



CENTER FOR
MACHINE PERCEPTION



CZECH TECHNICAL
UNIVERSITY

RESEARCH REPORT

ISSN 1213-2365

Curvature Prior for MRF-based Segmentation and Shape Inpainting

(Version 1.0)

Alexander Shekhovtsov¹

Pushmeet Kohli²

Carsten Rother²

shekhole@fel.cvut.cz pkohli@microsoft.com carrot@microsoft.com

¹Center for Machine Perception, K13133 FEE Czech Technical University

²Microsoft Research Cambridge

CTU-CMP-2011-11

September 8, 2011

A. Shekhovtsov was supported by EU project FP7-ICT-247870 NIFTi.

Research Reports of CMP, Czech Technical University in Prague, No. 11, 2011

Published by

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University

Technická 2, 166 27 Prague 6, Czech Republic

fax +420 2 2435 7385, phone +420 2 2435 7637, www: <http://cmp.felk.cvut.cz>

Abstract

Most image labeling problems such as segmentation and image reconstruction are fundamentally ill-posed and suffer from ambiguities and noise. Higher order image priors encode high level structural dependencies between pixels and are key to overcoming these problems. However, these priors in general lead to computationally intractable models. This paper addresses the problem of discovering compact representations of higher order priors which allow efficient inference. We propose a framework for solving this problem which uses a recently proposed representation of higher order functions where they are encoded as lower envelopes of linear functions. Maximum a Posterior inference on our learned models reduces to minimizing a pairwise function of discrete variables, which can be done approximately using standard methods. Although this is a primarily theoretical paper, we also demonstrate the practical effectiveness of our framework on the problem of learning a shape prior for image segmentation and reconstruction. We show that our framework can learn a compact representation that approximates a prior that encourages low curvature shapes. We evaluate the approximation accuracy, discuss properties of the trained model, and show various results for shape inpainting and image segmentation.

1 Introduction

Most computer vision problems can be formulated in terms of estimating the values of hidden variables from a given set of observations. In such a setting, probabilistic models are applied to represent the prior knowledge about hidden variables and their statistical relationship with observed variables.

A number of models encoding prior knowledge about scenes have been proposed in computer vision. The most popular ones have been in the form of a pairwise Markov Random Field (MRF). A *random field* is a strictly positive probability distribution of a collection of random variables. *Markov Random Field* (MRF) additionally satisfies some (or none) Markov (conditional independence) properties [2]. An important characteristic

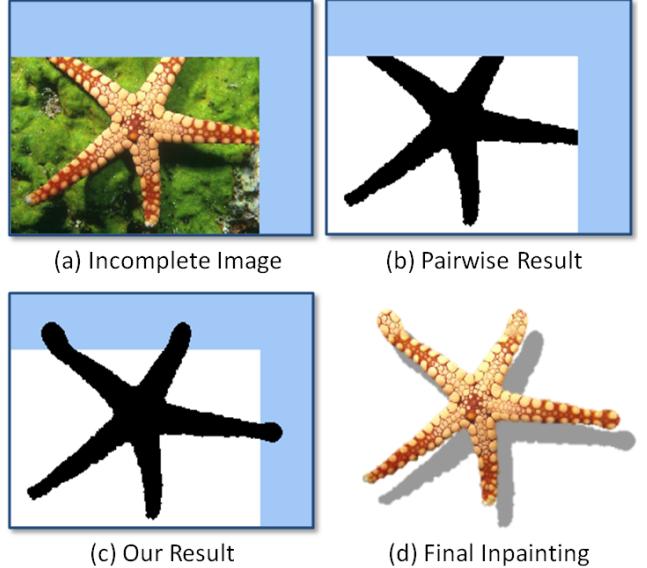


Figure 1: (a) Input image (area for completion of starfish is shown in blue). (b) The starfish was interactively segmented from the image. Then the three arms of the starfish, which touch the image borders, were completed with an 8-connected pairwise MRF which encodes a standard length prior. Note, with this prior no pixels in the blue completion area were assigned to the starfish arms. (c) Completion of the shape of the three starfish arms was done with our compact-higher-order prior which models curvature. (d) Finally, texture was added fully automatically using [1].

of an MRF is the factorization of the distribution into a product of factors. *Pairwise* MRFs can be written as a product of factors defined over two variables at a time. For discrete variables, this enables non-parametric representation of factors and the use of efficient optimization algorithms for approximate inference of the Maximum-a-Posteriori (MAP) solution. However, because of their restricted pairwise form, the model is not able to encode many types of powerful structural properties of images. Curvature is one such property which is known to be extremely helpful for inpainting (see figure 1), segmentation, and many other related problems.

Higher-order Priors There has been a lot of research into priors based on high-level structural dependencies between pixels such as curvature. These priors can be represented in the probabilis-

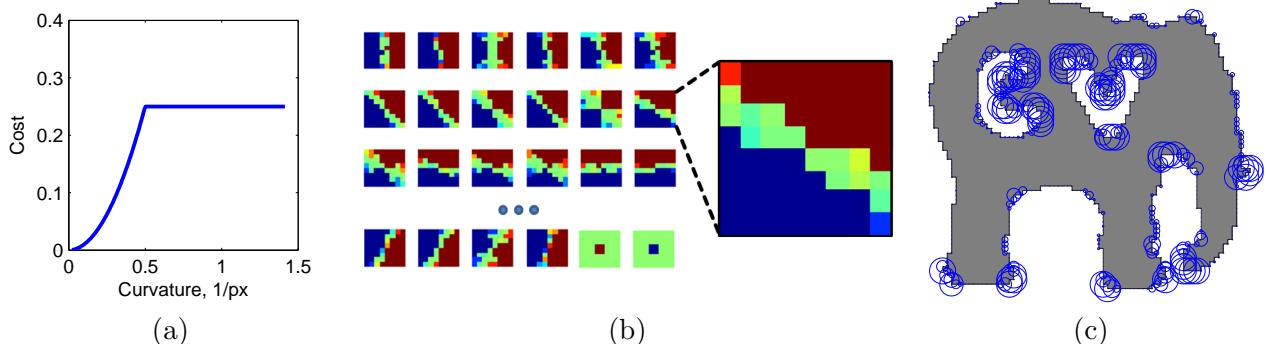


Figure 2: (a) A given cost function for curvature, which we want to approximate. (b) Our method learns a large set of ‘soft’ pattern-based potentials which implicitly model the curvature cost. Our learned MRF model has a higher-order 8×8 pattern-based potential at each pixel-location. Our model implicitly selects at each position the best-fitting pattern for a labeling. Intuitively, a pattern fits well (has low cost) if all foreground pixels match to blue pattern weights and all background pixels match to red pattern weights. Pixels which match to green pattern weights do not contribute and may be either background or foreground. The last two patterns encode the fact that if the higher-order potential is defined on top of a non-boundary location (all 4 center pixels are foreground or background), then the curvature cost is 0, i.e. is ignored. (c) An example demonstrating the curvature cost computed by our pattern based approximation at different parts of the object boundary. Circle radius correspond to the assigned cost.

tic model using factors which may depend on more than two variables at a time. The largest number of variables in a factor is called the *order* of the probabilistic model. Higher-order factors defined on discrete variables are computationally expensive to represent. In fact, the memory and time complexity for inferring the MAP solution with general inference algorithms grows exponentially with the order, and thus has limited the use of such models. The situation is a bit different for parametric models with continuous variables. Higher-order prior models such as Product of Experts (PoE) [3] or Field of Experts (FoE) [4] are differentiable in both parameters and hidden variables. These models thus enable inference using local gradient descent, and have led to impressive results for problems such as image restoration and optical flow.

Recent research on discrete higher-order models has focused on identifying families of higher-order factors which allow efficient inference. The factors can be categorized into 3 broad categories: (a) *Reducible factors*, which allow MAP inference to be reduced to the problem of minimizing a pairwise energy function of discrete variables with the addition of some auxiliary variables [5, 6, 7, 8, 9],

(b) *Message-enabled factors*, which allow efficient message computation and thus allow inference using message passing methods such as Belief Propagation (BP) and Tree Reweighted message passing (TRW) [10, 11, 12], and (c) *Constraint factors*, which impose global constraints that can be imposed efficiently in a relaxation framework [13, 14].

Pattern-based Representation Pattern and lower-envelope based representations proposed in [5, 11, 9] can represent some families of *Reducible factors*. The higher-order potentials of [11, 9] are defined by enumerating important configurations (patterns) in a local window. The model of [9] additionally enables deviations from encoded patterns, by using linear weighting functions. The above models are generalized by the representation proposed in [5] which encodes higher-order functions as lower (or upper) envelopes of linear (modular) functions of the variables. The complexity of representing and performing inference depends on the number of linear functions (or patterns) used for representing the higher-order factor. A number of higher-order priors can be encoded using few linear functions (or patterns) and thus allow efficient inference. However, the use of a general higher-order prior would require ex-

ponential (in the order of the factor) number of linear functions (or patterns).

Our Contribution This paper addresses the problem of discovering a compact representation of a general higher-order factor. More specifically, given a particular higher-order factor, we try to find the linear-envelope representation which best approximates it. Given a set of training examples of labeling and their corresponding desired costs, we find parameters of a linear-envelope representation that matches these costs. While the problem is difficult, we propose a simple yet effective algorithm.

We demonstrate the efficacy of our method on the problem of finding a compact ‘curvature prior’ for object boundaries. This prior encourages smooth boundaries by assigning a high cost to high curvature shape and a low cost of low curvature shapes. Given a set of training shapes, we find parameters of a linear-envelope representation that closely match their pre-specified curvature based cost¹. Figure 2 illustrates our discovered model. We then use the discovered higher-order factors in the problems of object segmentation and completion. The experimental results demonstrate that incorporation of these priors leads to much better results than those obtained using low-order (pairwise MRF) based models (see figure 1) and other state-of-the-art curvature formulations.

An outline of the paper follows. In section 2, we provide the notation, define the higher-order model, and explain the lower-envelope representation of higher-order factors. Section 3 reviews research on using curvature priors for labeling problems. Section 4 explains how we learn a lower-envelop representation of a curvature based higher-order prior model. Section 5 discusses the techniques we used to perform MAP inference in the pairwise model corresponding to the discovered higher-order model. Section 6 describes our experimental setup and provides the results. We conclude by summarizing our framework and list-

¹The purpose of this exercise is only to demonstrate the power of our representation scheme. A more difficult problem would be to learn the linear envelope model in an unsupervised way from examples of segmentations. We do not address this problem in this paper.

ing some directions for future work in Section 7.

2 Higher-order Model Representation

We consider a set of pixels $\mathcal{V} = \{1 \dots N_X\} \times \{1 \dots N_Y\}$ and a binary set of labels $\mathcal{L} = \{0, 1\}$, where 1 means that a pixel belongs to the foreground (shape) and 0 to the background. Let $\mathbf{x}: \mathcal{V} \rightarrow \mathcal{L}$ be the labeling for all pixels with individual components denoted by $x_v, v \in \mathcal{V}$. Furthermore, let $V(h) \subset \mathcal{V}$ denote a square window of size $K \times K$ at location h , and \mathcal{U} is the set of all window locations. Windows are located densely in all pixels. More precisely, all possible $K \times K$ windows are considered which are fully inside the 2D-grid \mathcal{V} (fig. 3 illustrates boundary locations). Let $\mathbf{x}_{V(h)}: V(h) \rightarrow \mathcal{L}$ denote a restriction of labeling \mathbf{x} to the subset $V(h)$.

We consider distribution of the form $p(\mathbf{x}) \propto \exp -E(\mathbf{x})$ with the following energy function:

$$E(\mathbf{x}) = \sum_{v \in \mathcal{V}} \theta_v(x_v) + \sum_{uv \in \mathcal{E}} \theta_{uv}(x_u, x_v) + \sum_{h \in \mathcal{U}} E_h(\mathbf{x}), \quad (1)$$

where notation uv stands for ordered pair (u, v) , $\theta_v: \mathcal{L} \rightarrow \mathbb{R}$ and $\theta_{uv}: \mathcal{L}^2 \rightarrow \mathbb{R}$ are unary and pairwise terms, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is a set of pairwise terms and E_h are higher-order terms. We consider the higher order terms E_h of the following form (equivalent to [9])

$$E_h(\mathbf{x}) = \min_{y \in P} \left(\langle \mathbf{w}_y, \mathbf{x}_{V(h)} \rangle + c_y \right). \quad (2)$$

This term is the minimum (lower envelope) of several modular functions of $\mathbf{x}_{V(h)}$. We call this model a *maxture* by the analogy with the mixture model discussed below. We refer to individual linear functions $\langle \mathbf{w}_y, \mathbf{x}_{V(h)} \rangle + c_y$ as “soft” patterns². Here $\mathbf{w}_y \in \mathbb{R}^{K^2}$ is a weight vector and $c_y \in \mathbb{R}$ is a constant term for the pattern. Vector \mathbf{w}_y is of the same size as the labeling patch $\mathbf{x}_{V(h)}$ and it can be visualized as an image (see fig. 2(b)). The variable $y \in P$ is called a pattern-switching variable. It is a discrete variable from the set $P = \{0, \dots, N_P\}$.

²The name “soft” refers to the fact that weights w_y can take arbitrary values. This is in contrast to other models, [9, 11], which constrain the weights w_y to certain values, as discussed later.

We let the pattern which corresponds to $y = 0$ have the associated weights $\mathbf{w}_0 = \mathbf{0}$. This pattern assigns a constant value c_0 to all labelings $\mathbf{x}_{V(h)}$ and it ensures that $E_h(\mathbf{x}) \leq c_0$ for all \mathbf{x} . It will be needed to express models [9, 11] in the form of (2), as these models explicitly define a cut-off value. It is also used in our curvature model, where it represents the maximal cost f^{\max} of the curvature cost function.

The minimization problem of energy (1) expresses as

$$\min_{\mathbf{x}} \left[E_0(\mathbf{x}) + \sum_{h \in U} \min_{y \in P} \left(\langle \mathbf{w}_y, \mathbf{x}_{V(h)} \rangle + c_y \right) \right], \quad (3)$$

where unary and pairwise terms are collected into E_0 . The problem can also be written as a minimization of a pairwise energy

$$\min_{\substack{\mathbf{x} \in \mathcal{L}^{\mathcal{V}} \\ \mathbf{y} \in \mathcal{P}^U}} \left[E_0(\mathbf{x}) + \sum_h \langle \mathbf{w}_{y_h}, \mathbf{x}_{V(h)} \rangle + c_{y_h} \right], \quad (4)$$

where $\mathbf{y} : \mathcal{U} \rightarrow P$ is the concatenated vector of all pattern switching variables³. Clearly, problem (4) is a minimization of a pairwise energy function of discrete variables \mathbf{x}, \mathbf{y} .

Problem (4) is NP-hard in general. A subclass [6] where minimization of (1) is solvable in polynomial time is described in Appendix A. However, this class is very restrictive and is not suitable for our purpose. In the general case a number of approximate MAP inference techniques for pairwise energies can be used, as discussed in sec. 5.

2.1 Pattern-based model

Let us relate the above model to the “hard” pattern-based model defined in [11] (and also special case in [9]). In [11] a potential of the following form is used:

$$E_h(\mathbf{x}) = \begin{cases} \tilde{c}_y & \text{if } \exists y \in \{1 \dots N_P\} \ \mathbf{x}_{V(h)} = \mathbf{p}^y \\ \tilde{c}_0 & \text{otherwise} \end{cases}. \quad (5)$$

This potential assigns cost \tilde{c}_y if the labeling matches exactly pattern $\mathbf{p}^y \in \mathcal{L}^{K^2}$ for some $y \in$

³We refer to components of \mathbf{y} by y_h , while y usually denotes an independent bound variable.

$\{1 \dots N_P\}$ and cost \tilde{c}_0 if none of the patterns are matched. The set of labels for this model is not necessarily binary. For binary labels (5) can be rewritten in the form (2) by setting

$$\begin{aligned} w_{y,v} &= \begin{cases} -B, & p_v^y = 1 \\ +B, & p_v^y = 0 \end{cases} \\ c_y &= \tilde{c}_y + B \sum_v p_v^y \\ w_{0,v} &= 0 \\ c_0 &= \tilde{c}_0, \end{aligned} \quad (6)$$

where B is a sufficiently large constant. This is a restricted model since deviations from the “hard” patterns are not allowed, in contrast to our model (2). However, this restricted model allows for an alternative optimization approach which was proposed in [11] and seems to correspond to a tighter relaxation than the standard relaxation for the pairwise model (4). Also, the hard pattern potential model [11] is too restrictive in the following sense. Function (2), as well as the special case (5), can exactly represent an arbitrary function of discrete variables $\mathbf{x}_{V(h)}$ if we allow $2^{|V(h)|}$ patterns. Obviously, in practice if $|V(h)|$ is large such an approach becomes computationally infeasible. The challenge is therefore to define a good model with a small number of patterns, for which case model (2) seems clearly a better choice.

2.2 Relation to Mixture Model

Minimization of a pairwise energy with auxiliary variables (4) can be interpreted as MAP inference in a pairwise MRF. The question arise then: why do we talk about higher order terms, rather than just introducing more hidden variables in a pairwise model? To answer this question we need to look at the associated probability distributions and the estimation problems. Here, for simplicity, we ignore unary and pairwise terms of E_0 and also ignore the observable variables. Let \mathbf{y} be auxiliary hidden variables in the following, new pairwise model

$$p(\mathbf{x}, \mathbf{y}) \propto \prod_{h \in U} \exp(-\langle \mathbf{w}_{y_h}, \mathbf{x}_{V(h)} \rangle - c_{y_h}). \quad (7)$$

The model of \mathbf{x} is then implied to be

$$\begin{aligned} p^{\text{mix}}(\mathbf{x}) &\propto \sum_{\mathbf{y} \in P^U} \prod_{h \in U} \exp(-\langle \mathbf{w}_{y_h}, \mathbf{x}_{V(h)} \rangle - c_{y_h}) \\ &= \prod_{h \in U} \sum_{y \in P} \exp(-\langle \mathbf{w}_y, \mathbf{x}_{V(h)} \rangle - c_y). \end{aligned} \quad (8)$$

It can be seen that factors in this model are *mixtures* of exponential distributions. The problem of MAP inference of \mathbf{x} , taking the logarithm, can be written as

$$\arg \max_{\mathbf{x}} \sum_h \bigoplus_{y \in P}^1 (-\langle \mathbf{w}_y, \mathbf{x}_{V(h)} \rangle - c_y), \quad (9)$$

where we define *log-sum-exp* operation \oplus^β as

$$a \oplus^\beta b = \frac{1}{\beta} \log(e^{\beta a} + e^{\beta b}), \quad (10)$$

which is commutative and associative binary operation, so $\bigoplus_{y \in P}^\beta a_y$ is also unambiguously defined.

The problem (9) is a discrete optimization with difficult objective, so by introducing auxiliary hidden variables we arrived at a complicated inference problem. A heuristic could be used to infer \mathbf{x} by solving the joint MAP in \mathbf{x} and \mathbf{y} and then to discard the estimate of \mathbf{y} , this would lead to the optimization problem of the desired form:

$$\arg \min_{\mathbf{x}, \mathbf{y}} \sum_h (\langle \mathbf{w}_{y_h}, \mathbf{x}_{V(h)} \rangle + c_{y_h}). \quad (11)$$

So we could have learned a mixture model and made a wrong use of it by replacing inference with (11). We prefer instead to state the model as

$$p^{\text{max}}(\mathbf{x}) = \prod_h \max_{y \in P} \exp(-\langle \mathbf{w}_y, \mathbf{x}_{V(h)} \rangle - c_y), \quad (12)$$

where the factors are *maxtures* of exponential distributions. Clearly, the model corresponds to (4) by rewriting it as

$$= \exp \left\{ - \sum_h \min_{y \in P} (\langle \mathbf{w}_y, \mathbf{x}_{V(h)} \rangle + c_y) \right\}. \quad (13)$$

The MAP inference of \mathbf{x} in this model directly correspond to (11). So this is a much cleaner correspondence of the model to the estimation problem.

In fact, there is a smooth transition between the two models. It is known that $\lim_{\beta \rightarrow \infty} a \oplus^\beta b = \max(a, b)$, which says that as distributions get sharper, their mixture turns into a maximum. It is also easy to see that under this limit the model p^{mix} transits to p^{max} and so is the corresponding MAP \mathbf{x} problem.

3 Curvature priors for Image Labeling

We will evaluate the usefulness of our curvature prior on the closely related problems of image segmentation and shape inpainting. Given an image region with a lack of observations, a good segmentation model should complete the segmentation in this region from the evidence outside of the region. This *shape inpainting* problem is related to inpainting of binary images which has been approached in the continuous setting with several curvature-related functionals [15, 16]⁴.

Image labeling with curvature regularization is an important topic of research, and both continuous and discrete formulations for the problem have been proposed. Continuous formulations offer accurate models, but they rely on numerical schemes which have to deal with highly nonlinear functions and need a good initialization to converge to a good local minimum of the cost functional⁵. Discrete methods for image labeling with curvature regularization build on quantization and enumeration of boundary elements. Until recently, they were applied only in restricted scenarios where it is possible to reduce the problem to a search of the minimal path or minimum ratio cycle [18]. These cases enjoy global optimality, however they restrict the topology or do not allow for arbitrary regional terms.

Recently, [19] proposed a discrete method for a general setting. This method is able to find globally optimal solutions for difficult segmentation problems. They formulate the problem as Integer Linear Programming (ILP), where variables are indicators of edge and region elements, while

⁴There is a vast literature on the general image inpainting problem, however these techniques, especially exemplar-based ones, do not extend to image segmentation problem, and are not relevant in the context of this paper.

⁵For instance, [17] works with discretized Euler-Lagrange equations of the 4th order.

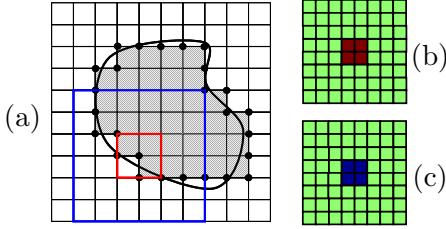


Figure 3: (a) Continuous shape and its discretization. Filled circles show boundary locations. Larger blue window illustrates $V(h)$ at location h . (b,c) Foreground and background patterns: green $w_{y,v} = 0$, red: $w_{y,v} = +B$, blue: $w_{y,v} = -B$, constants c_y are $-4B$ and $+4B$ respectively.

constraints make sure that these variables are consistent and do correspond to a shape. However, this method quantizes the directions of boundary elements, which, as we show in the experiment section, may result in large errors in the final segmentation. The complexity of the model [19] grows very fast with the number of directions, and it is not entirely clear how to build a cell complex with the required properties for more directions than [19] considers.

A recent work [20] claims to give fast optimal solution for curvature regularization. However, their model is a crude approximation to the curvature functional. Its 4-neighborhood variant essentially penalizes the number of ‘‘corners’’ in the segmentation – locations where a 2×2 window has 3 pixels foreground and 1 background or vice-versa. It assigns zero penalty for horizontal and vertical boundaries but diagonal boundaries have maximal penalty. The 8-neighborhood variant allows for diagonal lines at zero cost but penalizes vertical and horizontal lines. Our model with 2×2 windows can also implement the 4-neighborhood model [20]. However, as we argue, a larger window is necessary to capture the curvature of a shape represented by binary pixel labeling.

4 Learning a Curvature Cost Model

Suppose we are given a shape $S \subset \mathbb{R}^2$ such that we can calculate the curvature κ at every point of the boundary, ∂S . Let $f(\kappa) \geq 0$ be a curvature cost function, which defines a desired penalty on curvature, in this paper we consider $f(\kappa) = \min(\kappa^2, f^{\max})$. Let the total cost of the shape be

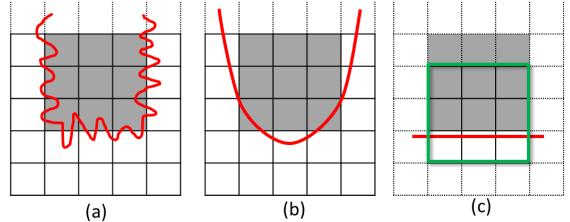


Figure 4: **Problem definition and motivation of large-sized windows.** Examples above show discrete labelings on a pixel grid with a corresponding red continuous curve. Note, there are infinitely many continuous curves which give rise to the same discrete labelling - two examples are given in (a) and (b). The red curve in (b) is probably the one with lowest curvature given the discrete labelling. Our goal is to find an energy function which maps every discrete labelling to the corresponding cost of the continuous curvature with lowest curvature. (c) makes the important point that larger sized windows have inherently a better chance of predicting well the curvature at the center of the window. In (c) the green window is of size 3×3 , while in (b) it is of size 5×5 . The underlying discrete labelling is identical in both cases and the red curve is the optimal (lowest curvature) continuous curve given the window. The crucial point is that the curvature of the continuous curve, at the center of the window, is very different in (b) and (c). Note, this problem is to some extend mitigated by the fact that the total cost of segmentation is the sum of costs along the boundary.

$\int_{\partial S} f(\kappa) dl$. Our goal is to approximate this integral by the sum

$$\sum_{h \in \mathcal{U}} E_h(\mathbf{x}), \quad (14)$$

where functions E_h operate over a discretized representation of the shape, \mathbf{x} , and are of the form (2) with weights \mathbf{w}, \mathbf{c} . Here \mathbf{w} and \mathbf{c} denote the concatenated vectors of all weights \mathbf{w}_y and c_y , respectively. The learning problem is to determine the pattern weights \mathbf{w}, \mathbf{c} such that the approximation is most accurate. Since the mapping of continuous to discrete curves is a many-to-one mapping, we further formalize our exact goal in figure 4. In the figure we also motivate the important aspect that larger windows are potentially superior.

We first restrict the sum in (14) to take into account only boundary locations. We call h a *boundary location* for shape \mathbf{x} if the 2×2 window at h contains some pixels which are labeled foreground as well as some pixels which are labeled background, as illustrated in fig. 3. We constrain all soft patterns to be non-negative ($\langle w_y, x \rangle + c_y \geq 0$) and introduce two special patterns (fig. 3b,c), which have cost 0 for locations where the 2×2 window at location h contains only background or foreground pixels. These patterns make $E_h(\mathbf{x})$ vanish over all non-boundary locations, therefore such locations do not contribute to the sum (14). The learning task is now to determine $E_h(\mathbf{x})$, such that at each boundary location the true cost $f(\kappa)$ is approximated. In this way (14) does correspond to the desired integral if we were to neglect the fact that the number of boundary locations does only approximate the true length of the boundary. Note, the number of boundary locations does correspond to the “Manhattan” length of the boundary. We will come back to this problem in sec. 6.

Point-wise learning procedure. Let us assume that in a local $K \times K$ window, shapes of low curvature can be well-approximated by simple quadratic curves⁶. The idea is to take many examples of such shapes and fit $E_h(\mathbf{x})$ to approximate their cost. We consider many quadratic shapes $(S^i)_{i=1}^N$ in the window $K \times K$ and derive their corresponding discretization on the pixel grid $(\mathbf{x}^i)_{i=1}^N$. Each continuous shape has an associated curvature cost $f^i = f(\kappa^i)$ at the central boundary location. We formulate the learning problem as minimization of the average approximation error:

$$\begin{aligned} & \arg \min_{\mathbf{w}, \mathbf{c}} \sum_i |E_h(\mathbf{x}^i) - f^i|, \\ \text{s.t. } & \begin{cases} \mathbf{w}_0 = 0, c_0 = f^{\max} \\ E_h(x) \geq 0 \quad \forall x \end{cases} \end{aligned} \quad (15)$$

where the first constraint represents the special implicit pattern ($w_0, c_0 = f^{\max}$), which ensures that $E_h(\mathbf{x}) \leq f^{\max}$. The second constraint makes sure that cost is non-negative. It is important

⁶Note, based on our definition in fig. 4 we select quadratic curves which are likely to be the ones of lowest curvature (among all curves) for the corresponding discrete labelling.

for the following reason: the formulation of the approximation problem does not explicitly take into account “negative samples”, i.e. labellings which do not originate from smooth curves, and which must have high cost in the model. However, requiring that all possible negative samples in a $K \times K$ window have high cost would make the problem too constrained. The introduced non-negativity constraint is tractable and not too restrictive. This problem appears difficult, since $E_h(\mathbf{x}^i)$ is itself a concave function in the parameters \mathbf{w}, \mathbf{c} . We approach (15) by a k-means like procedure with a specially constructed initialization:

Alg. 1. Iterative Factor Discovery

Input: $\mathbf{x}^i, f^i, \mathbf{w}, \mathbf{c}$

Repeat till convergence or maximum iterations:

1. For all training images i find best matching patterns $y^i = \arg \min_y [\langle \mathbf{w}_y, \mathbf{x}^i \rangle + c_y]$
2. For all $y \in 1 \dots N_P$ refit (\mathbf{w}_y, c_y) :

$$\begin{aligned} (\mathbf{w}_y, c_y) = & \arg \min_{\substack{\mathbf{w}_y, c_y \\ \xi}} \sum_{i|y^i=y} |\langle \mathbf{w}_y, \mathbf{x}^i \rangle + c_y - f^i| \\ \text{s.t. } & \begin{cases} \xi_v \leq w_{y,v} \\ \xi_v \leq 0 \\ \sum_v \xi_v + c_y \geq 0 \end{cases} \end{aligned} \quad (16)$$

The refitting step (16) is a linear optimization which can be solved exactly. The constraint in (16) is an equivalent representation of the constraint $\langle \mathbf{w}_y, \mathbf{x} \rangle + c_y \geq 0 \forall x$, imposed by (15).

The initialization and results of applying this learning procedure are discussed in the sec. 6.

5 Inference

We examined several standard MAP inference techniques for pairwise MRFs, and found the following two-stage procedure to work best for our problem. First, the TRW-S algorithm[21, 22] is run for a fixed number of iterations. Then an initial solution is obtained by rounding the tree min-marginals. Second, a Block-ICM procedure improves on this initial solution.

5.1 TRW-S

In contrast to many other pairwise MRFs encountered in computer vision, our model has a very large number of pairwise links. We developed the following memory-efficient implementation of TRW-S for models with pairwise interactions between variables with few states and variables with many states. For an image of size $N_Y \times N_X$ and model with N_P patterns of size $K \times K$, there are in total $O(N_Y N_X K^2)$ pairwise terms needed to represent the pattern potentials (pairwise energy (4) can be illustrated as a bipartite graph in fig. 5(a)). The algorithm in [21] needs to keep a message (vector of size N_P) for each edge. This makes the original procedure [21] extremely memory intensive. Since each pixel has only two labels and there are N_P labels for the pattern switching variable, we can improve on memory requirement with a little computation overhead. We make the following modification to the implementation [21], fig.3]. We store reparametrized unaries, $\hat{\theta}_s(x_s)$ and $\hat{\theta}_h(y_h)$, ($O(N_Y N_X N_P)$ memory) and messages only in the direction patterns \rightarrow pixels, $m_{vh}(x_v)$, ($O(N_Y N_X K^2)$ memory). When the reverse mes-

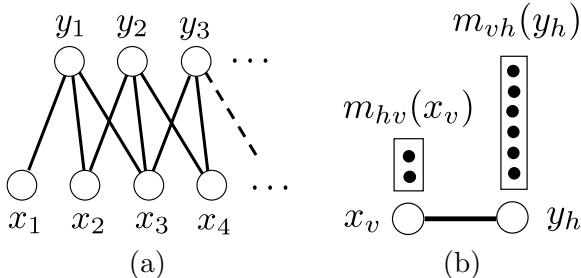


Figure 5: (a) Graphical model for the energy (4). (b) Pairwise terms connecting pixel labeling x_v and pattern-switching variable y_h . Circles show possible states: two states for x_v and N_P states for y_h .

sage is requested by the algorithm it is computed on the fly using the equation

$$m_{vh}(y_h) = \min_{x_v} [\gamma_{st} \hat{\theta}_s(x_s) - m_{hv}(x_v) + \theta_{vh}(x_v, y_h)], \quad (17)$$

which is $O(N_P)$ computations. To completely specify the algorithm we have to choose the ordering of variables and the parameters γ . We

can specify the ordering corresponding to longer chains, which potentially provide a faster convergence or the ordering where all x precede all y , corresponding to short 1-edge chains, which makes the computation parallelizable. The parameters γ are selected following the recommendation in [21].

This modified version of TRW-S requires only $O(N_Y N_X (K^2 + N_P))$ memory and runs in $O(N_Y N_X K^2 N_P)$ time per full-pass iteration. In practice this results in about 5 seconds per iteration for an image of size 158×128 , however a significant number of iterations was required to archive good results (e.g. up to 4000), which is a known issue for dense graphs [23].

5.2 Block-ICM

It often happens that the tree min-marginals of TRW-S for some pixels are “indecisive”, i.e. possibly many different labellings have a low energy. In this case the solution picked by our pixel-independent rounding schemes may be rather poor. We found that further local improvements can significantly decrease the energy. Block-ICM tries to improve the current labeling \mathbf{x} by switching states of a small block of k variables at a time. Obviously, its complexity grows exponentially in k , hence k must be low (we use $k=6$). During Block-ICM, the blocks are selected densely around the current boundary of \mathbf{x} . For an image of size 158×128 Block-ICM needs about 3 minutes to converge.

5.3 LP relaxation

TRW-S is a suboptimal dual solver for linear program relaxation of the discrete pairwise energy minimization. The relaxation (see e.g. [24] for an overview) is obtained by linearizing objective (4) and dropping the integrality constraints. In particular, it replaces binary variables $x_v \in \{0, 1\}$ with relaxed variables $\bar{x}_v \in [0, 1]$. The optimal relaxed labeling may be of interest even when the relaxation is not tight. However, TRW-S does not compute the primal relaxed labeling and it may get stuck in a suboptimal point. On the other hand solving the full primal LP seems infeasible since it requires $O(N_Y N_X K^2 N_P)$ variables. The following heuristic can be applied. Note, although, as we will see, the results may be improved with

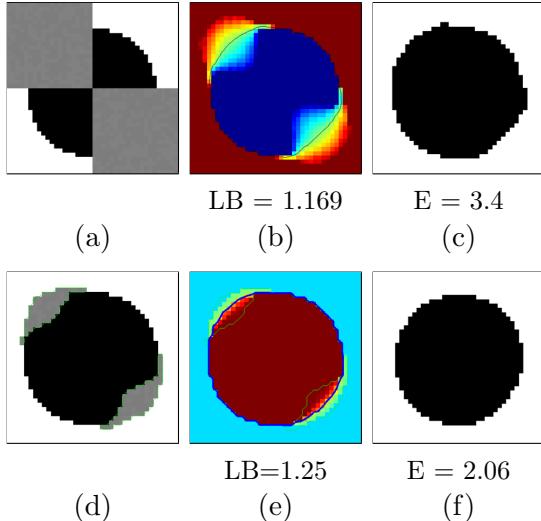


Figure 6: **Restricted LP.** (a) Unaries: black – foreground, white – background, gray – area to be inpainted; (b) Tree min-marginals of TRW-S: a thin line shows the 0-level contour; (c) Rounding of TRW-S solution; (d) The reduced problem; (e) Relaxed primal solution of the reduced problem (with 0-level contour); (f) Rounding of relaxed primal solution. We also show lower bound (LB) for relaxation and energie (E) for a discrete solution. Note, the discrete, ground truth circle has an energy of $E = 1.96$, which is still lower than the best solution found (f).

this heuristic, we did not use it in our experimental section due to its computational complexity. The procedure is that we use the dual solution of TRW-S to greedily fix a larger part of the primal relaxed variables (both corresponding to pixel labels \mathbf{x} and patterns \mathbf{y}). When the tree min-marginal for a label in a pixel is above a threshold compared to the minimal tree min-marginal in the pixel, we fix the corresponding primal variable to 0 and eliminate it. This gives a restricted linear program, which then can be solved by a primal method. Example in fig. 6 illustrates these steps for a circle inpainting problem of the type fig. 8. We make two observations: first, the restricted LP attains optimality in a “fractional” relaxed solution, so there exist an integrality gap and we can not obtain an optimal discrete solution. Second, the value of the objective at the optimum of restricted LP is different from the lower bound by

TRW-S, which means that TRW-S has converged to a sub-optimal dual solution.

Let us also briefly mention on the performance of Belief Propagation (BP), which we found inferior. We used the variant called sequential (min-sum) Belief Propagation (BP), obtained from TRW-S by setting all γ to 1 as described in [21]. In [9] it was reported that BP performs best for texture denoising with soft pattern-potentials, while TRW-S performed poorly. For our model we observed the opposite: BP may produce very poor results, especially for problems where there is a large area of uncertainty (shape-inpainting). Note, we also tried dumping and different re-ordering heuristics for BP, but without success. We believe that an interesting direction for future work is to thoroughly compare various optimization schema for various types of soft pattern-based potentials.

6 Experiments

We performed several kinds of experiments.

1. We discuss the learning procedure of our model and investigate the approximation quality it achieves. For the later we generated continuous shapes, for which the true total cost can be computed precisely, and compared that to our model. Note, to make inference and learning feasible we can only use a restricted number of patterns with limited size. We show that this gives a reasonable approximation of the desired cost functional. Note, in theory, by increasing the resolution (size of patterns) and the number of patterns one may archive approximation with an arbitrary accuracy.
2. The second set of experiments studies the task of “shape-inpainting” where the optimal segmentation (shape) has to be inferred, while only a few boundary conditions are given. It provides a good way of inspecting our prior shape model and assessing whether it corresponds to our intuitive notion of natural shapes.
3. The next experiment is on standard interactive image segmentation. Here we compare length versus curvature regularization.
4. Finally, we analyze the properties of the curvature model of [19] and provide some comparison with our model.

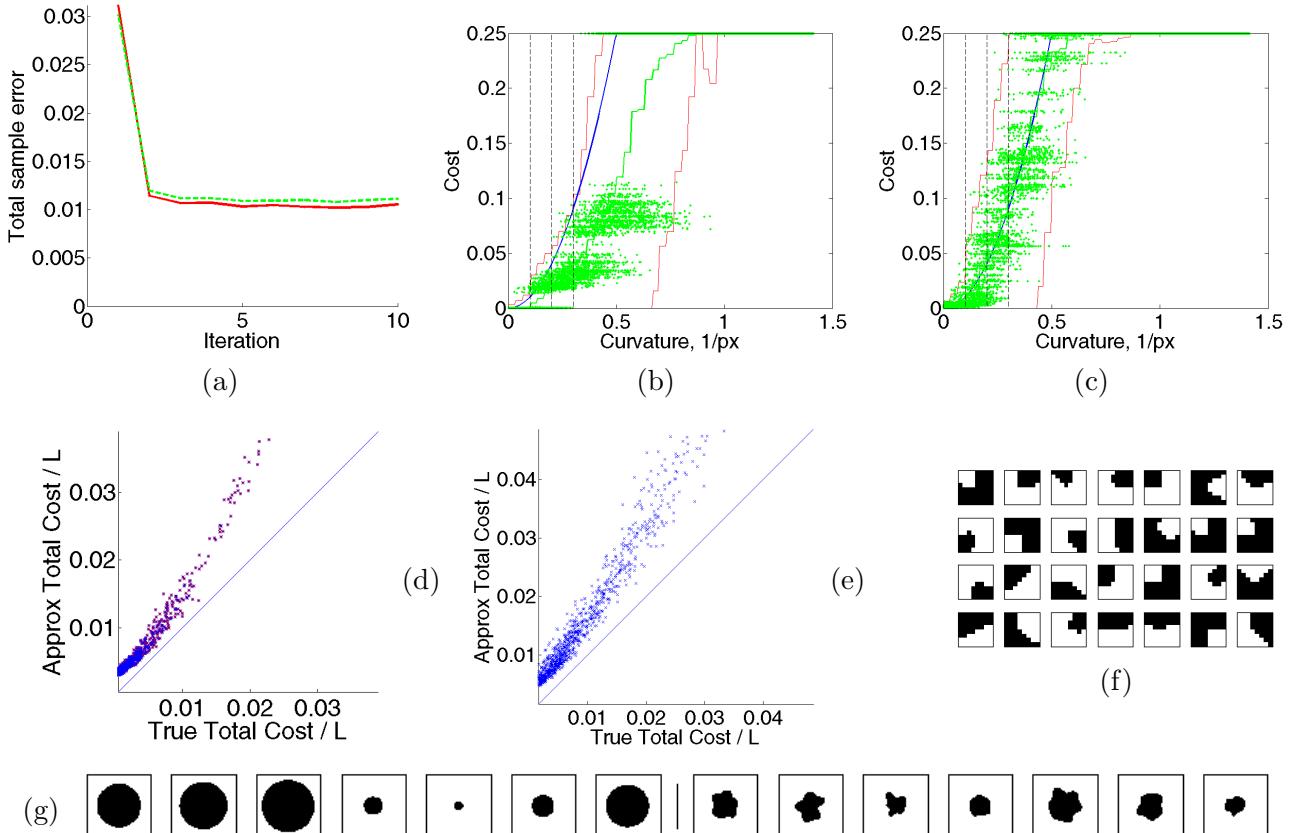


Figure 7: **Cost Approximation.** (a) The performance of Alg. 1 measured in terms of the objective (15). The red and green dashed curves show training and test error respectively. (b) Point-wise approximation cost with initial patterns, and (c) point-wise approximation cost after 10 iterations of Alg 1. In both (b,c): each green point is a test sample; the blue line shows the desired true cost; the green line the mean approximation cost; and the red lines show $3\times$ standard deviation bounds. (d,e) Approximate total cost / length vs. true total cost / length for circles (d) and Fourier shapes (e). (f) Examples of training and test patches used in (a-c). (g) Examples of discretized shapes for circular shapes and Fourier shapes used in (d,e).

6.1 Cost approximation

To learn our curvature model we used 96 patterns of size $K = 8$. For the learning we sampled $N = 10000$ random quadratic curves passing close to the center in a $K \times K$ pixel patch (fig. 7(f)). The initial model is build as follows. We split the training patches into 32 orientations, using the tangent of a curve, and also 3 different curvature intervals. This gives in total 96 bins. For each bin we fit a separate linear function using step 2 of Alg.1, which results in an initial set of 96 patterns. We then run several iterations of Alg.1. The estimated error of the objective (15) is shown in fig. 7(a). We

see that both training and test error decrease over time.

The initial and final point-wise cost (i.e. for a patch) is illustrated in Fig. 7 (b) and (c) respectively. It can be seen that the mean of the approximated cost is very close to the true cost function, however, the variance is considerably large. However, this problem should be mitigated when the cost is summed up along the full boundary, as the errors average out. To verify this, we sampled larger shapes for which we can compute the true total cost exactly and then compared to that of our model. Fig. 7(d,e) shows this experiment for two classes of shapes: (d) circular shapes of

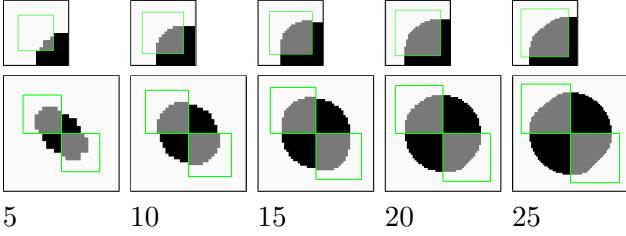


Figure 8: Inpainting of a corner and a circle. The green boxes show the area to be inpainted, where the size in pixels of the length of green box is below the figures. Pixels in gray show the estimated solution. Note, the boundary conditions are different: right-angle boundary condition (top), circle boundary condition (bottom).

size 100×100 with random radius (uniformly sampled in [5 50] pixels) and subpixel shift; (e) complex shapes created using Fourier series $\rho(\alpha) = a_0 + \sum_{k=1}^5 a_k \sin(k\alpha) + b_k \cos(k\alpha)$ in polar coordinates with random coefficients (a, b) . Derivatives, curvature and the total cost integral can be computed accurately for these shapes. We then measure the approximation error relative to the true length of the curve. Figure 7(d,e) shows that the variance is reduced, especially for shapes with low average curvature which are pre-dominant. The plots do also reveal the fact that we consistently overestimate the true curvature cost. This problem is related to the fact that we approximate the integral along the boundary by the sum over boundary locations, which corresponds to the ‘‘Manhattan’’ length, which is usually higher than the Euclidean length. While this is not essential for getting a useful model for the shape prior, we discuss in Appendix B a way of reducing this error by adjusting the pattern costs.

6.2 Shape Inpainting

The goal is to reconstruct the full shape, while only some parts of the shape are visible to the algorithm. This is a useful test to inspect our shape prior. Let $F \subset \mathcal{V}$ be the set of pixels restricted to foreground (shape) and $B \subset \mathcal{V}$ pixels restricted to background. The unary terms of (1), $\theta_v(x_v)$, are set to ∞ if label x_v contradicts to the constraints and 0 otherwise. This ensures that the correct segmentation is inferred in the region $F \cup B$.

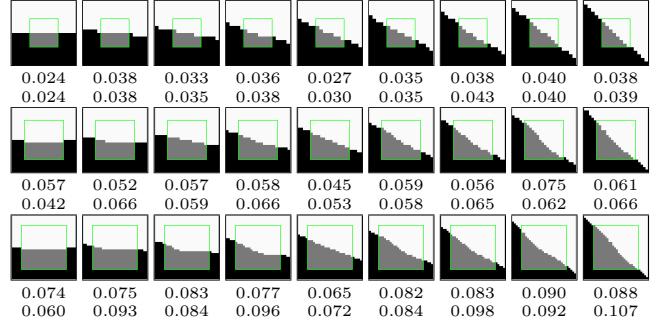


Figure 9: Inpainting with a straight line boundary condition. The green box is of size 8, 16 and 24 pixels respectively from top to bottom. The numbers show the model cost of the estimated solution (top line) and the cost of the ground truth (discretized) straight line (bottom line).

In the unknown region $\mathcal{V} \setminus (F \cup B)$ all unaries are exactly 0. Fig. 8,9 show results for different inpainting problems with various boundary conditions corresponding to inpainting of some simple shapes. The main conclusion is that all results look reasonably good. Note, for a tiny circle in fig. 8(bottom left) the reconstruction looks more like an oval than a circle. This is an expected result, and visually acceptable, since the boundary condition (black pixels) may correspond to either of the shapes and an oval has a lower cost. Also, we see that for some line inpainting examples in fig. 9 the result deviates slightly from the ground truth (straight line). Given that the cost of our solution is almost always lower than the cost of the ground truth, it is quite likely that the problem is due to a non-perfect model and not due to a local minimum of the optimization. As mentioned before, one way to overcome this problem is to allow for more patterns. We also tested our algorithm on inpainting real world images, and compared its results with those obtained by using a pairwise Markov Random Field formulation that tries to reduce the boundary length. The results can be seen in figures 10 and 1. It can be seen that the higher-order model that encodes curvature produces shape completions with smooth boundaries. An example combining curvature and length priors is shown in fig. 11.

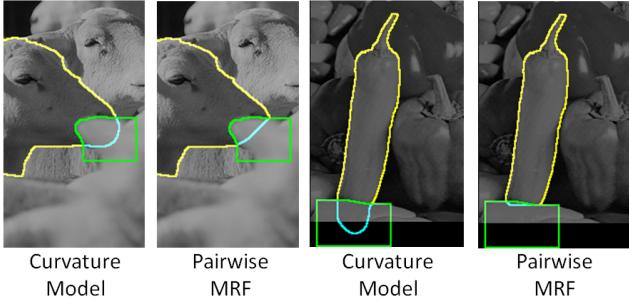


Figure 10: Two example for automatic shape completions of an occluded object. In both cases the left result is with a pure curvature prior and the right result with a pure length prior (8-connected). Note, the yellow curve (and a part of the green curve) indicate the original user-defined segmentation. Then the user defines the green area. Inside the green area, the method automatically finds the shape completion (blue curve).

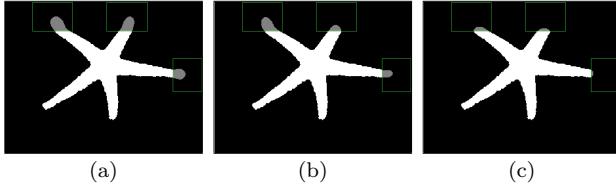


Figure 11: Combining length and curvature for inpainting: (a) pure curvature, (b) curvature + length, (c) curvature + more length.

6.3 Image Segmentation

We use a simple model for the task of interactive FG/BG image segmentation, similar to [25]. Based on the user brush strokes (fig. 12(a)) we compute likelihoods using a Gaussian mixture model (GMM) with 10 components. The difference of the unaries $\theta_v(1) - \theta_v(0)$ correspond to the negative log-likelihood ratio of foreground and background. Fig. 12(e) shows results when using a simple pairwise MRF (8-connectivity), which puts a prior on the length of the boundary. By varying the strength of the prior we achieve various results, however, none of the results is satisfying. Note, the length prior is, in contrast to [25], not gradient-sensitive since the legs of the giraffe do not have an edge with sharp contrast. Results for our curvature model for various strengths of the prior are shown in fig. 12(f). Note, no additional length prior is added. We clearly see that the cur-

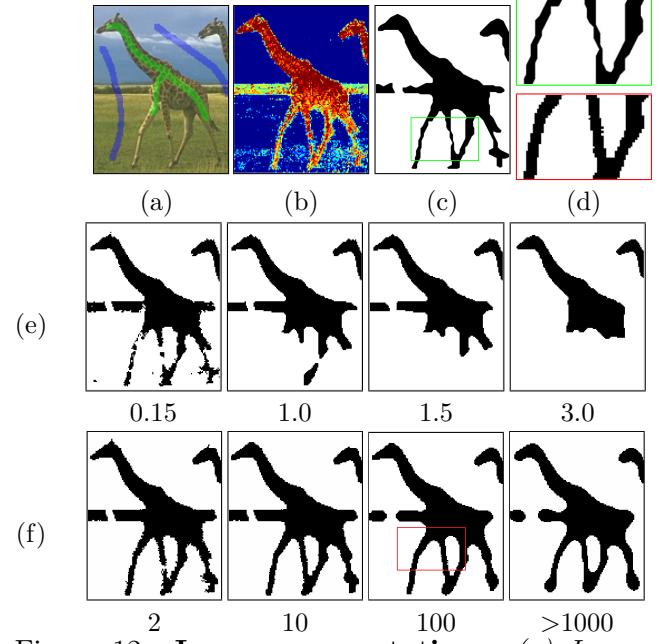


Figure 12: **Image segmentation.** (a) Image with foreground (green) and background (blue) seeds; (b) Color based unary potential costs (red implies foreground-favoring, blue implies background-favoring). (c) Detail of segmentation result from [19] (top) and our result (f,100) (bottom). (d) Detail of segmentation [19] and our. (e) Segmentation with length prior (8-connected model) for various strength of the prior (numbers below figure). (f) Segmentation with curvature prior (our model) for various strength of the prior.

vature prior is able to properly segment the legs of the giraffe, compared to the length prior. Increasing the strength of the prior above some limit (1000) has almost no effect on the smoothness of the solution, because each local 8×8 window is already maximally smooth according to the model. Note, that our result, e.g. fig. 12(f,100), is visually superior to [19], fig. 12(c), despite the fact that we use a grid with much coarser resolution (see detailed discussion below).

6.4 Analysis of the model of Schönemann et al. [19]

The main property of the model of [19] is that there is a pre-defined set of quantized directions. For our analysis, we considered a restricted scenario (Fig. 13) where it is evident that the optimal shape has to be described by a path in the

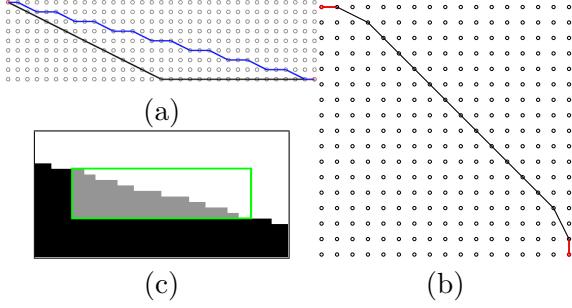


Figure 13: **Properties of [19]**. In (a,b) the problem is to find the optimal shape with lowest curvature given the boundary conditions: the 2 endpoints in (a) and 2 terminal edge-elements in (b). In (a,b) the black line shows an optimal solution of model [19]. The blue line in (a) is the closest approximation of a straight line. Note, in both cases (a,b) the model [19] has multiple solutions: (a) two solution with each having one corner point; (b) a family of optimal solutions with each having 4 corner points. (c) Inpainting results by our model for the same problem as in (a).

graph. We used a 16-connected graph. For two consequent edge elements of the boundary, we approximated the squared curvature as $A^2 \frac{l_1 + l_2}{l_1 l_2}$, [26] (functional G_2), where A is the angle between the line segments and l_1, l_2 their lengths. This is similar to [19] but a symmetric form. Another (not essential) difference of our re-implementation of [19] is that they construct edge elements by subdividing each pixel, whereas we model edge element by end-points in a discrete grid (where edges can also intersect). Fig. 13 reveals the problem of discretized directions. We observed that lines at directions which are not perfectly modeled in [19] (e.g. the line at 1/4 slope in fig. 13(a)) have a very large approximation error. Indeed, the best approximation to the line in fig. 13(a) has many small ‘steps’, whereas the optimal boundary in the model of [19] makes only one large ‘step’. We believe that this effect is the reason for the visual artifacts in the segmentation result of the giraffe in fig. 12(c), where the legs are approximated with a few straight lines.

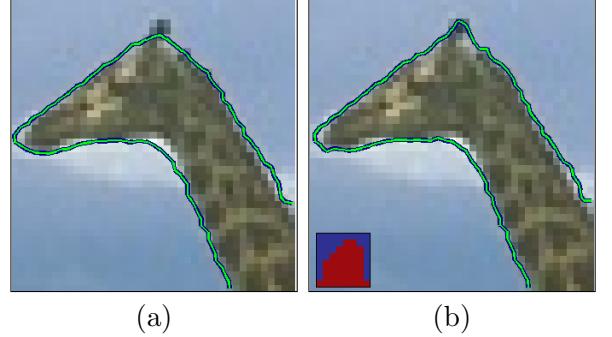


Figure 14: (a) Segmentation of the giraffe’s head, where we show the zero level line of tree marginals. The curvature model smoothes out the giraff’s ear. (b) Segmentation result with one pattern added (shown in the corner).

6.5 Generic Patterns

As a demonstration of the extendibility of our model we made the following simple experiment. The giraff’s ear is smoothed out by our model (fig. 14(a)), since it is of high curvature and has weak support in the color model. We included one additional pattern which fits well to the ear. As all other patterns, this new pattern is available in all locations. The segmentation of the head, fig. 14(b), clearly improves around the ear.

7 Conclusions and Discussion

This paper shows how to compute compact representations of higher order priors which enable the use of standard algorithms for MAP inference. We have demonstrated our method on the problem of learning a ‘curvature-based’ shape prior for image inpainting and segmentation. Our higher-order shape prior operates on a large set (neighbourhood) of pixels and is less sensitive to discretization artifacts. The applicability of our method is not limited to 2D image segmentation and inpainting; it could also be used for 3D completion. More generally, it can be used to obtain tractable representations for higher order priors for general labelling problems such as optical flow, stereo, and image restoration.

It would be interesting to extend our approach to incorporate other types of local shape properties, not necessarily defined by an analytic function but perhaps by exemplars. Such a general-

ization would very likely require a more general learning technique, which is an interesting direction for future research.

A Solvable subclass

Here we give a short review of the result in [6]. We show that a higher-order term of binary variables which is a composition of a piecewise-linear concave $\mathbb{R} \rightarrow \mathbb{R}$ function and a modular function with non-negative coefficients can be represented as min-projections of submodular quadratic functions. In the context of model (1), each higher-order term (2) must be of the form

$$E_h(x) = \min_y (a_y \langle l, \mathbf{x}_{V(h)} \rangle + c_y), \quad (18)$$

where $l \in \mathbb{R}_+^{K^2}$ and a and c are arbitrary. In this case problem (19), where pairwise terms are submodular as well, is solvable in polynomial time. Note, obviously one of the key differences to our general model is that the weight vector \mathbf{w}_y in (2) can have arbitrary entries for different y , and of different signs for a single y . This is a crucial component for modeling some aspects, e.g. curvature, of the boundary of a segmentation.

Consider the problem

$$\min_{x \in \{0,1\}^V} [A(x) + P(x) + G(x)], \quad (19)$$

where $A(x)$ is a linear term: $A(x) = \sum_{s \in V} a_s x_s$ and $P(x)$ is quadratic and submodular. We are seeking for functions G which can be represented as

$$\min_{y \in \{0,1\}^V} [B(x) + C(y) + Q(x, y) + D], \quad (20)$$

where B and C are linear and Q is quadratic submodular. In this case problem (19) would reduce to minimization of a submodular quadratic function, which is easily solvable by max-flow/min-cut.

Functions suggested by [6] may be described as follows. A function of the form

$$G(x) = \min(0, L(x) + D), \quad (21)$$

where $L(x) = \sum_s l_s x_s$ has all coefficients l_s non-positive and $D \in \mathbb{R}$ can be represented as

$$\begin{aligned} G(x) &= \min_{y_1 \in \{0,1\}} y_1(L(x) + D) \\ &= \min_{y_1 \in \{0,1\}} \left[\sum_s l_s x_s y_1 + D y_1 \right], \end{aligned} \quad (22)$$

where the quadratic term in (x, y) is submodular. In the case when all coefficients l_s are non-negative, it can be represented as

$$\begin{aligned} &\min_{y_1 \in \{0,1\}} (1 - y_1)(L(x) + D) \\ &= \min_{y_1 \in \{0,1\}} L(x) + D + \left[\sum_s -l_s x_s y_1 - D y_1 \right], \end{aligned} \quad (23)$$

where the quadratic term is again submodular. This allows us to represent also function $\min(L^1(x) + D^1, L^2(x) + D^2) = \min(L^1(x) - L^2(x) + D^1 - D^2, 0) + L^2(x) + D^2$, under the condition that coefficients of $L^1 - L^2$ are all non-positive or all non-negative.

When coefficients of $L^1 - L^2$ are of indefinite sign it seems impossible to represent $\min(L^1, L^2)$. To give an example, why it is so restrictive, consider $G(x_1, x_2) = \min(x_1, x_2)$. This function is not submodular, indeed, $G(1, 1) + G(0, 0) = 1 \not\geq 0 = G(1, 0) + G(0, 1)$ and therefore can not be represented as min-projection of a submodular one.

Now, consider a more limited case, when $L^1 = a^1 \sum_s l_s x_s + b^1$ and $L^2 = a^2 \sum_s l_s x_s + b^2$, where $l_s \geq 0$. Then $\min(L^1, L^2)$ is always representable, because coefficients $L^1 - L^2$ are all either positive or negative, depending on the sign of $a^1 - a^2$. It is also easy to see that any $\mathbb{R} \rightarrow \mathbb{R}$ concave piecewise linear function of $\sum_s l_s x_s$ can be represented as sum of minima where each minimum is of two linear functions satisfying the conditions and thus it is itself representable.

B Accounting for overlap

Let $bnd(\mathbf{x})$ be the set of boundary locations. It is easy to see that for a closed discrete contour, the number of points on the contour, $|bnd(\mathbf{x})|$ is the same as the number of edges in a 4-connected grid (i.e. the number of neighboring pixels with different labels). Clearly, if $E_h(\mathbf{x})$ is approximating the cost of the curvature in the neighborhood of locations h , then the sum (14) will be an inaccurate approximation of the continuous integral, at least because it measures the length in the 4-connected graph metric. However, because each 8×8 pattern actually “sees” a larger neighborhood of the

boundary, its weights may be adjusted so that the sum of pattern costs approximate the desired integral better. For example, weights of the pattern matching a diagonal line at 45 degrees may be scaled by $\frac{1}{\sqrt{2}}$. Because neighboring patterns overlap and because they have to model any arbitrary complex boundary, such an adjustment has to be done jointly for all patterns. Note, the main focus of [4] was to address learning of overlapping terms, in the context of field of experts.

We now propose a second training method, called Algorithm 2, which attempts to deal with these problems. The goal is to adjust the pattern weights such that the total curvature cost of a shape is approximated as well as possible.

In particular, given a set of larger images, indexed by i , e.g. of size 100×100 , where each image depicts a different continuous shapes S^i , with discretization \mathbf{x}^i , and total curvature t^i , we now formulate the learning problem as:

$$\arg \min_{\mathbf{w}, \mathbf{c}} \sum_i \left| \sum_h E_h(\mathbf{x}^i) - t^i \right|. \quad (24)$$

Let us use the same trick as in Alg. 1. If we linearize all terms E_h by fixing the current best pattern for each location in each image, the problem simplifies. We therefore can iterate the following two steps:

Algorithm 2

Input: $\mathbf{x}^i, t^i, \mathbf{w}, \mathbf{c}$

1. For all training images i find matching patterns at all locations $h \in \mathcal{U}$:

$$y_h^i = \arg \min_y \left[\sum_{v \in V(h)} w_{y_h^i, v} x_v^i + c_{y_h^i} \right] \quad (25)$$

2. Refit all patterns:

$$\begin{aligned} (\mathbf{w}, \mathbf{c}) = & \\ \arg \min_{\mathbf{w}, \mathbf{c}} \sum_i & \left| \left(\sum_h \sum_{v \in V(h)} w_{y_h^i, v} x_v^i + c_{y_h^i} \right) - t^i \right| \\ \text{s.t. } & \left[\min_{\mathbf{x} \in \{0,1\}^{K^2}} \mathbf{w}_j^T \mathbf{x} \right] + c_j \geq 0 \quad \forall j \in P. \end{aligned} \quad (26)$$

The constraint in the second step enforces that the pattern cost for each arbitrary labeling is at least 0, which is the lowest value of the function f .

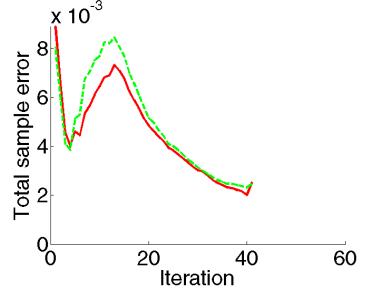


Figure 15: Objective of (24) during iterations of Alg. 2. Red curve is evaluated on training data and green dashed curve is evaluated on the independent test data.

The main difference to Alg. 1 is that the refitting step does no longer decouple into estimating patterns weights independently. Step (26) will now be solved with an LP.

We tested Alg. 2 to retrain costs \mathbf{c} on the class of Fourier shapes (see fig 7g(right)) so as the total cost assigned by the model would match better to the true cost. It is initialized with patterns (\mathbf{w}, \mathbf{c}) which were learned using Alg. 1. Here we kept the weights \mathbf{w} fixed and only update the weights \mathbf{c} in step 2. Figure 15 shows the progress of Alg. 2. Figure 16 evaluates the new trained model on independent samples of circles and Fourier shapes. We see that the bias of the model to overestimate the total cost was reduced, compared to fig. 7(d,e). As expected, this is especially true for the Fourier shapes, since only Fourier shapes were used in training. Note, since retraining with Alg 2 gave only a mild improvement in approximation error, we did not use these patterns, but rather the ones obtained after Alg. 1 for all of our experiments.

References

- [1] Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: a randomized correspondence algorithm for structural image editing. ACM Trans. Graph. **28** (2009) 1
- [2] Lauritzen, S.L.: Graphical Models. Oxford Statistical Science Series. Oxford Science Publications (1998) 1
- [3] Hinton, G.E.: Products of experts. In: ICANN. (1999) 2
- [4] Roth, S., Black, M.: Fields of experts. IJCV **82** (2009) 2, 15
- [5] Kohli, P., Kumar, M.: Energy minimization for linear envelope MRFs. In: CVPR. (2010) 2

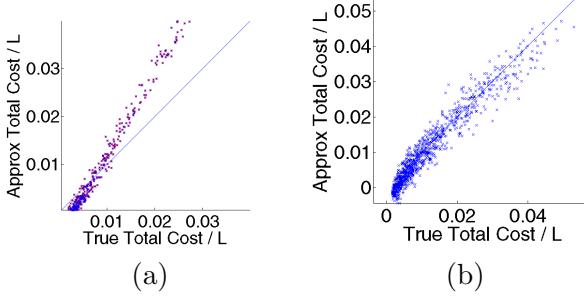


Figure 16: Results after applying Alg. 2 to refit constant terms. We show approximate total cost / length vs. true total cost / length, for circles (a) and Fourier shapes (b) respectively. As can be seen the bias is removed

- [6] Kohli, P., Ladicky, L., Torr, P.: Robust higher order potentials for enforcing label consistency. *IJCV* **82** (2009) 2, 4, 14
- [7] Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? *PAMI* **26** (2004) 2
- [8] Ramalingam, S., Kohli, P., Alahari, K., Torr, P.: Exact inference in multi-label CRFs with higher order cliques. In: *CVPR*. (2008) 2
- [9] Rother, C., Kohli, P., Feng, W., Jia, J.: Minimizing sparse higher order energy functions of discrete variables. In: *CVPR*. (2009) 2, 3, 4, 9
- [10] Gupta, R., Diwan, A.A., Sarawagi, S.: Efficient inference with cardinality-based clique potentials. In: *ICML*. (2007) 2
- [11] Komodakis, N., Paragios, N.: Beyond pairwise energies: Efficient optimization for higher-order MRFs. In: *CVPR*. (2009) 2, 3, 4
- [12] Tarlow, D., Zemel, R., Frey, B.: Flexible priors for exemplar-based clustering. In: *UAI*. (2008) 2
- [13] Nowozin, S., Lampert, C.: Global connectivity potentials for random field models. In: *CVPR*. (2009) 2
- [14] Vicente, S., Kolmogorov, V., Rother, C.: Joint optimization of segmentation and appearance models. In: *ICCV*. (2009) 2
- [15] Bertozzi, A., Esedoglu, S., Gillette, A.: Inpainting of binary images using the Cahn-Hilliard equation. *IP* **16** (2007) 5
- [16] Chan, T.F., Shen, J.: Non-texture inpainting by curvature-driven diffusions (CDD). *JVCIR* **12** (2001) 5
- [17] Esedoglu, S., March, R.: Segmentation with depth but without detecting junctions. *JMIV* **18** (2003) 5
- [18] Schoenemann, T., Cremers, D.: Introducing curvature into globally optimal image segmentation: Minimum ratio cycles on product graphs. In: *ICCV*. (2007) 5

- [19] Schoenemann, T., Kahl, F., Cremers, D.: Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation. In: *ICCV*. (2009) 5, 6, 9, 12, 13
- [20] El-Zehiry, N.Y., Grady, L.: Fast global optimization of curvature. In: *CVPR*. (2010) 6
- [21] Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. *PAMI* **28** (2006) 7, 8, 9
- [22] Wainwright, M., Jaakkola, T., Willsky, A.: Exact MAP estimates by (hyper)tree agreement. In: *NIPS*. Volume 15. (2003) 7
- [23] Kolmogorov, V., Rother, C.: Comparison of energy minimization algorithms for highly connected graphs. In: *ECCV*. (2006) 8
- [24] Werner, T.: A linear programming approach to max-sum problem: A review. *PAMI* **29** (2007) 8
- [25] Boykov, Y., Jolly, P.: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In: *ICCV*. (2001) 12
- [26] Bruckstein, A., Netravali, A., Richardson, T.: Epiconvergence of discrete elastica. *Applicable Analysis* **79** (2001) 13