Non-parametric Higher-order Random Fields for Image Segmentation

Pablo Márquez-Neila[1], Pushmeet Kohli[2], Carsten Rother[3], and Luis Baumela[1]

[1]Universidad Politécnica de Madrid [2]Microsoft Research
[3]Technische Universität Dresden
{p.mneila, lbaumela}@upm.es, pkohli@microsoft.com,
carsten.rother@tu-dresden.de

Abstract. Models defined using higher-order potentials are becoming increasingly popular in computer vision. However, the exact representation of a general higher-order potential defined over many variables is computationally unfeasible. This has led prior works to adopt parametric potentials that can be compactly represented. This paper proposes a non-parametric higher-order model for image labeling problems that uses a patch-based representation of its potentials. We use the transformation scheme of [11, 25] to convert the higher-order potentials to a pair-wise form that can be handled using traditional inference algorithms. This representation is able to capture structure, geometrical and topological information of labels from training data and to provide more precise segmentations. Other tasks such as image denoising and reconstruction are also possible. We evaluate our method on denoising and segmentation problems with synthetic and real images.

Keywords: random fields; biomedical image analysis; higher-order models; image denoising; image segmentation; structured prediction

1 Introduction

Conditional and Markov random fields (CRF/MRF) are popular models for representing regularized solutions to many computer vision problems, such as object segmentation, optical flow and disparity estimation [28]. One variant of these models, the *pairwise random field*, has been extensively used in computer vision because it allows efficient inference of its Maximum a Posterior (MAP) solution. However, the pairwise random fields only allow the incorporation of statistical relationships between pairs of random variables and are unable to enforce the high-level structural dependencies between pixels that have been shown to be extremely useful for a variety of computer vision problems. Some approaches try to overcome the limitations of pairwise terms with dense, fully-connected CRFs [16].

The last few years have seen the successful application of higher-order CRFs and MRFs to some low-level vision problems such as image restoration, disparity estimation and object segmentation [6, 10, 17, 18, 23, 24, 29]. These models are composed of higher-order potentials, i.e., potentials defined over multiple variables, which have higher modeling power. In general, it is computationally unfeasible to exactly represent a higher-order potential function defined over many variables. Representation of a general m order potential function of k-state discrete variables requires k^m parameter values. This has led researchers to propose a number of parametric families of higher order potentials that can be compactly represented [4, 8–12, 14, 17, 21, 24–26].

In this paper, we propose a non-parametric pattern-based higher-order random field. The higher-order potentials used in our model are defined using a data driven approach. We use a pattern based representation [11, 14, 25] to encode the structure and shape of the labels. This allows us to use the transformation scheme of [11, 25] to convert the higher-order potentials to a general pairwise form that can be handled using traditional inference algorithms such as belief propagation (BP) [23] and tree-reweighted message passing (TRW) [13]. We evaluate the performance of our method in synthetic images, medical images and the MSRCv2 dataset, and compare our results with conventional pairwise energy regularization, a higher-order method [25] and structured random forests [15].

Although we adopt the transformation scheme in [25], the resulting algorithm is more general. The approach in [25] uses higher-order potentials to define a prior for binary texture denoising. These potentials are defined over each patch in the image, encouraging the pixels in the patch to take a joint labeling from a predefined global set of patterns. To make the problem computationally tractable, the size of the global set of patterns is limited to a small number. This makes this approach inadequate for tasks such as segmentation. In contrast, the potentials in our model are conditioned on the data. This means that every potential can choose the most suitable joint labeling from a local set of patterns selected according to the observations. Since this local set can be different for every potential, the global set of patterns that our model considers can be as large as required by the application. Hence, the expressive power of our model is much greater than the one in [25], at the same computational cost.

A number of methods in the literature have also adopted a data-driven philosophy to solve image labeling problems. These methods generally work by finding, for the image patch under consideration, the closest matches in the training dataset [3, 5, 6, 19]. Instead, our approach uses the labeling candidates from the matching patches to define a higher order energy whose minimization performs the label aggregation to obtain a consistent solution.

Higher-order potentials have also been used for curvature regularization. In [22] each potential considers an exhaustive enumeration of possible joint labelings for its pixels. Since the labeling enumeration is exponential in the size of the patches, patches must be small and the potentials can only impose a weak regularization. In our work, however, possible joint labelings are learned from data and are, in consequence, sparser than exhaustive enumeration. This permits larger patches and more expressive potentials.

Our higher order potential can be seen as encoding a higher order likelihood [7] function that takes into account all patches in the training set.

2 Non-parametric Higher-order Random Field (NHRF)

The energy of the pattern-based model for texture denoising is [25]

$$E(\mathbf{y}) = \sum_{i \in \mathcal{V}} \phi_i(\mathbf{y}_i|\mathbf{x}) + \sum_{c \in \mathcal{P}} \phi(\mathbf{y}_c), \tag{1}$$

where $\mathcal{V}$ is the set of pixels of the image, $\mathbf{x}$ is the observed image data, $\mathbf{y}$ is the vector of labels and $\mathcal{P} \subset 2^{\mathcal{V}}$ represents a set of cliques in the pixels of the image. In this model there are two kinds of potentials: unary potentials ϕ_i defined over individual pixels and higher-order potentials ϕ defined over many pixels. The expressions $\mathbf{x}_c$ and $\mathbf{y}_c$ represent the elements of the image $\mathbf{x}$ and the labeling $\mathbf{y}$ that correspond to the clique c. Notice that only the unary potentials ϕ_i, and not the higher-order potentials, are dependent on the data $\mathbf{x}$.

In our model, however, the higher order potentials defined over a set of variables directly depend on the pixel observations. The energy of our model is a sum of higher order potentials ϕ_c,

$$E(\mathbf{y}) = \sum_{c \in \mathcal{P}} \phi(\mathbf{y}_c \mid \mathbf{x}_c) = \sum_{c \in \mathcal{P}} \phi_c(\mathbf{y}_c), \tag{2}$$

The cliques in $\mathcal{P}$ can overlap, have different sizes and shapes and be centered on any pixel. For simplicity we work with square, fixed sized and overlapping cliques centered on a grid of pixels with a given separation among them, that we call *stride*. This layout has proven to be powerful enough for all our experiments. The size of the $m \times m$ clique and the stride of the grid s are hyper-parameters of the model. It is required that $s < m$, or otherwise there would be pixels not affected by any clique. The order of the energy is the number of pixels of every clique in $\mathcal{P}$, i.e., m^2. Figure 1(a) shows the factor graph of our model.

The key idea for our higher-order potentials ϕ_c is that they have a data-driven, non parametric representation based on a set of $m \times m$ patterns $\mathcal{Y} = \{\mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \ldots, \mathbf{Y}^{(t)}\}$. As a first approach, the potential $\phi_c(\mathbf{y}_c)$ is defined so that $\mathbf{y}_c$ can only be equal to one of the patterns in $\mathcal{Y}$. Otherwise, if the value of $\mathbf{y}_c$ is not in $\mathcal{Y}$, $\phi_c(\mathbf{y}_c)$ will be infinity. To encode the fact that not all patterns equally suit the observations $\mathbf{x}_c$ of each patch, we use a set of costs $\{\theta_1^c, \ldots, \theta_t^c\}$ associated to each pattern. The cost θ_q^c will be small when the pattern q provides a good explanation for the observations $\mathbf{x}_c$.

Given these considerations, we could define the potential ϕ_c as

$$\phi_c(\mathbf{y}_c) = \begin{cases} \theta_q^c & \text{if } y_i = Y_i^{(q)} \quad \forall i \in c \\ \infty & \text{otherwise} \end{cases}. \tag{3}$$

Hence, the variables of every patch can only have values that perfectly match one of the patterns in $\mathcal{Y}$, which we will call the *active pattern* of the patch. This model is constrained to use the patterns in $\mathcal{Y}$, something very restrictive in practice. To alleviate this restriction we will also allow deviations from patterns $\mathcal{Y}$, but we will penalize those deviations using a set of *deviation cost* functions $d_1^c, \dots, d_t^c : \mathcal{L}^{|c|} \to \mathbb{R}$. Since their input is discrete, they can be defined as:

$$d_q^c(\mathbf{y}_c) = \sum_{i \in c, l \in \mathcal{L}} w_{qil}^c \delta(y_i = l), \tag{4}$$

where w_{qil}^c is the cost of assigning the label l to the variable y_i of the clique c when that clique is considered to be associated to the pattern $\mathbf{Y}_q^c$. When a variable has no deviation from the active pattern, the corresponding deviation cost is 0.

With the deviation functions we can define our potentials as

$$\phi_c(\mathbf{y}_c) = \min_{q \in \{1,\dots,t\}} \theta_q^c + d_q^c(\mathbf{y}_c), \tag{5}$$

where θ_q^c and d_q^c depend on observation $\mathbf{x}_c$. Thus, given a labeling $\mathbf{y}_c$, the potential $\phi_c(\mathbf{y}_c)$ will be the cost of the best pattern for the labeling plus the costs of the deviations from that pattern.

Since the patches overlap, a pixel i can be included in multiple patches, and it may occur that the labelings for those patches do not agree. The energy minimization solves these disagreements by assigning to y_i the label that minimizes the sum of deviations of the potentials that share the pixel.

The higher-order random field defined in this section does not specify the structure model of the problem. Instead, the structure itself, and not only a set of parameters, is learned from data. Thus, this is a *non-parametric higher-order random field* (NHRF).

2.1 Transformation to a Pair-wise Form

The energy (2) cannot be minimized directly. Instead, we use the sparse nature of the potentials to transform the higher-order energy into a pairwise one by introducing a pattern selection variable $z_c \in \{1, \dots, t\}$ for every patch c. This variable selects the active pattern in that patch. This allows the transformation of potentials to the equivalent form

$$\phi_c(\mathbf{y}_c) = \min_{z_c} h_c(z_c) + \sum_{i \in c} g_c(z_c, y_i), \tag{6}$$

that has only unary and pairwise terms. The unary term $h_c(z) = \theta_z^c$ encodes the cost of choosing the pattern z, and the pairwise terms $g_c(z, y_i) = w_{ziy_i}^c$ encode the deviation costs. Figure 1 depicts how this transformation changes the appearance of the factor graph.

The global energy function (2) becomes

$$E(\mathbf{y}) = \min_{\mathbf{z}} \sum_{c \in \mathcal{P}} h_c(z_c) + \sum_{i \in c} g_c(z_c, y_i), \tag{7}$$

and computing the MAP estimation $\mathbf{y}^* = \operatorname{argmin}_{\mathbf{y}} E(\mathbf{y})$ is just a minimization of a pairwise energy over the labels $\mathbf{y}$ and the selection variables $\mathbf{z}$. We resort to the standard inference algorithms tree-reweighted (TRW) message passing and belief propagation (BP) to minimize it.

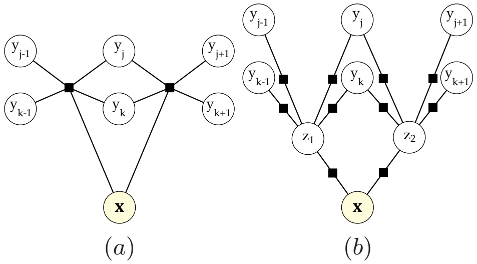


Fig. 1. (a) Factor graph of a simple higher-order random field. (b) Transformation of the higher-order random field to a pairwise random field.

3 Training Non-parametric Higher-order Random Fields

Just like other structured prediction methods, our model requires pairs $(\mathbf{x}, \mathbf{y})$ of images and their corresponding segmentations for training. The learning consists on inferring the pattern set $\mathcal{Y}$ from data, as well as a method for estimating the costs of the patterns for an observation of a patch $\mathbf{x}_c$. We could also learn the deviation costs w^c_{qil} from data, but the huge amount of parameters needed would complicate both the model and the learning algorithm. We have seen in our experiments that a single cost α for all deviations,

$$w^c_{qil} = \begin{cases} 0 & \text{if } l = Y^{(q)}_i \\ \alpha & \text{otherwise} \end{cases}, \tag{8}$$

suffices for all practical cases. The deviation cost α is a hyper-parameter of our model. It could be learned by cross-validation, but we have verified in our experiments that large changes in α affect little or nothing the results. Therefore, an arbitrary value such as $\alpha = 1$ is typically used.

The training data consists of a set of images $\mathcal{M} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(p)}\}$ and their labelings $\mathcal{N} = \{\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(p)}\}$. From training data, we extract many pairs of $m \times m$ image and label patches. We consider all overlapping patches centered on a grid of pixels with a given stride. We will call $\mathcal{X} = \{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(t)}\}$ to the set of patches extracted from training images, and the corresponding set of patches extracted from labelings is the set of patterns $\mathcal{Y}$ described in previous section.

The patches in $\mathcal{X}$ and $\mathcal{Y}$ are used to estimate the costs when a new patch $\mathbf{x}_c$ from a testing image arrives. The costs θ^c_q are computed by a dissimilarity function

$$\theta^c_q = d(\mathbf{x}_c, \mathbf{X}^{(q)}, \mathbf{Y}^{(q)}) \tag{9}$$

that measures how different the testing patch $\mathbf{x}_c$ is from the training patch with image $\mathbf{X}^{(q)}$ and labeling $\mathbf{Y}^{(q)}$. Therefore, equation (9) assigns higher costs to patterns with higher dissimilarity with $\mathbf{x}_c$. This framework is very flexible and many different dissimilarity functions can be considered depending on the characteristics of the problem. We will discuss two of them.

Perhaps the simplest approach consists on ignoring the dependence on the labeling patch and performing the dissimilarity computations only in the space of image patches:

$$\theta_q^c = d_X(\mathbf{x}_c, \mathbf{X}^{(q)}). \tag{10}$$

The function d_X is in this case just a metric in the space of image patches. This metric could be the standard Euclidean distance but, unless the relationship between images and labels is simple, more involved alternatives such as learned metrics are required. An interesting advantage of using d_X as the dissimilarity function is that, as a metric, it permits using a kd-tree for fast search of patches.

A second, more interesting approach is dropping the dependence on the image patch and computing the dissimilarity directly with the labeling patch that will be used for defining the corresponding higher-order potential:

$$\theta_q^c = d_Y(\mathbf{x}_c, \mathbf{Y}^{(q)}). \tag{11}$$

Since d_Y is not a metric, we need a way to relate the image observations from patch $\mathbf{x}_c$ to the labels in $\mathbf{Y}^{(q)}$. We model this relationship with a probability function $P(\mathbf{Y}^{(q)} \mid \mathbf{x}_c)$, that estimates the probability that the pattern $\mathbf{Y}^{(q)}$ explains the observation $\mathbf{x}_c$, and define

$$d_Y(\mathbf{x}_c, \mathbf{Y}^{(q)}) = -\log P(\mathbf{Y}^{(q)}|\mathbf{x}_c). \tag{12}$$

To deal with the number of parameters needed by this probability function, we assume independence between variables, what leads to the factorization:

$$P(\mathbf{Y}^{(q)} \mid \mathbf{x}_c) = \prod_i P(y_i^{(q)} \mid \mathbf{x}_c). \tag{13}$$

This may seem a very strong assumption, since the labels in a pattern are strongly correlated. However, label dependencies are already implicitly encoded in the set of possible patterns $\mathcal{Y}$ and we do not need to learn them again in the joint probability.

Every factor $P(y_i^{(q)} \mid \mathbf{x}_c)$ is just the probability of a label in a single pixel given the observations. A pixel-wise classifier is responsible of learning this probability. In our experiments, we have used a variety of pixel-wise classifiers ranging from simple Gaussian classifiers to random forests.

A very convenient consequence of dropping the dependence on $\mathbf{X}^{(q)}$ from the dissimilarity function is that the cardinality of the pattern set $\mathcal{Y}$ can be greatly reduced. Indeed, it is not necessary to store the image patches from $\mathcal{X}$ to compute the costs with d_Y. Also, the set $\mathcal{Y}$ has lots of very similar or repeated elements. Therefore, a clustering on $\mathcal{Y}$ is able to reduce the number of patterns by removing duplicates and similar patterns.

Inference Given a new testing image $\mathbf{x}$, first the set of cliques $\mathcal{P}$ must be defined. Then, for every patch $\mathbf{x}_c$ corresponding to a clique c in the testing image, we should compute the costs, d (9), of all patches extracted in the training step and build the higher-order potentials of the random field with them. However, such an amount of data per potential function is prohibitive in terms of memory.

In practice, we do not keep all t costs and patterns in every higher-order potential ϕ_c, but only a subset of the t' patterns with the lowest costs, with $t' \ll t$. The subset of t' patterns with lowest costs for a clique c of the testing image will be called the set of *local patterns*, or *candidates*, of that clique. The rest of patterns in $\mathcal{Y}$ with larger costs are simply ignored, assuming than their costs are too large to be considered. For reference, t' is in the order of tens or, at most, hundreds.

The NHRF is defined once every potential ϕ_c has been fully determined with its candidates and their associated costs. Then, it is converted to pair-wise form as defined in Section 2.1 and inference is performed via energy minimization with TRW or BP.

To clarify previous discussion, Figure 2 shows an example of the training of a NHRF and Figure 3 shows some details of a NHRF built for a testing image.

4 Experiments

By using patterns previously seen in training data, the NHRFs implicitly integrate high-order geometric and topological information. We have conducted several experiments with both synthetic and real images to assess the power of the NHRFs in image denoising and segmentation, respectively.

4.1 Occluded Squares

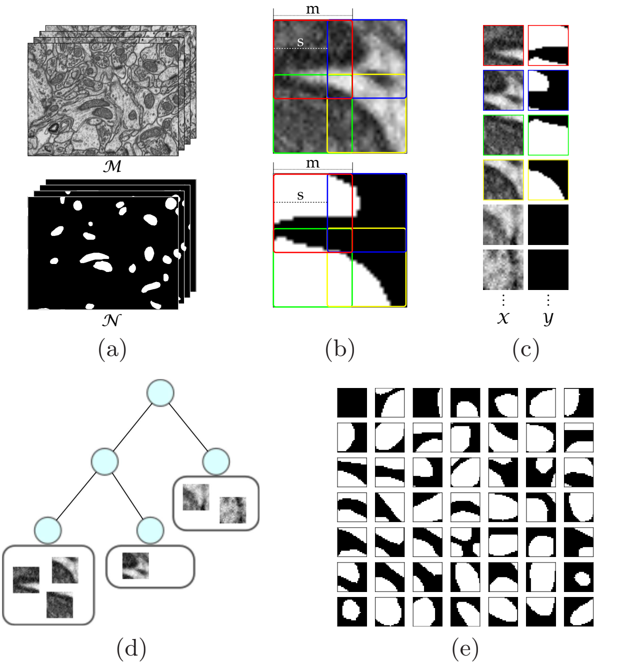
In this experiment we analyze the performance of NHRFs and compare it with other approaches using synthetic images. These images feature occlusions and a high level of noise. We use a dataset of 150×150 images with 50×50 squares in several orientations. The images are highly perturbed with noise and circular holes that *occlude* the squares. Figure 4(a,b) shows some images of the testing dataset and the corresponding labelings.

We use 500 images with their labelings for training. We extract all possible 21×21 patches with a stride of 2 pixels. We end up with more than 2 million image and label patches. Since we will use the Euclidean distance in the space of image patches (i.e., d_X) as our dissimilarity function, we build a kd-tree with the image patches.

For a new image, we define our set of cliques $\mathcal{P}$ as all the 21×21 squared cliques with a stride $s = 2$ pixels. For every clique, we look for the nearest $t' = 15$ candidates in the space of image patches.

Figure 4(d) shows the segmentation obtained for a test image. The accuracy of the method is 99.7%, and the average Jaccard index is 97.28%. The NHRF model is able to discover that circular structures are not part of the objects of interest, while straight lines and corners are.

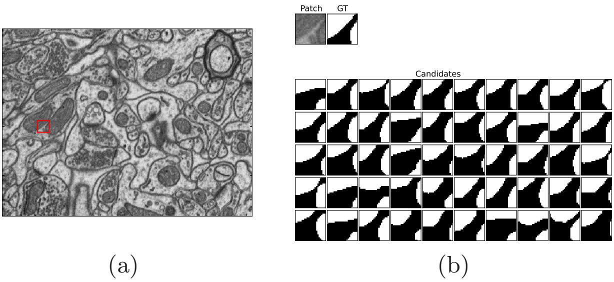
We have also segmented these images using a pixel-wise random forest classifier trained with the same training dataset. The results of the classifier are then



(a) (b) (c)

(d) (e)

Fig. 2. Training of a NHRF. Given the training data (a), we extract all $m \times m$ patches with stride s as shown in (b). For clarity, (b) shows only a small 40×40 fragment from training data. In this example, $m = 24$ and $s = 16$ pixels. The patches extracted from images and labelings form the sets $\mathcal{X}$ and $\mathcal{Y}$, respectively. (c) shows some elements of these sets. When costs are estimated with dissimilarity d_X, a kd-tree with elements of $\mathcal{X}$ is built (d). When d_Y is used instead, clustering over $\mathcal{Y}$ is performed to obtain a reduced set of patterns without repeated elements. (e) shows some patterns obtained after clustering.



(a) (b)

Fig. 3. Example of a higher-order potential in a NHRF. Given a testing image (a), $m \times m$ cliques are configured in a grid with stride s in a similar way to Figure 2(b). For every clique, we look for the t' patterns with lowest costs. These are the candidates for those cliques. (b) shows the candidates found for the the clique marked in (a).

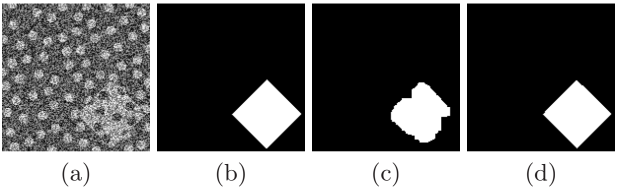


Fig. 4. Results for the square dataset. (a) Testing images. (b) Ground-truth. (c) Results with a random forest pixel-wise classifier with pairwise regularization. (d) NHRF.

regularized minimizing a 4-connected pair-wise energy via graph-cuts [2]. The capacities of the edges are equal across the grid and set via cross-validation. The accuracy reached with pair-wise regularization is 98.45% and the Jaccard index is 86.89%. Figure 4(c) shows the results obtained. Despite the good quantitative results, this figure proves that pixel-wise classifiers and pair-wise regularization are not powerful enough to regularize the segmentation of objects where high-level shape information is present. Instead, the NHRF method exploits that kind of information to obtain better labelings with straight lines and sharp corners. Moreover NHRFs do not present the undesirable effect of the *shrinking bias* and *metrication errors* present in pair-wise regularization (see Figure 4(c)).

4.2 Binary Image Reconstruction and Denoising

In this experiment we examine the performance in image reconstruction and denoising. We use the Brodatz D101 texture shown in Figure 5. A fragment of that texture is used both as the training image $\mathcal{M} = \{\mathbf{x}^{(1)}\}$ and as labeling image $\mathcal{N} = \{\mathbf{y}^{(1)}\}$ (Figure 5(a)). A different fragment is selected as ground-truth (Figure 5(c)). The ground-truth is perturbed with 30% of noise, (Figure 5(b)) and used as input. The patch size is $m = 10$ pixels, and the stride between consecutive patches is $s = 1$ pixel. The costs of the patterns θ_q^c are computed using the dissimilarity d_X. The deviation cost is set to an arbitrarily large number $\alpha = 1000$. The resulting energy is finally minimized using TRW. Figure 5(f) shows the result of the minimization.

We also reconstructed the input image in Figure 5(b) using, first, a standard pair-wise regularization approach with 4-connected pixels, submodular terms and equal capacities for all edges and, second, the global patterns algorithm introduced in [25]. Figure 5(e,f) shows the results obtained with these methods.

Quantitative pixel error of the reconstruction for the NHRF is 5.96%. The error of pair-wise regularization and global patterns are respectively 9.68% and 9.32%.

Qualitatively, the results show that pair-wise regularization does not maintain the overall image structure. The global pattern method of [25] partially maintains the image structure, although it makes some small holes disappear. Moreover, the reconstruction also keeps part of the noise, specially visible in the jagged image boundaries, due to the unary terms. The reconstruction obtained

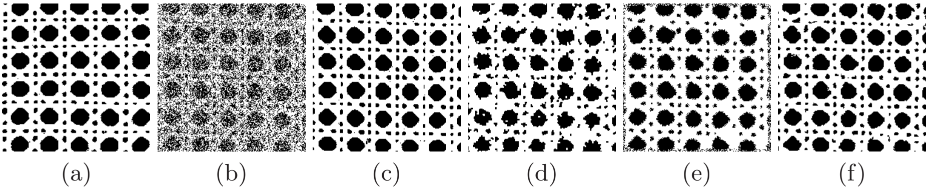


Fig. 5. Example of binary image reconstruction. The training is done with image (a) and the testing with image (b), whose ground-truth is (c). (d) Results with pair-wise regularization, (e) global patterns [25], (f) NHRF. The size of all images is 200 × 200 pixels.

with the NHRF is superior to the others both in terms of maintenance of image structure and noise removal.

This experiment is a good example of the limitations of the global patterns [25]. As discussed in Section 1, the number of global patterns is restricted by practical considerations. In this experiment the chosen number of global patterns, 50, cannot model the variety of shapes and structures that occur in the simple repetitive texture under consideration. In the segmentation of real world images this limitation is expected to be even more pronounced. However, for the NHRF, 50 local patterns for every potential chosen according to the observations, are more than adequate for this problem.

4.3 Mitochondria Segmentation

The segmentation of electron microscopy (EM) imaging of the brain is one of the areas where the NHRFs can provide better results. In the first place, brain structures such as mitochondria have a very characteristic shape and topology: they are simply connected structures with no holes and tubular-like shapes. The NHRFs capacity for learning shape and topology makes them a suitable tool for this application.

We will use the EM dataset from [20]. This dataset is a labeled sample of the rat hippocampus (see Figure 2(a)). The dataset is divided in two stacks of the same size for training and testing. Each stack consists of 165 slices with size 384 × 512 pixels.

We use the the dissimilarity function d_Y to estimate the costs of candidates. From the labeling slices of the training stack we extract all $m \times m$ patches with a stride of s pixels. We have performed experiments with several patch sizes. For reference, in the case of $m = 24$ pixels this gives about 2 million patches. After clustering, the number of patterns drops to 12261. Some of them are shown in Figure 2(e). For our pixel-wise classifier, we use boosted context cues [1], a set of features that has proved to perform specially well for the segmentation of synaptic junctions and other brain structures.

We compare the NHRF method with the performance of the pixel-wise classifier with and without pair-wise regularization. We also include the *structured random forest* (SRF) method from [15] in our comparison. The SRF is an extension to the standard random forest that aims to integrate structural label information of the images to segment. Table 1(left) presents results for some combinations of patch size m and deviation cost α.

As expected, the NHRF model improves the results obtained by the pixel-wise classifier alone and with pair-wise regularization. These results are, to the best of our knowledge, the state-of-the-art in this dataset.

The performance of the SRF is poor in this dataset. This could be attributed to the fact that SRFs are good learning the relative position and relations among the labels, but they do not learn the shapes and topological features of each label, as the NHRFs do.

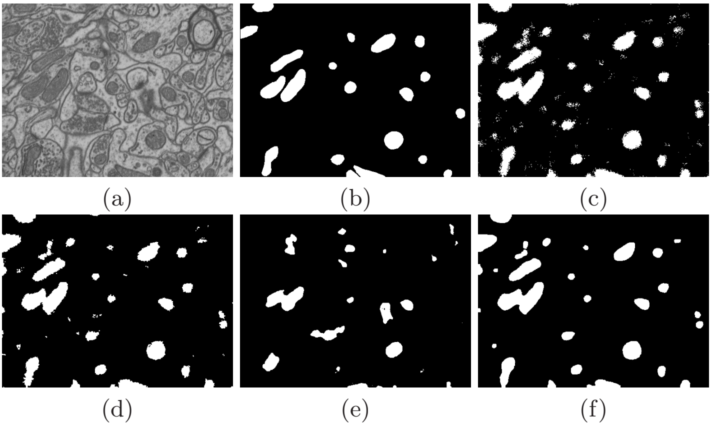
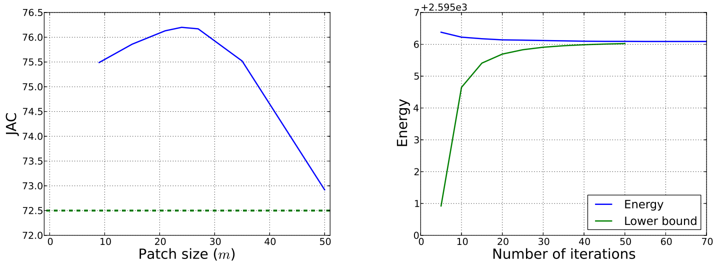


Fig. 6. Results for the hippocampus dataset. (a) Testing image. (b) Ground-truth. (c) Pixel-wise classifier. (d) Pair-wise regularization. (e) Structured random forest [15], (f) NHRF ($m = 24$ pixels, $\alpha = 1$).

Although the quantitative results prove the good performance of the NHRFs, the qualitative results give complementary insights. In Figure 6 the effects of the higher-order regularization are very noticeable. The regions obtained with pair-wise regularization do not resemble the appearance of real mitochondria. The boundaries are ragged and background regions with arbitrary shapes are still present. The NHRF leads to much more realistic looking results. Most regions have smooth, rounded boundaries like real mitochondria. With the SRF the shapes of the regions look less alike real mitochondria.

The running time depends on the size of the patch. For the best size $m = 24$, the inference takes around 10 minutes per image using the TRW implementation from [13]. For $m = 15$ this time falls to 4 minutes.



(a) (b)

Fig. 7. (a) Jaccard score vs. patch size m. The horizontal dashed line marks the performance of the pixel-wise classifier. (b) Energy and lower bound evolution during the inference procedure of a slice. Both plots are for the mitochondria dataset.

Figure 7 clarifies some aspects of our method. The patch size m noticeably affects to the results of the segmentation (see Table 1(a)). However, the method is robust to the choice of m, providing similar performance in a broad range of values ($m \in [10, 30]$) as seen in Figure 7(a). The performance falls for larger patch sizes since the available training data is insufficient to provide a representative sample of the space of labeling patches. However, even for patches as large as $m = 50$ the NHRF performs better than the pixel-wise classifier.

Missing the energy optimum is another issue commonly raised with minimization methods such as TRW or BP. Figure 7(b) plots the evolution of the energy and the lower bound computed by TRW for a slice of the mitochondria dataset. The energy comes very close to the lower bound. This has been the case in all our experiments.

4.4 MSRCv2 Dataset

The MSRCv2 dataset [27] consists of 591 images annotated with 21 classes. The annotations are coarse and incomplete, with areas marked as *void* where none of the classes is valid. The 591 images are split into 315 training and 276 testing images (roughly 55%-45%).

From the training images we extract all patches centered at pixels of a grid with a stride of 5 pixels. We repeat this procedure with two different values for the patch size m: 11 and 21 pixels. This leads to approximately 800K patches extracted for $m = 11$ pixels and 750K for $m = 21$ pixels. As in the previous experiment, we use the dissimilarity d_Y to compute costs. After clustering, we obtain $t = 8417$ patterns for the patch of size $m = 11$ pixels and $t = 19549$ patterns for the patch of size $m = 21$ pixels.

From the training data we also train a random forest classifier with HOG, texture and color features. The HOG features are extracted with 6×6 pixels per cell and 4×4 cells per patch, with 9 bins for each histogram. For every cell we also include 4 texture descriptors and 2 color values (the a and b channels of the

CIELAB color space), leading to a total of $(9+4+2) \times 16 = 240$ dimensions of the feature vector.

We perform segmentation and regularization of the testing images with NHRFs for the mentioned patches size, and for different values of the deviation costs. Table 1(right) summarizes the results and compares the performance with other algorithms. In this table we report the global accuracy (the percentage of pixels that were correctly classified) and the average Jaccard index over all classes. As in [15], we ignore the pixels annotated as *void* in the ground-truth from the estimation of all validation metrics.

Table 1. Quantitative results for hippocampus (left) and MSRCv2 (right) datasets. Numbers in parentheses indicate parameters (m, α) of the NHRF.

Hippocampus			**MSRCv2**		
Method	JAC	ACC	Method	Avg. JAC	ACC
Boost context cues (BCC)	72.50	98.20	Random forest (RF)	23.6	56.6
BCC+pw regularization	73.62	98.31	RF+pw regularization	24.6	58.4
BCC+NHRF (24, 1)	**76.20**	**98.51**	RF+NHRF (11, 1)	24.5	58.6
BCC+NHRF (24,100)	75.94	98.49	RF+NHRF (21, 1)	25.7	**59.9**
BCC+NHRF (50, 1)	72.92	98.32	RF+NHRF (21, 10)	25.7	59.9
SRF [15]	31.68	94.27	SRF [15]	**27.0**	57.6

We also compare our results with the SRF [15]. We use the same training parameters given in [15]: feature patch size of 24×24, 10 trees and 500 iterations per node stopping when less than 5 samples per leaf were available. The results obtained with this method are better for this dataset than for the hippocampus. This is reasonable, since this dataset relies on the relative positions of the classes much more than on the shapes of the classes. In fact, the shapes and geometry of the classes are rather unimportant in this dataset due to the coarse labeling of the training data. This affects negatively the performance of the NHRF method, which is very dependent on shapes. Nevertheless, the NHRF results are still compelling. This proves that they are also able to learn and make use of the relative positions of the classes in a similar way as the structured random forests do. Moreover, the MSRCv2 dataset has been manually segmented in a coarse way, with loose boundaries and imprecise shapes. The Jaccard index is very sensitive to differences in the segmentation of boundaries with respect to the ground-truth, so this affects its reliability in this particular dataset. Hence, the performance differences related to this index are not very informative.

Figure 8 shows a qualitative comparison of the segmentations obtained with different methods for several images. The NHRFs get good results in many images of the dataset. Thanks to the learned shapes and relative positions of labels, they are able to overcome the noisy segmentations produced by the random forests, and in many cases their results are better than the ones obtained with the SRFs.

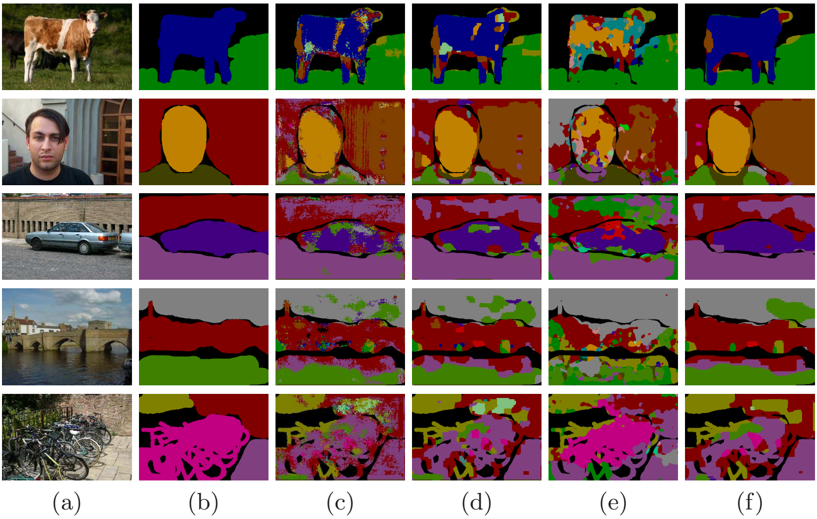


Fig. 8. Qualitative results. (a) Test images. (b) Ground-truth. (c) Random forests. (d) Pair-wise regularization. (e) Structured random forests [15]. (f) NHRF with $m = 21$ pixels and $\alpha = 1$. The last row presents a difficult case where the NHRF fails.

5 Conclusions

Higher-order potentials are required to capture structural, geometric and topological information that weaker pair-wise potentials are unable to exploit. However, parameterizing higher-order potentials is hard. In this paper we propose to use a soft and sparse representation of higher-order potentials based on a set of patterns extracted from training data. Our higher-order potentials are directly conditioned on data and no unary terms are required. This allows us to define a set of local patterns for every higher-order potential, making our method more expressive than approaches with global patterns.

A NHRF is defined as the sum of these higher-order potentials. The inference procedure in a NHRF is constrained to use the patterns of its potentials with small deviations to build the resulting labeling.

Our experiments prove, both in synthetic and real datasets, that NHRFs provide better results than pixel-wise classifiers alone and with pair-wise regularization. Moreover, our results are comparable or better than those of the SRF, that was designed to learn labeling structure, but not shape or topology. The NHRFs have also applications in areas other than segmentation, such as image denoising and reconstruction, where they get appealing results.

Acknowledgments This research was funded by the *Cajal Blue Brain Project.* The authors thank Carlos Becker for the Boosted Context Cues code, Peter Kontschieder for the Structured Random Forests code and Graham Knott for providing the mitochondria dataset.

References

1. Becker, C.J., Ali, K., Knott, G., Fua, P.: Learning Context Cues for Synapse Segmentation. IEEE Transactions on Medical Imaging 32(10), 1864–1877 (2013)
2. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 26 (2004)
3. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: Proc. International Conference on Computer Vision and Pattern Recognition (CVPR). pp. 60–65 (2005)
4. Delong, A., Osokin, A., Isack, H.N., Boykov, Y.: Fast approximate energy minimization with label costs. In: Proc. International Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2173–2180 (2010)
5. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: SIGGRAPH. pp. 341–346 (2001)
6. Fitzgibbon, A., Wexler, Y., Zisserman, A.: Image-based rendering using image-based priors. In: Proc. International Conference on Computer Vision (ICCV). pp. 1176–1183 (2003)
7. Glocker, B., Heibel, T.H., Navab, N., Kohli, P., Rother, C.: Triangleflow: Optical flow with triangulation-based higher-order likelihoods. In: Proc. European Conference on Computer Vision (ECCV). pp. 272–285 (2010)
8. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Computation 14(8), 1771–1800 (2002)
9. Hoiem, D., Rother, C., Winn, J.M.: 3d layoutCRF for multi-view object class recognition and segmentation. In: Proc. International Conference on Computer Vision and Pattern Recognition (CVPR) (2007)
10. Kohli, P., Kumar, M., Torr, P.: P^3 and beyond: Solving energies with higher order cliques. In: Proc. International Conference on Computer Vision and Pattern Recognition (CVPR) (2007)
11. Kohli, P., Kumar, M.P.: Energy minimization for linear envelope MRFs. In: Proc. International Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1863–1870 (2010)
12. Kohli, P., Ladicky, L., Torr, P.H.S.: Robust higher order potentials for enforcing label consistency. International Journal of Computer Vision (IJCV) 82(3), 302–324 (2009)
13. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 28(10), 1568–1583 (2006)
14. Komodakis, N., Paragios, N.: Beyond pairwise energies: Efficient optimization for higher-order MRFs. In: Proc. International Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2985–2992 (2009)
15. Kontschieder, P., Bulo, S., Bischof, H., Pelillo, M.: Structured class-labels in random forests for semantic image labelling. In: Proc. International Conference on Computer Vision (ICCV). pp. 2190–2197 (Nov 2011)
16. Krahenbuhl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K. (eds.) Advances in Neural Information Processing Systems 24, pp. 109–117 (2011), http://papers.nips.cc/paper/4296-efficient-inference-in-fully-connected-crfs-with-gaussian-edge-potentials.pdf

17. Ladicky, L., Russell, C., Kohli, P., Torr, P.H.S.: Graph cut based inference with co-occurrence statistics. In: Proc. European Conference on Computer Vision (ECCV). pp. 239–253 (2010)
18. Lan, X., Roth, S., Huttenlocher, D., Black, M.: Efficient belief propagation with learned higher-order markov random fields. In: Proc. European Conference on Computer Vision (ECCV). pp. 269–282 (2006)
19. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing via label transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 33(12), 2368–2382 (2011)
20. Lucchi, A., Li, Y., Fua, P.: Learning for Structured Prediction Using Approximate Subgradient Descent with Working Sets. In: Proc. International Conference on Computer Vision and Pattern Recognition (CVPR) (2013)
21. Nowozin, S., Rother, C., Bagon, S., Sharp, T., Yao, B., Kohli, P.: Decision tree fields. Proc. International Conference on Computer Vision (ICCV) pp. 1668–1675 (2011)
22. Olsson, C., Ulen, J., Boykov, Y., Kolmogorov, V.: Partial enumeration and curvature regularization. In: Proc. International Conference on Computer Vision (ICCV). pp. 2936–2943 (Dec 2013)
23. Potetz, B.: Efficient belief propagation for vision using linear constraint nodes. In: Proc. International Conference on Computer Vision and Pattern Recognition (CVPR) (2007)
24. Roth, S., Black, M.: Fields of experts: A framework for learning image priors. In: Proc. International Conference on Computer Vision and Pattern Recognition (CVPR). pp. 860–867 (2005)
25. Rother, C., Kohli, P., Feng, W., Jia, J.: Minimizing sparse higher order energy functions of discrete variables. Proc. International Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1382–1389 (2009)
26. Sharp, T., Rother, C., Nowozin, S., Jancsary, J.: Regression tree fields – an efficient, non-parametric approach to image labeling problems. Proc. International Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2376–2383 (2011)
27. Shotton, J., Winn, J., Rother, C., Criminisi, A.: *TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Proc. European Conference on Computer Vision (ECCV). vol. 1, pp. 1–15 (2006)
28. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C.: A comparative study of energy minimization methods for Markov random fields. In: Proc. European Conference on Computer Vision (ECCV). pp. 16–29 (2006)
29. Woodford, O., Torr, P., Reid, I., Fitzgibbon, A.: Global stereo reconstruction under second order smoothness priors. In: Proc. International Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8 (2008)