Saarland University
Faculty of Natural Science and Technology I
Department of Computer Science

Master's Thesis

Symmetry Detection in Images Using Belief Propagation

submitted by

Silke Jansen

on
July 1, 2010

Supervisor
Dr. Michael Wand

Advisor
Dr. Ruxandra Lasowski

Reviewers
Dr. Michael Wand
Prof. Dr. Hans-Peter Seidel

Statement in Lieu of an Oath

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis.

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken,

Abstract

In this thesis a general approach for detection of symmetric structures in images is presented. Rather than relying on some feature points to extract symmetries, symmetries are described using a probabilistic formulation of image self-similarity. Using a Markov random field we obtain a joint probability distribution describing all assignments of the image to itself. Due to the high dimensionality of this joint distribution, we do not examine this distribution directly, but approximate its marginals in order to gather information about the symmetries with the image. In the case of perfect symmetries this approximation is done using belief propagation. A novel variant of belief propagation is introduced allowing for reliable approximations when dealing with approximate symmetries. We apply our approach to several images ranging from perfect synthetic symmetries to real-world scenarios, demonstrating the capabilities of probabilistic frameworks for symmetry detection.

Acknowledgements

During the time of writing this thesis I have been supported by several people whom I would like to thank sincerely.

I want to thank Prof. Dr. Hans-Peter Seidel for providing such excellent working conditions. Likewise, I want to express my gratitude to Michael Wand for numerous insightful discussions. Special thanks goes to Ruxandra Lasowski for her advice on belief propagation.

Last but not least, I would like to thank David Spieler for his never-ending encouragement and support.

Contents

1 Introduction

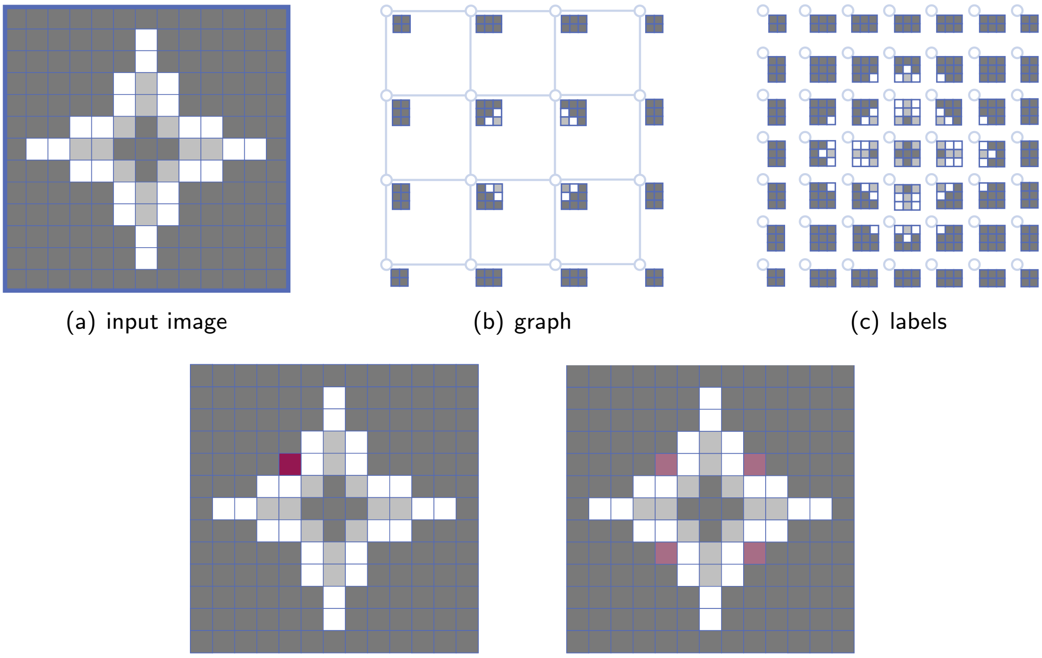
> *"Symmetry is what we see at a glance"*
>
> Blaise Pascal (1623-1662)
> mathematician, physicist, and philosopher

Symmetries and recurring parts are prevalent everywhere around us. They can be observed in nature as well as in man-made objects such as architecture or arts. The ability to recognize symmetric structures is a key-concept to understand the world surrounding us. It is not surprising that the capability to detect symmetries efficiently can be very useful for object recognition [26]. Furthermore, symmetry detection can be beneficial in various other applications, such as compression, noise removal, shape completion, and content variation.

Despite the ability of humans to perceive symmetries or symmetric structures instantaneously, automated discovery of such symmetries is a challenging task. Unsurprisingly a lot of research in computer vision and computer graphics has been devoted to the task of symmetry detection [1, 2, 6, 10, 17, 20–22, 25, 27, 30–32].

Formally we understand symmetry as the invariance under rigid transformations and mirroring. The symmetries considered in this thesis can be partial, approximative or both. The aim of this thesis is to extract symmetries of a given image based on a probabilistic formulation of image self-similarity rather than heuristics. Hence, no prior knowledge of the symmetry such as shape, location, or number of repetitions is needed, yielding a very general approach for symmetry detection.

The conceptual idea of our approach is illustrated in Figure 1.1. Inspired by the work of Lasowski et al. [17], we use a Markov random field (MRF) to describe a probability distribution over all possible matches of an image to itself (cf. Figures 1.1(a) - 1.1(c)). The symmetries within the image are revealed by this joint probability distribution. However, the probability space is of exponential size, making direct extraction of symmetries impractical. Fortunately, this distribution typically has only a few peaks, and consequently, we can examine the distribution after projection to a smaller subspace. We do this by computing marginal distributions of image points (cf. Figure 1.1(e)). The marginal probabilities of a point are obtained by fixing candidate point matches and summing the joint probabilities of all assignments containing the match. Hence, the resulting marginal probabilities indicate whether the matched points constitute mating points of symmetric parts. Given these marginal distributions, the actual symmetries can be extracted by region growing (cf. [17]).



(a) input image (b) graph (c) labels

(d) query point (e) marginal probability

Figure 1.1: Basic Idea. The input image is sampled twice, resulting in a set of *variables* or *nodes* (b) and a set of *labels* (c). Each variable ranges over the set of labels. The MRF describes a joint probability distribution over all possible assignments: each variable should locally be well-described by the assigned label; the distance between neighboring nodes, i.e. those connected by an edge in (b), should be approximately retained by the labels. Marginal probabilities (d) are computed by fixing one assignment and summing (marginalizing) over all other possibilities. Marginal distributions are a good indicator for symmetric parts.

Since exact computation of marginals requires summation of an exponential amount of terms, approximations have to be used. In the case of perfect symmetries, loopy belief propagation yields good results. In order to obtain reliable results when dealing with approximate symmetries, marginals of several spanning trees of the original underlying graph (cf. Figure 1.1(b)) are taken into account.

In this thesis we will discuss how the symmetric structure of an image can be characterized using an MRF. The marginals of the joint probability distribution described by the MRF provide valuable evidence for symmetries in the image. Hence, we will focus on retrieving good approximations to these marginal distributions.

1.1 Related Work

A number of approaches for symmetry detection in images have been developed [6, 10, 20, 21, 30, 32]. Most of these approaches focus on certain types of symmetries, such as rotational symmetries [6, 30], reflective symmetries [10, 20], or both [21].
Cornelius and Loy [6] detect rotational symmetries under affine projections by using feature pairs to predict possible centers of rotation. The centers are grouped to detect dominant rotational symmetries. In [21] Loy and Eklund use ideas based on the Hough transform in order to detect reflective and rotational symmetries in images. Gofman and Kiryati [10] use a global optimization approach rather than enumeration to extract all reflective symmetries. However this approach is limited to reflective symmetries.
In [32] Tuytelaars et al. consider a wider range of symmetries. By using invariant based hashing and Hough transforms, regular repetitions of planar patterns, including periodicities, can be detected. However, their approach can not handle rotations.

Symmetry detection within 3D objects has recently gained some attention in graphics research [1, 2, 17, 22, 25, 27, 31]. As for images, many methods are based on transformation voting, e.g. [22, 27]. Mitra et al. [22] use pairs of points with similar shape signature as evidence for potential symmetries. Clustering in transformation space is employed to extract candidate symmetries, followed by a verifying step. A major drawback of transformation voting techniques is their restriction to transformations that can be characterized by only few parameters. By using feature-based graph matching, Berner et al. [2] circumvent this restriction. However, their method is sensitive to the quality of the extracted feature lines.

This thesis is based on a more general approach to symmetry detection closely related to the works of Lasowski et al. [17] and Anguelov et al. [1]. In [17] the authors use a Markov random field model describing a probability distribution over all matches of a shape to itself. The intrinsic symmetries are extracted by approximating the marginals using loopy belief propagations, followed by peak tracking and region growing.

1.2 Organization

The remainder of this thesis is organized as follows. Chapter 2 and 3 provide short introductions to graphical models, focusing on Markov random fields (MRFs), and loopy belief propagation (LBP), respectively. Chapter 4 describes how the task of symmetry detection can be formulated as an MRF, suggesting the use of LBP to approximate the desired marginals. However, it will turn out that LBP only provides valuable approximations in case of perfect symmetries, as shown in chapter 5. The chapter proceeds to consider variants of LBP to overcome this shortcoming and ends by proposing a novel approach using spanning trees yielding better results. An evaluation and comparison of both approaches is given in chapter 6. Chapter 7 concludes the thesis by providing an outlook to some possible future work.

Notations and Preliminary Remarks. In the scope of this thesis, X_i will denote a random variable ranging over values $x_i \in \mathcal{X}_i$. We will write $p(x_i)$ instead of $p(X_i = x_i)$ whenever the X_i is obvious from the context. For set S, X_S denotes the set of random variables $\{X_s | s \in S\}$, and $x_S = \{x_s | s \in S\}$ denotes their corresponding values. The joint probability distribution of a set of random variables $X_S = \{X_{s_1}, \ldots, X_{s_n}\}$ is written as $p(X_S) = p(X_{s_1}, \ldots, X_{s_n}\}$. Bold letters represent vectors and subscripts refer to their elements, i.e. $p(\mathbf{x}) = p(x_1, \ldots, x_N) = p(X_1 = x_1, \ldots, X_N = x_N)$.

For the sake of simplicity we will only consider discrete random variables; however, the theoretical concepts introduced can also be generalized for continuous random variables.

2 Graphical models

Graphical models [3, 33] provide a concise and intuitive representation of joint probability distributions. They describe the probabilistic relationship of random variables using graphs. The nodes of the graph represent random variables, and the conditional independence assumptions are described by the edges, or more precisely, by the absence of edges. Figures 2.1, 2.2, and 2.3 show examples of different types of graphical models.

The two most common forms of graphical model are *directed graphical models*, usually referred to as *Bayesian networks* [23, 28], or *Belief networks*, and *undirected graphical models*, also known as *Markov random fields* [14, 23, 33].

Although this thesis only involves undirected graphical networks, a short introduction to the fundamentals of Bayesian networks is included for the sake of completeness. Furthermore, a unifying representation for Bayesian networks and Markov random fields, called *factor graph* [15, 23], is introduced in this chapter.

2.1 Bayesian Networks

A *directed graph* $G = (V, A)$ is a pair of vertices (or nodes) V and arcs (or directed edges) $A \subseteq V \times V$. We call G *acyclic* if there is no path from a vertex to itself, i.e. there is no sequence $s_1, \ldots, s_k$ with $(s_k, s_1) \in A$ and $(s_i, s_{i+1}) \in A$ for $i = 1, \ldots, k-1$. For vertex $v \in V$, let $\pi(v) = \{t | (t, v) \in A\}$ denote the set of *parents* of this node.

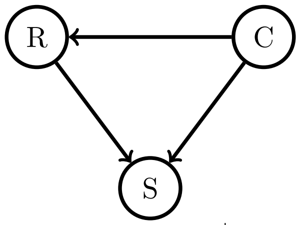
A *Bayesian network* consist of a directed acyclic graph and a collection of conditional probability distributions $(p_i)_{i \in V}$. Each vertex i of the graph represents a random variable X_i ranging over values x_i in some space $\mathcal{X}_i$. The conditional probability distribution for X_i given its parents $X_{\pi(i)}$ is given by p_i and often represented by a conditional probability table, as can be seen in Figure 2.1.

The joint probability distribution described by the Bayesian network is given by

$$p(\mathbf{x}) = \prod_{s \in V} p_s(x_s | x_{\pi(s)}) \,. \tag{2.1}$$

Accordingly, the joint probability distribution for the Bayesian network depicted in Figure 2.1 is given by $p(G, S, R) = p(G|S, R)p(S|R)p(R)$.

Cloudy	Rain T	Rain F
F	0.01	0.99
T	0.4	0.6



Cloudy T	Cloudy F
0.5	0.5

Rain	Cloudy	Shorts T	Shorts F
F	F	0.8	0.2
F	T	0.4	0.6
T	F	0.3	0.7
T	T	0.01	0.99

Figure 2.1: Example of a Bayesian network describing the effect of weather on clothing. Wearing shorts (S) is probably not a good idea when it is cloudy (C) and raining (R). The later one is usually dependent on the presence of clouds.

2.2 Markov Random Fields

An *undirected graph* $G = (V, E)$ is a pair of vertices V and a set of undirected edges $E \subseteq \{\{i, j\} | i, j \in V, i \neq j\}$. Two vertices i and j are said to be *adjacent*, written $i \sim j$, if they are connected by an edge, i.e. $\{i, j\} \in E$. Let $\mathcal{N}(i) = \{j | i \sim j\}$ denote the set of *neighbors* (or *neighborhood*) of i.

A family of random variables $(X_i)_{i \in V}$ is said to form a *Markov random field* (MRF) with respect to $G = (V, E)$ if its joint probability is strictly positive and one of the equivalent Markov properties holds, i.e. $p(x_i | x_j, i \neq j) = p(x_i | x_j, j \in \mathcal{N}(i))$.

Unfortunately, there is no direct way to compute the joint probability distribution given the conditional distribution [12]. The *Hammerlsey-Clifford theorem* [4, 11], however, guarantees that the probability distribution will factorize into a product of clique potentials.

2.2.1 Alternative Formulation

We call $C \subseteq V$ a *clique* of G if its is fully connected, i.e. $i \sim j$ for all $i, j \in C$. Let $\mathcal{C}$ denote the set of all cliques of graph G. A clique C is *maximal* if no more vertices can be added to it, i.e. C is not a strict subset of any other clique.

A *Markov random field* consists of an undirected graph G and a collection of distributions $(\psi_I)_{I\in\mathcal{C}}$. Each $i \in V$ represents a random variable taking values in $\mathcal{X}_i$; each clique C is associated with a *potential function* $\psi_C : \mathcal{X}_C \to \mathbb{R}_+$ (also called *compatibility function* or *clique potential*).

The joint probability distribution described by the MRF is given by

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{C\in\mathcal{C}} \psi_C(x_C)\,, \tag{2.2}$$

where Z is a constant given by

$$Z = \sum_{\mathbf{x}} \prod_{C\in\mathcal{C}} \psi_C(x_C) \tag{2.3}$$

ensuring correct normalization of the distribution. The set $\mathcal{C}$ can also be taken as the set of maximal cliques, without the loss of generality.

Since the potential functions are strictly positive, they can also be expressed as

$$\psi_C = \exp\{-E(C)\}\ ,$$

illustrating the relation to Gibbs distributions underlying the Hammerlsey-Clifford theorem [19].

2.2.2 Pairwise Markov Random Fields

Pairwise MRFs have been studied in statistical physics (Ising model [13]) and applied to various computer vision problems [9]. In pairwise MRFs, we are usually given some observed quantities y_i for the nodes, e.g. the gray-value at the corresponding pixel. The aim is to gain some knowledge about the underlying scene variables x_i.

An example of a pairwise MRF is shown in Figure 2.2. The MRF is depicted using two different representations: on the right, the observed quantities are drawn explicitly (filled nodes), while they are omitted in the left representation.

In the case of pairwise MRFs we have two types of potential functions. The potential function $\phi_i(x_i, y_i)$, often called *evidence* for x_i, expresses the statistical dependency between x_i and y_i, for each node i. The compatibility of neighboring variables is assessed by the *compatibility function* $\psi_{ij}(x_i, x_j)$. The joint probability of the hidden scene and the observations is given by

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_{i} \phi_i(x_i, y_i) \prod_{i\sim j} \psi_{ij}(x_i, x_j). \tag{2.4}$$

Considering the observed variables to be fixed, we can rewrite the joint probability of the unknown variables as

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{i} \phi_i(x_i) \prod_{i\sim j} \psi_{ij}(x_i, x_j). \tag{2.5}$$

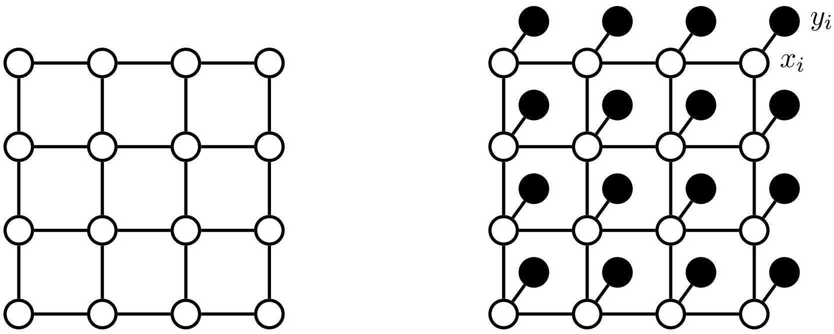


Figure 2.2: Square lattice pairwise MRF. The representation on the right explicitly represents the observed variable nodes, which are filled, while the *hidden* ones are denoted by empty circles.

Please note that (2.5) is a special case of (2.2). Therefore, ψ_{ij} should be read as $\psi_{\{i,j\}}$ and accordingly $\psi_{ij} = \psi_{ji}$ holds for all $i \sim j$.

2.3 Factor Graphs

The factorization properties of undirected and directed graphical models (equations 2.1 and 2.2, respectively) lead to a unifying representation, called *factor graphs*.

Assume that $g(x_1, \ldots, x_n)$ factorizes into a product of *local functions* (or *factors*) $(f_I)_{I \in F}$:

$$g(x_1, \cdots, x_n) = \prod_{I \in F} f_I(x_{N_I}), \tag{2.6}$$

where N_I is a subset of variable indices associated with factor index I, and f_I takes the elements of $\mathcal{X}_{N_I}$ as arguments.

A *factor graph* is a bipartite graph $G = (V, F, E)$ that expresses the structure of a factorization (2.6). A factor graph has a *variable node* $i \in V$ for each variable x_i, a *factor node* $I \in F$ for each local function f_I, and an undirected edge $\{i, I\} \in E$ connecting variable node i and factor node I, if and only if, f_I depends on x_i, i.e. $i \in N_I$.

Factor graphs equivalent to the graphical models given in Figures 2.1 and 2.2 are shown in Figure 2.3. Variable nodes are represented by circles and factor nodes are denoted by rectangles.

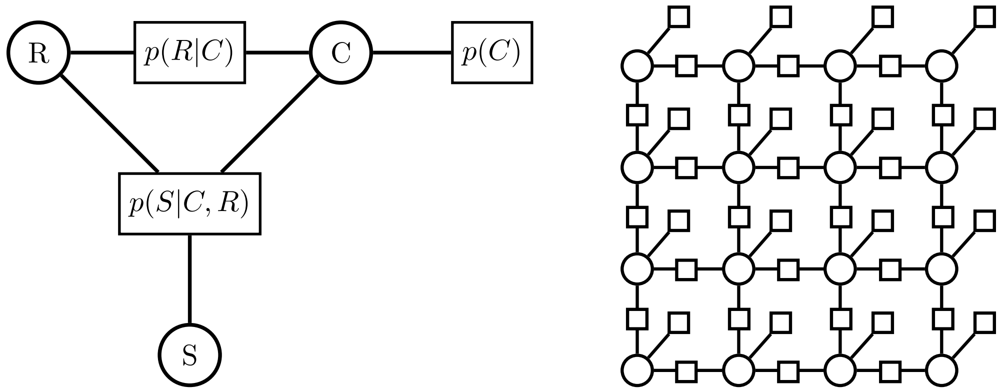


Figure 2.3: Factor graphs corresponding to the Bayesian network and the MRF given in Figures 2.1 and 2.2, respectively.

2.4 Conversion of Graphical Models

As already mentioned, both directed and undirected graphical models can be represented by factor graphs (see Figure 2.3 for examples). For Bayesian networks, the factors correspond to the conditional probability distributions. In the case of MRFs, the factors directly relate to the clique potentials. Figure 2.4 illustrates the conversion of a pairwise MRF into a factor graph.

We end this section by noting that arbitrary factor graphs can be converted into equivalent MRFs or directed graphical models, where the later is more involved [3,39]. Hence, we can freely choose a representation without any loss of generality.

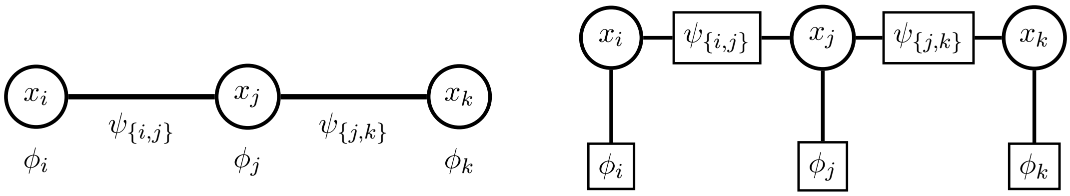


Figure 2.4: Conversion of a pairwise MRF into a factor graph.

3 Inference in Markov Random Fields

As we have seen in the previous chapter, the joint probability distribution defined by a pairwise MRF can be written as

$$p(\mathbf{x}) = \frac{1}{Z} \prod_i \phi_i(x_i) \prod_{i \sim j} \psi_{ij}(x_i, x_j)\,.$$

The normalizing constant Z is usually unknown. Furthermore, even if Z was known, direct examination of this exponential sized distribution is typically not feasible. Thus we rely on the marginals of the distribution, hinting at areas with high probabilities. Marginals are obtained by summing the joint distribution over all variables except one. Accordingly, the marginal probability distribution of X_i is given by

$$p(X_i = x_i) = \sum_{x_1} \cdots \sum_{x_{i-1}} \sum_{x_{i+1}} \cdots \sum_{x_N} p(\mathbf{x})\,. \tag{3.1}$$

In the case of symmetries, a high marginal probability $p(X_i = x_i)$ indicates that the points associated with the variable X_i and the assigned value x_i are likely to be corresponding points of symmetric parts.

In the scope of this thesis, *inference* will refer to the task of retrieving marginal probabilities of random variables. Exact inference is often computationally infeasible and approximations have to be used. A wide-spread approach for approximate inference is *belief propagation* (BP) [3, 39]. Belief propagation has been independently developed in different fields, such as statistical physics [24], error-correcting codes [8], or artificial intelligence [29].

This chapter starts with exact inference on chains and trees, motivating the belief propagation algorithm which can be applied to general graphs.

3.1 Exact Inference

In general, exact computation of (3.1) is intractable.[1] Consider the case of an undirected graphical network containing N discrete random variables X_i, each ranging over K states. To compute the marginal distribution of a single variable X_i, K^N operations are necessary, K^{N-1} operations for each possible state of X_i.

[1] In [5] Copper proves that for Bayesian networks (strongly related to MRFs, cf. section 2.4), probabilistic inference is NP-hard.

However, this naive way of computing the marginal distribution does not take the conditional independence structure into consideration. Taking the structure of the graph into account, exact computation of marginal distributions becomes feasible for some types of graphical models.

3.1.1 Inference on a Chain

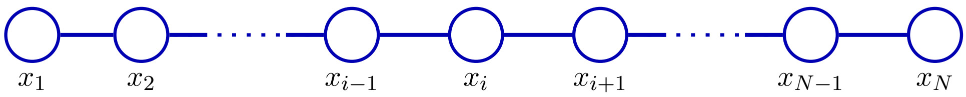


Figure 3.1: MRF with N random variables X_i each ranging over $|\mathcal{X}_i| = K$ values.

Figure 3.1 shows a pairwise MRF based on a chain structured graph. Since this linear graph contains no vertex with more than two neighbors, each variable associated to a node of the graph can participate in at most two compatibility terms.

Accordingly, the joint probability distribution described by the MRF takes the form

$$p(\mathbf{x}) = \frac{1}{Z}\phi_1(x_1)\dots\phi_N(x_N)\cdot\psi_{1,2}(x_1,x_2)\psi_{2,3}(x_2,x_3)\dots\psi_{N-1,N}(x_{N-1},x_N).$$

For random variable X_i, the marginals are given by

$$p(x_i) = \frac{1}{Z}\sum_{x_1}\sum_{x_2}\cdots\sum_{x_{i-1}}\sum_{x_{i+1}}\cdots\sum_{x_N}\phi_1\dots\phi_N\cdot\psi_{1,2}\psi_{2,3}\dots\psi_{N-1,N}\,,$$

where $\psi_{i,j}$ and ϕ_i denote $\psi_{i,j}(x_i,x_j)$ and $\phi_i(x_i)$, respectively, for reasons of legibility.

By changing the order of summation and multiplication, this term can be rewritten as:

$$\begin{aligned}
p(x_i) =& \frac{1}{Z}\cdot\phi_i\cdot\left[\sum_{x_1}\sum_{x_2}\cdots\sum_{x_{i-1}}\phi_1\dots\phi_{i-1}\cdot\psi_{1,2}\psi_{2,3}\dots\psi_{i-1,i}\right] \\
&\left[\sum_{x_{i+1}}\cdots\sum_{x_N}\phi_{i+1}\dots\phi_N\cdot\psi_{i,i+1}\dots\psi_{N-1,N}\right] \\
=& \frac{1}{Z}\cdot\phi_i\cdot\left[\sum_{x_{i-1}}\phi_{i-1}\psi_{i-1,i}\dots\left[\sum_{x_2}\phi_2\psi_{2,3}\left[\sum_{x_1}\phi_1\psi_{1,2}\right]\right]\dots\right] \\
&\left[\sum_{x_{i+1}}\phi_{i+1}\psi_{i,i+1}\dots\left[\sum_{x_{N-1}}\phi_{N-1}\psi_{N-2,N}\left[\sum_{x_N}\phi_N\psi_{N-1,N}\right]\right]\dots\right]
\end{aligned}$$

Using this decomposition, we can reduce the computational effort from exponential in N (for the naive approach) to linear in the number of vertices:

In order to compute $\sum_{x_2}\phi_2\psi_{2,3}\sum_{x_1}\phi_1\psi_{1,2}$, we first have to evaluate $\sum_{x_1}\phi_1(x_1)\psi_{1,2}(x_1,x_2)$ for every $x_2 \in \mathcal{X}_2$, requiring $|\mathcal{X}_2|\cdot|\mathcal{X}_1| = K^2$ operations. The result is a K-dimensional vector, which we will denote by $m_{1\to 2}$. This vector is passed onto the next summation, yielding $\sum_{x_2}\phi_2(x_2)\psi_{2,3}(x_2,x_3)\cdot m_{1\to 2}(x_2)$. Since $m_{1\to 2}(x_2)$ is a scalar factor, evaluation of this sum again requires $O(K^2)$ steps. This scheme is continued until $m_{i-1\to i}$ is reached. Analogously, $m_{i+1\to i}$ is obtained, leading to a total cost of $O(NK^2)$ to compute

$$p(x_i) = \frac{1}{Z}\cdot\phi_i\cdot\underbrace{\left[\sum_{x_{i-1}}\phi_{i-1}\psi_{i-1,i}\cdots\underbrace{\left[\sum_{x_1}\phi_1\psi_{1,2}\right]}_{m_{1\to 2}(x_2)}\cdots\right]}_{m_{i-1\to i}(x_i)}\underbrace{\left[\sum_{x_{i+1}}\phi_{i+1}\psi_{i,i+1}\cdots\underbrace{\left[\sum_{x_N}\phi_N\psi_{N-1,N}\right]}_{m_{N\to N-1}(x_{N-1})}\cdots\right]}_{m_{i+1\to i}(x_i)}.$$

The vectors $m_{i\to j}$ are commonly called *messages*, since $m_{i\to j}$ can be thought of a message from node i to node j about what state j should be in.

Assuming that all variables range over the same set of values, i.e. $\mathcal{X}_j = \mathcal{X}_i$, we can use the messages computed so far rather than performing the above computations separately for each variable.

E. g. consider the marginal distribution of $X_{i+1} = X_j$:

$$p(x_j) = \frac{1}{Z}\psi_j(x_j)\left[\sum_{x_i}\phi_i(x_i)\psi_{i,j}(x_i,x_j)\cdot m_{i-1\to i}(x_i)\right]\cdot m_{j+1\to j}(x_j)$$

This leads to an efficient, yet exact algorithm to compute all marginals of any chain structured graph:

Given a chain-structured graph $G = (V, E)$.
(W. l. o. g. $V = \{1,\ldots,N\}, E = \{\{i, i+1\}|i = 1,\ldots,N-1\}$)

1. start with computing $m_{1\to 2}$, continue until $m_{N-1\to N}$ is reached
2. analogously compute all opposite messages, starting with $m_{N\to N-1}$

   Given all messages, we can calculate $p(x_j)$ for any X_j by

$$p(x_j) = \frac{1}{Z}\psi_j(x_j)\cdot m_{(j-1)\to j}(x_j)\cdot m_{(j+1)\to j}(x_j)\,,$$

   where $m_{0\to 1} = m_{N+1\to N} = \mathbf{1}$.

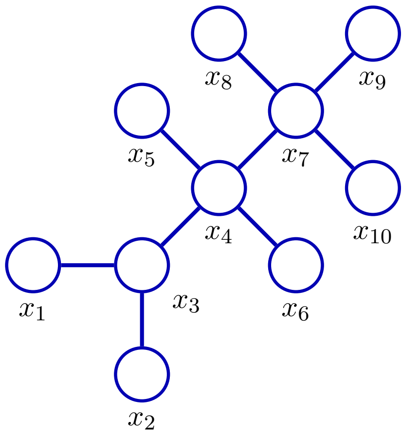


Figure 3.2: A small tree-structured MRF.

3.1.2 Inference on Trees

A similar algorithm can be derived for cycle-free graphs. Consider the MRF based on the tree-structured graph given in Figure 3.2. Again we can rewrite the marginals, e.g. for X_4, by changing the order of summation and multiplication.

$$
\begin{aligned}
p(x_4) =& \frac{1}{Z} \sum_{x_1} \cdots \sum_{x_3} \sum_{x_5} \cdots \sum_{x_{10}} \phi_1 \ldots \phi_{10} \psi_{1,3} \psi_{2,3} \psi_{3,4} \psi_{4,5} \psi_{4,6} \psi_{4,7} \psi_{7,8} \psi_{7,9} \psi_{7,10} \\
=& \frac{1}{Z} \cdot \phi_4 \cdot \left[\sum_{x_1} \sum_{x_2} \sum_{x_3} \phi_1 \phi_2 \phi_3 \psi_{1,3} \psi_{2,3} \psi_{3,4} \right] \left[\sum_{x_5} \phi_5 \psi_{4,5} \right] \left[\sum_{x_6} \phi_6 \psi_{4,6} \right] \\
& \left[\sum_{x_7} \sum_{x_8} \sum_{x_9} \sum_{x_{10}} \phi_7 \phi_8 \phi_9 \phi_{10} \psi_{4,7} \psi_{7,8} \psi_{7,9} \psi_{7,10} \right] \\
=& \frac{1}{Z} \cdot \phi_4 \cdot \underbrace{\left[\sum_{x_3} \phi_3 \psi_{3,4} \underbrace{\left[\sum_{x_1} \phi_1 \psi_{1,3} \right]}_{m_{1\to3}} \underbrace{\left[\sum_{x_2} \phi_2 \psi_{2,3} \right]}_{m_{2\to3}} \right]}_{m_{3\to4}} \underbrace{\left[\sum_{x_5} \phi_5 \psi_{4,5} \right]}_{m_{5\to4}} \underbrace{\left[\sum_{x_6} \phi_6 \psi_{4,6} \right]}_{m_{6\to4}} \\
& \underbrace{\left[\sum_{x_7} \phi_7 \psi_{4,7} \left[\sum_{x_8} \phi_8 \psi_{7,8} \right] \left[\sum_{x_9} \phi_9 \psi_{7,9} \right] \left[\sum_{x_{10}} \phi_{10} \psi_{7,10} \right] \right]}_{m_{7\to4}}
\end{aligned}
$$

As in the previous case, we can reuse some of the messages calculated so far for computing the marginal distribution of other variables:

$$
p(x_3) = \frac{1}{Z} \cdot \phi_3(x_3) \cdot m_{1\to3}(x_3) \cdot m_{2\to3}(x_3) \cdot \left[\sum_{x_4} \phi_4(x_4) \psi_{3,4}(x_3, x_4) \cdot m_{5\to4}(x_4) \cdot m_{6\to4}(x_4) \cdot m_{7\to4}(x_4) \right]
$$

We can express these messages for an arbitrary cycle-free graph by

$$m_{i\to j}(x_j) = \sum_{x_i} \phi_i(x_i)\psi_{i,j}(x_i, x_j) \prod_{k\in\mathcal{N}_{i\setminus j}} m_{k\to i}(x_i)\,, \tag{3.2}$$

where $\mathcal{N}_{i\setminus j}$ is a shorthand notation for $\mathcal{N}(i) \setminus j$.

Using (3.2) we can algorithmically compute marginal probabilities for tree-structured graphs.

Given an undirected graphical model without cycles

1. pick some node i to be the root of the tree
2. propagate messages, using (3.2), from the leaves to the root
3. propagate messages starting at the root to the leaves

Given all messages, the marginal probabilities can be computed by

$$p\left(x_i\right) = \tfrac{1}{Z} \cdot \phi_i\left(x_i\right) \prod_{j\in\mathcal{N}(i)} m_{j\to i}\left(x_i\right)\,. \tag{3.3}$$

This scheme ensures that all incoming messages ($m_{j\to i}$) of variable node i have been computed before any outgoing message ($m_{i\to j}$) is evaluated. Accordingly, each message needs to be calculated exactly once to obtain all marginal distributions.

The propagation of messages in trees is illustrated in Figure 3.3.

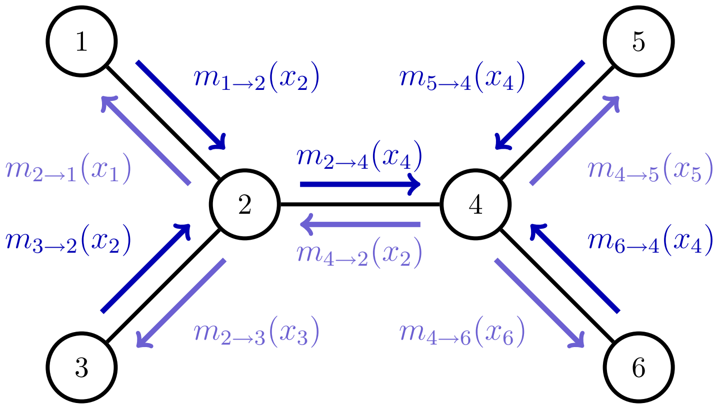


Figure 3.3: Message passing on trees. First, messages are propagated inwards towards the root (4), followed by outward propagation of messages.

3.2 Belief Propagation

The algorithm defined by (3.2) and (3.3) is known as the *sum-product algorithm* or *belief propagation* (BP). The messages are usually defined by recursive updates:

$$m_{i\to j}(x_j) \leftarrow \sum_{x_i} \phi_i(x_i)\psi_{i,j}(x_i, x_j) \prod_{k\in\mathcal{N}_{i\setminus j}} m_{k\to i}(x_i) \tag{3.4}$$

Typically, the messages are updated in *parallel*, i.e. all updates are computed at once using the messages of the previous iteration, but other schedules (e.g. *serial*) can be used as well. [2]

For parallel updates, (3.5) can be also written as:

$$m^t_{i\to j}(x_j) = \sum_{x_i} \phi_i(x_i)\psi_{i,j}(x_i, x_j) \prod_{k\in\mathcal{N}_{i\setminus j}} m^{t-1}_{k\to i}(x_i) \tag{3.5}$$

When using a parallel update schedule, the messages need to be initialized. Typically uniform initialization is used:

$$m^0_{k\to l}(x_l) = \tfrac{1}{|\mathcal{X}|}\,,$$

for all messages $m_{k\to l}$ and labels x_l.

Once the updates have converged, the marginals can be calculated using equation 3.3. For tree-structured graphs it can be shown [29] that the updates will converge, and that the corresponding marginals given by (3.3) are indeed exact.

3.2.1 Loopy Belief Propagation

Although only guaranteed to converge on cycle-free graphs, BP may also be applied to general graphs. The algorithm is then typically referred to as *loopy belief propagation* (LBP) to emphasize its usage on graphs containing loops.

Since the computed marginals are only approximative, they are usually called *beliefs*. The beliefs for node i are given by

$$b\left(x_i\right) = k \cdot \phi_i\left(x_i\right) \prod_{j\in\mathcal{N}(i)} m_{j\to i}\left(x_i\right), \tag{3.6}$$

where k is a normalization constant, s.t. $\sum_{x_i} b(x_i) = 1$.

When applying LBP to graphs containing cycles, it is not possible to compute all messages before they are needed to update another message (cyclic dependencies). Hence, all messages need to be initialized - typically using uniform initialization.

[2] The schedule used in section 3.1.2 is a special case of a serial schedule.

3.2.2 The Max-Product Algorithm

Replacing the sum in the update rule (3.5) by the maximum operator, the *max-product algorithm* [33] is obtained. For chain-structured graphs, the algorithm is usually known as the Viterbi algorithm [7].

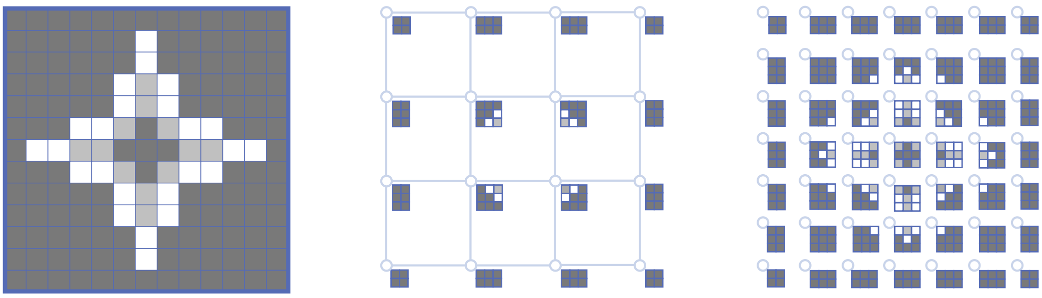
The max-product algorithm computes (or approximates, in the case of cyclic graphs) the max-marginals

$$\nu_s(x_s) = \max_{\{\mathbf{x}' | x'_s = x_s\}} p(x_1, \dots, x_n)$$

of the distribution, which can be used to compute the mode of the distribution.

4 Pairwise MRFs for Symmetry Detection in Images

Given some gray-scale image, an MRF describing its symmetry structure is constructed as shown in Figure 4.1. A set of nodes and a set of labels (Figures 4.1(b) and 4.1(c) respectively) are sampled from the image. The MRF defines a probability distribution over all possible assignments of the nodes to the labels. For a high probability of an assignment the nodes should be locally well described by the assigned labels and neighboring nodes, i.e. nodes connected by an edge, should behave geometrically consistently.



(a) input image (b) graph (c) labels

Figure 4.1: Construction of a MRF for symmetry detection.

A *continuous gray-scale image* can be represented by a mapping $f : \Omega \rightarrow \mathbb{R}$, where $\Omega \subset \mathbb{R}^2$ is a rectangular domain, i.e. $\Omega = (0, a1) \times (0, a2)$ [35]. In order to obtain a digital image $\{f_{ij} | i = 0, \dots, N-1; j = 0, \dots, M-1\}$, Ω is discretized by sampling and the range is commonly quantized to $\{0, 1, \dots, 255\}$ (for gray-scale images).

The single points (i, j) of the discretized image are called pixels. For pixel $a = (i, j)$, let a_x and a_y stand for i and j, respectively, and let f_a denote the gray-value at this position, i.e. f_{ij}.
The distance between two pixels a and b is given by $d(a, b) = ((a_x - b_x)^2 + (a_y - b_y)^2)^{\frac{1}{2}}$.

This chapter explains how the pairwise MRF is constructed, given an image of interest. Some aspects specific to images, e.g. image boundaries, have to be considered when constructing the MRF. After pointing out the necessity to take these aspects into account, possibilities to handle them adequately will be discussed.

4.1 Underlying Graph

In order to construct an MRF for symmetry detection, we need to define the underlying graph describing the conditional dependencies of the variables.

Given some image $\{f_{ij} | i = 0, \ldots, N-1; j = 0, \ldots, M-1\}$, we sample the pixel positions uniformly at a given sample distance s_V to obtain the set of nodes

$$V \subseteq \big\{(i,j) \big| i = 0, \ldots, N-1; j = 0, \ldots, M-1\big\} .$$

The vertices are horizontally and vertically connected by edges:

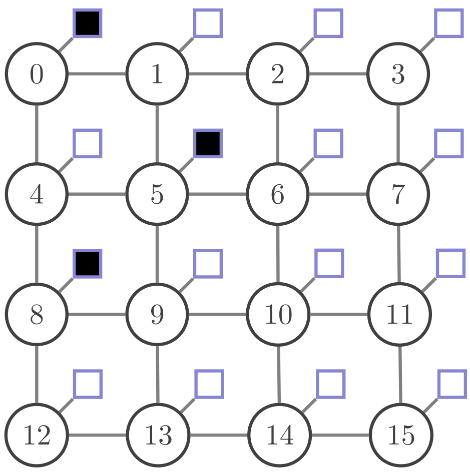
$$E = \ \big\{(i,j),(i+s_V,j) \big| (i,j),(i,j+s_V) \in V\big\} \cup \big\{(i,j),(i,j+s_V) \big| (i,j),(i+s_V,j) \in V\big\}.$$

Accordingly, the graph underlying the MRF is given by $G = (V, E)$.

4.1.1 Neighborhoods

Ideally, the neighborhoods of nodes should be defined such that non-rigid mappings of the image are only possible, i.e. their joint probability is significantly larger than zero, if the compatibility functions ψ_{ij} allow for some flexibility. This is however not the case if we use the neighborhoods as defined above.

Consider example 4.1 motivating the need for larger neighborhoods. Since the potential functions are very strict, no variation in the distance between neighboring nodes nor in the gray values associated with the node and the assigned label are tolerated. Therefore, the joint probability should only be non-zero for $\mathbf{x} = (0, 1, \ldots, 15)$. In particular the marginal distribution of X_0 should be zero for $X_0 \neq 0$.



$$\psi_{ij}(kl) = \begin{cases} 1 & d(i,j) = d(k,l) \\ 0 & \text{else} \end{cases}$$

$$\phi_i(j) = \begin{cases} 1 & f_i = f_j \\ 0 & \text{else} \end{cases}$$

Example 4.1: Small example demonstrating the necessity for larger neighborhoods. For simplicity, $\mathcal{X}$ and V are taken to be $\{0, \ldots, 15\}$.

But having a closer look at example 4.1 reveals:

$$p(X_0 = 5) = \sum_{\mathbf{x}, x_0=5} p(\mathbf{x}) \geq p(X_0 = 5, X_1 = 1, \dots, X_{15} = 15)$$
$$= \tfrac{1}{Z} \cdot \Big[\phi_0(5) \cdot \prod_{i=1}^{15} \phi_i(i)\Big] \cdot \Big[\psi_{0,1}(5,1)\psi_{0,4}(5,4) \prod_{\substack{i\sim j \\ i,j\neq 0}} \psi_{i,j}(i,j)\Big] = \tfrac{1}{Z} \neq 0$$

Intuitively, this is caused by folding the grid diagonally which keeps all distances between neighboring vertices unchanged. This can be avoided by extending the neighborhood of a vertex by including its diagonal grid neighbors (cf. Figure 4.2(a)).

When using diagonal neighbors, $p(X_0 = 5)$ is indeed zero. But still the desired marginal distribution for X_0 is not obtained, since the grid can still fold together horizontally:

$$p(X_0 = 8) \geq p(X_0 = 8, X_1 = 9, X_2 = 10, X_3 = 11, X_4 = 4, \dots, X_{15} = 15)$$
$$= \tfrac{1}{Z} \cdot \Big[\phi_0(8)\phi_1(9)\phi_2(10), \phi_3(11) \cdot \prod_{i=4}^{15} \phi_i(i)\Big] \cdot \Big[\prod_{i\sim j} \psi_{i,j}(i,j)\Big]$$
$$= \tfrac{1}{Z} \cdot \Big[\psi_{0,1}(8,9)\psi_{0,4}(8,4)\psi_{0,5}(8,5) \cdot \prod_{\substack{i\sim j \\ i,j\neq 0}} \psi_{i,j}(i,j)\Big] = \cdots = \tfrac{1}{Z} \neq 0\,.$$

Adding additional horizontal and vertical neighbors (cf. Figure 4.2(b)) helps to circumvent this problem. When using the neighborhoods as depicted in Figure 4.2(c), the desired joint distribution is obtained.[1]

The graphs corresponding to the extended neighborhoods introduced above are depicted in Figure 4.2. Figure 4.3 clarifies the effect of using different neighborhoods in example 4.1 by providing experimental results using accordant settings.

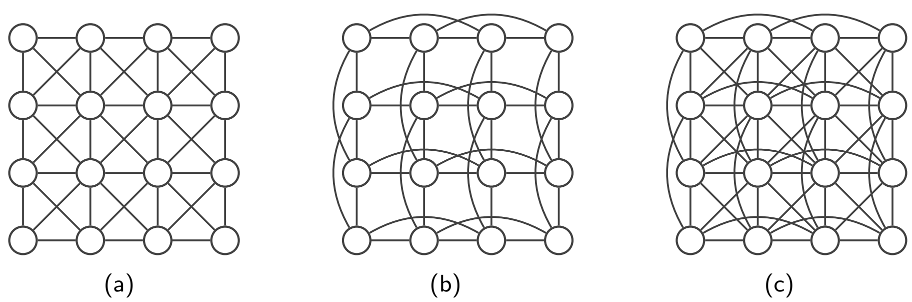


Figure 4.2: Extended Neighborhoods.

[1] In this specific example the neighborhood of 4.2(b) already gives the desired results, since ψ_{ij} does not allow for any deviation of the distance. Usually, ψ_{ij} allows for some deviations, cf. section 4.3.2, making 4.2(c) necessary.

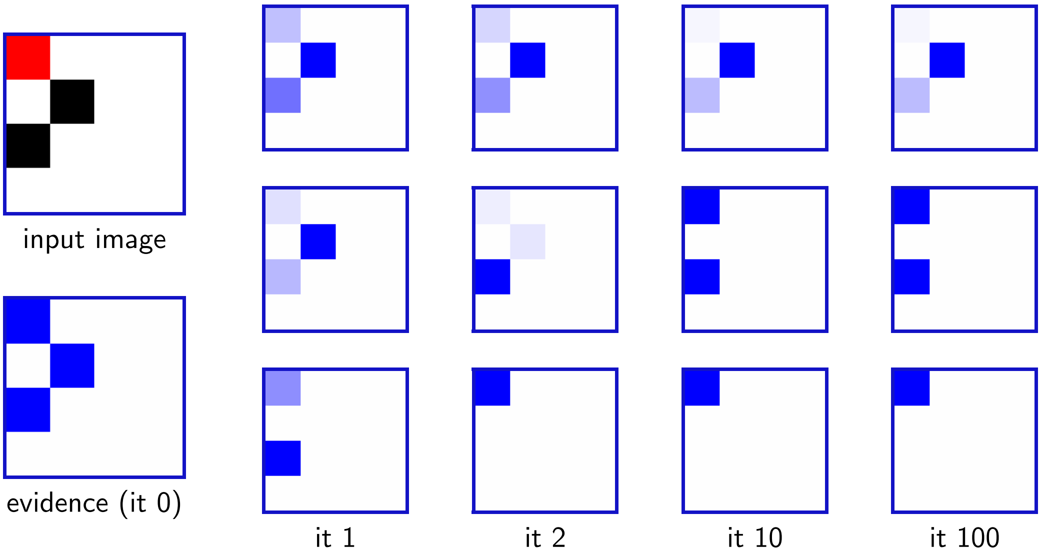


Figure 4.3: Effect of extended neighborhoods. Beliefs drawn after transformation to $[0, 1]$. For details on representation of results please refer to chapter 6.
Beliefs for the marked point when using direct grid neighbors (1^{st} row), additional diagonal neighbors (2^{nd} row), and full extended neighborhood (3^{rd} row).

In general, larger neighborhoods will lead to better results but will also cause an overhead in computation time (cf. section 6.4). The trade-off between accuracy and computation time has to be decided depending on the image under consideration. For instance, if background regions are dominating the image a large neighborhood is advisable.

4.2 Random Variables

Each vertex i of the graph described in section 4.1 represents a random variable X_i ranging over $\mathcal{X}_i$. In this thesis, we will usually refer to the elements of $\mathcal{X}_i$ as labels.

For the means of symmetry detection in images, pixels of the image are related to each other. Hence, the labels are sampled from the pixel positions of the image as well (recall, the same was done for the vertices of the graph, cf. section 4.1). Furthermore, all random variable X_i range over the same set of labels

$$\mathcal{X} \subseteq \big\{(i, j)\big|i = 0, \ldots, N-1; j = 0, \ldots, M-1\big\}\,.$$

Typically, we sample $\mathcal{X}$ denser than V, i.e. $|\mathcal{X}| \gg |V|$. This is not mandatory, but necessary to reliably detect symmetries, in particular rotational symmetries. Some examples in this thesis, however, will assume $V = \mathcal{X}$ in order to illustrate certain aspects more clearly.

4.3 Potentials

We have two types of potential functions: *single potentials* $\phi_i(x_i)$, usually called *evidence* for x_i; and *pairwise potentials* $\psi_{ij}(x_i, x_j)$, referred to as *compatibility functions*.

4.3.1 Evidence

The evidence $\phi_i(x_i)$ evaluates how well the pixel associated with node i is described by the pixel corresponding to x_i. Thus, descriptors are needed to characterize the image locally around these pixels. To account for rotational symmetries the descriptors need to be rotationally invariant up to some discretization error.

Let $desc(x)$ be a function computing a local descriptor for the pixel corresponding to x. Then, the evidence for x_i is given by

$$\phi_i(x_i) = \exp\left(-\frac{(desc(i) - desc(x_i))^2}{\sigma^2}\right) . \quad (4.1)$$

As a local descriptor we use a (approximately) Gaussian-weighted average of the neighborhood of the pixel under consideration. This actually amounts to taking the local gray-value after applying a low-pass filter with a Gaussian convolution kernel. Low-pass filters smooth images by eliminating noise and unimportant small-scale details [35], helping to reliably detect symmetries without getting "distracted" by the presence of noise. Furthermore, using a Gaussian kernel ensures the required rotational invariance.

We use a binomial kernel (e.g. Figure 4.4) as a discrete approximation to the Gaussian. The 5×5 binomial kernel is given in Figure 4.4. The actual size of the kernel is a user-defined parameter and should be adapted to the sampling distance, as well as to the amount of noise present in the image. In order to compute descriptors for pixels close to the boundary, the image is extended. Depending on the image under consideration, this is done by constant extensions or by assuming periodic or reflecting image boundaries.

$$\frac{1}{256} \times \begin{array}{|c|c|c|c|c|} \hline 1 & 4 & 6 & 4 & 1 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 6 & 24 & 36 & 24 & 6 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array}$$

Figure 4.4: (5×5) Binomial Kernel.

4.3.2 Pairwise Potentials

The compatibility function $\psi_{ij}(x_i, x_j)$ assesses how well the distance between the nodes is maintained by the labels and is given by

$$\psi_{ij}\left(x_i, x_j\right) = \exp\left(-\frac{\left(\|i,j\|_2 - \|x_i, x_j\|_2\right)^2}{\sigma^2}\right), \tag{4.2}$$

where the amount of deviation tolerated is influenced by the parameter σ.

Global symmetries can be reliably described using these compatibility terms. However, this does not hold for local symmetries. If an instance of a local symmetry is matched to another instance, it is not possible to maintain all distances on the graph. Hence, either many distance constraints are slightly violated or a few distances are not maintained at all. In both cases the joint probability of the assignment will be neglectable, since the product includes either many small terms or a few almost zero factors. Accordingly, marginal probabilities corresponding to local symmetries will be almost zero, unless ψ_{ij} allows large discrepancies in the distance.

In order to detect local symmetries we need to accept solutions which are geometrically consistent up to some global discontinuities. This is done by using truncation for the pairwise potentials to obtain geometrically piecewise consistent solutions:

$$\psi_{ij}\left(x_i, x_j\right) = \max\left\{\exp\left(-\frac{\left(\|i,j\|_2 - \|x_i, x_j\|_2\right)^2}{\sigma^2}\right), \rho^2\right\} \tag{4.3}$$

Using these modified compatibility functions, assignments with a few violations of the geometric compatibility can still receive a relatively high probability.

Besides allowing for local symmetries, truncation also has a positive effect on computation time. Consider the following decomposition of (3.5):

$$\begin{aligned}
m_{i\to j}\left(x_j\right) &\leftarrow \sum_{x_i} \phi_i\left(x_i\right)\psi_{ij}\left(x_i, x_j\right) \prod_{k\in\mathcal{N}_{i\setminus j}} m_{ki}\left(x_i\right) \\
&\leftarrow \sum_{x_i} \phi_i\left(x_i\right)\rho^2 \prod_{k\in\mathcal{N}_{i\setminus j}} m_{k\to i}\left(x_i\right) \;+\; \sum_{x_i} \phi_i\left(x_i\right)\left(\psi_{ij}\left(x_i, x_j\right) - \rho^2\right) \prod_{k\in\mathcal{N}_{i\setminus j}} m_{ki}\left(x_i\right) \\
&\leftarrow \underbrace{\rho^2 \sum_{x_i} \phi_i\left(x_i\right) \prod_{k\in\mathcal{N}_{i\setminus j}} m_{k\to i}\left(x_i\right)}_{\text{independent of } x_j} \;+\; \sum_{x_i : \psi_{ij} > \rho^2} \phi_i\left(x_i\right)\left(\psi_{ij}\left(x_i, x_j\right) - \rho^2\right) \prod_{k\in\mathcal{N}_{i\setminus j}} m_{ki}\left(x_i\right)
\end{aligned}$$

The first summation of the decomposition is independent of the actual x_j and hence only needs to be computed once per message update. To obtain the individual elements $m_{i\to j}\left(x_j\right)$ of the current message, we just need to sum over all of the labels x_i that yield a compatibility greater than ρ^2.

Furthermore, ψ_{ij} only depends on the relative positions, i.e. $\|i, j\| = \|k, l\| \Rightarrow \psi_{ij} = \psi_{kl}$. Hence, the compatibilities can be precomputed for all distances between neighboring nodes and only compatibilities greater than ρ^2 need to be stored.

4.4 Image Boundaries

Even when using the extended neighborhood as described earlier in this chapter, the joint probability, described by the MRF, might not characterize the symmetric structure as intended. As we will see, this is caused by insufficient treatment of image boundaries.

Figure 4.5 provides an example with very pronounced boundary artifacts. The marginal probabilities of the marked corner point are higher for points closer to the image center than for the corners.

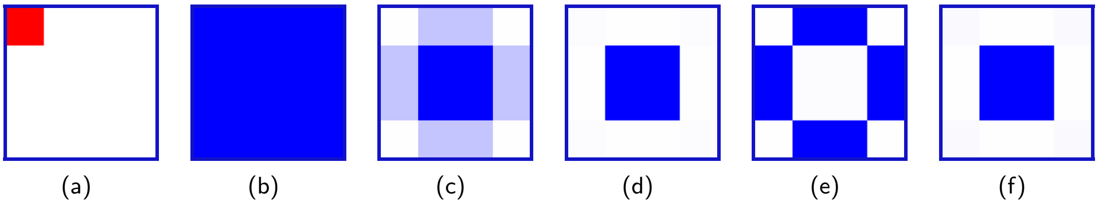


Figure 4.5: A simple example exhibiting boundary artifacts: (a) input image, query point marked, (b) evidence for marked point, (c) scaled beliefs after the first iteration, (d) - (f) scaled beliefs after 5, 6, and 7 iterations, respectively.
Extended neighborhoods as in Figure 4.2(c); ψ; $\sigma = 0.01$, $\rho = 0$.

The cause for these boundary artifacts can be revealed by having a closer look at the message update rule (3.5) and its decomposition:

$$m_{i\to j}(x_j) \leftarrow \underbrace{\rho^2 \sum_{x_i} \phi_i(x_i) \prod_{k\in\mathcal{N}_{i\setminus j}} m_{k\to i}(x_i)}_{m_{i\to j}^{min}} + \sum_{x_i : \psi_{ij}>\rho^2} \phi_i(x_i) \left(\psi_{ij}(x_i, x_j) - \rho^2\right) \prod_{k\in\mathcal{N}_{i\setminus j}} m_{ki}(x_i)$$

Due to truncation, $m_{i\to j}(x_j)$ will be at least $m_{i\to j}^{min}$, independent of the actual value of x_j. Hence, the discriminative part of the message only depends on the sum over the x_i with $\psi_{ij}(x_i, x_j) > \rho^2$. In the following, we will call these labels the *support* of x_j. The dependency of the position of a label and its support is illustrated in Figure 4.6. Since only labels within the support can add positively to $m_{i\to j}(x_j)$, labels with a larger support are more likely to be "preferred" by the messages.

Similar boundary artifacts can be observed when no truncation is used, even though all labels have full support. The sum of the $\psi_{ij}(x_i, x_j)$ over all labels x_i is still smaller if label x_j is closer to the boundary, and therefore similar effects can be observed.

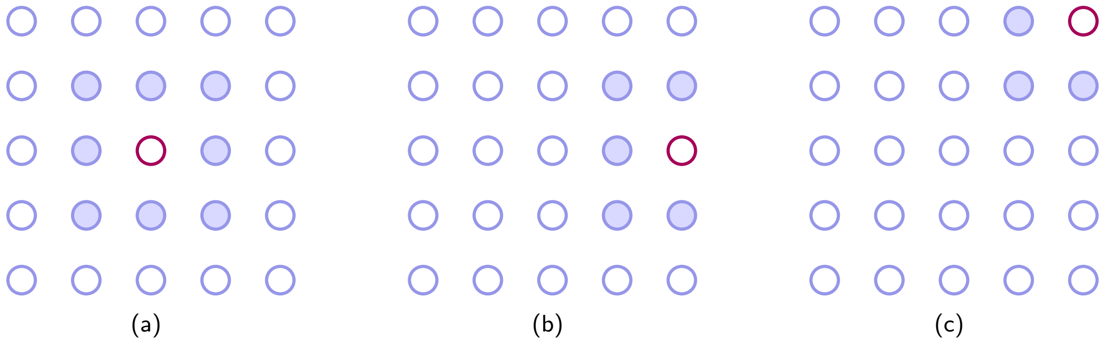


Figure 4.6: Illustration of the support of labels. Support of a label in the middle of the image (a), at the boundary (b), and at the corner (c).

4.4.1 Boundary Treatments

There are two principle approaches to tackle boundaries, either by adapting the grid or by modifying the potentials. They differ in how they interpret image boundaries. The first method interprets boundaries as features, while the second one assumes that the image is just a part of a larger scene, i.e. it is continued beyond the boundaries.

4.4.1.1 Adaptation of the Grid

A simple, yet effective approach to avoid boundary artifacts is to add edges connecting periodic neighbors to the graph, as shown in Figure 4.7.

In the case of "strict" pairwise potentials (small σ, $\rho = 0$), the success of periodical neighbors is quite obvious. Due to the strictness of the potentials, the distance between periodical neighbors must be approximately retained, forcing nodes close to the boundary to relate only to labels close to the boundary.

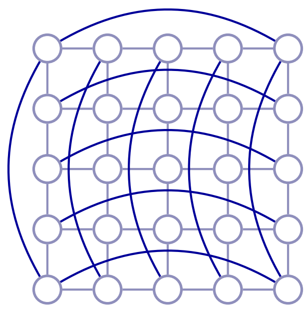
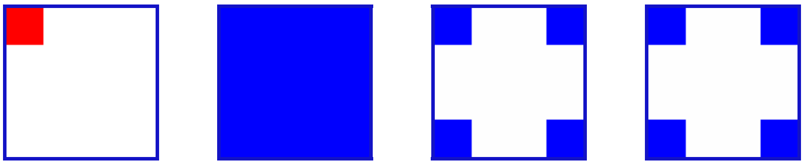


Figure 4.7: Periodic Neighbors.

Boundary artifacts are also avoided when the compatibility functions allow for variations of the original distance ($\rho > 0$). In order for variables to take values leading to violations of the distance, there must be some favorable local evidence making up for ρ^2.

Figure 4.8 demonstrates the impact of periodic neighbors. The pairwise potentials are chosen to be very strict. Hence, addition of periodic neighbors forces corner points to be mapped to corner points only.



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

Figure 4.8: Effect of periodic neighbors: (a) input image (query point marked), (b) evidence for marked point, (c) scaled beliefs after the first iteration, (d) after 2 iteration.

4.4.1.2 Modification of Potentials

The principle idea is to modify the potential, such that the sum over the contributions for all labels within the support is equal for all labels x_j, i.e. $\sum_{x_i} \widetilde{\psi_{ij}} (x_i, x_j) = k$. Intuitively, we try to predict how the image is continued beyond its boundaries.

There are different strategies to achieve this:

Uniform Scaling. The idea of uniform scaling is to give pairwise potentials for labels closer to the boundary higher weights. These scaled potentials are given by

$$\widetilde{\psi_{ij}} (x_i, x_j) = \alpha_{x_j} \psi_{ij} (x_i, x_j) \ ,$$

where α_{x_j} is a scalar factor depending on the label x_j and is typically given by

$$\alpha_{x_j} = \frac{1}{\sum_{x_i} \psi_{ij} (x_i, x_j)} \ .$$

Using uniform scaling, all labels are geometrically equally preferred. However, the symmetry property ($\psi_{ij} = \psi_{ji}$) of the pairwise potentials is usually lost when scaling is used.

Mirroring. Rather than mirroring the labels directly, which would not solve the problem since boundaries are just shifted, we modify the potentials to simulate mirroring. This is done by adding $\psi_{ij} (x_i^m, x_j)$ for all mirrored positions x_i^m of x_i to $\psi_{ij} (x_i, x_j)$. Hence, a label can be interpreted as a representative for all its mirrored positions.

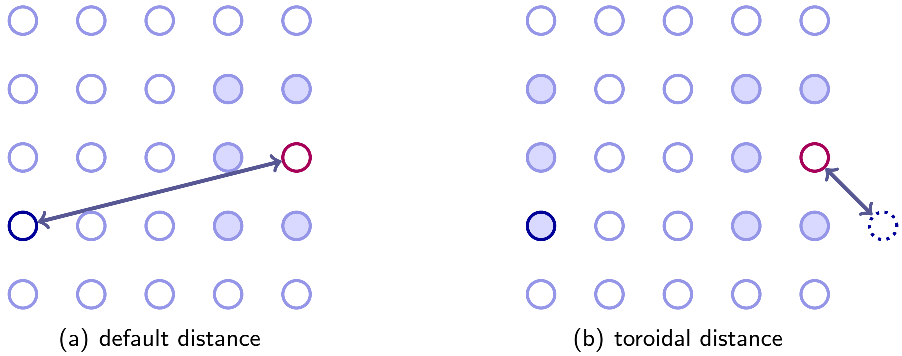


Figure 4.9: Effect of toroidal distance on the support of a label.

Toroidal Distances. We can also modify the potentials by using a different distance measure such as toroidal distances in the potentials.
Toroidal distances are given by

$$d^{toroid}(i, j) = \sqrt{(g_x(|i.x - j.x|))^2 + (g_y(|i.y - j.y|))^2},$$

where $g_x(z) = \min\{z, (width + 1) - z\}$, and $g_y(z) = \min\{z, (height + 1) - z\}$.

Figure 4.9 provides a graphical depiction of toroidal distances and demonstrates their influence on the support of labels. Intuitively, one can think of the image as a toroid (except for the distortions). Thus, it makes sense to use toroidal distances in combination with periodic neighbors.

In Figure 4.10 the marginals are computed using toroidal distance. The marginal probability is equal for all labels, since the toroidal distances allow the corner of the rigid grid to be mapped to every label.

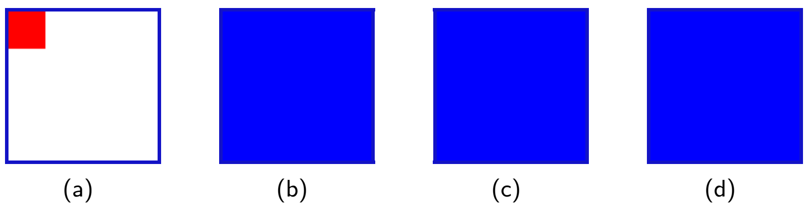


Figure 4.10: Effect of toroidal distances: (a) input image (query point marked), (b) evidence for marked point, (c) scaled beliefs after the first iteration, (d) after 2 iteration.

Please note that in the case of a completely white image, as shown above in Figure 4.10, uniform scaling and mirroring yield the same results as well.

5 Inference for Symmetry Detection

Given the MRF describing the joint distribution, we want to infer the marginals, which in turn can be used to extract symmetries of the image under consideration (cf. [17]).

When applied to graphs containing loops, belief propagation is neither guaranteed to converge, nor to compute good approximations of the marginals. Despite this fact, belief propagation has been successfully applied in various previous applications [23]. So we hope to obtain considerable approximations after a finite number of iterations. The beliefs do not have to be accurate, but have to provide a reasonable basis for symmetry extraction.

Indeed, LBP yields suitable marginals in the case of perfect symmetries. Non-synthetic images, however, typically exhibit approximate symmetries due to sampling, noise, and other means. Applying LBP to such approximate symmetries usually does not deliver the expected results.

If LBP converges to a deficient solution, we call these beliefs *pseudo marginals*. An example for pseudo marginals is shown in Figure 5.1. Similar effects can already be established with much smaller images, helping to understand what causes the shortcomings in the case of approximate symmetries.

In the remainder of this chapter we will consider a small example and compare the exact marginals with their approximations obtained by LBP. Furthermore, we will discuss the consequences of the findings and possible modifications of BP to overcome the problems.

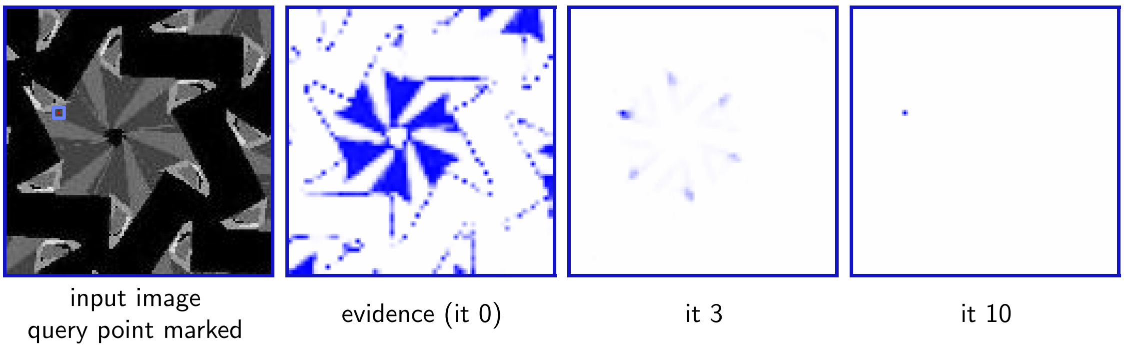
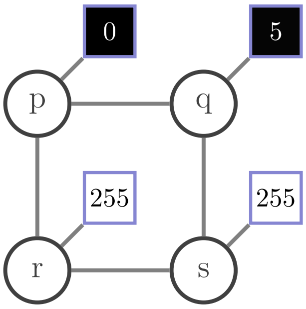


Figure 5.1: Beliefs for an image exhibiting approximate symmetries.
Beliefs drawn scaled to [0,1]; ψ: σ=1.0, ρ=0.05; periodic neighbors
Image taken from the PSU Near-Regular Texture Database [18].

5.1 A small Example

The MRF used in Figure 5.1 is rather complex, making direct examination impractical. However, the problems arising can be reproduced using a smaller MRF. Example 5.1 exhibits similar behavior as Figure 5.1. The potential functions are designed such that the exact marginals can be computed. Given the marginals, we can compare them to the beliefs obtained with LBP.



$$\psi_{ij}(kl) = \begin{cases} 1 & d(i,j) = d(k,l) \\ 0 & \text{else} \end{cases}$$

$$\phi_i(j) = \begin{cases} 1 & f_i = f_j \\ 0.9 & |f_i - f_j| \le 10 \\ 0 & else \end{cases}$$

Example 5.1: Example helping to elucidate the reasons of the shortcomings of LBP. The sets V and $\mathcal{X}$ are taken to be $\{p, q, r, s\}$.

5.1.1 Exact Solution

Since in example 5.1 the set of nodes as well as the set of labels are both small, exact computation of the marginals is possible. Due to the nature of the chosen compatibility functions, most of the 256 assignments have zero probability. Consequently, the marginals can be computed by hand, as follows:

$$\begin{aligned}
p(x_q) &= \sum_{x_p,x_r,x_s} p(x_p, x_q, x_r, x_s) \\
&= \frac{1}{Z} \cdot \sum_{x_p,x_r,x_s} \phi_p(x_p)\phi_q(x_q)\phi_r(x_r)\phi_s(x_s)\psi_{pq}(x_p, x_q)\psi_{qs}(x_q, x_s)\psi_{sr}(x_s, x_r)\psi_{rp}(x_r, x_p) \\
&= \frac{1}{Z} \cdot \Bigg[\sum_{x_r,x_s} \phi_p(p)\phi_q(x_q)\phi_r(x_r)\phi_s(x_s)\psi_{pq}(p, x_q)\psi_{qs}(x_q, x_s)\psi_{sr}(x_s, x_r)\psi_{rp}(x_r, p) \\
&\qquad + \sum_{x_r,x_s} \phi_p(q)\phi_q(x_q)\phi_r(x_r)\phi_s(x_s)\psi_{pq}(q, x_q)\psi_{qs}(x_q, x_s)\psi_{sr}(x_s, x_r)\psi_{rp}(x_r, q) \Bigg] \\
&= \frac{1}{Z} \cdot \Big[\phi_p(p)\phi_q(x_q)\psi_{pq}(p, x_q)\psi_{qs}(x_q, s) + \phi_p(q)\phi_q(x_q)\psi_{pq}(q, x_q)\psi_{qs}(x_q, r) \Big] ,
\end{aligned}$$

where the partition function is given by $Z = 1 + \phi_p(q)\phi_q(p)$.

Accordingly, the marginal distribution for X_q is given by

$$p(X_q = q) = \tfrac{1}{1+0.81} \approx 0.5525, \; p(X_q = p) = \tfrac{0.81}{1+0.81} \approx 0.4475, \text{ and } p(X_q = r, s) = 0 \,.$$

5.1.2 Beliefs

Having calculated the exact marginals, we can use them to evaluate the approximation obtained with loopy belief propagation. In order to determine the fixed point of BP, we first need to consider the message updates. Since the updating schedule only influences convergence properties [23], a parallel schedule was chosen for the computations below. By successively applying the message update rules, we get

$$\begin{aligned} m^t_{p\to q}(x_q) &= \sum_{x_p} \left[\phi_p\left(x_p\right)\psi_{p,q}\left(x_p, x_q\right) \cdot m^{t-1}_{r\to p}(x_p)\right] \\ &= \sum_{x_p} \left[\phi_p\left(x_p\right)\psi_{p,q}\left(x_p, x_q\right) \cdot \sum_{x_r} \left[\phi_r\left(x_r\right)\psi_{r,p}\left(x_r, x_p\right) \cdot m^{t-2}_{s\to r}(x_r)\right]\right], \end{aligned}$$

where $m^{t-2}_{s\to r}(x_r) = \sum_{x_s} \left[\phi_s\left(x_s\right)\psi_{s,r}\left(x_s, x_r\right) \cdot \sum_{x_q} \left[\phi_q\left(x_q\right)\psi_{q,s}\left(x_q, x_s\right) \cdot m^{t-4}_{p\to q}(x_q)\right]\right].$

When expanding the sum, we obtain

$$\begin{aligned} m^t_{p\to q}(x_q) &= \phi_p(p)\psi_{p,q}(p, x_q)\phi_r(r)\psi_{r,p}(r,p)\phi_s(s)\psi_{s,r}(s,r)\phi_q(q)\psi_{q,s}(q,s) \cdot m^{t-4}_{p\to q}(x_q) \\ &\quad + \phi_p(q)\psi_{p,q}(q, x_q)\phi_r(s)\psi_{r,p}(s,q)\phi_s(r)\psi_{s,r}(r,s)\phi_q(p)\psi_{q,s}(p,r) \cdot m^{t-4}_{p\to q}(x_q) \\ &= \left[\psi_{p,q}(p, x_q) + \psi_{p,q}(q, x_q)\phi_p(q)\phi_q(p)\right] \cdot m^{t-4}_{p\to q}(x_q) = \left[\phi_q(x_q)\right]^2 \cdot m^{t-4}_{p\to q}(x_q) \\ &= \left[\phi_q(x_q)\right]^{2\cdot\lfloor t/4\rfloor} \cdot m^{(t \bmod 4)}_{p\to q}(x_q). \end{aligned}$$

Likewise, $m^t_{s\to q}(x_q) = \left[\phi_q(x_q)\right]^2 \cdot m^{t-4}_{s\to q}(x_q) = \left[\phi_q(x_q)\right]^{2\cdot\lfloor t/4\rfloor} \cdot m^{(t \bmod 4)}_{s\to q}(x_q)$.

Due to the loopy structure of the grid, messages depend on earlier versions of themselves. As a consequence some of the evidence terms are over-used for the approximation of the beliefs:

$$b^t_q(x_q) = k \cdot \left[\phi_q(x_q)\right]^{2*2\cdot\lfloor t/4\rfloor+1} \cdot m^{(t \bmod 4)}_{p\to q}(x_q) \cdot m^{(t \bmod 4)}_{s\to q}(x_q).$$

With increasing numbers of iterations higher powers of evidence terms are used to compute the beliefs. Accordingly, the beliefs after 20 (4,16,40,100) iterations, assuming uniform initialization, are given by

$$b^{20}_q(q) = k \cdot \phi_q(q) \cdot \left[\phi_q(q)\right]^{20} \cdot m^0_{p\to q}(q) \cdot m^0_{s\to q}(q) = \tfrac{1}{1+0.9^{21}} \approx 0.9$$

$$b^4_q(q) \approx 0.629,\ b^{16}_q(q) \approx 0.857,\ b^{40}_q(q) \approx 0.987,\ \text{and}\ \ b^{100}_q(q) \approx 0.99998.$$

When using large powers of evidence terms rather than the term itself, an initial preference (i.e. highest evidence) for a label is exaggerated more and more, yielding beliefs with a single peak at that label.

Independent of the (strictly positive) initialization, the beliefs will always converge towards these pseudo marginals. Using uniform initialization, the exact marginals are only obtained after the first and second iteration:

$$\begin{aligned} b^1_q(q) &= k \cdot \phi_q(q) \cdot \left[\phi_p(p) m^0_{r\to p}(p) \cdot \phi_s(s) m^0_{r\to s}(s)\right] \frac{1}{1 + \phi_p(q)\phi_q(p)} \\ &= b^2_q(q) = k \cdot \phi_q(q) \cdot \left[\phi_p(p)\phi_r(r) m^0_{s\to r}(r) \cdot \phi_s(s)\phi_r(r) m^0_{p\to r}(r)\right] = p(X_q = q). \end{aligned}$$

5.2 Consequences for Symmetry Detection

The effects observed in the example above can be generalized for any graphs containing multiple loops. After a few iterations it is almost impossible to tell whether some information was already known before. Actual old information is mistakenly presumed to be new, and as a consequence differences are amplified more and more with increasing number of iterations. Accordingly, the beliefs tend to have single peaks when the evidence just slightly deviates which is almost compulsory for approximate symmetries.

Even though LBP performs well on perfect global symmetries, these findings have still to be taken into account. If the set of labels is sub-sampled from the image, the image is only approximated, and therefore the symmetries as well. Hence, every single pixel position has to be in $\mathcal{X}$ in order for LBP to succeed on perfect symmetries.

5.3 Possible Workarounds

Before introducing the idea that helps to circumvent the problems caused by the loopy structure of the graph, a short summary of other approaches taken into consideration is given.

For small networks, like the one shown in the example, one could quantify the amount of double counting and apply revision techniques as discussed in [36] to overcome the problem. However, for images of reasonable size the graph will contain too many cycles of different lengths.

One could also think of discretizing the evidence, but it is hard to tell how to do that exactly without knowing the given image in detail.

Generalized belief propagation [38] yields better approximations on loopy graphs by sending messages between groups of nodes. This requires to sum over several variables at once, which is feasible if the variables range over only few values (in our setting, $\mathcal{X}$ is usually large).

Tree-reweighted belief propagation [34] takes the opposing message as well as edge appearances into account. Furthermore, this approach has been successfully applied for nonrigid image registration [16].

Tree-reweighted BP uses modified message update rule

$$m_{i\to j}^{t+1}(x_j) = \sum_{x_i} \phi_i(x_i) \left[\psi_{i,j}(x_i, x_j)\right]^{\frac{1}{\mu_{ji}}} \frac{\prod_{k\in\mathcal{N}_{i\setminus j}} \left[m_{k\to i}^{t}(x_i)\right]^{\mu_{ki}}}{\left[m_{j\to i}^{t}(x_i)\right]^{1-\mu_{ij}}},$$

where μ_{ij} is the edge appearance probability of $\{i, j\} \in E(T)$ over all spanning trees.

Unfortunately, this approach did not succeed in computing appropriate beliefs for the example above (cf. Figure 5.4), and therefore, was not considered further.

5.4 Approximation of Beliefs using Spanning Trees

Our goal is to obtain good approximations to the marginal distributions

$$p(x_k) = \frac{1}{Z} \sum_{\mathbf{x}';x'_k=x_k} \prod_i \phi_i(x_i) \prod_{i\sim j} \psi_{ij}(x'_i, x'_j)\,.$$

Unfortunately, the results obtained by standard LBP are not satisfactory when considering approximate symmetries. However, the first iterations of LBP are quite promising when using a tree-based update schedule (i.e. only update messages along random trees). See Figure 5.2 for an example.

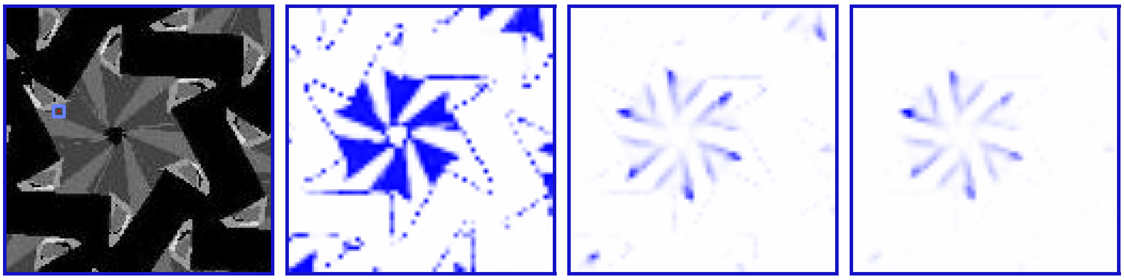


Figure 5.2: Initial iterations when using a tree-based update schedule.

Let $T = (V, E_T)$ be a spanning-tree of $G = (V, E)$. Consider the MRF with T as the underlying graph. The corresponding marginal distribution of X_k is given by

$$p^T(x_k) = \frac{1}{Z_T} \sum_{\mathbf{x}';x'_k=x_k} \prod_i \phi_i(x_i) \cdot \prod_{\{i,j\}\in E_T} \psi_{ij}(x_i, x_j)\,.$$

Recall that BP computes exact marginals on tree-structured graphs; and hence, we can apply BP to compute $p^T(x_k)$. Furthermore, BP can be used to compute $Z_T \cdot p^T(x_k) = \tilde{p}^T(x_k)$ if no normalization is employed.

Since $\psi_{ij}(x_i, x_j) \in [0, 1]$ and $\phi_i(x_i) \geq 0$, we get:

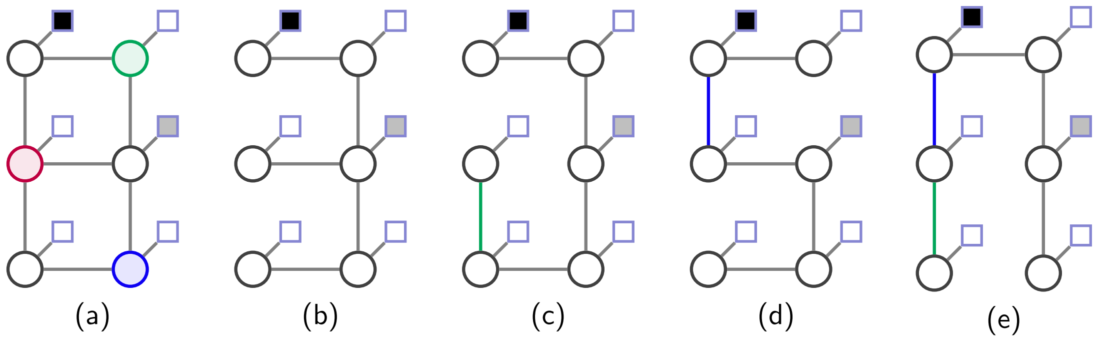
$$\begin{aligned}
\tilde{p}^T(x_k) &= \sum_{\mathbf{x}';x'_k=x_k} \prod_i \phi_i(x'_i) \cdot \prod_{\{i,j\}\in E_T} \psi_{ij}(x'_i, x'_j) \\
&\geq \sum_{\mathbf{x}';x'_k=x_k} \prod_i \phi_i(x'_i) \cdot \prod_{\{i,j\}\in E_T} \psi_{ij}(x'_i, x'_j) \cdot \prod_{\{i,j\}\in E\setminus E_T} \psi_{ij}(x'_i, x'_j) \\
&= \sum_{\mathbf{x}';x'_k=x_k} \prod_i \phi_i(x'_i) \cdot \prod_{\{i,j\}\in E} \psi_{ij}(x'_i, x'_j) \;=\; Z \cdot p(x_k)
\end{aligned}$$

Thus, we can use the un-normalized marginals of spanning trees as sound over-approximations for the actual un-normalized marginals of the graph. We will refer to beliefs approximated by spanning tree marginals as *spanning tree beliefs*.

5.4.1 Minimal Spanning Tree Beliefs Approach

When approximating marginals by using a spanning tree, we simply "forget" some compatibility terms (corresponding to the deleted edges). In case of good compatibilities (i.e. $\psi_{ij}(x_i, x_j) \approx 1$), this has hardly any effect on the result. However, when discarding low compatibilities some marginal probabilities get over-approximated. We can interpret the edges corresponding to these low compatibilities $\psi_{ij}(x_i, x_j)$ as a contradiction to the respective assignments $(X_i = x_i, X_j = x_j)$. Thus, we hope to find a spanning tree containing most of these contradictory edges.

Consider example 5.2. We wish to approximate the marginal distribution of the red marked variable. Using the spanning tree shown in (b), the marginal probability for the green and blue labels are approximated insufficiently. Spanning trees (c) and (e) yield good approximations for the green label. Proper approximations for the blue label are obtained when using (d) or (e). Please note that in more complex cases a "perfect" spanning tree, like (e), yielding good results for all labels might not always exist; hence, we use individual spanning trees for each label.



Example 5.2: Example illustrating the concept of contradictory edges. When assigning the blue label to the red node, the blue edge in (d) and (e) yields a contradiction (low probability of this assignment). Analogously for green.

Algorithm 5.1: Minimal Spanning Tree Beliefs Algorithm

Input: MRF with $G = (V, E)$; number N of spanning trees

1. initialize beliefs: $b(x_k) \leftarrow 1$
2. **for** $i = 0$ to N **do**
3. randomly pick spanning tree T_i of G
4. calculate un-normalized beliefs $\tilde{b}^{T_i}(x_k)$
5. **for all** X_k, $x_k \in \mathcal{X}$ **do**
6. $b(x_k) \leftarrow \min\{b(x_k), \tilde{b}^{T_i}(x_k)\}$
7. **end for**
8. **end for**
9. normalize beliefs

These observations can be used to derive a new strategy for the approximation of marginals. The basic idea is sketched in Algorithm 5.1. For a set of N spanning trees $\{T_i\}_{i \in N}$, all corresponding un-normalized marginals are computed. For each variable and each label of the original graph, the un-normalized belief is given by the minimum over all respective beliefs:

$$b(x_k) = l \cdot \min_{i \in N} \left\{ p^{T_i} \right\} ,$$

where l is a normalizing constant.

Since the beliefs are approximated by the minimum over a set of spanning tree beliefs, we refer to this strategy as the *minimal spanning tree belief* (MSTB) approach.

An example for an application of the MSTB Algorithm is given in Figure 5.3.

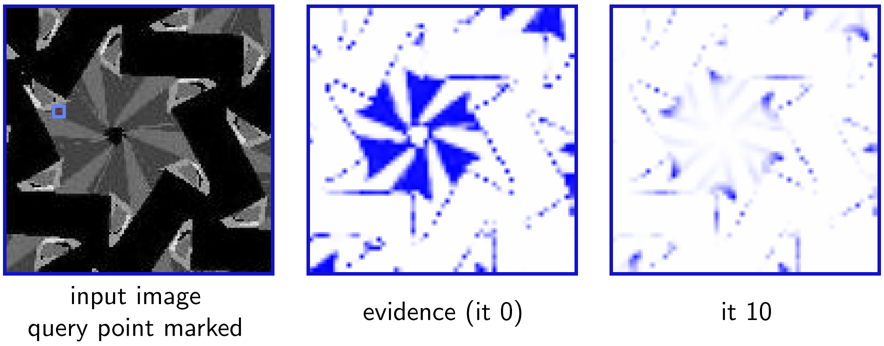


Figure 5.3: Beliefs obtained by the MSTB approach.

We conclude this chapter by returning to the small example considered in section 5.1. Approximations obtained using different approaches are compared in Figure 5.4. Only the beliefs calculated by the MSTB approach capture the marginals sufficiently.

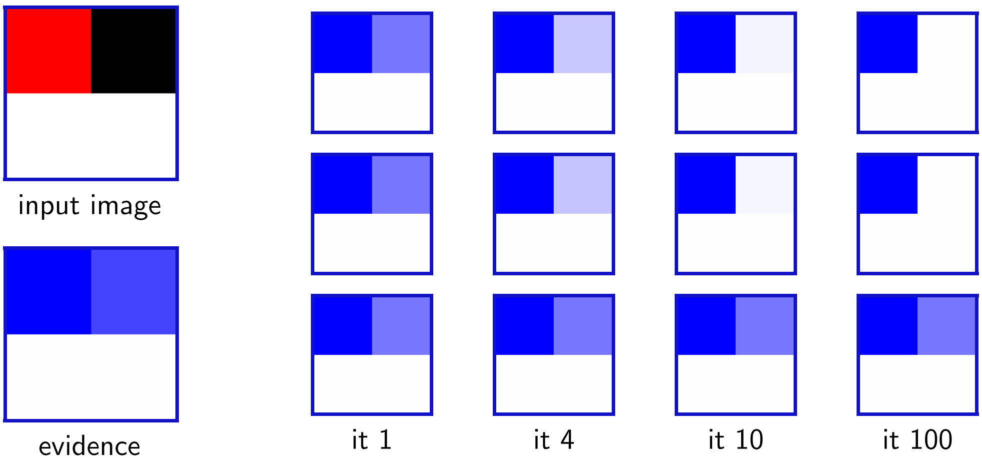


Figure 5.4: Comparison of different methods for approximation of marginal distributions. first row: LBP (parallel schedule); second row: TRW-LBP; third row: MSTB.

6 Results and Discussion

For purpose of evaluation, both versions of the approach were applied to a set of test images ranging from synthetic images to real photographs. Except for some small bitmap images, all pictures were taken from the PSU Near-Regular Texture Database [18] and converted to gray-value images.[1]

Given the beliefs of variable node X_i, we represent the beliefs by adapting the saturation (in HSV color space) of the associated pixels accordingly. The saturation of the pixel corresponding to x_i is given by

$$255 \cdot b(x_i) \, .$$

For pixels not contained in the set of labels, the saturation is determined by linear interpolation between labels in close proximity. A strong belief ($b(x_i) \approx 1$) is represented by a saturated blue, while pixels with low beliefs are drawn white.

Since the beliefs are approximations to marginal distributions, they must sum up to one. Hence, for reasonably sized sets of labels, the beliefs are typically very small, resulting in almost white images. Thus, the beliefs need to be transformed before visualization. This can be achieved by *gamma correction* [35] or by mapping to the unit interval.

Using gamma correction, the saturation for pixel x_i is given by

$$255 \cdot [b(x_i)]^{\frac{1}{\gamma}} \, ,$$

where γ typically takes values between 2 and 5, depending on the number of labels. For $\gamma > 1$, gamma correction is a concave mapping, spreading smaller values while compressing larger ones. In Figure 6.1 $[b(x_i)]^{1/\gamma}$ is plotted for different values of gamma. Thus, gamma correction can be used to uncover small-scale details. However, γ needs to be chosen appropriately depending on the image under consideration. Furthermore, the actual differences between beliefs get distorted when using gamma correction.

Let $b_{min} = \min\limits_{x_i \in \mathcal{X}}$ and $b_{max} = \max\limits_{x_i \in \mathcal{X}}$ be the smallest and largest belief of X_i.
We map the interval $[b_{min}, b_{max}]$ to the unit interval $[0, 1]$ by mapping $b(x_i)$ to:

$$\frac{b(x_i) - b_{min}}{b_{max} - b_{min}} \, .$$

Given the rare case that $b_{min} = b_{max}$, we set every belief $b(x_i)$ to 1.

[1] The approach is not restricted to gray-value images in general, but to handle color images appropriate descriptors are needed.

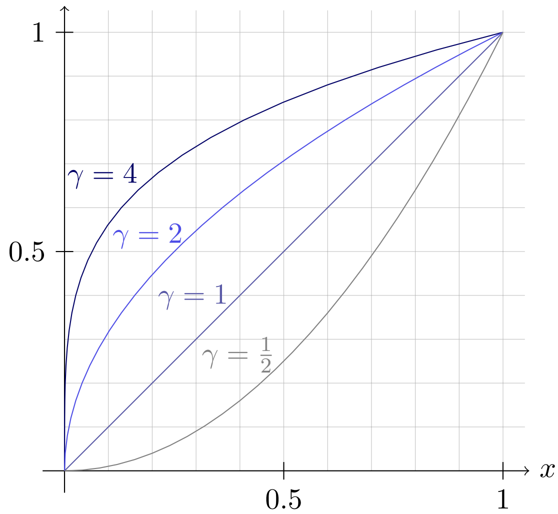


Figure 6.1: Gamma correction.

Typically, this simple transformation suffices to make the beliefs visually perceptible without adjusting an additional parameter. Furthermore, all relative differences between the beliefs are preserved by this mapping.

Hence, if not noted otherwise, the beliefs will be presented after transformation to the unit interval.

6.1 Perfect Symmetries

To asses the quality of approximations obtained, we will first regard simple perfect symmetries allowing to predict the actual marginals.

Consider the perfect global symmetry given in Figure 6.2. When using very strict compatibility functions, LBP as well as the MSTB approach compute beliefs agreeing with the actual marginals.

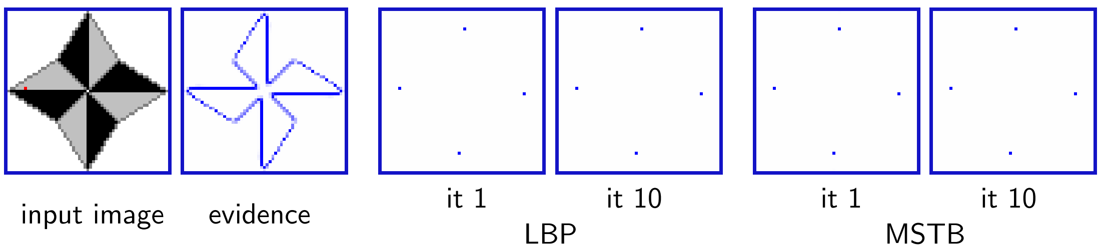


Figure 6.2: Example exhibiting perfect, global symmetries. Both variants agree on their beliefs. ψ: σ=0.01; ρ=0; extended neighborhood; periodic neighbors.

The MRF used in Figure 6.2 only allows for rigid mappings of the grid. However, using such an MRF already assumes that the image under consideration contains perfect global symmetries. Furthermore, when considering a completely rigid grid the marginals can be computed exactly since most assignments have zero probability. Accordingly, there is no need for approximations of the marginals.

When not using prior knowledge, and accordingly, compatibility functions allowing for more flexibility, the approximation techniques should still be able to recognize the symmetric structure of the image. Using a less strict compatibility function should result in four high peaks corresponding to the actual instances of the point, as well as some smaller peaks in close proximity to these points. Beliefs obtained using such a compatibility function are shown in Figure 6.3. The beliefs computed by LBP are more localized than the expected marginal distribution, while the beliefs obtained by the MSTB approach are much less localized.

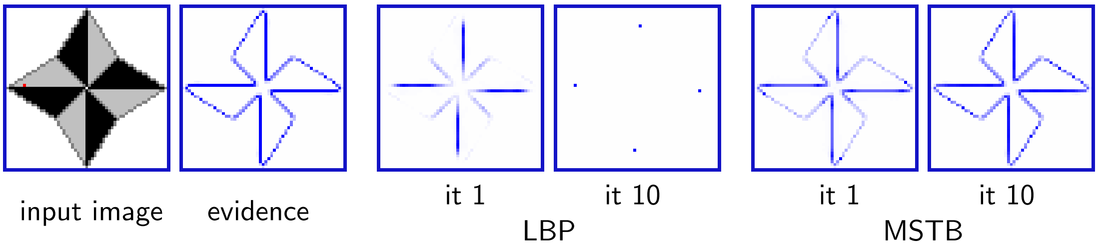


Figure 6.3: Example exhibiting perfect, global symmetries. Using a compatibility function allowing for more flexibility: σ=0.2; ρ=0.1

Since not all compatibilities terms are considered when using spanning trees to approximate the beliefs, discontinuities of the geometric consistency are much "cheaper" than actually modeled by the MRF. Consequently, marginals approximated using the MSTB approach tend to be less localized than specified.

Even though LBP does not approximate the marginals correctly, the approach still captures all symmetric structures within the image. However, when considering local symmetries, the reason underlying these very localized beliefs leads useless beliefs.

The beliefs computed by LBP shown in Figure 6.4 converge towards pseudo marginals even though all pixels of the image are used as labels (in order to introduce no approximation by sampling). The pairwise potential do no allow for variations in the local structure (σ=0.01), however, for some global discontinuities (ρ=0.1). Accordingly, one would expect beliefs similar to those obtained after 20 iterations in Figure 6.4.

The reasoning used to explain the cause of these artifacts is very similar to the one used in chapter 5. Due to the loopy structure of the underlying graph, the messages also depend on earlier versions of themselves. This does not only lead to double counting of some evidence terms (as seen in chapter 5), but also to over-usage of compatibility terms. It might be interesting to note that the beliefs did not converge towards the identity mapping in Figure 6.4, but to some other symmetric instance of the point.

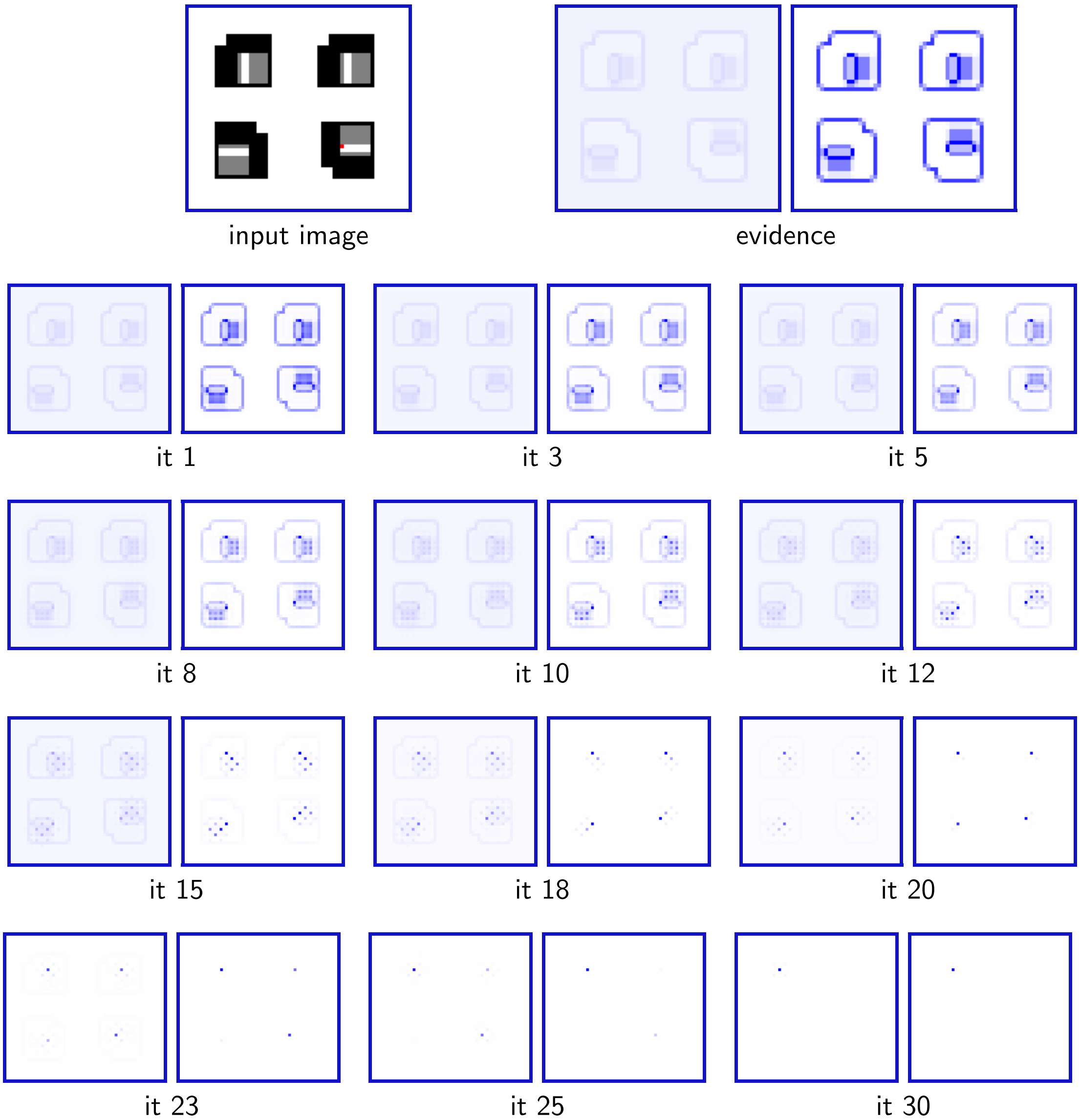


Figure 6.4: Perfect, local symmetries. The beliefs computed by LBP converge to pseudo marginals. The left panel of each iteration shows beliefs after applying gamma correction (γ=3). Compatibility functions: σ:0.01; ρ:0.1; periodic neighbors.

Figure 6.5 shows the results received when using the MSTB algorithm rather than LBP. However, when using the settings as in Figure 6.4, the beliefs are not very localized. Due to missing some compatibilities when approximating the belief by spanning trees, truncation becomes more likely (some of the ρ^2 are simply left out in the computation). Using a smaller truncation parameter helps to compensate for this effect to some extend by making geometric discontinuities less preferred (see Figure 6.5, right frames).

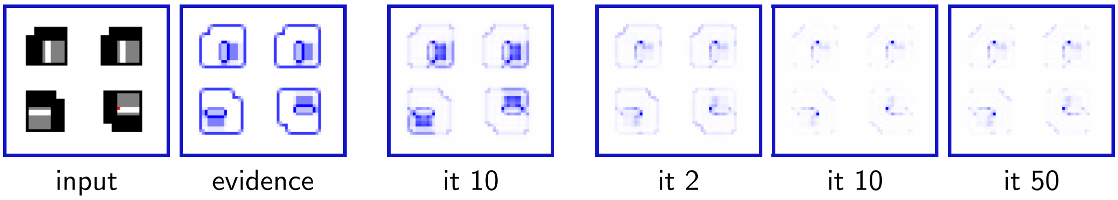


Figure 6.5: Perfect, local symmetries. Beliefs obtained using the MSTB approach. Beliefs shown in the middle are computed using the same setting as in Figure 6.4. The three frames on the right were calculated using less truncation: ρ=0.01.

6.2 Approximate Symmetries in Synthetic Scenes

Next, we will evaluate the performance of the approach on images containing symmetries which only slightly deviate from perfect symmetries.

Figure 6.6 provides another example of typical pseudo marginals obtained when using LBP on approximate symmetries. Using the MSTB approach rather than LBP in case of approximate symmetries yields much better results, as shown in Figure 6.7.

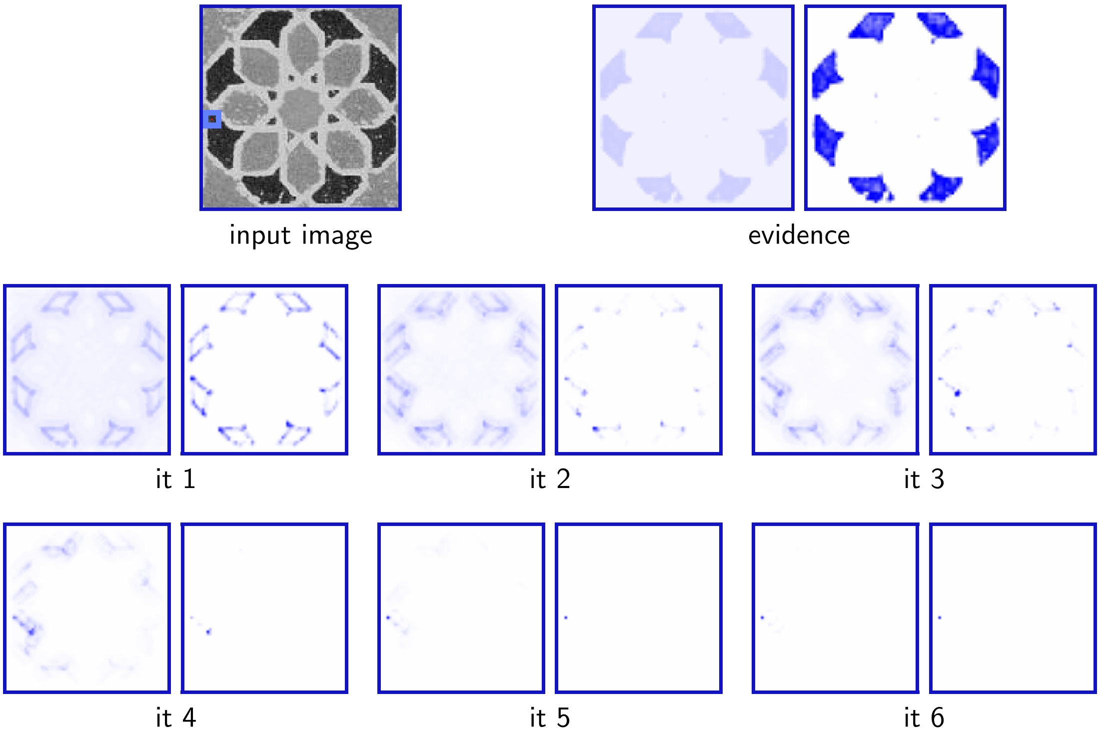


Figure 6.6: Image containing a global approximate symmetry. Pseudo marginals given by LBP. Left panels after gamma correction, γ=4. ψ: σ:0.4; ρ:0.01

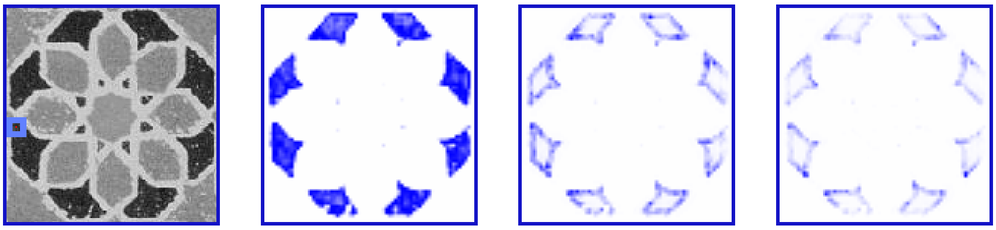


Figure 6.7: Image containing a global approximate symmetry. MSTB approach used.

Considering the beliefs after different numbers of iterations of the MSTB approach (given in Figure 6.8), two aspects typical for this approach can be observed:

1. Many iterations of the MSTB algorithm do not yield a better approximation.
2. Sometimes the beliefs seem to get worse, e.g. it 10 compared to 13.

The first aspect is caused by the naive sampling of spanning trees. By just drawing a random sample, there is no guaranty that this spanning tree will contribute to a better approximation of the beliefs. While the second is just a matter of representation. The MSTB approach tries to minimize the un-normalized versions of the beliefs. Whenever a spanning tree yields better (smaller) approximations for some of the un-normalized beliefs, their total sum is decreased as well. Accordingly, the normalized versions of beliefs that have not been updated are larger than before even though the approximation has been improved.

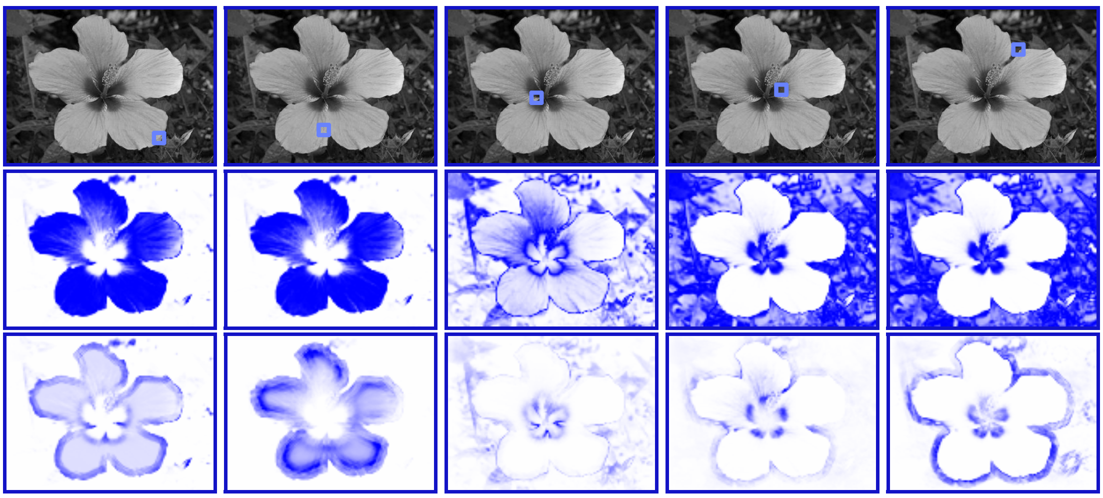


Figure 6.9: Picture exhibiting approximate symmetries. Beliefs after 25 iterations using the MSTB approach (row 3) for different query points (row 1).
Evidence as shown in row 2; compatibility functions: σ=0.2; ρ=0.01.

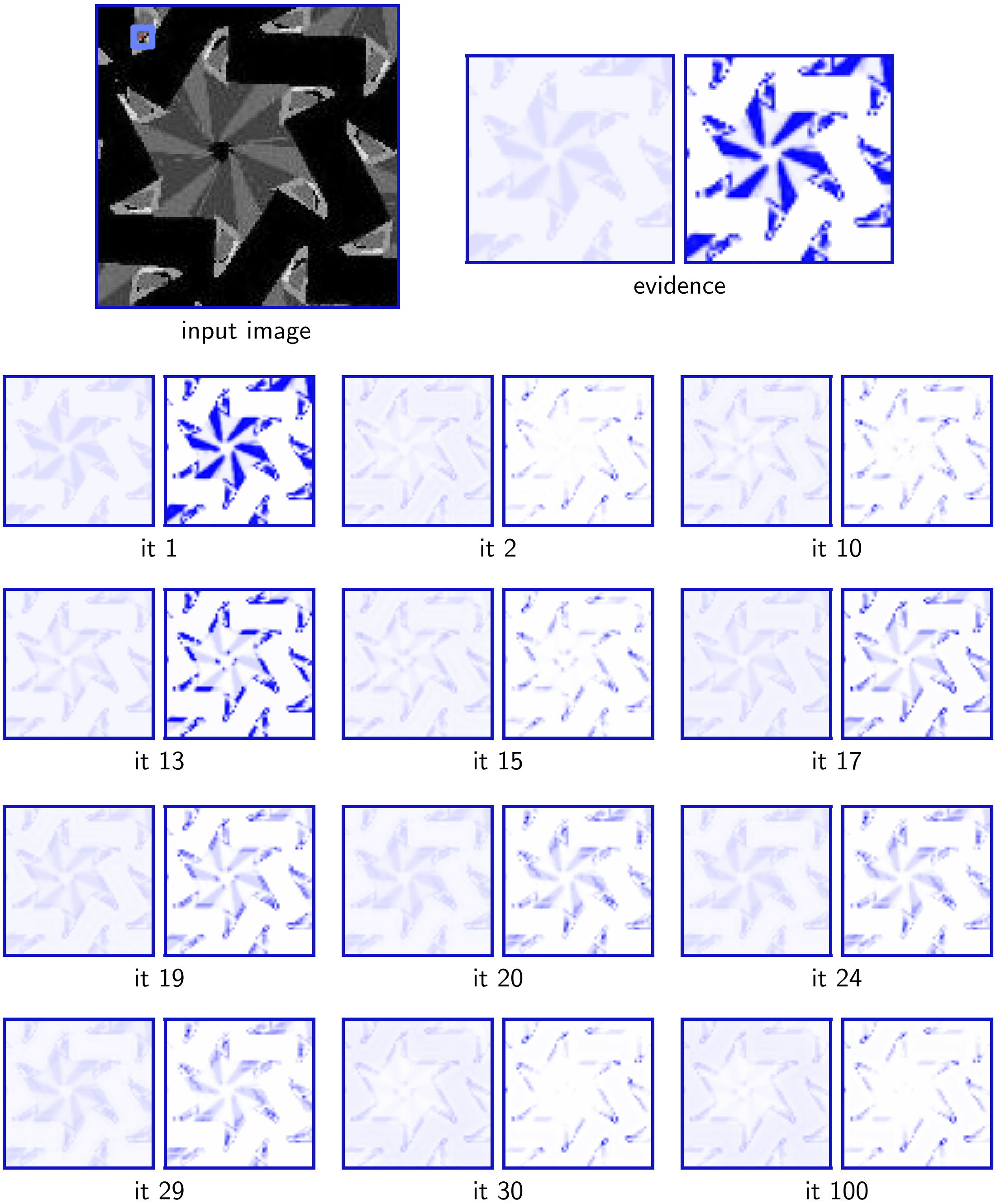


Figure 6.8: Image exhibiting approximate symmetries. Several iterations of the MSTB approach depicted. Left panels: γ=3.3. Compatibility functions: σ=0.05; ρ=0.01.

6.3 Approximate Symmetries in Real-World Images

Finally we turn to symmetric structures in real-world images. Figures 6.9-6.11 provide examples of application of the MSTB approach to real photographs in increasing order of difficulty. As in the previous examples, the beliefs tend to be more flexible than the actual marginals. However, our goal is to obtain beliefs providing good evidence for potential symmetries rather than calculating perfect approximations to the marginals

In the scope this thesis we only focused on obtaining suitable approximations to the marginals. The next step is to extract the actual symmetries as described by Lasowski et al. in [17]. When extracting symmetries based on (approximated) marginal distributions, peaks of the marginal distribution of some variable are used as seeds. The regions are grown if the marginals of neighboring variable nodes agree with these peaks (i.e. distance of nodes is approximately maintained by a pair of peaks).

Hence, in order for this extraction to succeed, it suffices if the beliefs for some nodes are properly localized. Examples for fairly well localized beliefs can be observed in Figure 6.9 (middle) or Figure 6.10 (in both panels on the right).

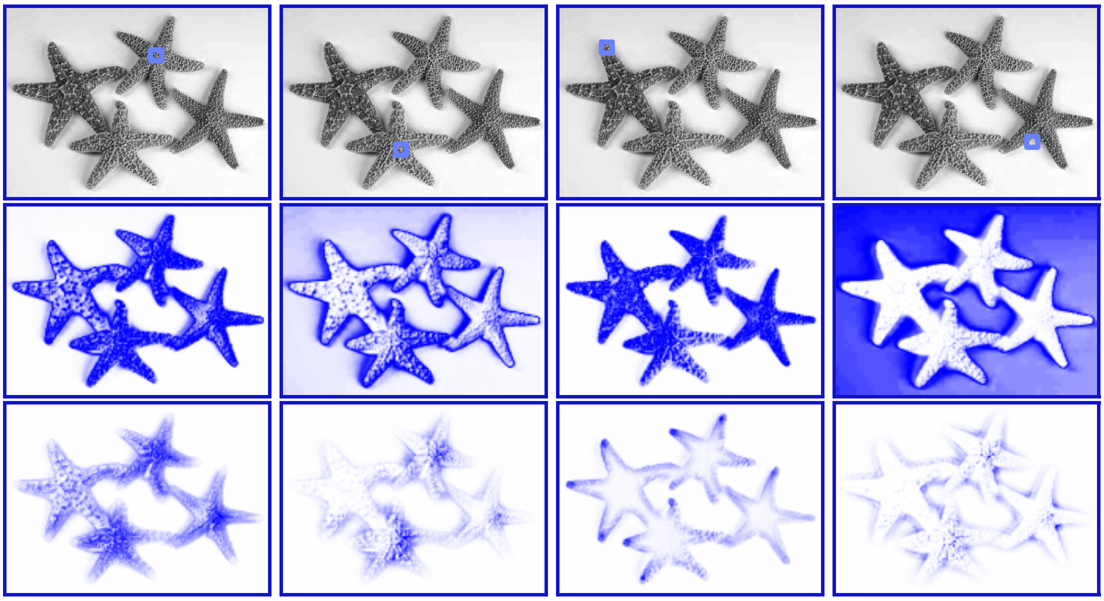


Figure 6.10: Approximate symmetries with different degrees of similarity. Using 50 iterations of the MSTB algorithm to approximate the marginals. 1^{st} row: query points, 2^{nd} row: evidence, 3^{rd} row: beliefs. Compatibility functions: σ=0.2; ρ=0.01.

Even though the picture shown in Figure 6.11 contains perspective skew, the different regions of the flowers are roughly recognized. The correct marginals would only point at symmetries with similar size. Hence, when considering images with perspective skew, the (unwanted) flexibility of the approximation can turn out to be beneficial.

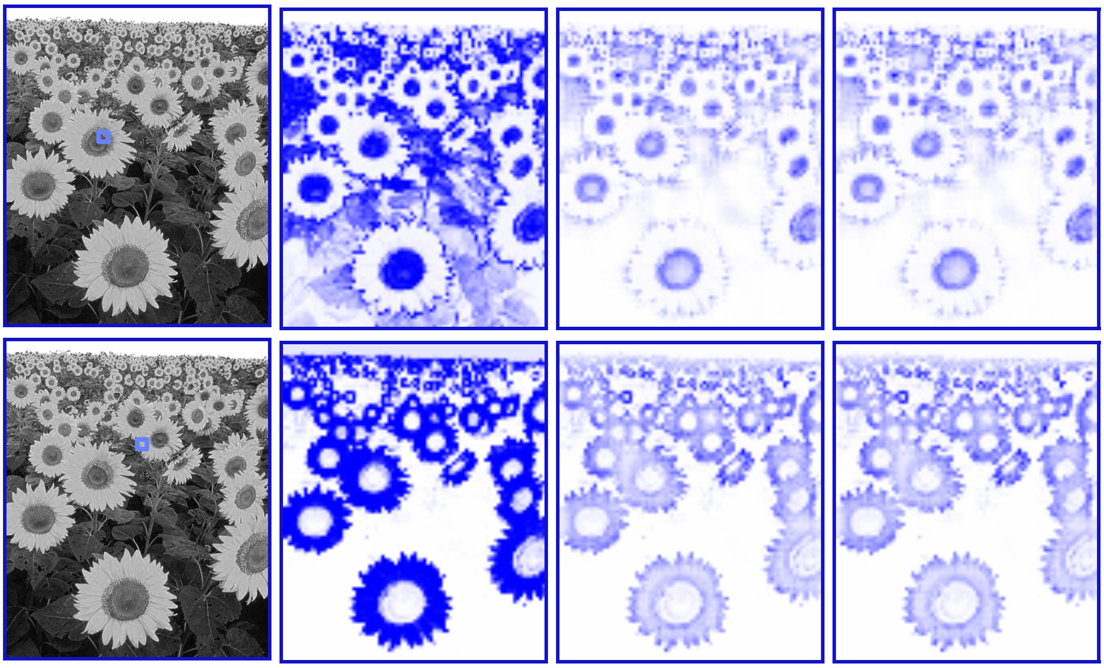


Figure 6.11: Photograph containing symmetric structures under perspective skew. 50 MSTB iterations used. ψ: σ=0.01; ρ=0.01.

6.4 Computation Times

We end this chapter by having a look at the time needed to compute beliefs. All times were measured on a 3.16GHz machine with 7.87GB of RAM using a C++ implementation of the approach without any further optimizations.

Tables 6.1-6.3 show the effect of the different parameters on computation time. The MSTB approach is typically two times faster than ordinary LBP.

When using larger neighborhoods or more nodes, the time needed by LBP scales with approximately the same factor since the number of messages is increased (cf. table 6.1). For the MSTB approach, only the size of the variable nodes has a significant influence on computation time.

Increasing the number of labels, increases the computation time needed drastically as shown in table 6.2. Hence, the number of labels used is currently a limiting factor of our approach. Please note that although table 6.2 uses no truncation, not all pairs of labels are considered in the current implementation since compatibilities less than 10^{-30} are discarded (assumed to be zero).

Table 6.3 illustrates the effect of σ and ρ on computation time. Increasing σ and/or decreasing ρ requires longer computation times.

$\lvert\mathcal{N}\rvert$	$\lvert V\rvert$	$\lvert\mathcal{X}\rvert$	MSTB	LBP
≤ 16	256	1024	21.234s	109.28s
≤ 8	256	1024	18.390s	66.265s
≤ 4	256	1024	16.218s	30.656s
≤ 4	64	1024	5.204s	8.782s
≤ 4	16	1024	1.781s	2.906s

Table 6.1: Effect of the numbers of messages on computation time.
σ=1.0; ρ=0.1

$\lvert V\rvert$	$\lvert\mathcal{X}\rvert$	MSTB	LBP
64	64	0.140s	0.312s
64	256	0.799s	1.329s
64	1024	8.782s	15.203s
64	4096	232.186s	371.95s

Table 6.2: Effect of the numbers of labels on computation time.
$\lvert\mathcal{N}\rvert \leq 4$; σ=1.0; ρ=0

σ	ρ	MSTB	LBP
0.1	0.1	48.253s	90.063s
1.0	0.1	55.219s	106.624s
1.0	0.01	62.015s	119.078s
1.0	0	162.735s	317.140s
10.0	0.1	845.489s	1533.85s

Table 6.3: Effect of σ and ρ on computation time.
$\lvert\mathcal{N}\rvert \leq 4$; $\lvert V\rvert$=$\lvert\mathcal{X}\rvert$=1024

7 Conclusion

In this thesis we have presented a general approach to symmetry detection in images. Using a Markov random field (MRF), we model a joint probability describing the self-similarity of the image. The MRF uses a graph to depict the variables which should behave geometrically consistently. In order to extract locations of potential symmetries from this joint probability distribution, we refer to the marginals of its variables.

The first version of the approach uses loopy belief propagation (LBP) to obtain approximations of the marginals. As we have seen, LBP only provides sufficient approximations when considering perfect symmetries. Due to the very loopy structure of the graph used by the MRF, LBP tends to exaggerate initial preferences for certain values of the variables when applied to approximate symmetries.

Motivated by the shortcomings of LBP, we developed a novel variant of belief propagation: the minimal spanning tree beliefs (MSTB) approach. By calculating approximations based on spanning trees of the original graph, the MSTB approach bypasses the problems encountered by LBP.

Even though the marginals can not always be reliably approximated, we were able to demonstrate that a simple probabilistic description of similarity can be used for symmetry detection.

7.1 Future Work

As we have seen in the previous chapter, a major drawback of the approach proposed in this thesis is the required computational effort limiting the practicability of the approach. Thus, an implementation on graphics hardware could help to accelerate computations making the approach more applicable to real-world imaged. In addition, both versions of the approach are memory intensive. For every variable node the corresponding messages and beliefs, each containing an entry for every label, have to be stored. Using a more sparse representation of messages and beliefs could help to reduce the required memory and at the same time accelerate computations.

Even though the results obtained by the MSTB approach are typically superior to those acquired by using LBP, there is still room for improvement in this approach.

Currently the spanning trees used in the MSTB algorithm are randomly sampled from the space of all spanning trees. This naive sampling also yields spanning trees not contributing to a better approximation of the marginals. Consequently, there is no inherent stopping criteria for this approach. In order to make some assumptions about convergence, it might be necessary to keep track of the spanning trees used in prior iterations. Furthermore, the spanning trees typically allow for much more flexibility, and hence, approximated marginals are typically not as localized as intended. Requiring that at least one spanning tree contains all edges directly connected to the variable node of interest, might already help to obtain better results. One could also think of ways to group edges, (similar to [37] where nodes are grouped) and ensure that each spanning tree contains at least one representative of each group. Hence, future work in these directions would consider different strategies to draw samples from the space of all spanning trees.

However, in order to obtain significantly better approximations, other approaches for computation of marginal distributions need to be considered.

Bibliography

[1] D. Anguelov, P. Srinivasan, H.-C. Pang, D. Koller, S. Thrun, and J. Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 33–40. MIT Press, Cambridge, MA, 2005.

[2] A. Berner, M. Bokeloh, M. Wand, A. Schilling, and H.-P. Seidel. Generalized intrinsic symmetry detection. Research Report MPI-I-2009-4-005, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, August 2009.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*, chapter 8. Graphical Models. Springer-Verlag, 2006.

[4] P. Clifford. *Disorder in Physical Systems: A Volume in Honour of John M. Hammersley*, chapter Markov Random Fields in Statistics. Oxford University Press, 1990.

[5] G. F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks (research note). *Artif. Intell.*, 42(2-3):393–405, 1990.

[6] H. Cornelius and G. Loy. Detecting rotational symmetry under affine projection. *Pattern Recognition, International Conference on*, 2:292–295, 2006.

[7] G. D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[8] R. G. Gallager. *Low-Density Parity-Check Codes.* MIT Press, 1963.

[9] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Journal of Applied Statistics*, pages 452–472, 1984.

[10] Y. Gofman and N. Kiryati. Detecting symmetry in grey level images: The global optimization approach. *Pattern Recognition, International Conference on*, 1:889, 1996.

[11] J. M. Hammersley and P. Clifford. Markov field on finite graphs and lattices. `http://www.statslab.cam.ac.uk/~grg/books/hammfest/hamm-cliff.pdf`, 1971.

[12] O. Ibe. *Markov processes for stochastic modeling.* Elsevier Academic Press, 2009.

[13] E. Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei*, 31(1):253–258, 02 1925.

[14] R. Kindermann and J. L. Snell. *Markov Random Fields and Their Applications.* American Mathematical Society, 1980.

[15] F. Kschischang, S. Member, B. J. Frey, and H. andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519, 2001.

[16] D. Kwon, K. J. Lee, I. D. Yun, and S. U. Lee. Nonrigid image registration using dynamic higher-order mrf model. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 373–386, Berlin, Heidelberg, 2008. Springer-Verlag.

[17] R. Lasowski, A. Tevs, H.-P. Seidel, and M. Wand. A probabilistic framework for partial intrinsic symmetries in geometric data. In *IEEE International Conference on Computer Vision (ICCV'09)*, page toappear, Koyoto, Japan, 2009. IEEE Computer Society.

[18] S. Lee and Y. Liu. Psu near-regular texture database. `http://vivid.cse.psu.edu`, 2005.

[19] S. Li. *Markov Random Field Modeling in Computer Vision.* Springer-Verlag, 1995.

[20] Y. Liu, J. H. Hays, Y.-Q. Xu, and H.-Y. Shum. Digital papercutting. In *Technical Sketch, SIGGRAPH.* 2005.

[21] G. Loy and J. olof Eklundh. Detecting symmetry and symmetric constellations of features. In *In ECCV*, pages 508–521, 2006.

[22] N. J. Mitra, L. J. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3d geometry. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 560–568, New York, NY, USA, 2006. ACM.

[23] J. M. Mooij. *Understanding and Improving Belief Propagation.* PhD thesis, Radboud University Nijmegen, May 2008.

[24] K. Nakanishi. Two- and three-spin cluster theory of spin-glasses. *Phys. Rev. B*, 23(7):3514–3522, Apr 1981.

[25] M. Ovsjanikov, J. Sun, and L. Guibas. Global intrinsic symmetries of shapes. *Computer Graphics Forum*, 27(5):1341–1348, July 2008.

[26] M. Park, S. Lee, P. chun Chen, S. Kashyap, A. A. Butt, and Y. Liu. Performance evaluation of state-of-the-art discrete symmetry detection algorithms. In *in Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[27] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. Guibas. Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics*, 27(3):#43, 1–11, 2008.

[28] J. Pearl. Bayesian networks: A model of self-activated memory for evidential reasoning. In *Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine*, pages 329–334, August 1985.

[29] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[30] V. S. N. Prasad and L. S. Davis. Detecting rotational symmetries. *Computer Vision, IEEE International Conference on*, 2:954–961, 2005.

[31] D. Raviv, A. Bronstein, M. Bronstein, and R. Kimmel. Symmetries of non-rigid shapes. pages 1–7, 2007.

[32] T. Tuytelaars, A. Turina, and L. V. Gool. Noncombinatorial detection of regular repetitions under perspective skew. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:418–432, 2003.

[33] M. Wainwright and M. Jordan. Graphical Models, Exponential Families, and Variational Inference. In *Foundations and Trends in Machine Learning* [34], pages 1–305.

[34] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree-reweighted belief propagation algorithms and approximate ml estimation via pseudo-moment matching. In *Workshop on Artificial Intelligence and Statistics*, January 2003.

[35] J. Weickert. Image processing and computer vision. lecture notes, 2009.

[36] Y. Weiss. Belief propagation and revision in networks with loops. Technical report, 1997.

[37] L. Xiong, F. Wang, and C. Zhang. Multilevel belief propagation for fast inference on markov random fields. In *ICDM '07: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 371–380, Washington, DC, USA, 2007. IEEE Computer Society.

[38] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *IN NIPS 13*, pages 689–695. MIT Press, 2000.

[39] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. pages 239–269, 2003.