Int J Comput Vis (2009) 82: 302–324
DOI 10.1007/s11263-008-0202-0

Robust Higher Order Potentials for Enforcing Label Consistency

Pushmeet Kohli · L'ubor Ladický · Philip H.S. Torr

Received: 17 August 2008 / Accepted: 16 December 2008 / Published online: 24 January 2009

Abstract This paper proposes a novel framework for labelling problems which is able to combine multiple segmentations in a principled manner. Our method is based on higher order conditional random fields and uses potentials defined on sets of pixels (image segments) generated using unsupervised segmentation algorithms. These potentials enforce label consistency in image regions and can be seen as a generalization of the commonly used pairwise contrast sensitive smoothness potentials. The higher order potential functions used in our framework take the form of the Robust P^n model and are more general than the P^n Potts model recently proposed by Kohli et al. We prove that the optimal *swap* and *expansion* moves for energy functions composed of these potentials can be computed by solving a st-mincut problem. This enables the use of powerful graph cut based move making algorithms for performing inference in the framework. We test our method on the problem of multi-class object segmentation by augmenting the conventional CRF used for object segmentation with higher order potentials defined on image regions. Experiments on challenging data sets show that integration of higher order potentials quantitatively and qualitatively improves results leading to much better definition of object boundaries. We believe that this method can be used to yield similar improvements for many other labelling problems.

Keywords Discrete energy minimization · Markov and conditional random fields · Object recognition and segmentation

P. Kohli (✉)
Microsoft Research, Cambridge, UK
e-mail: pkohli@microsoft.com

L. Ladický · P.H.S. Torr
Oxford Brookes University, Oxford, UK

L. Ladický
e-mail: lladicky@brookes.ac.uk

P.H.S. Torr
e-mail: philiptorr@brookes.ac.uk

1 Introduction

In recent years an increasingly popular way to solve various image labelling problems like object segmentation, stereo and single view reconstruction is to formulate them using image segments (so called superpixels) obtained from unsupervised[1] segmentation algorithms (He et al. 2006; Hoiem et al. 2005a; Rabinovich et al. 2006). These methods are inspired from the observation that pixels constituting a particular segment often have the same label; for instance, they may belong to the same object or may have the same surface orientation. This approach has the benefit that higher order features based on all the pixels constituting the segment can be computed and used for classification.[2] Further, it is also much faster as inference now only needs to be performed over a small number of superpixels rather than all the pixels in the image.

Methods based on grouping segments make the assumption that segments are consistent with object boundaries in the image (He et al. 2006), i.e. segments do not contain multiple objects. As observed by Hoiem et al. (2005b) and Rus-

[1] By unsupervised, we mean that the segmentation algorithm does not use information from object recognition.

[2] In some sense this causes the problem of scene understanding to be decoupled from the image resolution given by the hardware; it is conducted using more natural primitives that are independent of resolution.

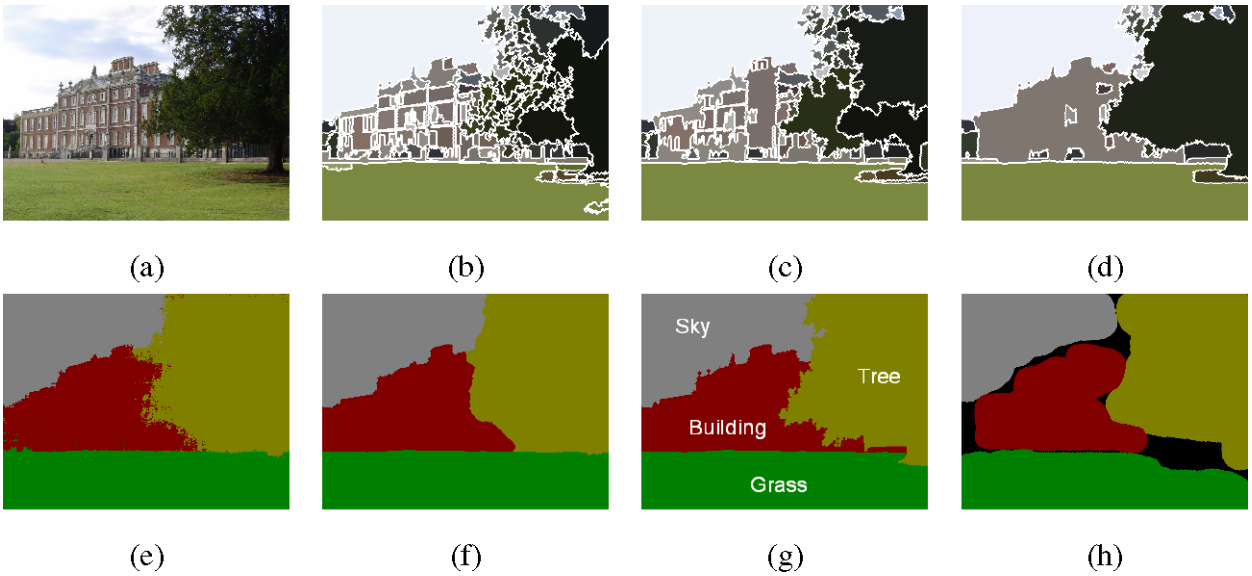


Fig. 1 Incorporating higher order potentials for object segmentation. (**a**) An image from the MSRC-21 dataset. (**b**), (**c**) and (**d**) Unsupervised image segmentation results generated by using different parameters values in the mean-shift segmentation algorithm (Comaniciu and Meer 2002). (**e**) The object segmentation obtained using the unary likelihood potentials from TextonBoost. (**f**) The result of performing inference in the pairwise CRF defined in Sect. 2. (**g**) Our segmentation result obtained by augmenting the pairwise CRF with higher order potentials defined on the segments shown in (**b**), (**c**) and (**d**). (**h**) The rough hand labelled segmentations provided in the MSRC data set. It can be clearly seen that the use of higher order potentials results in a significant improvement in the segmentation result. For instance, the branches of the tree are much better segmented

sell et al. (2006) this is not always the case and segments obtained using unsupervised segmentation methods are often wrong. To overcome these problems (Hoiem et al. 2005b) and (Russell et al. 2006) use multiple segmentations of the image (instead of only one) in the hope that although most segmentations are bad, some are correct and thus would prove useful for their task. They merge these multiple superpixels using heuristic algorithms which lack any optimality guarantees and thus may produce bad results. In this paper we propose an algorithm that can compute the solution of the labelling problem (using features based on image segments) in a principled manner. Our approach couples potential functions defined on sets of pixels with conventional unary and pairwise cues using higher order CRFs. We test the performance of this method on the problem of object segmentation and recognition. Our experiments show that the results of our approach are significantly better than the ones obtained using pairwise CRF models (see Fig. 1).

1.1 Object Segmentation and Recognition

Combined object segmentation and recognition is one of the most challenging and fundamental problems in computer vision. The last few years have seen the emergence of object segmentation algorithms which integrate *object specific* top-down information with *image based* low-level features (Borenstein and Malik 2006; He et al. 2004; Huang et al. 2004; Kumar et al. 2005; Levin and Weiss 2006). These methods have produced excellent results on challenging data sets. However, they typically only deal with one object at a time in the image independently and do not provide a framework for understanding the whole image. Further, their models become prohibitively large as the number of classes increases. This prevents their application to scenarios where segmentation and recognition of many object classes is desired.

Shotton et al. (2006) recently proposed a method (*TextonBoost*) to overcome this problem. In contrast to using explicit models to encode object shape they used a boosted combination of *texton* features which jointly modeled shape and texture. They combine the result of textons with colour and location based likelihood terms in a conditional random field (CRF). Although their method produced good segmentation and recognition results, the rough shape and texture model caused it to fail at object boundaries. The problem of extracting accurate boundaries of objects is considerably more challenging. In what follows we show that incorporation of higher order potentials defined on superpixels dramatically improves the object segmentation result. In particular, it leads to segmentations with much better definition of object boundaries as shown in Fig. 1.

1.2 Higher Order CRFs

Higher order random fields are not new to computer vision. They have been long used to model image textures (Lan et

al. 2006; Paget and Longstaff 1998; Roth and Black 2005). The initial work in this regard has been quite promising and higher order CRFs have been shown to improve results for problems such as image denoising and restoration (Roth and Black 2005), and texture segmentation (Kohli et al. 2007). However their use has been quite limited due to lack of efficient algorithms for performing inference in these models.

Traditional inference algorithms such as BP are quite computationally expensive for higher order cliques although recent work has improved their performance for certain classes of potential functions. Lan et al. (2006) proposed approximation methods for BP to make efficient inference possible in higher order MRFs. This was followed by the recent work of Potetz (2007) in which he showed how belief propagation can be efficiently performed in graphical models containing moderately large cliques. However, as these methods were based on BP, they were quite slow and took minutes or hours to converge.

Kohli et al. (2007) recently showed how certain higher order clique potentials can be minimized using move making algorithms for approximate energy minimization such as α-expansion and $\alpha\beta$-swap (Boykov et al. 2001). They introduced a class of higher order potentials called the P^n Potts model and showed how the optimal *expansion* and *swap* moves for energy functions containing these potentials can be computed in polynomial time by solving a st-mincut problem. The complexity of their algorithm increased linearly with the size of the clique and thus it was able to handle cliques composed of thousands of latent variables.

The higher order energy functions characterizing the higher order CRFs arising from our work are more general in form than the P^n Potts model and thus cannot be minimized efficiently using the algorithm of Kohli et al. (2007). We introduce a new family of higher order potentials which is a generalization of the P^n Potts class. The potential functions belonging to this family are parameterized with a truncation parameter Q which controls their *robustness*. We will show how energy functions composed of these *robust* potentials can be minimized using move making algorithms such as α-expansion and $\alpha\beta$-swap. Specifically, we show how the optimal swap and expansion moves for such potentials can be found using algorithms for computing the st-mincut.

1.3 Organization of the Paper

This paper proposes a general framework for solving labelling problems which has the ability to utilize higher order potentials defined on segments.[3] We test this framework on the problem of object segmentation and recognition by integrating label consistency and shape based terms defined on segments with conventional unary and pairwise potentials. We show how inference in this framework can be efficiently performed by extending the recent work on minimizing energy function with higher order cliques (Kohli et al. 2007). To summarize, the novelties of our approach include:

(1) The method for efficiently solving a new family of higher order potentials which we call the robust P^n model, and is a generalization of the P^n Potts model.
(2) A novel higher order region consistency potential which is a strict generalization of the commonly used pairwise contrast sensitive smoothness potential.
(3) The application of higher order CRFs for object segmentation and recognition which integrate the above mentioned higher order potentials with conventional unary and pairwise potentials based on colour, location, texture, and smoothness.

An outline of the paper follows. In Sect. 2 we discuss the basic theory of conditional random fields. We then show how pairwise CRFs can be used to model labelling problems like object segmentation. In Sect. 3 we augment the pairwise CRF model by incorporating novel higher order potentials based on super-pixel segmentations. In Sect. 4 we review the work on move making algorithms for solving higher order energy functions. The potential functions which can be solved using our method are described in Sect. 5. Finally, in Sect. 6 we show how the optimal expansion and swap moves for energy functions composed of such potentials can be computed by solving a st-mincut problem. The experimental results of our method are given in Sect. 7. These include qualitative and quantitative results on well known and challenging data sets for object segmentation and recognition. The conclusions and directions for future work are listed in Sect. 8. The proofs of the theorems stated in the paper are given in Appendix B.

2 Preliminaries

We start by providing the basic notation used in the paper. Consider a discrete random field $\mathbf{X}$ defined over a lattice $\mathcal{V} = \{1, 2, \ldots, N\}$ with a neighbourhood system $\mathcal{E}$. Each random variable $X_i \in \mathbf{X}$ is associated with a lattice point $i \in \mathcal{V}$ and takes a value from the label set $\mathcal{L} = \{l_1, l_2, \ldots, l_k\}$. The neighborhood system $\mathcal{E}$ is the set of edges connecting variables in the random field. A clique c is a set of random variables $\mathbf{X}_c$ which are conditionally dependent on each other. Any possible assignment of labels to the random variables will be called a *labelling* (denoted by $\mathbf{x}$) which takes values from the set $\mathbf{L} = \mathcal{L}^N$.

The probability $\Pr(X = \mathbf{x})$ of any labelling $\mathbf{x}$ of the random variables will be referred to as $\Pr(\mathbf{x})$. From the Hammersley Clifford theorem, the posterior distribution $\Pr(\mathbf{x}|\mathbf{D})$

[3] An earlier version of this paper appeared as Kohli et al. (2008).

over the labellings of a Markov Random Field (MRF) is a *Gibbs* distribution and can be written as

$$\Pr(\mathbf{x}|\mathbf{D}) = \frac{1}{Z}\exp\left(-\sum_{c\in\mathcal{C}}\psi_c(\mathbf{x}_c)\right), \tag{1}$$

where $\psi_c(\mathbf{x}_c)$ are potential functions defined over the variables ($\mathbf{x}_c = \{x_i, i \in c\}$) constituting the clique c, Z is a normalizing constant known as the partition function, and $\mathcal{C}$ is the set of all cliques (Lauristzen 1996). The corresponding Gibbs energy is defined as

$$E(\mathbf{x}) = -\log\Pr(\mathbf{x}|\mathbf{D}) - \log Z = \sum_{c\in\mathcal{C}}\psi_c(\mathbf{x}_c). \tag{2}$$

The maximum a posteriori (MAP) labelling $\mathbf{x}^*$ of the random field is defined as

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathbf{L}}\Pr(\mathbf{x}|\mathbf{D}) = \arg\min_{\mathbf{x}\in\mathbf{L}}E(\mathbf{x}). \tag{3}$$

A conditional random field (CRF) may be viewed as an MRF globally conditioned on the data (Lafferty et al. 2001). In this case, the potential functions are conditioned by the data and are thus should be written as $\psi_c(\mathbf{x}_c|\mathbf{D})$. To be concise, we will drop $\mathbf{D}$, and just use $\psi_c(\mathbf{x}_c)$ to denote the potential functions of a CRF.

2.1 Pairwise CRFs for Object Segmentation

The CRF models commonly used for object segmentation are characterized by energy functions defined on unary and pairwise cliques as:

$$E(\mathbf{x}) = \sum_{i\in\mathcal{V}}\psi_i(x_i) + \sum_{(i,j)\in\mathcal{E}}\psi_{ij}(x_i, x_j). \tag{4}$$

Here $\mathcal{V}$ corresponds to the set of all image pixels, while $\mathcal{E}$ is the set of all edges connecting the pixels $i, j \in \mathcal{V}$. The edge set is commonly chosen to define a 4 or 8 neighbourhood. The labels constituting the label set $\mathcal{L}$ represent the different objects. The random variable x_i denotes the labelling of pixel i of the image. Every possible assignment of the random variables $\mathbf{x}$ (or configuration of the CRF) defines a segmentation.

The unary potential ψ_i of the CRF is defined as the negative log of the likelihood of a label being assigned to pixel i. It can be computed from the colour of the pixel and the appearance model for each object. However, colour alone is not a very discriminative feature and fails to produce accurate segmentations. This problem can be overcome by using sophisticated potential functions based on colour, texture, location, and shape priors as shown by Blake et al. (2004), Bray et al. (2006), Kumar et al. (2005), Rother et al. (2004), Shotton et al. (2006). The unary potential used by us can be written as:

$$\psi_i(x_i) = \theta_T\psi_T(x_i) + \theta_{col}\psi_{col}(x_i) + \theta_l\psi_l(x_i) \tag{5}$$

where θ_T, θ_{col}, and θ_l are parameters weighting the potentials obtained from TextonBoost(ψ_T) (Shotton et al. 2006), colour(ψ_{col}) and location(ψ_l) respectively.

The pairwise terms ψ_{ij} of the CRF take the form of a contrast sensitive Potts model:

$$\psi_{ij}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j, \\ g(i,j) & \text{otherwise,} \end{cases} \tag{6}$$

where the function $g(i, j)$ is an edge feature based on the difference in colors of neighboring pixels (Boykov and Jolly 2001). It is typically defined as:

$$g(i,j) = \theta_p + \theta_v\exp(-\theta_\beta\|I_i - I_j\|^2), \tag{7}$$

where I_i and I_j are the colour vectors of pixel i and j respectively. θ_p, θ_v, and θ_β are model parameters whose values are learned using training data. We refer the reader to Boykov and Jolly (2001), Rother et al. (2004), Shotton et al. (2006) for more details.

Inferring the Most Probable Segmentation The object segmentation problem can be solved by finding the least energy configuration of the CRF defined above. As the pairwise potentials of the energy function (4) are of the form of a Potts model it can be minimized approximately using the well known α-expansion algorithm (Boykov et al. 2001). The resulting segmentation can be seen in Fig. 1. We also tried other energy minimization algorithms such as sequential tree-reweighted message passing (TRW-S) (Kolmogorov 2006; Wainwright et al. 2005). The α-expansion algorithm was preferred because it was faster and gave a solution with lower energy compared to TRW-S.

Need for Higher Order CRFs The use of Potts model (Boykov et al. 2001) potentials in the CRF model makes it favour smooth object boundaries. Although this improves results in most cases it also introduces an undesirable side effect. Smoothness potentials make the model incapable of extracting the fine contours of certain object classes such as trees and bushes. As seen in the results, segmentations obtained using pairwise CRFs tend to be oversmooth and quite often do not match the actual object contour. In the next section we show how these results can be significantly improved by using higher order potentials derived from multiple segmentations obtained from an unsupervised image segmentation method.

3 Incorporating Higher Order Potentials

Methods based on grouping regions for segmentation generally make the assumption that all pixels constituting a particular segment (or region) belong to the same object (He et al. 2006). This is not always the case, and image segments quite often contain pixels belonging to multiple object classes. For instance, in the segmentations shown in Fig. 2 the bottom image segment contains some 'building' pixels in addition to all the grass pixels.

Unlike other object segmentation algorithms which use the label consistency in segments as a hard constraint, our method uses it as a *soft constraint*. This is done by using higher order potentials defined on the image segments generated using unsupervised segmentation algorithms. Specifically, we augment the pairwise CRF model explained in the previous section by incorporating higher order potentials defined on sets or regions of pixels. The Gibbs energy of this higher order CRF can now be written as:

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j) + \sum_{c \in \mathcal{S}} \psi_c(\mathbf{x}_c), \quad (8)$$

where $\mathcal{S}$ refers to a set of image segments (or super-pixels), and ψ_c are higher order potentials defined on them. In our experiments, the set $\mathcal{S}$ consisted of all segments of multiple segmentations of an image obtained using an unsupervised image segmentation algorithm such as mean-shift (Comaniciu and Meer 2002) (see Sect. 3.4 for more details). We will now describe in detail how these higher order potentials are defined.

3.1 Region Based Consistency Potential

The *region consistency potential* is similar to the smoothness prior present in pairwise CRFs (Boykov and Jolly 2001). It favours all pixels belonging to a segment taking the same label, and as will be shown later is particularly useful in obtaining object segmentations with fine boundaries. It takes the form of a $\mathcal{P}^n$ Potts model (see (16)) (Kohli et al. 2007):

$$\psi_c^p(\mathbf{x}_c) = \begin{cases} 0 & \text{if } x_i = l_k,\ \forall i \in c, \\ \theta_p^h |c|^{\theta_\alpha} & \text{otherwise} \end{cases} \quad (9)$$

where $|c|$ denotes the cardinality of the pixel set c which in our case is the number of pixels constituting superpixel c, while θ_p^h and θ_α are parameters of the potential. The expression $\theta_p^h |c|^{\theta_\alpha}$ gives the label inconsistency cost, i.e. the cost added to the energy of a labelling in which different labels have been assigned to the pixels constituting the segment. The parameters θ_p^h and θ_α are learned from the training data by cross validation as described in Sect. 7. The reader should note that this potential takes multiple variables as argument

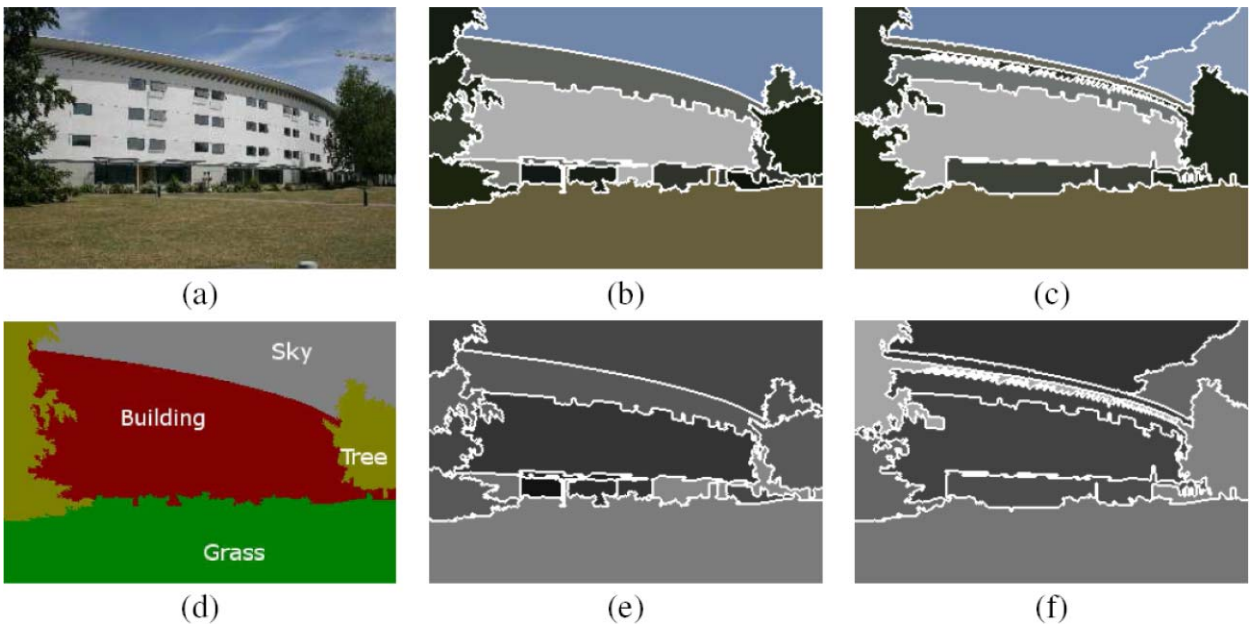


Fig. 2 Quality sensitive region consistency prior. (**a**) An image from the MSRC data set. (**b**) and (**c**) Two different segmentations of the image obtained using different parameter values for the mean-shift algorithm. (**d**) A hand labelled object segmentation of the image. (**e**) and (**f**) The value of the variance based quality function $G(c)$ (see (11)) computed over the segments of the two segmentations. Segments with high quality values are darker. It can be clearly seen that segments which contain multiple object classes have been assigned low quality. For instance, the top segment of the left tree in segmentation (**c**) includes a part of the building and thus is brighter in the image (**f**) indicating low quality. Potentials defined on such segments will have a lower labelling inconsistency cost and will have less influence in the CRF

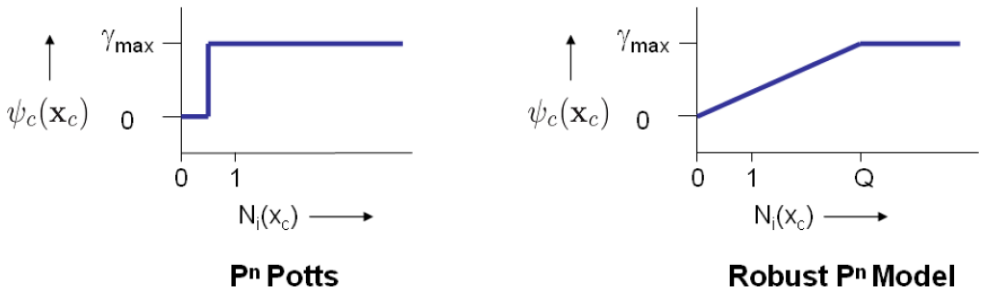


Fig. 3 Behaviour of the rigid P^n Potts potential and the Robust P^n model potential. The figure shows how the cost enforced by the two higher order potentials changes with the number of variables in the clique not taking the dominant label i.e. $N_i(\mathbf{x}_c) = \min_k(|c| - n_k(\mathbf{x}_c))$

and thus cannot be expressed in the conventional pairwise CRF model.

3.2 Quality Sensitive Consistency Potential

Not all segments obtained using unsupervised segmentation are equally good, for instance, some segments may contain multiple object classes. A region consistency potential defined over such a segment will encourage an incorrect labelling of the image. This is because the potential (9) does not take the quality or *goodness* of the segment into account. It assigns the same penalty for breaking 'good' segment as it assigns to 'bad' ones. This problem of the consistency potential can be overcome by defining a quality sensitive higher order potential (see Fig. 2). This new potential works by modulating the label inconsistency cost with a function of the quality of the segment (which is denoted by $G(c)$). Any method for estimating the segment quality can be used in our framework. A good example would be the method of Ren and Malik (2003) which uses inter and intra region similarity to measure the quality or goodness of a segment. Formally, the potential function is written as:

$$\psi_c^v(\mathbf{x}_c) = \begin{cases} 0 & \text{if } x_i = l_k,\ \forall i \in c, \\ |c|^{\theta_\alpha}(\theta_p^h + \theta_v^h G(c)) & \text{otherwise.} \end{cases} \tag{10}$$

For our experiments, we use the variance of the response of a unary feature evaluated on all constituent pixels of a segment to measure the quality of a segment, i.e.

$$G(c) = \exp\left(-\theta_\beta^h \frac{\| \sum_{i \in c}(f(i) - \mu)^2 \|}{|c|}\right), \tag{11}$$

where $\mu = \frac{\sum_{i \in c} f(i)}{|c|}$ and $f()$ is a function evaluated on all constituent pixels of the superpixel c. If we restrict our attention to only pairwise cliques i.e. $|c| = 2$, the variance sensitive potential becomes

$$\psi_c^v(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j, \\ |c|^{\theta_\alpha}\left(\theta_p^h + \theta_v^h \exp\left(-\theta_\beta^h \frac{\|f(i) - f(j)\|^2}{4}\right)\right) & \\ \text{otherwise.} & \end{cases} \tag{12}$$

This is the same as the pairwise potential (6) commonly used in pairwise CRFs for different image labelling problems (Boykov and Jolly 2001; Rother et al. 2004). Thus, the variance sensitive potential can be seen as a higher order generalization of the contrast preserving potential. The variance function response over two segmentations of an image is shown in Fig. 2.

3.3 Making the Potentials Robust

The P^n Potts model enforces label consistency rigidly. For instance, if all but one of the pixels in a super-pixel take the same label then the same penalty is incurred as if they were all to take different labels. Due to this strict penalty, the potential might not be able to deal with inaccurate super-pixels or resolve conflicts between overlapping regions of pixels. This phenomenon is illustrated in Fig. 4 wherein a part of the bird is merged with the 'sky' super-pixel and results in an inaccurate segmentation. Intuitively, this problem can be resolved using the *Robust* higher order potentials defined as:

$$\psi_c^v(\mathbf{x}_c) = \begin{cases} N_i(\mathbf{x}_c)\frac{1}{Q}\gamma_{\max} & \text{if } N_i(\mathbf{x}_c) \le Q, \\ \gamma_{\max} & \text{otherwise,} \end{cases} \tag{13}$$

where $N_i(\mathbf{x}_c)$ denotes the number of variables in the clique c not taking the dominant label i.e. $N_i(\mathbf{x}_c) = \min_k(|c| - n_k(\mathbf{x}_c))$, $\gamma_{\max} = |c|^{\theta_\alpha}(\theta_p^h + \theta_v^h G(c))$, and Q is the truncation parameter which controls the rigidity of the higher order clique potential. We will show in Sect. 4 how energy functions composed of such potentials can be minimized using move making algorithms such as α-expansion and $\alpha\beta$-swap.

Unlike the standard P^n Potts model, this potential function gives rise to a cost that is a linear truncated function of the number of inconsistent variables (see Fig. 3). This enables the robust potential to allow some variables in the clique to take different labels. In the image shown in Fig. 4, the robust P^n potentials allows some pixels of the 'sky' segment to take the label 'bird' thus producing a much better segmentation. Experimental results are shown for multiple values of the truncation parameter Q. More qualitative results can be seen in Fig. 13.

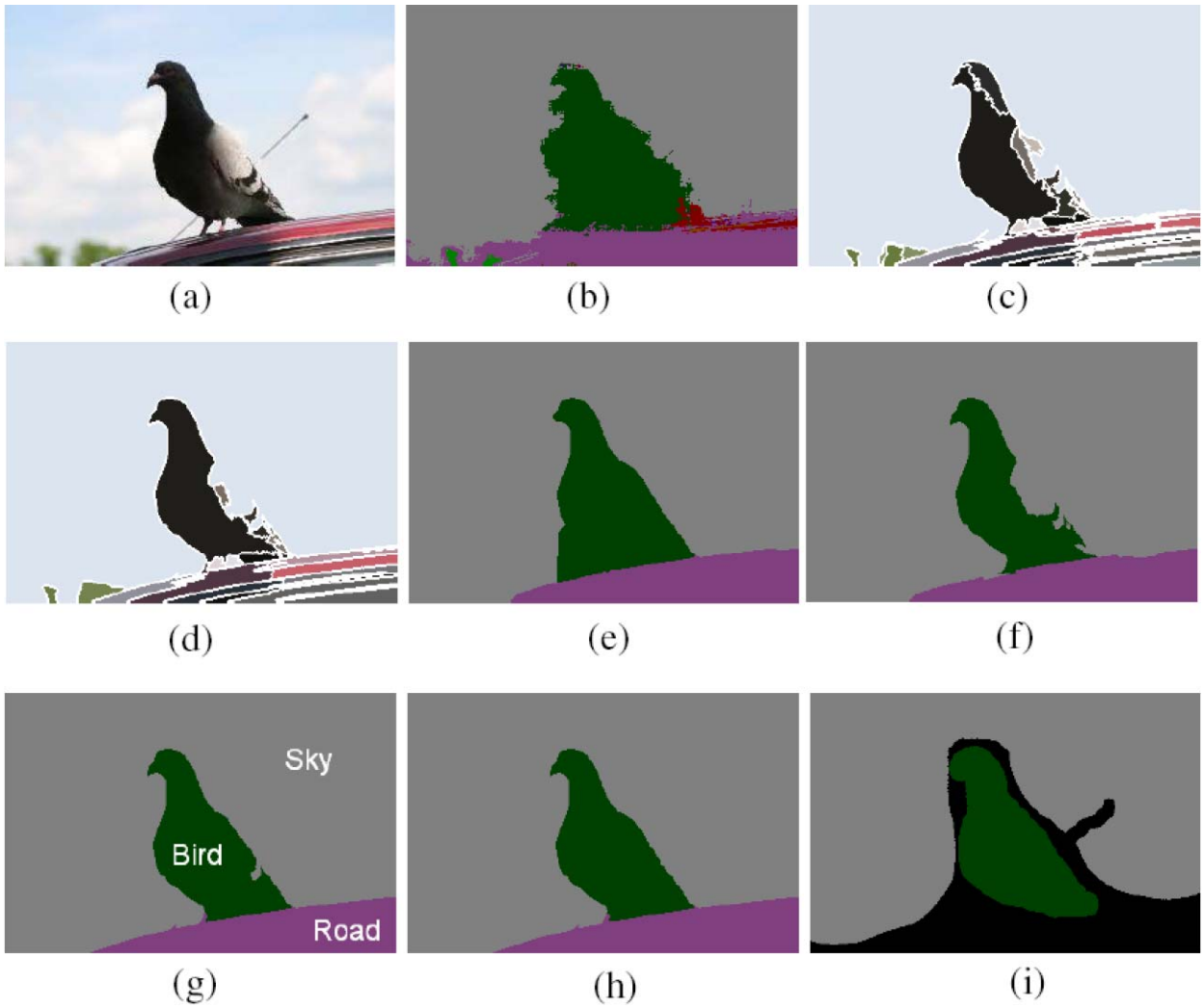


Fig. 4 Object segmentation and recognition using the Robust P^n higher order potentials (13). (**a**) Original image. (**b**) Labelling from unary likelihood potentials from TextonBoost (Shotton et al. 2006). (**c**) and (**d**) Segmentations obtained by varying the parameters of the Mean shift algorithm for unsupervised image segmentation (Comaniciu and Meer 2002). (**e**) Result obtained using pairwise potential functions as described in Shotton et al. (2006). (**f**) Result obtained using P^n Potts model potentials defined on the segments (or superpixels) shown in (**c**) and (**d**). These higher order potentials encourage all pixels in a superpixel to take the same label. The P^n Potts model rigidly enforces label consistency in regions thus causing certain pixels belonging to the 'bird' to erroneously take the label 'sky' as they were included in the 'sky' superpixel. This problem can be overcome by using the Robust P^n model potentials defined in (13) which are robust and allow some variables in the clique to take different labels. (**g**) and (**h**) Show results obtained by using the robust potentials with truncation parameter Q equal to $0.1|c|$ and $0.2|c|$ respectively. Here $|c|$ is equal to the size of the superpixel over which the Robust P^n model potential is defined. (**i**) Hand labelled segmentation from the MSRC dataset

3.4 Generating Multiple Segmentations

We now explain how the set $\mathcal{S}$ of segments used for defining the higher order energy function (8) was generated. Our framework is quite flexible and can handle multiple overlapping or non-overlapping segments. The computer vision literature contains algorithms for sampling the likely segmentations of an image (Tu and Zhu 2002) or for generating multi-scale segmentations (Sharon et al. 2001). However, following in the footsteps of Russell et al. (2006) we choose to generate multiple segmentations by varying the parameters of the mean shift segmentation algorithm (Comaniciu and Meer 2002). This method belongs to the class of unsupervised segmentation algorithms which work by clustering pixels on the basis of low level image features (Shi and Malik 2000; Comaniciu and Meer 2002; Felzenszwalb and Huttenlocher 2004). They have been shown to give decent results which have proved to be useful for many applications (Hoiem et al. 2005a, 2005b; Wang et al. 2005).

The kernel used in the mean shift algorithm is defined as the product of spatial and range kernels. The spatial do-

Fig. 5 (Color online) Generating multiple segmentations. The figure shows the segmentations obtained by using different parameters in the mean-shift algorithm. The parameters used for generating the segmentation are written below it in the format (h_s, h_r), where h_s and h_r are the bandwidth parameters for the spatial and range (colour) domains

main contains the (x, y) coordinates, while the range domain contains pixel colour information in LUV space. An assumption of Euclidian metric in both of them allows the use of a single bandwidth parameter for each domain, h_s for spatial and h_r for range. The segmentation results obtained using 2 different spatial $\{7, 18\}$ and 3 different range parameter values $\{6.5, 9.5, 15\}$ are shown in Fig. 5. It can be seen that the results do not change dramatically on small images by modifying h_s. The only difference occurs on very noisy parts of the image like trees and bushes. By increasing the range parameter h_r we can get a range of segmentations which vary from over-segmented to under-segmented. We decided to use three segmentations with parameters $(h_s, h_r) = \{(7, 6.5), (7, 9.5), (7, 15)\}$.

4 Inference in Higher Order CRFs

The problem of inferring the most probable solution of a higher order CRF is equivalent to minimizing an energy function. In general, the energy minimization problem is NP-hard (Kolmogorov and Zabih 2004). However, there exist classes of functions which can be solved exactly in polynomial time. Two well known classes of tractable functions are: submodular functions, and functions defined over graphs with bounded tree width. However, most energies encountered in practical problems do not belong to these families. They are instead solved using algorithms for approximate energy minimization. These algorithms can be divided into two broad categories: message passing algorithms such as belief propagation and its variants (Kolmogorov 2006; Wainwright et al. 2005; Yedidia et al. 2000), and move making algorithms such as the graph cut based α-expansion and $\alpha\beta$-swap (Boykov et al. 2001). Message passing algorithms have been shown to produce excellent results for many energy functions. However, their runtime complexity increases exponentially with the size of the largest clique in the random field, making them inapplicable to functions defined over large cliques. Efficient graph cut based α-expansion and $\alpha\beta$-swap move algorithms have been successfully used to minimize energy functions composed of pairwise potential functions. In this paper, we show how they can be applied to a large and useful class of higher order energy functions.

4.1 Expansion and Swap Move Algorithms

Move making algorithms start from an initial solution and proceed by making a series of changes which lead to solutions having lower energy. At each step, the algorithms search a move space and choose the move which leads to the solution having the lowest energy. This move is referred to as the *optimal* move. The algorithm is said to converge when no lower energy solution can be found.

The size of the move space is a key characteristic of these algorithm. A large move space means that bigger changes to the current solution can be made. This makes the algorithm less prone to getting stuck in local minima and also results in a faster rate of convergence. This paper deals with two particular *large* move making algorithms, namely *α-expansion* and *$\alpha\beta$-swap* (Boykov et al. 2001) whose move space size increases exponentially with the number of variables involved in the energy function. We will use the notation of Kohli et al. (2007) to describe how these algorithms work. The moves of the expansion and swap algorithms can be encoded as a vector of binary variables $\mathbf{t} = \{t_i, \forall i \in \mathcal{V}\}$. The

transformation function $T(\mathbf{x}^p, \mathbf{t})$ of a move algorithm takes the current labelling $\mathbf{x}^p$ and a move $\mathbf{t}$ and returns the new labelling $\mathbf{x}^n$ which has been induced by the move.

An α-expansion move allows any random variable to either retain its current label or take label 'α'. One iteration of the algorithm involves making moves for all α in $\mathcal{L}$ in some order successively. The transformation function $T_\alpha()$ for an α-expansion move transforms the label of a random variable X_i as

$$T_\alpha(x_i, t_i) = \begin{cases} \alpha & \text{if } t_i = 0, \\ x_i^p & \text{if } t_i = 1. \end{cases} \tag{14}$$

An $\alpha\beta$-swap move allows a variable whose current label is α or β to either take label α or β. One iteration of the algorithm involves performing swap moves for all α and β in $\mathcal{L}$ in some order successively. The transformation function $T_{\alpha\beta}()$ for an $\alpha\beta$-swap transforms the label of a random variable x_i as

$$T_{\alpha\beta}(x_i, t_i) = \begin{cases} \alpha & \text{if } x_i = \alpha \text{ or } \beta \text{ and } t_i = 0, \\ \beta & \text{if } x_i = \alpha \text{ or } \beta \text{ and } t_i = 1. \end{cases} \tag{15}$$

The energy of a move $\mathbf{t}$ is the energy of the labelling $\mathbf{x}$ the move $\mathbf{t}$ induces i.e. $E_m(\mathbf{t}) = E(T(\mathbf{x}, \mathbf{t}))$. The move energy is a pseudo-boolean function ($E_m : \{0, 1\}^n \rightarrow \mathbb{R}$) and will be denoted by $E_m(\mathbf{t})$. At each step of the expansion and swap move algorithms, the *optimal* move $\mathbf{t}^*$, i.e. the move decreasing the energy of the labelling by the most amount is computed. This is done by minimizing the move energy i.e. $\mathbf{t}^* = \arg\min_{\mathbf{t}} E(T(\mathbf{x}, \mathbf{t}))$. The optimal move $\mathbf{t}^*$ can be computed in polynomial time if the move function $E_m(\mathbf{t})$ is submodular.

4.2 Recent Developments

The last few years have seen a lot of interest in graph cut based move algorithms for energy minimization. Komodakis et al. (2005, 2007) recently gave an alternative interpretation of the α-expansion algorithm. They showed that α-expansion works by solving the dual of a linear programming relaxation of the energy minimization problem. Using this theory, they developed a new algorithm (FAST-PD) which was faster than vanilla α-expansions and produced the exact same solution. Alahari et al. (2008) have recently proposed a similar but simpler method which achieves the same performance.

Researchers have also proposed a number of novel move encoding strategies for solving particular forms of energy functions. Veksler (2007) proposed a move algorithm in which variables can choose any label from a range of labels. They showed that this move space allowed them to obtain better minima of energy functions with truncated convex pairwise terms. Kumar and Torr (2008) have since shown that the range move algorithm achieves the same guarantees as the ones obtained by methods based on the standard linear programming relaxation. More recently, Lempitsky et al. (2007) proposed an algorithm which encoded labels by a binary string. During each move, the variables were allowed to change a particular bit of the binary string. They showed that this particular move encoding strategy results in a substantial speedup when minimizing energy functions with large label sets.

4.3 Computing Moves Using Graph Cuts

We had discussed in Sect. 4.1 that the optimal expansion and swap moves can be computed by minimizing a (move) function of binary variables. Functions of binary variables ($F : \{0, 1\} \rightarrow \mathbb{R}$) are usually referred to as *Pseudo-boolean* functions. It is known that a move function can be minimized in polynomial time if it is submodular (Orlin 2007) (see Appendix A). Submodular set functions are encountered in many areas of research and are particularly useful in combinatorial optimization, probability and geometry (Fujishige 1991; Lovasz 1983). Many optimization problems relating to submodular functions can be solved efficiently. In this respect they are similar to convex/concave functions encountered in continuous optimization.

Algorithms for submodular function minimization have high runtime complexity. Although recent work has been successful in reducing the runtime complexity of these algorithms, they are still quite computationally expensive and cannot be used to minimize large functions. For instance, the complexity of the current best algorithm for general submodular function minimization is $O(n^5\mathcal{T} + n^6)$ where $\mathcal{T}$ is the time taken to evaluate the function (Orlin 2007). This algorithm improved upon the previous best algorithm by a factor of $n \log n$.

Some submodular functions can be minimized much more efficiently by solving an st-mincut problem (Boros and Hammer 2002). Specifically, all submodular functions of binary variables of order at most 3 can be minimized in this manner (Boros and Hammer 2002; Kolmogorov and Zabih 2004). Researchers have shown that certain higher order functions can be transformed into submodular functions of order 2, and thus can also be minimized (Boros and Hammer 2002; Freedman and Drineas 2005). The same transformation technique can be used to minimize some functions of multi-valued variables (Flach 2002; Ishikawa 2003; Schlesinger and Flach 2006).

Solving pairwise CRFs using move making algorithms involves computing optimal moves by minimizing quadratic pseudo-boolean move functions. Boykov et al. (2001) showed that all expansion move functions encountered while minimizing an energy function composed of metric potential functions are submodular. As these functions are

quadratic, they can be efficiently minimized by solving an equivalent st-mincut problem. They also showed that all $\alpha\beta$-swap move functions can be exactly minimized if the pairwise potentials of the CRF are semi-metric.

5 Robust Higher Order Potentials

Kohli et al. (2007) recently characterized a class of higher order clique potentials for which the optimal expansion and swap moves can be computed by minimizing a submodular function. However, as discussed earlier, minimizing a general submodular function is quite computationally expensive and it is infeasible to apply this procedure to minimize energy functions encountered in computer vision problems, which generally involve millions of random variables. In the same work, they also introduced a class of higher order clique potentials called the P^n Potts model and showed that the optimal *expansion* and *swap* moves for energy functions containing these potentials can be computed in polynomial time by solving a st-mincut problem. The P^n Potts model was defined as:

$$\psi_c(\mathbf{x}_c) = \begin{cases} \gamma_k & \text{if } x_i = l_k,\ \forall i \in c, \\ \gamma_{\max} & \text{otherwise,} \end{cases} \tag{16}$$

where $\gamma_{\max} \geq \gamma_k,\ \forall l_k \in \mathcal{L}$. This potential is a higher order generalization of the widely used Potts model potential which is defined over cliques of size two as $\psi_{ij}(a, b) = \gamma_k$ if $a = b = l_k$ and $\gamma_{\max}$ otherwise.

In this paper we introduce a novel family of higher order potentials which we call the Robust P^n model. This family contains the P^n Potts model as well as its *robust* variants, and can be used for modelling many computer vision problems. We show that the optimal swap and expansion move energy functions for any Robust P^n model potential can be transformed into a second order submodular function by the addition of at most two binary auxiliary variables. This transformation enables us to find the optimal swap and expansion moves in polynomial time.[4]

The Robust P^n model potentials take the form:

$$\psi_c(\mathbf{x}_c) = \min\Big\{\min_{k\in\mathcal{L}}((|c| - n_k(\mathbf{x}_c))\theta_k + \gamma_k), \gamma_{\max}\Big\} \tag{17}$$

where $|c|$ is the number of variables in clique c, $n_k(\mathbf{x}_c)$ denotes the number of variables in clique c which take the label k in labelling $\mathbf{x}_c$, and $\gamma_k, \theta_k, \gamma_{\max}$ are potential function parameters which satisfy the constraints:

$$\theta_k = \frac{\gamma_{\max} - \gamma_k}{Q} \quad \text{and} \quad \gamma_k \leq \gamma_{\max}, \quad \forall k \in \mathcal{L}. \tag{18}$$

Q is called the truncation parameter of the potential and satisfies the constraint $2Q < |c|$. It can be seen that the Robust P^n model (17) becomes a P^n Potts model (16) when the truncation parameter is set to 1.

Example 1 Consider the set of clique variables $\mathbf{X} = \{X_1, X_2, \ldots, X_7\}$ where each $X_i, i \in \{1, 2, \ldots, 7\}$ takes a value from the label set $\mathcal{L} = \{a, b, c\}$. If the clique potential takes the form a P^n Potts model, it assigns a cost $\gamma_{\max}$ to all labellings of the random variables except those where all variables X_i take the same label. Thus, the configuration $\mathbf{x} = (a, a, b, a, c, a, a)$ will be assigned cost $\gamma_{\max}$ even though there are only 2 variables (specifically, X_3 and X_5) which are assigned labels (b and c) different from the dominant label a. In contrast, if the clique potential takes the form of the Robust P^n model with truncation 3 i.e., $Q = 3$, it assigns the cost: $\gamma_a + \frac{(\gamma_{\max} - \gamma_a)}{3} \times 2$ to the same configuration.

The region based consistency potentials (13) used in our higher order CRF takes the form of a Robust P^n model where the constants γ_k have the same value for all labels $k \in \mathcal{L}$. In this case, the higher order potential can be seen as encouraging all the variables in the clique c to take the same label. In other words, the potential tries to reduce the number of variables in the clique not taking the dominant label i.e., $N_i(\mathbf{x}_c) = \min_k(|c| - n_k(\mathbf{x}_c))$. In what follows we will refer to these variables as *inconsistent*.

Unlike the P^n Potts model that rigidly enforces label consistency, the Robust P^n Potts model gives rise to a cost that is a linear truncated function of the number of inconsistent variables (see Fig. 3). This enables the robust potential to allow some variables in the clique to take different labels.

5.1 Approximating Concave Consistency Potentials

Multiple Robust P^n model potentials can be combined to approximate any non-decreasing concave consistency potential up to an arbitrary accuracy. This potential takes the form:

$$\psi_c(\mathbf{x}_c) = \min\Big\{\min_{k\in\mathcal{L}} \mathcal{F}_c((|c| - n_k(\mathbf{x}_c))), \gamma_{\max}\Big\} \tag{19}$$

where $\mathcal{F}_c$ is a non-decreasing concave function.[5] This is illustrated in Fig. 6.

5.2 Generalized form of Robust Higher Order Potentials

We now provide a characterization of a larger class of functions for which at most two auxiliary variables are sufficient

[4] All second order submodular functions of binary variables can be minimized exactly in polynomial time by solving an st-mincut problem (Boros and Hammer 2002; Kolmogorov and Zabih 2004).

[5] A function $f(x)$ is concave if for any two points (a, b) and λ where $0 \leq \lambda \leq 1$: $\lambda f(a) + (1 - \lambda) f(b) \leq f(\lambda a + (1 - \lambda)b)$.

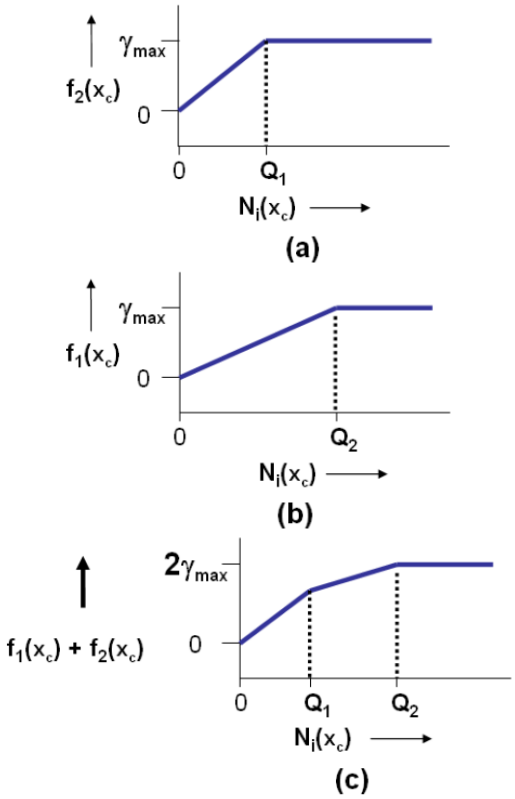


Fig. 6 Approximating Concave Consistency Potentials. The figure shows the result of combining two robust higher order potentials (**a**) and (**b**). The resulting potential function is shown in (**c**)

to transform the higher order swap and expansion move energy functions to second order functions. The potentials belonging to this new family have the form:

$$\psi_c(\mathbf{x}_c) = \min\left\{\min_{k\in\mathcal{L}}((P - f_k(\mathbf{x}_c))\theta_k + \gamma_k), \gamma_{\max}\right\} \tag{20}$$

where the parameter P and functions $f_k(\mathbf{x}_c)$ are defined as:

$$P = \sum_{i\in c} w_i^k, \quad \forall k \in \mathcal{L}, \tag{21}$$

$$f_k(\mathbf{x}_c) = \sum_{i\in c} w_i^k \delta_k(x_i) \tag{22}$$

$$\text{where } \delta_k(x_i) = \begin{cases} 1 & \text{if } x_i = k, \\ 0 & \text{otherwise,} \end{cases} \tag{23}$$

and weights $w_i^k \geq 0, i \in c, k \in \mathcal{L}$ encode the relative importance of different variables in preserving consistency of the labelling of the clique. The parameters $\gamma_k, \theta_k, \gamma_{\max}$ of the potential function satisfy the constraints:

$$\theta_k = \frac{\gamma_{\max} - \gamma_k}{Q_k} \quad \text{and} \quad \gamma_k \leq \gamma_{\max}, \quad \forall k \in \mathcal{L}. \tag{24}$$

$Q_k, k \in \mathcal{L}$ are the truncation parameters of the potential functions and satisfy the constraints $Q_a + Q_b < P, \forall a \neq b \in \mathcal{L}$.

If we assume that $w_i^k = w_i \geq 0$ and $Q_k = Q$ for all $k \in \mathcal{L}$, the potential family (20) can be seen as weighted version of the Robust P^n model. The weights can be used to specify the relative importance of different variables. For instance, this can be used to change the robust region consistency potential (13) to reduce the inconsistency cost for pixels on the segment boundary by reducing their weights. We will show that for the case of symmetric weights i.e. $w_i^k = w_i$, the higher order swap and expansion and move energy functions for the potentials (20) can be transformed to a submodular second order binary energy.[6]

6 Minimizing Higher Order Move Functions Using Graph Cuts

We will now explain how the optimal swap and expansion moves for energy functions containing potential functions of the form (20) can be computed using graph cuts. The computation of the optimal moves requires the minimization of higher order move functions. This is done by first transforming the higher order move functions to quadratic submodular functions by adding auxiliary binary variables, and then minizing them using graph cuts.

6.1 Transforming Higher Order Move Energies

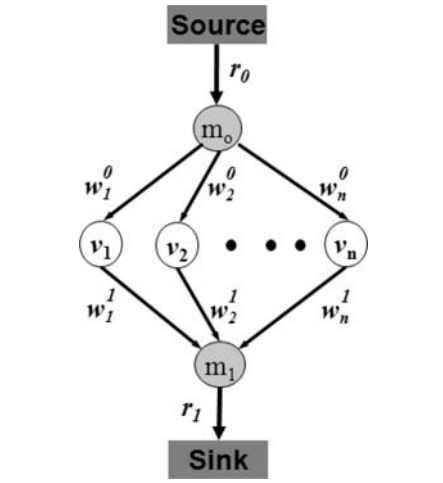
The problem of transforming a general submodular higher order function to a second order one has been well studied (Boros and Hammer 2002). It is known that in the worst case this may require the addition of exponential number of auxiliary binary variables. Due to the special form of the Robust P^n model (10), the method described in the paper only needs to add two binary variables per higher order potential to transform the move energy to a submodular quadratic function. This allows for the efficient computation of the optimal swap and expansion moves. The complexity of our algorithm for computing the optimal move increases linearly with the size of the clique. This enables us to handle potential functions defined over very large cliques.

The important observation that inspired our method is the fact that higher order pseudo-boolean functions of the form:

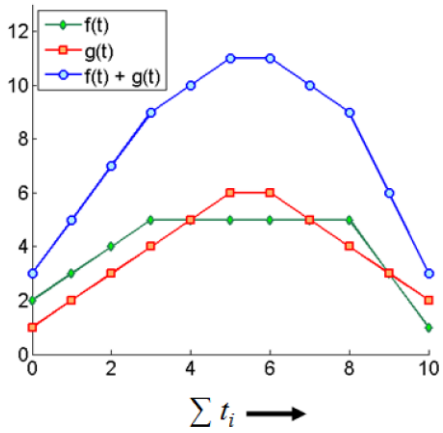
$$f(\mathbf{t}_c) = \min\left(\theta_0 + \sum_{i\in c} w_i^0(1 - t_i), \theta_1 + \sum_{i\in c} w_i^1 t_i, \theta_{\max}\right) \tag{25}$$

can be transformed to submodular quadratic pseudo-boolean functions, and hence can be minimized using graph cuts.

[6] Higher order potentials with asymmetric weights can also be transformed to quadratic functions in a similar manner. We restrict our attention to potentials with symmetric weights for a cleaner exposition.



(a)



(b)

Fig. 7 (**a**) Graph construction for minimizing higher order quadratic functions of the form (25) using the transformation given in Theorem 1. For every binary variable t_i in the energy function there is a corresponding graph node v_i. The minimum cost source sink cut (st-mincut) divides the graph into two sets: the source set ($\mathcal{S}$) and the sink set ($\mathcal{T}$). $v_i \in (\mathcal{S})$ implies $t_i = 1$ while $v_i \in (\mathcal{T})$ implies $t_i = 0$. (**b**) Graph illustrating how two higher order potentials of the form (25) can be summed together to attain any function of the generalized concave form (29). Function f and g are defined over 10 binary variables ($c = \{1, 2, \ldots, 10\}$). They are defined as: $f(\mathbf{t}) = \min(1 + \sum_{i \in c} 2(1 - t_i), 2 + \sum_{i \in c} t_i, 5)$ and $g(\mathbf{t}) = \min(2 + \sum_{i \in c}(1 - t_i), 1 + \sum_{i \in c} t_i, 6)$

Here, $\mathbf{t}_c = \{t_i \in \{0, 1\},\ i \in c\}$ is the set of binary random variables included in the clique c, and $w_i^0 \geq 0$, $w_i^1 \geq 0$, θ_0, θ_1, $\theta_{\max}$ are parameters of the potential satisfying the constraints $\theta_{\max} \geq \theta_0$, $\theta_{\max} \geq \theta_1$, and

$$\left(\left(\theta_0 + \sum_{i \in c} w_i^0 (1 - t_i) \geq \theta_{\max}\right) \vee \left(\theta_1 + \sum_{i \in c} w_i^1 t_i \geq \theta_{\max}\right)\right) = 1, \quad \forall \mathbf{t}_c \in \{0, 1\}^{|c|} \tag{26}$$

where $\vee$ is a boolean OR operator. The transformation to a quadratic pseudo-boolean function requires the addition of only two binary auxiliary variables making it computationally efficient.

Theorem 1 *The higher order pseudo-boolean function*:

$$f(\mathbf{t}_c) = \min\left(\theta_0 + \sum_{i \in c} w_i^0 (1 - t_i), \theta_1 + \sum_{i \in c} w_i^1 t_i, \theta_{\max}\right) \tag{27}$$

can be transformed to the submodular quadratic pseudo-boolean function:

$$f(\mathbf{t}_c) = \min_{m_0, m_1} \left(r_0(1 - m_0) + m_0 \sum_{i \in c} w_i^0 (1 - t_i) + r_1 m_1 + (1 - m_1) \sum_{i \in c} w_i^1 t_i - K\right) \tag{28}$$

by the addition of binary auxiliary variables m_0 and m_1. Here, $r_0 = \theta_{\max} - \theta_0$, $r_1 = \theta_{\max} - \theta_1$ and $K = \theta_{\max} - \theta_0 - \theta_1$.

Proof in Appendix B.

The graph construction for minimizing the quadratic pseudo-boolean function (28) is shown in Fig. 7(a).

Multiple higher order potentials of the form (25) can be summed together to obtain higher order potentials of the more general form

$$f(\mathbf{t}_c) = F_c\left(\sum_{i \in c} t_i\right) \tag{29}$$

where $F_c : \mathbb{R} \to \mathbb{R}$ is any concave function. See Fig. 7(b) for an illustration.

In what follows we show that any swap or expansion move energy for higher order potentials of the form (20) can be converted to a submodular pairwise function if $w_i^k = w_i$ for all $k \in \mathcal{L}$. Our transformation requires the addition of only two binary auxiliary variables. To proceed further, we will need to define the function $W(s)$, $s \subseteq c$:

$$W(s) = \sum_{i \in s} w_i. \tag{30}$$

It can be seen from constraint (21) that $W(c) = P$.

6.2 Swap Moves

Recall from the definition of the swap move transformation function that only variables which are currently assigned labels α or β can take part in a $\alpha\beta$-swap move. We call these variables *active* and denote the vector of their indices by c_a. $\mathbf{t}_{c_a}$ will be used to denote the corresponding vector of move variables. Similarly, variables in the clique which do not take

part in the swap move are called *passive*, and the set of their indices is denoted by c_p. Let functions $f_k^m(\mathbf{t}_{c_a}), k \in \{0, 1\}$ be defined as:

$$f_k^m(\mathbf{t}_{c_a}) = \sum_{i \in c_a} w_i \delta_k(t_i). \tag{31}$$

The move energy of a $\alpha\beta$-swap move from the current labelling $\mathbf{x}_c^p$ is equal to the energy of the new labelling $\mathbf{x}_c^n$ induced by the move and is given as

$$\psi_c^m(\mathbf{t}_{c_a}) = \psi_c(\mathbf{x}_c^n). \tag{32}$$

The new labelling $\mathbf{x}_c^n$ is obtained by combining the old labelling of the passive variables $\mathbf{X}_{c_p}$ with the new labelling of the active variables $\mathbf{X}_{c_a}$ as:

$$\mathbf{x}_c^n = \mathbf{x}_{c_p}^p \cup T_{\alpha\beta}(\mathbf{x}_{c_a}^p, \mathbf{t}_{c_a}). \tag{33}$$

On substituting the value of $\mathbf{x}_c^n$ from (33) in (32), and using the definition of the higher order potential functions (20) we get:

$$\psi_c^m(\mathbf{t}_{c_a}) = \psi_c(\mathbf{x}_{c_p}^p \cup T_{\alpha\beta}(\mathbf{x}_{c_a}^p, \mathbf{t}_{c_a})) \tag{34}$$

$$= \min\Big\{\min_{k \in \mathcal{L}}(z_k\theta_k + \gamma_k), \gamma_{\max}\Big\} \tag{35}$$

where $z_k = P - f_k(\mathbf{x}_{c_p}^p \cup T_{\alpha\beta}(\mathbf{x}_{c_a}^p, \mathbf{t}_{c_a}))$.

It can be easily observed that if conditions:

$$W(c_a) < P - Q_\alpha \quad \text{and} \quad W(c_a) < P - Q_\beta, \tag{36}$$

are satisfied, then the expression:

$$(P - f_k(\mathbf{x}_{c_p}^p \cup T_{\alpha\beta}(\mathbf{x}_{c_a}^p, \mathbf{t}_{c_a})))\theta_k + \gamma_k \tag{37}$$

is greater than $\gamma_{\max}$ for both $k = \alpha$ and $k = \beta$. Thus, in this case the move energy $\psi_c^m(\mathbf{t}_{c_a})$ is independent of $\mathbf{t}_{c_a}$ and is equal to the constant:

$$\eta = \min\Big\{\min_{k \in \mathcal{L} \setminus \{\alpha,\beta\}}((P - f_k(\mathbf{x}_c^p))\theta_k + \gamma_k), \gamma_{\max}\Big\} \tag{38}$$

which can be ignored while computing the swap moves. However, if constraints (36) are not satisfied, the move energy becomes:

$$\psi_c^m(\mathbf{x}_{c_a}^p, \mathbf{t}_{c_a}) = \min\big\{(W(c_a) - f_0^m(\mathbf{t}_{c_a}))\theta_\alpha + \lambda_\alpha, (W(c_a) - f_1^m(\mathbf{t}_{c_a}))\theta_\beta + \lambda_\beta, \lambda_{\max}\big\} \tag{39}$$

where $\lambda_\alpha = \gamma_\alpha + R_{\alpha\beta}\theta_\alpha$, $\lambda_\beta = \gamma_\beta + R_{\alpha\beta}\theta_\beta$, $\lambda_{\max} = \gamma_{\max}$ and $R_{\alpha\beta} = W(c - c_a)$.

The higher order move energy (39) has the same form as the function defined in (27), and can be transformed to a pairwise function by introducing binary auxiliary variables m_0 and m_1 as:

$$\psi_c^m(\mathbf{t}_c) = \min_{m_0, m_1}\Big(r_0(1 - m_0) + \theta_\beta m_0 \sum_{i \in c_a} w_i(1 - t_i) + r_1 m_1 + \theta_\alpha(1 - m_1)\sum_{i \in c_a} w_i t_i - \delta\Big), \tag{40}$$

where $r_0 = \lambda_\alpha + \delta$, $r_1 = \lambda_\beta + \delta$, and $\delta = \lambda_{\max} - \lambda_\alpha - \lambda_\beta$.

The properties $\gamma_{\max} \geq \gamma_k, \forall k \in \mathcal{L}$ and $w_i \geq 0$ of the clique potential (20) imply that all coefficients of the energy function (40) are non-negative. The function is thus *submodular* and can be minimized by solving a st-mincut problem (Kolmogorov and Zabih 2004). The graph construction for minimizing the energy function (40) is shown in Fig. 8(a). The constant δ in (40) does not affect the minimization problem i.e. it does not change the move having the least energy and thus is ignored.

6.3 Expansion Moves

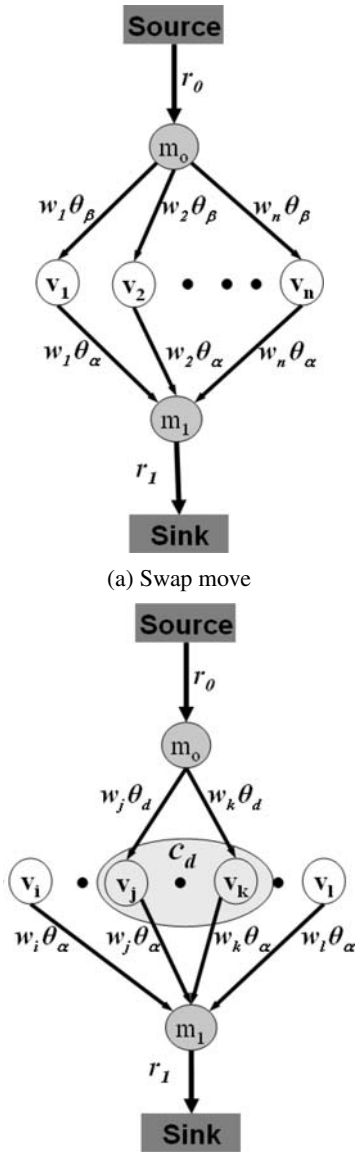
We now describe how the optimal expansion moves can be computed for the higher order potentials (20). Let c_k denote the set of indices of variables in clique c that have been assigned label k in the current solution $\mathbf{x}_c^p$. We find the *dominant* label $d \in \mathcal{L}$ in $\mathbf{x}_c^p$ such that $W(c_d) > P - Q_d$ where $d \neq \alpha$. The constraints $Q_a + Q_b < P, \forall a \neq b \in \mathcal{L}$ of the higher order potentials (20) make sure that there is at most one such label. If we find such a label in the current labelling, then the expansion move energy can be written as:

$$\begin{aligned}&\psi_c^m(\mathbf{t}_c) = \psi_c(T_\alpha(\mathbf{x}_c^p, \mathbf{t}_c)) \quad \text{or,}\\ &\psi_c^m(\mathbf{t}_c) = \min\Big\{\lambda_\alpha + \theta_\alpha \sum_{i \in c} w_i t_i, \lambda_d + \theta_d \sum_{i \in c_d} w_i(1 - t_i), \lambda_{\max}\Big\},\end{aligned} \tag{41}$$

where $\lambda_\alpha = \gamma_\alpha$, $\lambda_d = \gamma_d + R_d\theta_d$, $\lambda_{\max} = \gamma_{\max}$ and $R_d = W(c - c_d)$. Without the minimization operator the function (41) becomes:

$$\psi_c^m(\mathbf{t}_c, \mathbf{t}_{c_d}) = \begin{cases} K_\alpha & \text{if } f_0^m(\mathbf{t}_c) > P - Q_\alpha, \\ K_d & \text{if } f_0^m(\mathbf{t}_{c_d}) \leq Q_d - R_d, \\ \lambda_{\max} & \text{otherwise} \end{cases} \tag{42}$$

where $K_\alpha = \lambda_\alpha + (P - f_0^m(\mathbf{t}_c))\theta_\alpha$, and $K_d = \lambda_d + f_0^m(\mathbf{t}_{c_d})\theta_d$. Next we will show that this higher order move energy can be written as a second order submodular function with the addition of the auxiliary binary variables m_0 and m_1.



(a) Swap move

(b) Expansion move

Fig. 8 (**a**) Graph construction for minimizing the swap energy function (40). (**b**) Graph construction for minimizing the expansion move energy function (43). For every binary move variable t_i in the energy function there is a corresponding graph node v_i. The minimum cost source sink cut (st-mincut) divides the graph into two sets: the source set ($\mathcal{S}$) and the sink set ($\mathcal{T}$). $v_i \in (\mathcal{S})$ implies $t_i = 1$ while $v_i \in (\mathcal{T})$ implies $t_i = 0$

Theorem 2 *The expansion move energy* (42) *can be transformed into the pairwise function*:

$$\psi_c^m(\mathbf{t}_c) = \min_{m_0,m_1} \Bigg(r_0(1-m_0) + \theta_d m_0 \sum_{i\in c_d} w_i(1-t_i) + r_1 m_1 + \theta_\alpha(1-m_1)\sum_{i\in c} w_i t_i - \delta \Bigg), \tag{43}$$

where $r_0 = \lambda_\alpha + \delta$, $r_1 = \lambda_d + \delta$, *and* $\delta = \lambda_{\max} - \lambda_\alpha - \lambda_d$.

Proof in Appendix *B*.

The energy function (43) is submodular and can be minimized by finding the st-mincut in the graph shown in Fig. 8(b).

If a dominant label cannot be found then the move energy can be written as:

$$\psi_c^m(\mathbf{t}_c) = \min\left\{\lambda_\alpha + \theta_\alpha \sum_{i\in c} w_i t_i, \lambda_{\max}\right\}, \tag{44}$$

where $\lambda_\alpha = \gamma_\alpha$, and $\lambda_{\max} = \gamma_{\max}$. This can be transformed to the binary pairwise energy:

$$\psi_c^m(\mathbf{t}_c) = r_1 m_1 + \theta_\alpha(1-m_1)\sum_{i\in c} w_i t_i + \lambda_\alpha, \tag{45}$$

where $r_1 = \lambda_{\max} - \lambda_\alpha$. The proof for this transformation is similar to the one shown for Theorem 2.

7 Experiments

In this section we provide the details of our experiments which are divided in two parts. The first set of experiments analyze the performance of our algorithm for minimizing higher order energy functions, while those in the second set deal with evaluating the performance of using our higher order potentials for the problem of object segmentation.

7.1 Computational Performance

We have tested our methods on randomly generated higher order energy functions. We compare the performance of our methods with the Iterated Conditional Modes (ICM) algorithm. Comparison with the conventional factor graph formulation (Lan et al. 2006) of message passing algorithms like BP and TRW-S was infeasible due to the large size of cliques defining the energy functions used in our tests.[7] Our experiments show that the graph cut based expansion and swap move algorithms for the Robust P^n model potentials produce solutions with lower energy compared to the ICM algorithm. Further, they required much less time to converge to the final solution compared to ICM. The graph in Fig. 10 show how the energy of the solutions obtained from different minimization algorithms changes with time. The graphs in Fig. 9 show how the convergence time for the different algorithm is influenced by parameters of the energy function.

[7] It should be noted that the Robust P^n model potentials can be transformed into pairwise potentials by the addition of multi-label auxiliary variables. This enables the use of message passing algorithms for minimizing energy functions composed of them. However, the analysis of these transformations, and the subsequent study of the performance of message passing algorithms on the transformed functions lies outside the scope of this paper.

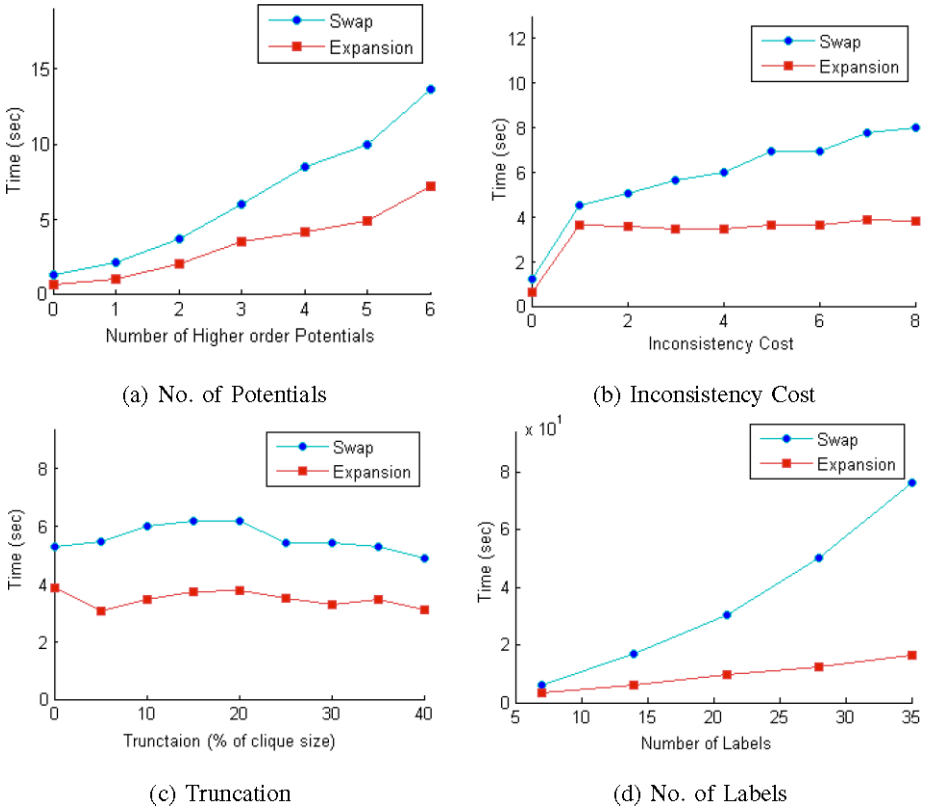


Fig. 9 Convergence times of the swap and expansion move algorithms. The graphs show how the convergence times of the move algorithms is affected by changes in the higher order energy function. The experiments were performed on grids corresponding to 10 randomly selected images from the Sowerby dataset for Object Segmentation. The size of the grid was 96×64. The unary and pairwise potentials were generated using the method proposed in Shotton et al. (2006). Higher order potentials of the form of the Robust P^n were generated randomly and incorporated in the conditional random field. A detailed description of the graphs can be found in the text. The average convergence time (in seconds) of the expansion, swap and ICM algorithms (in this order) for the different experiments were: (**a**) 3.3, 6.4, 312.3, (**b**) 3.5, 5.6, 384.6, (**c**) 3.3, 5.8, 318.4, (**d**) 9.6, 35.9, 298.1

The graph in Fig. 9(a) shows how the convergence time is affected by the number of higher order potentials. In the energy functions used for this experiment, each random variable is included in the same number of higher order cliques. The x-axes of the graph shows the number of higher order potentials each variable is involved in. As expected the convergence time increases with the number of higher order potentials in the energy function. The graph in Fig. 9(b) shows the effect of parameter $\gamma_{\max}$ of the Robust P^n model potentials on the convergence time. The graph in Fig. 9(c) shows the effect of the truncation parameter Q of the Robust P^n model. Q is specified as the percentage of the size of the higher order clique. The change in convergence time of the move making algorithms with the increase in size of the label set of the random variables can be seen in Fig. 9(d).

7.2 Object Segmentation Results

For comparative evaluation of our method we implemented the state of the art TextonBoost (Shotton et al. 2006) algorithm which uses a pairwise CRF. We then augmented the CRF model by adding higher order potentials defined on segments obtained from mean-shift (Comaniciu and Meer 2002).

Datasets We tested both the pairwise CRF and higher order CRF models on the MSRC-21 (Shotton et al. 2006) and Sowerby-7 (He et al. 2006) datasets. The MSRC dataset contains 23 object classes and comprises of 591 colour images of 320×213 resolution. The Sowerby dataset contains 7 object classes and comprises of 104 colour images of 96×64 resolution. In our experiments, 50% of the images in the dataset were used for training and the remaining were used for testing.

7.3 Setting CRF parameters

The optimal values for different parameters of the higher order CRF were found in a manner similar to the one used for the pairwise CRF in Shotton et al. (2006). The model parameters were learned by minimizing the overall pixelwise classification error rate on a set of validation images—a subset of training images which were not used for training unary potentials.

A simple method for selecting parameter values is to perform cross-validation for every combination of unary, pairwise and higher order parameters within a certain discretized range. Unfortunately, the space of possible parameter values is high dimensional and doing an exhaustive search is infeasible even with very few discretization levels for each parameter. We used a heuristic to overcome this problem. First we learned the weighting between unary potentials from colour, location and TextonBoost. Then we

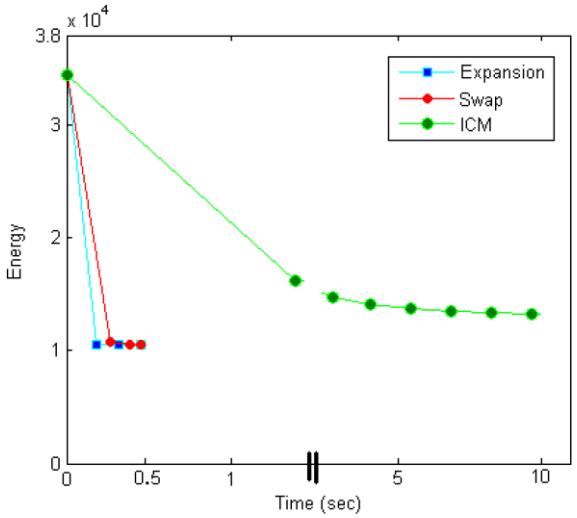


Fig. 10 Comparison of solution energy with respect to runtime. For the experiment, we used a CRF defined over a rectangular grid of 6000 random variables with a label set of size 7. The pairwise terms of the random field enforced 8 connectivity. The energy function used in the experiment had higher order potentials defined over a random number of cliques. These cliques were generated so that each variable of the random field is included in exactly one higher order clique. The graph shows how the energy of the solution obtained from different minimization algorithms changes with time when we use a Robust P^n model with the truncation parameter Q equal to one tenth of the clique size i.e. $Q = 0.1|c|$. It can be seen that expansion and swap algorithms are much faster and produce better solutions than ICM

kept these weights constant and learned the optimal parameters for pairwise potentials. Pairwise and higher order potentials have similar functionality in the framework, thus learning of higher order parameters from the model with optimal unary and pairwise parameters would lead to very low weights of higher order potentials. Instead we learned optimal higher order parameters in CRF with only unary and higher order potentials and in the last step the ratio between pairwise and higher order potentials. The final trained coefficients for the MSRC dataset were $\theta_T = 0.52$, $\theta_{col} = 0.21$, $\theta_l = 0.27$, $\theta_p = 1.0$, $\theta_v = 4.5$, $\theta_\beta = 16.0$, $\theta_\alpha = 0.8$, $\theta_p^h = 0.2$, $\theta_v^h = 0.5$, $\theta_\beta^h = 12.0$.[8] Parameter learning for higher order CRFs is an ongoing topic of research.

7.4 Quantitative Segmentation Results

The results of our experiments show that integration of higher order P^n Potts model potentials quantitatively and qualitatively improves segmentation results. The use of the robust potentials lead to further improvements (see Figs. 4, 11, 13 and 15). Inference on both the pairwise and higher order CRF model was performed using the graph cut based expansion move algorithm. The optimal expansion moves for the energy functions containing the Robust P^n potential (13) were computed using the method described in the previous section.

Effect of Multiple Segmentations The use of multiple segmentations allows us to obtain accurate segmentations of objects with thin structures. For instance, consider the image shown in Fig. 12(a). Our method produces an accurate segmentation (Fig. 12(g)) of the bird which, unlike the solution of the pairwise CRF (Fig. 12(f)), also contains the bird's leg. This result does not require that many super-pixels contain both: a part of the bird's leg, and a part of the bird's body. In fact, as shown in Figs. 12(b) and (c), many super-pixels contain only the leg and many other super-pixels contain only (a part of) the bird without the leg. As we explain below, our method can work even if only one super-pixel contains both the bird body and leg together.

The reader should observe that solution of the higher order CRF (Fig. 12(g)) is roughly *consistent*[9] with all super-pixels present in the multiple segmentations (Figs. 12(b), (c) and (d)). The solution is thus assigned a low cost by the higher order label consistency potentials. Now consider the solution of the pairwise CRF (Fig. 12(f)). This labelling is consistent with super-pixels in two segmentations (Figs. 12(b) and (c)), but is inconsistent with regards to the segmentation shown in Fig. 12(d). It assigns 'bird' and 'water' labels to pixels constituting the super-pixel which contained the bird, and is thus assigned a high cost by the higher order label consistency potential defined on that super-pixel.

Use of Image Specific Appearance Models Shotton et al. (Shotton et al. 2006) used the segmentation result obtained from the pairwise CRF to build an image specific colour appearance model for the different object classes. They added unary potentials derived from these models in their pairwise CRF model. The appearance models were also iteratively refined (as proposed by Rother et al. 2004) to obtain the final segmentation result. In our experiments, we observed that although the use of image specific models leads to better segmentations for some of the images, it led to worse solutions for some others. Therefore, while comparing results of pairwise and higher order random field models, we decided against using this technique to avoid obfuscation of the results.

[8] The magnitude of the learned parameter values does not correctly reflect the relative strength (importance) of higher order potentials vis a vis pairwise potentials. Higher order potential costs (see (9) and (10)) are multiplied by a term dependent on the size of the clique (segment). This is typically a large number and makes the cost of higher order potentials high compared to that of the pairwise potentials.

[9] The solution does not assign different labels to many pixels belonging to the same super-pixel.

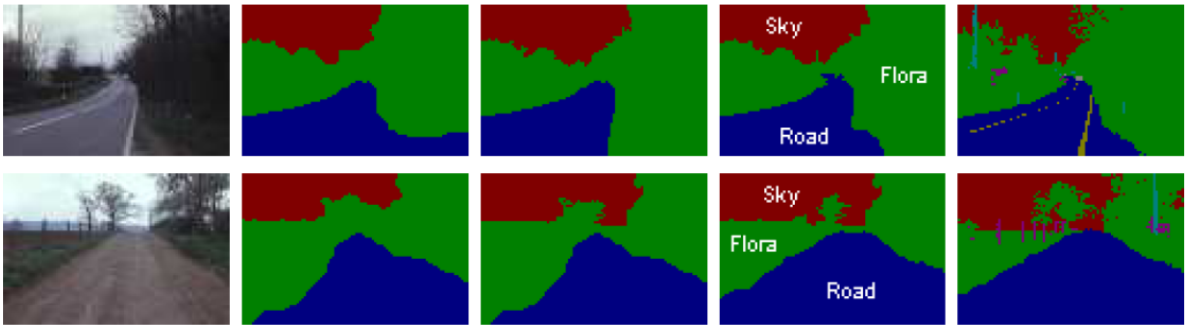


Fig. 11 Qualitative object segmentation and recognition results. The first column shows the original image from the Sowerby-7 dataset. Column 2 shows the result of performing inference in the pairwise CRF model described in Sect. 2. The result obtained using the P^n Potts potential (10) is shown in column 3. The results of using the Robust P^n potential (13) is shown in column 4. The hand labelled segmentation used as ground truth is shown in column 5

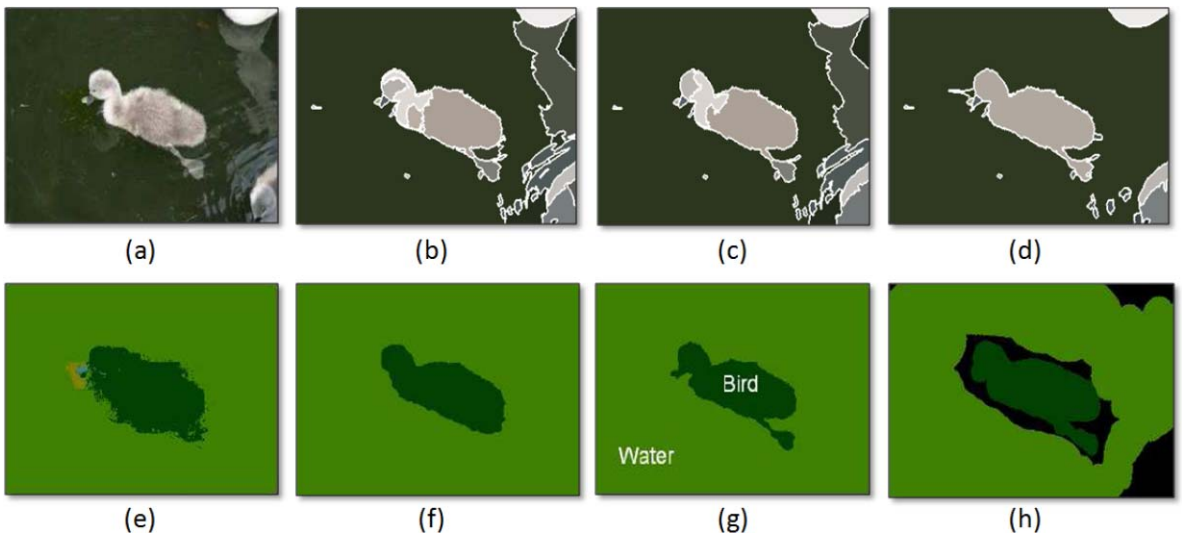


Fig. 12 Segmenting objects with thin structures using multiple segmentations. (**a**) An images from the MSRC-21 dataset. (**b**), (**c**) and (**d**) Multiple segmentations of the image obtained by varying the parameters of the mean shift algorithm (Comaniciu and Meer 2002). (**e**) Labelling from TextonBoost unary potentials (see Sect. 2). (**f**) Result of the pairwise CRF (see Sect. 2). (**g**) Results obtained by incorporating the Robust P^n higher order potential (13) defined on the segments. (**h**) Hand labelled result used as ground truth

Ground Truth The hand labelled 'ground truth' images that come with the MSRC-23 data set are quite rough. In fact qualitatively they always looked worse than the results obtained from our method. The hand labelled images suffer from another drawback. A significant numbers of pixels in these images have not been assigned any label. These unlabelled pixels generally occur at object boundaries and are critical in evaluating the accuracy of a segmentation algorithm. It should be noted that obtaining an accurate and fine segmentation of the object is important for many tasks in computer vision.

In order to get a good estimate of our algorithm's accuracy, we generated accurate segmentations which preserved the fine object boundaries present in the image. Generating these segmentations is quite time consuming. It takes between 15–60 minutes to hand label one image. We hand labelled 27 images from the MSRC data set. Figure 14 shows the original hand labelled images of the MSRC data set and the new segmentations manually labelled by us which were used as ground truth.

Evaluating Accuracy Typically the performance of a segmentation algorithm is measured by counting the total number of mislabelled pixels in the image. We believe this measure is not appropriate for measuring the segmentation accuracy if the user is interested in obtaining accurate segmentations as alpha mattes with fine object boundaries. As only a small fraction of image pixels lie on the boundary of an object, a large qualitative improvement in the quality of the segmentation will result in only a small increase in the per-

Fig. 13 Some qualitative results. Please view in colour. *First row:* Original image. *Second row:* Unary likelihood labelling from TextonBoost (Shotton et al. 2006). *Third row:* Result obtained using a pairwise contrast preserving smoothness potential as described in Shotton et al. (2006). *Fourth row:* Result obtained using the P^n Potts model potential (Kohli et al. 2007). *Fifth row:* Results using the Robust P^n model potential (13) with truncation parameter $Q = 0.1|c|$, where $|c|$ is equal to the size of the superpixel over which the Robust P^n higher order potential is defined. *Sixth row:* Hand labelled segmentations. Observe that the results obtained using the Robust P^n model are significantly better than those obtained using other methods. For instance, the leg of the sheep and bird have been accurately labelled which was missing in other results. Same can be said about the tail and leg of the dog, and the wings of the aeroplane

centage pixel-wise accuracy. This phenomenon is illustrated in Fig. 16.

With this fact in mind, we evaluate the quality of a segmentation by counting the number of pixels misclassified in the region surrounding the actual object boundary and not over the entire image. The error was computed for different widths of the evaluation region. The evaluation regions for some images from the MSRC dataset are shown in Fig. 17. The accuracy of different segmentation methods is plotted in the graph shown in Fig. 18.

8 Conclusions and Future Work

In this paper we proposed a novel framework for labelling problems which is capable of utilizing features based on sets of pixels in a principled manner. We also introduced a novel family of higher order potentials which we call the robust P^n model. We showed that energy functions composed of such potentials can be minimized using the graph cut based expansion and swap move algorithms. Our methods for computing the optimal expansion and swap moves are extremely

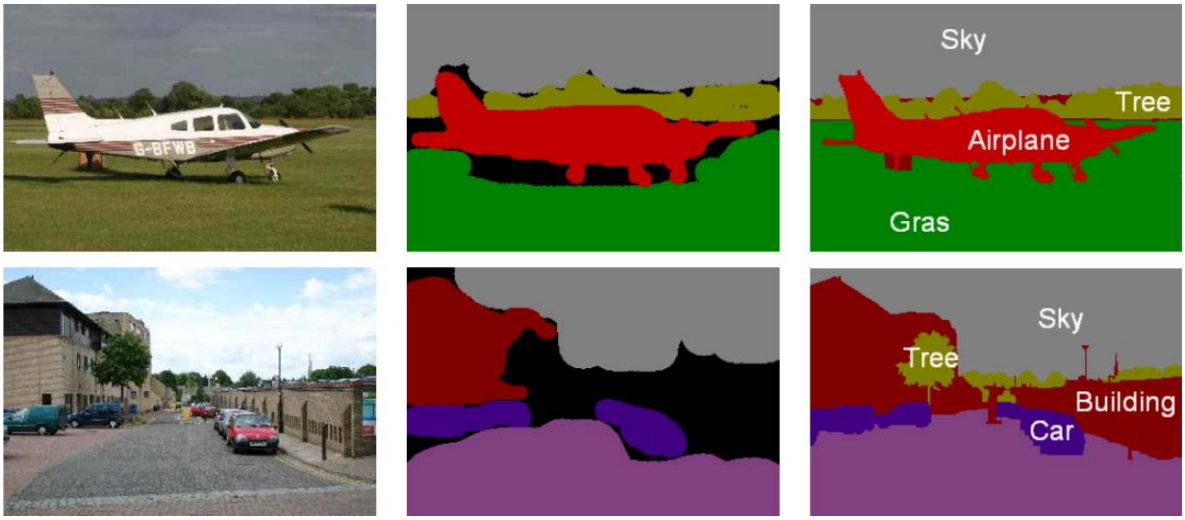


Fig. 14 Accurate hand labelled segmentations which were used as ground truth. The *figure* shows some images from the MSRC data set (*column* 1), the hand labelled segmentations that came with the data set (*column* 2), and the new segmentations hand labelled by us which were used as ground truth (*column* 3)

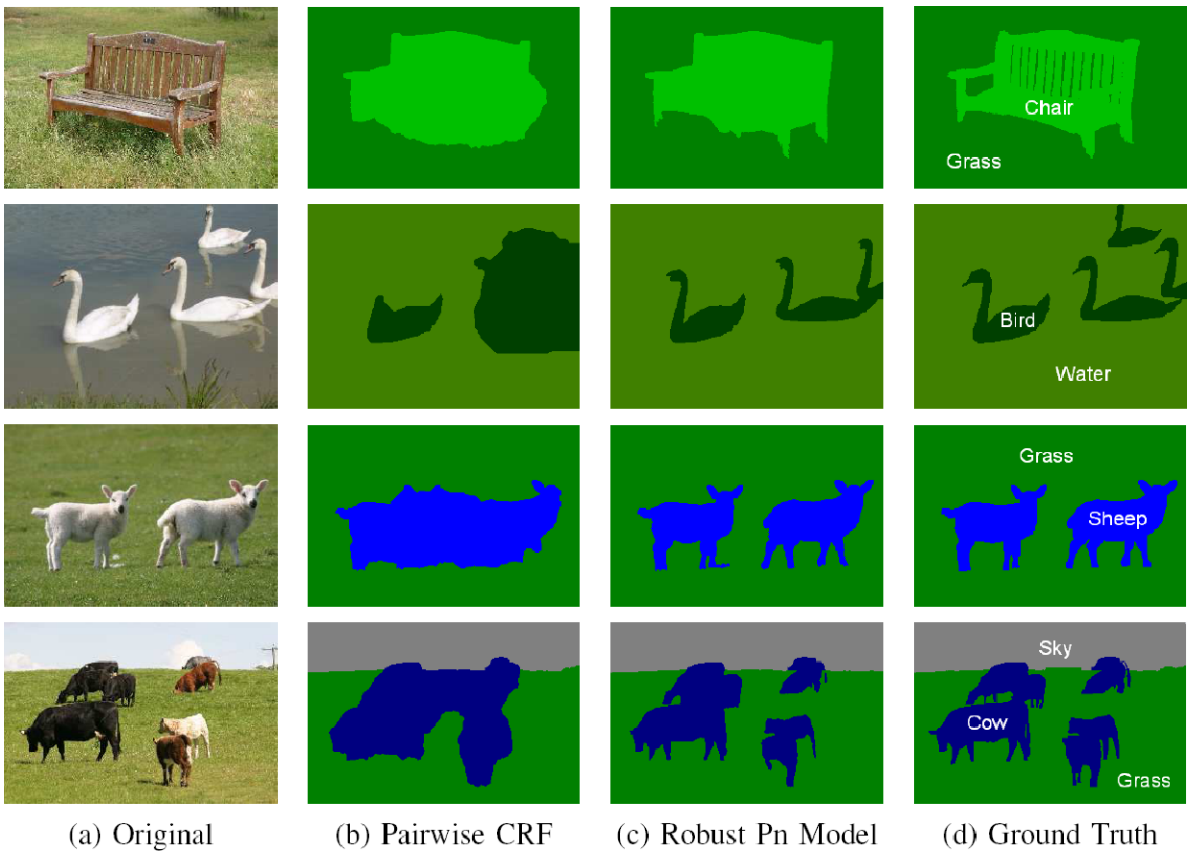


Fig. 15 Qualitative results of our method. (**a**) Original images. (**b**) Segmentation result obtained using the pairwise CRF (explained in Sect. 2). (**c**) Results obtained by incorporating the robust P^n higher order potential (13) defined on segments. (**d**) Hand labelled result used as ground truth

efficient. They can handle potentials defined over cliques of thousands of random variables.

We tested this approach on the problem of multi-class object segmentation and recognition. Our experiments showed that incorporation of P^n Potts and robust P^n model type potential functions (defined on segments) in the conditional random field model for object segmentation improved results. We believe this method is generic and can be used to

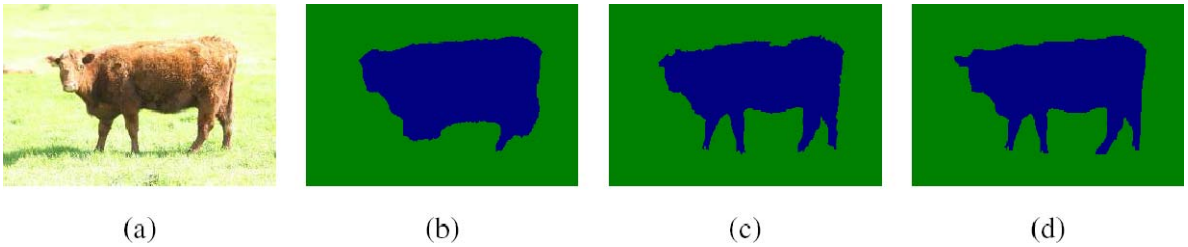


Fig. 16 The relationship between qualitative and quantitative results. (**a**) Original image. (**b**) Segmentation result obtained using the pairwise CRF (explained in Sect. 2). Overall pixelwise accuracy for the result is 95.8%. (**c**) Results obtained by incorporating the Robust P^n higher order potential (13) defined on segments. Overall pixelwise accuracy for this result is 98.7%. (**d**) Hand labelled result used as ground truth. It can be seen that even a small difference in the pixelwise accuracy can produce a massive difference in the quality of the segmentation

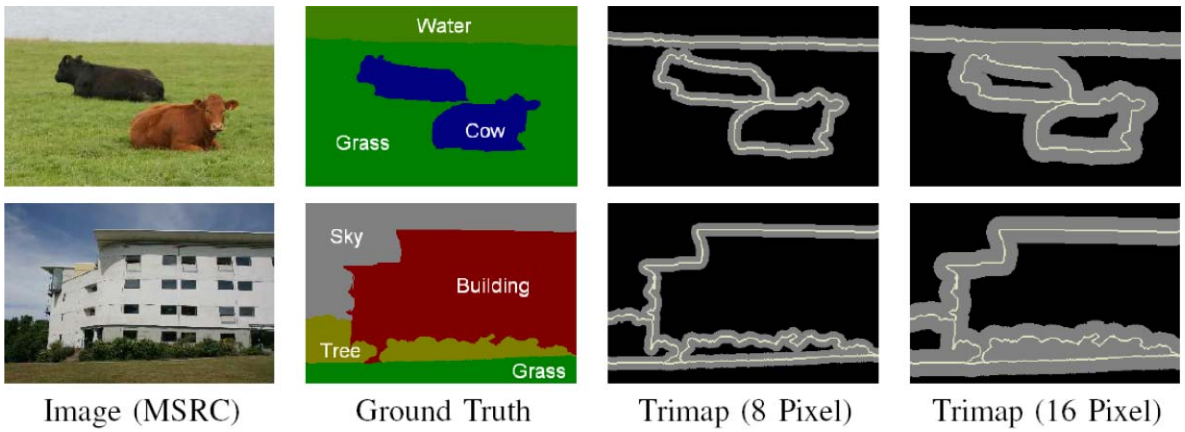


Fig. 17 Boundary accuracy evaluation using trimap segmentations. The first column shows some images from the MSRC dataset (Shotton et al. 2006). The ground truth segmentations of these image are shown in column 2. Column 3 shows the trimaps used for measuring the pixel labelling accuracy. The evaluation region is coloured gray and was generated by taking an 8 pixel band surrounding the boundaries of the objects. The corresponding trimaps for an evaluation band width of 16 pixels is shown in column 4

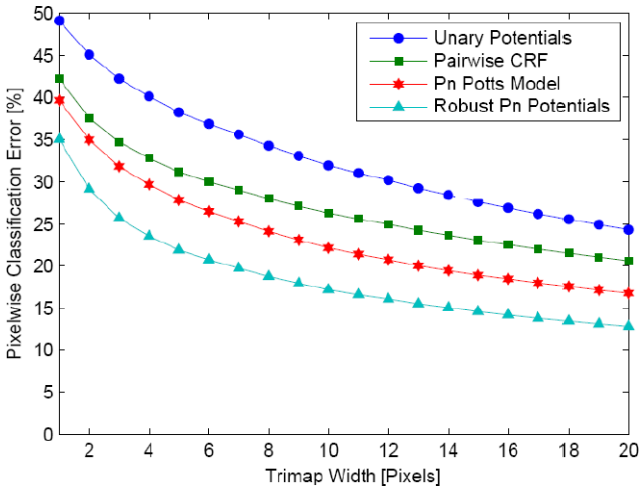


Fig. 18 Pixelwise classification error in our results. The graph shows how the overall pixelwise classification error varies as we increase the width of the evaluation region

solve many other labelling problems. In the future we would like to investigate the use of more sophisticated higher order potentials based on the shape and appearance of image segments. We believe that such potentials would be more discriminative and will result in even better performance.

Up until now, the work on solving higher order potentials using move making algorithms has targeted particular classes of potential functions. Developing efficient large move making for exact and approximate minimization of general higher order energy functions is an interesting and challenging problem for future research. Another interesting direction would be study and use of primal-dual schema (such as Fast-PD Komodakis et al. 2007) for efficiently minimizing the class of higher order potentials proposed in this paper.

Acknowledgements This work was supported by the EPSRC research grant $GR/T21790/01(P)$, HMGCC and the IST Programme of European Community, under the PASCAL Network of Excellence. We also would like to thank Sarah Mercer and Howard Cummings for support and encouragement. Professor Torr is in receipt of a Royal Society Wolfson Research Merit Award, and would like to acknowledge support from the Royal Society and Wolfson foundation.

Appendix A: Submodular Functions

To provide a formal definition of a submodular function, we will first need to define the concept of a projection of a function. A projection of a function $f : \mathcal{L}^n \rightarrow \mathbb{R}$ on s variables is a function $f^p : \mathcal{L}^s \rightarrow \mathbb{R}$ which is obtained by fixing the values of $n - s$ arguments of $f(\cdot)$. A function of one binary variable is always submodular. A function $f(x_1, x_2)$ of two binary variables $x_1, x_2 \in \mathbb{B} = \{0, 1\}$ is submodular if and only if: $f(0, 0) + f(1, 1) \leq f(0, 1) + f(1, 0)$. A function $f : \mathbb{B}^n \rightarrow R$ is submodular if and only if all its projections on 2 variables are submodular (Boros and Hammer 2002; Kolmogorov and Zabih 2004).

Appendix B: Proofs

Theorem 1 *The higher order pseudo-boolean function*:

$$f(\mathbf{t}_c) = \min\left(\theta_0 + \sum_{i\in c} w_i^0(1-t_i), \theta_1 + \sum_{i\in c} w_i^1 t_i, \theta_{\max}\right) \tag{46}$$

can be transformed to the submodular quadratic pseudo-boolean function:

$$f(\mathbf{t}_c) = \min_{m_0,m_1}\Big(r_0(1-m_0) + m_0\sum_{i\in c} w_i^0(1-t_i) + r_1 m_1 + (1-m_1)\sum_{i\in c} w_i^1 t_i - K\Big) \tag{47}$$

by the addition of binary auxiliary variables m_0 *and* m_1. *Here*, $r_0 = \theta_{\max} - \theta_0$, $r_1 = \theta_{\max} - \theta_1$, $K = \theta_{\max} - \theta_0 - \theta_1$, $\mathbf{t}_c = \{t_i \in \{0, 1\},\ i \in c\}$ *is the set of binary random variables included in the clique* c, *and* $w_i^0 \geq 0$, $w_i^1 \geq 0$, θ_0, θ_1, $\theta_{\max}$ *are function parameters that take values in* $\mathbb{R}$.

Proof We decompose the function (47) as:

$$f(\mathbf{t}_c) = \mathcal{F}^0(\mathbf{t}_c) + \mathcal{F}^1(\mathbf{t}_c) - K \quad \text{where} \tag{48}$$

$$\mathcal{F}^0(\mathbf{t}_c) = \min_{m_0} r_0(1-m_0) + m_0\sum_{i\in c} w_i^0(1-t_i). \tag{49}$$

It can be seen that the above function can be written without the minimization operator as:

$$\mathcal{F}^0(\mathbf{t}_c) = \begin{cases} \sum_{i\in c} w_i^0(1-t_i) & \text{if } \sum_{i\in c} w_i^0(1-t_i) \leq r_0, \\ r_0 & \text{otherwise.} \end{cases} \tag{50}$$

Similarly, the function $\mathcal{F}^1(\mathbf{t}_{c_a})$ can be written as:

$$\mathcal{F}^1(\mathbf{t}_c) = \begin{cases} \sum_{i\in c} w_i^1 t_i & \text{if } \sum_{i\in c} w_i^1 t_i \leq r_1, \\ r_1 & \text{otherwise.} \end{cases} \tag{51}$$

Adding equations (50) and (51) and using the constraint 26, we get

$$\mathcal{F}^0(\mathbf{t}_{c_a}) + \mathcal{F}^1(\mathbf{t}_{c_a}) = \begin{cases} r_0 + \sum_{i\in c} w_i^1 t_i & \text{if } \sum_{i\in c} w_i^1 t_i \leq r_1, \\ r_1 + \sum_{i\in c} w_i^0(1-t_i) & \text{if } \sum_{i\in c} w_i^0(1-t_i) \leq r_0, \\ r_0 + r_1 & \text{otherwise.} \end{cases} \tag{52}$$

Substituting this in (48), we get

$$f(\mathbf{t}_c) = \begin{cases} \theta_1 + \sum_{i\in c} w_i^1 t_i & \text{if } \sum_{i\in c} w_i^1 t_i \leq r_1, \\ \theta_0 + \sum_{i\in c} w_i^0(1-t_i) & \text{if } \sum_{i\in c} w_i^0(1-t_i) \leq r_0, \\ \theta_{\max} & \text{otherwise.} \end{cases} \tag{53}$$

This equation can alternatively be written as:

$$f(\mathbf{t}_c) = \min\left(\theta_0 + \sum_{i\in c} w_i^0(1-t_i), \theta_1 + \sum_{i\in c} w_i^1 t_i, \theta_{\max}\right). \tag{54}$$

□

Theorem 2 *The expansion move energy* (42) *can be transformed into the pairwise function*:

$$\psi_c^m(\mathbf{t}_c) = \min_{m_0,m_1}\Big(r_0(1-m_0) + \theta_d m_0\sum_{i\in c_d} w_i(1-t_i) + r_1 m_1 + \theta_\alpha(1-m_1)\sum_{i\in c} w_i t_i - \delta\Big) \tag{55}$$

where

$$r_0 = \lambda_\alpha + \delta,\ r_1 = \lambda_d + \delta, \quad \textit{and} \quad \delta = \lambda_{\max} - \lambda_\alpha - \lambda_d.$$

Proof We decompose the move energy (55) as:

$$\psi_c^m(\mathbf{t}_c) = \mathcal{F}^0(\mathbf{t}_{c_d}) + \mathcal{F}^1(\mathbf{t}_c) - \delta \quad \text{where} \tag{56}$$

$$\mathcal{F}^0(\mathbf{t}_{c_d}) = \min_{m_0} r_0(1-m_0) + f_0^m(\mathbf{t}_{c_d})\theta_d m_0 \tag{57}$$

$$= \min_{m_0}(\lambda_\alpha + \delta)(1-m_0) + \theta_d m_0 f_0^m(\mathbf{t}_{c_d}) \tag{58}$$

$$= \min_{m_0}(\gamma_{\max} - \gamma_d - R_d\theta_d)(1-m_0) + \frac{\gamma_{\max} - \gamma_d}{Q_d} m_0 f_0^m(\mathbf{t}_{c_d})$$

(Recall $R_d = W(c - c_d)$)

$$= \min_{m_0}(\gamma_{\max} - \gamma_d)(1-m_0) + \frac{\gamma_{\max} - \gamma_d}{Q_d} m_0(f_0^m(\mathbf{t}_{c_d}) + R_d) - R_d\theta_d$$

$$= \begin{cases} \lambda_{\max} - \lambda_d & \text{if } f_0^m(\mathbf{t}_{c_d}) > Q_d - R_d, \\ f_0^m(\mathbf{t}_{c_d})\theta_d & \text{if } f_0^m(\mathbf{t}_{c_d}) \leq Q_d - R_d. \end{cases} \tag{59}$$

Similarly,

$$\mathcal{F}^1(\mathbf{t}_c) = \min_{m_1} r_1 m_1 + f_1^m(\mathbf{t}_c)\theta_\alpha(1 - m_1) \tag{60}$$

$$= \min_{m_1} (\lambda_d + \delta)m_1 + \theta_\alpha(1 - m_1) f_1^m(\mathbf{t}_c) \tag{61}$$

$$= \min_{m_1} (\gamma_{\max} - \gamma_\alpha)m_1 + \frac{\gamma_{\max} - \gamma_\alpha}{Q_\alpha}(1 - m_1) f_1^m(\mathbf{t}_c) \tag{62}$$

$$= \begin{cases} \lambda_{\max} - \lambda_\alpha & \text{if } f_1^m(\mathbf{t}_c) \geq Q_\alpha, \\ f_1^m(\mathbf{t}_c)\theta_\alpha & \text{if } f_1^m(\mathbf{t}_c) < Q_\alpha \end{cases} \tag{63}$$

$$= \begin{cases} \lambda_{\max} - \lambda_\alpha & \text{if } f_0^m(\mathbf{t}_c) \leq P - Q_\alpha, \\ f_1^m(\mathbf{t}_c)\theta_\alpha & \text{if } f_0^m(\mathbf{t}_c) > P - Q_\alpha. \end{cases} \tag{64}$$

Adding (59) and (64) and using the relations[10]

$$f_0^m(t_{c_d}) \leq Q_d - R_d \implies f_0^m(\mathbf{t}_c) \leq P - Q_d, \tag{65}$$

$$f_0^m(\mathbf{t}_c) > P - Q_d \implies f_0^m(t_{c_d}) > Q_d - R_d \tag{66}$$

we get:

$$\mathcal{F}^0(\mathbf{t}_{c_a}) + \mathcal{F}^1(\mathbf{t}_{c_a}) = \begin{cases} \lambda_{\max} - \lambda_d + (P - f_0^m(\mathbf{t}_c))\theta_\alpha \\ \quad \text{if } f_0^m(\mathbf{t}_c) > P - Q_\alpha, \\ f_0^m(\mathbf{t}_{c_d})\theta_d + \lambda_{\max} - \lambda_\alpha \\ \quad \text{if } f_0^m(\mathbf{t}_{c_d}) \leq Q_d - R_d, \\ \lambda_{\max} - \lambda_\alpha + \lambda_{\max} - \lambda_d \\ \quad \text{otherwise.} \end{cases} \tag{67}$$

Substituting in (56) and simplifying we get

$$\psi_c^m(\mathbf{t}_c, \mathbf{t}_{c_d}) = \begin{cases} \lambda_\alpha + (P - f_0^m(\mathbf{t}_c))\theta_\alpha \\ \quad \text{if } f_0^m(\mathbf{t}_c) > P - Q_\alpha, \\ \lambda_d + f_0^m(\mathbf{t}_{c_d})\theta_d \\ \quad \text{if } f_0^m(\mathbf{t}_{c_d}) \leq Q_d - R_d, \\ \lambda_{\max} \\ \quad \text{otherwise} \end{cases} \tag{68}$$

which is the same as (42). □

[10] These relations are derived from the constraints $Q_i + Q_j < P, \forall i \neq j \in \mathcal{L}$ and $W(s) \leq P, \forall s \subseteq c$.

References

Alahari, K., Kohli, P., & Torr, P. (2008). Reduce, reuse and recycle: efficiently solving multi-label MRFs. In *IEEE conference on computer vision and pattern recognition*.

Blake, A., Rother, C., Brown, M., Perez, P., & Torr, P. (2004). Interactive image segmentation using an adaptive GMMRF model. In *European conference on computer vision* (pp. I: 428–441).

Borenstein, E., & Malik, J. (2006). Shape guided object segmentation. In *IEEE conference on computer vision and pattern recognition* (pp. 969–976).

Boros, E., & Hammer, P. (2002). Pseudo-boolean optimization. *Discrete Applied Mathematics*, *123*(1–3), 155–225.

Boykov, Y., & Jolly, M. (2001). Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *International conference on computer vision* (pp. I: 105–112).

Boykov, Y., Veksler, O., & Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(11), 1222–1239.

Bray, M., Kohli, P., & Torr, P. (2006). Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In *European conference on computer vision* (pp. 642–655).

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(5), 603–619.

Felzenszwalb, P., & Huttenlocher, D. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, *59*(2), 167–181.

Flach, B. (2002). *Strukturelle bilderkennung* (Tech. Rep.). Universit at Dresden.

Freedman, D., & Drineas, P. (2005). Energy minimization via graph cuts: Settling what is possible. In *IEEE conference on computer vision and pattern recognition* (pp. 939–946).

Fujishige, S. (1991). *Submodular functions and optimization*. Amsterdam: North-Holland.

He, X., Zemel, R., & Carreira-Perpiñán, M. (2004). Multiscale conditional random fields for image labeling. In *IEEE conference on computer vision and pattern recognition (2)* (pp. 695–702).

He, X., Zemel, R., & Ray, D. (2006). Learning and incorporating top-down cues in image segmentation. In *European conference on computer vision* (pp. 338–351).

Hoiem, D., Efros, A., & Hebert, M. (2005a). Automatic photo pop-up. *ACM Transactions on Graphics*, *24*(3), 577–584.

Hoiem, D., Efros, A., & Hebert, M. (2005b). Geometric context from a single image. In *International conference on computer vision* (pp. 654–661).

Huang, R., Pavlovic, V., & Metaxas, D. (2004). A graphical model framework for coupling MRFs and deformable models. In *IEEE conference on computer vision and pattern recognition* (Vol. 11, pp. 739–746).

Ishikawa, H. (2003). Exact optimization for Markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*, 1333–1336.

Kohli, P., Kumar, M., & Torr, P. (2007). P^3 and beyond: solving energies with higher order cliques. In *IEEE conference on computer vision and pattern recognition*.

Kohli, P., Ladicky, L., & Torr, P. (2008). Robust higher order potentials for enforcing label consistency. In *CVPR*.

Kolmogorov, V. (2006). Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(10), 1568–1583.

Kolmogorov, V., & Zabih, R. (2004). What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(2), 147–159.

Komodakis, N., & Tziritas, G. (2005). A new framework for approximate labeling via graph cuts. In *International conference on computer vision* (pp. 1018–1025).

Komodakis, N., Tziritas, G., & Paragios, N. (2007). Fast, approximately optimal solutions for single and dynamic MRFs. In *CVPR*.

Kumar, M., & Torr, P. (2008). Improved moves for truncated convex models. In *Proceedings of advances in neural information processing systems*.

Kumar, M., Torr, P., & Zisserman, A. (2005). Obj cut. In *IEEE conference on computer vision and pattern recognition (1)* (pp. 18–25).

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labelling sequence data. In *International conference on machine learning* (pp. 282–289).

Lan, X., Roth, S., Huttenlocher, D., & Black, M. (2006). Efficient belief propagation with learned higher-order Markov random fields. In *European conference on computer vision* (pp. 269–282).

Lauristzen, S. (1996). *Graphical models*. Oxford: Oxford University Press.

Lempitsky, V., Rother, C., & Blake, A. (2007). Logcut—efficient graph cut optimization for Markov random fields. In *ICCV*.

Levin, A., & Weiss, Y. (2006). Learning to combine bottom-up and top-down segmentation. In *European conference on computer vision* (pp. 581–594).

Lovasz, L. (1983). Submodular functions and convexity. In *Mathematical programming: the state of the art* (pp. 235–257).

Orlin, J. (2007). A faster strongly polynomial time algorithm for submodular function minimization. In *Proceedings of integer programming and combinatorial optimization* (pp. 240–251).

Paget, R., & Longstaff, I. (1998). Texture synthesis via a noncausal nonparametric multiscale Markov random field. *IEEE Transactions on Image Processing*, *7*(6), 925–931.

Potetz, B. (2007). Efficient belief propagation for vision using linear constraint nodes. In *IEEE conference on computer vision and pattern recognition*.

Rabinovich, A., Belongie, S., Lange, T., & Buhmann, J. (2006). Model order selection and cue combination for image segmentation. In *IEEE conference on computer vision and pattern recognition (1)* (pp. 1130–1137).

Ren, X., & Malik, J. (2003). Learning a classification model for segmentation. In *International conference on computer vision* (pp. 10–17).

Roth, S., & Black, M. (2005). Fields of experts: A framework for learning image priors. In *IEEE conference on computer vision and pattern recognition* (pp. 860–867).

Rother, C., Kolmogorov, V., & Blake, A. (2004). Grabcut: interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics* (pp. 309–314).

Russell, B., Freeman, W., Efros, A., Sivic, J., & Zisserman, A. (2006). Using multiple segmentations to discover objects and their extent in image collections. In *IEEE conference on computer vision and pattern recognition (2)* (pp. 1605–1614).

Schlesinger, D., & Flach, B. (2006). *Transforming an arbitrary minsum problem into a binary one* (Tech. Rep. TUD-FI06-01). Dresden University of Technology, April 2006.

Sharon, E., Brandt, A., & Basri, R. (2001). Segmentation and boundary detection using multiscale intensity measurements. In *IEEE conference on computer vision and pattern recognition (1)* (pp. 469–476).

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 888–905.

Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2006). TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European conference on computer vision* (pp. 1–15).

Tu, Z., & Zhu, S. (2002). Image segmentation by data-driven Markov chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(5), 657–673.

Veksler, O. (2007). Graph cut based optimization for MRFs with truncated convex priors. In *CVPR*.

Wainwright, M., Jaakkola, T., & Willsky, A. (2005). Map estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*, *51*(11), 3697–3717.

Wang, J., Bhat, P., Colburn, A., Agrawala, M., & Cohen, M. (2005). Interactive video cutout. *ACM Transactions on Graphics*, *24*(3), 585–594.

Yedidia, J., Freeman, W., & Weiss, Y. (2000). Generalized belief propagation. In *NIPS* (pp. 689–695).