\hat{f} = \arg\min E(f')$ among $f'$ within one $\alpha$-$\beta$ swap move of $f$
            if $E(f') < E(f)$ then
                $f = f'$
                improved = True
            else
                improved = False
            end
        end
    until not improved
end
```

Algorithm 5: Alpha-beta swap

$\mathcal{S}_{\alpha\beta} = \mathcal{S}_\alpha \cup \mathcal{S}_\beta$. Then the set of vertices contains two terminals α and β and sites with label α or β. Each vertex is connected to the terminals by edges t_p^α and t_p^β, respectively. Between each pair of neighboring sites $\{p, q\} \in \mathcal{N}$ an edge $e_{\{p,q\}}$ is created. An example of a graph constructed this way is shown in fig. 2.12 and weights of individual edges are shown in table 2.3.

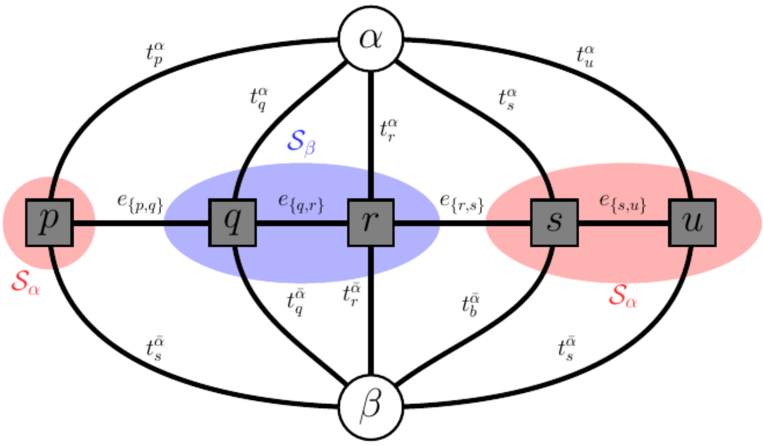
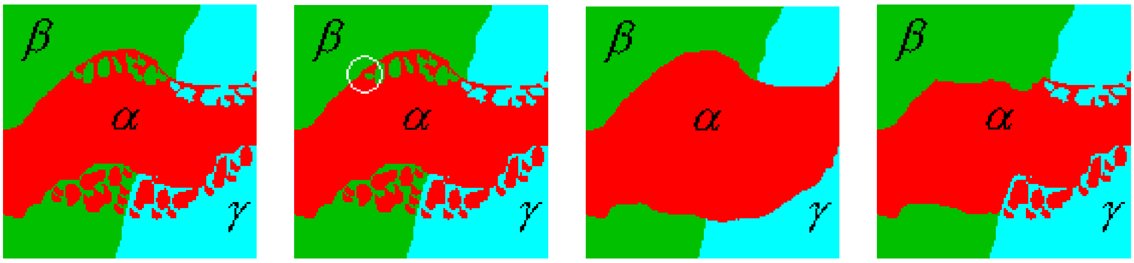


Figure 2.12: An example of $\mathcal{G}_{\alpha\beta}$ with sites p, q, r, s and u and terminals α and β. Two label segments are shown in this graph: $\mathcal{S}_\alpha = \{p, s, u\}$ and $\mathcal{S}_\beta = \{q, r\}$.

While in the α expansion algorithm have a guaranteed optimality properties, a solution found by α-β swap algorithm can be arbitrarily far from the global minimum. On the other hand it can be used for optimization of wider class of energies because the interaction term could be "only" a semimetric. An interaction term is said to be a semimetric if only the equations 2.29 and 2.30 are satisfied, i.e. the triangular inequality doesn't had to be hold.

edge	weight	for
t_p^{α}	$D_p(\alpha) + \sum_{q \in \mathcal{N}_p,\, q \notin \mathcal{S}_{\alpha\beta}} V_{p,q}(\alpha, f_q)$	$p \in \mathcal{S}_{\alpha\beta}$
t_p^{β}	$D_p(\beta) + \sum_{q \in \mathcal{N}_p,\, q \notin \mathcal{S}_{\alpha\beta}} V_{p,q}(\beta, f_q)$	$p \in \mathcal{S}_{\alpha\beta}$
$e_{\{p,q\}}$	$V_{p,q}(\alpha, \beta)$	$\{p, q\} \in \mathcal{N},\; p, q \in \mathcal{S}_{\alpha\beta}$

Table 2.3: Edge weights for a $\mathcal{G}_{\alpha\beta}$ used in α-β swap algorithm.



(a) (b) (c) (d)

Figure 2.13: Comparison of standard and large moves from given labeling (a). The α expansion (c) and the α-β swap (d) change many site labels simultaneously where one standard move of simulated annealing (b) changes label of a single site (in the circled area). Images were taken from [Boykov et al., 2001].

In the paper [Boykov et al., 2001] a possibility of achieving a solution within a known factor from the global optimum even when the interaction term is a semimetric is discussed. The approach suggests to approximate the semimetric and use α expansion algorithm to get a local minimum. This minimum is then used as a starting point for successive application of α-β swap algorithm.

Bibliography

[Amelio and Pizzuti, 2012] Amelio, A. and Pizzuti, C. (2012). An evolutionary and graph-based method for image segmentation. In Coello, C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., and Pavone, M., editors, *Parallel Problem Solving from Nature - PPSN XII*, volume 7491 of *Lecture Notes in Computer Science*, pages 143–152. Springer Berlin Heidelberg.

[Bäck, 1996] Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* Oxford University Press, Oxford, UK.

[Besag, 1974] Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236.

[Blake et al., 2011] Blake, A., Kohli, P., and Rother, C. (2011). *Markov Random Fields for Vision and Image Processing.* MIT Press.

[Boykov and Kolmogorov, 2001] Boykov, Y. and Kolmogorov, V. (2001). *An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Computer Vision.*

[Boykov et al., 2001] Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239.

[Chow, 1962] Chow, C. K. (1962). A recognition method using neighbor dependence. *Electronic Computers, IRE Transactions on*, EC-11(5):683–690.

[Ford and Fulkerson, 1956] Ford, L. R. and Fulkerson, D. R. (1956). Maximal flow through a network. *Canadian Journal of Mathematics*, 8(3):399–404.

[Ford and Fulkerson, 1962] Ford, L. R. and Fulkerson, D. R. (1962). *Flows in Networks.* Princeton University Press.

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

[Goldberg and Tarjan, 1988] Goldberg, A. V. and Tarjan, R. E. (1988). A new approach to the maximum-flow problem. *J. ACM*, 35(4):921–940.

[Goldberg, 1989] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition.

[Hammersley and Clifford, 1971] Hammersley, J. M. and Clifford, P. E. (1971). Markov random fields on finite graphs and lattices.

[Holland, 1975] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems.* University of Michigan Press.

[Kennedy and Eberhart, 1995] Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948 vol.4.

[Kirkpatrick et al., 1983] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *SCIENCE*, 220(4598):671–680.

[Li, 2009] Li, S. Z. (2009). *Markov Random Field Modeling in Image Analysis.* Springer Publishing Company, Incorporated, 3rd edition.

[Metropolis et al., 1953] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. 21:1087–1092.

[Moscato, 1989] Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical Report C3P Report 826, California Institute of Technology.

[Park and Song, 1998] Park, Y. and Song, M. (1998). A genetic algorithm for clustering problems. In Koza, J. R., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D. B., Garzon, M. H., Goldberg, D. E., Iba, H., and Riolo, R., editors, *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 568–575. Morgan Kaufmann.

[Pavlidis, 1986] Pavlidis, T. (1986). A critical survey of image analysis methods. *ICPR*, pages 502–511.

[Perez, 1998] Perez, P. (1998). Markov random fields and images. *CWI Quarterly*, pages 413–437.

[Sonka et al., 2007] Sonka, M., Hlavac, V., and Boyle, R. (2007). *Image Processing, Analysis, and Machine Vision*, volume 3. PWS Publishing.

[Černý, 1985] Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45:41–51.