

Robust Higher Order Potentials for Enforcing Label Consistency

Pushmeet Kohli

Microsoft Research, Cambridge, UK

L'ubor Ladický Philip H. S. Torr

Oxford Brookes University, Oxford, UK

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 proposed recently in [3]. 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.

Index Terms

Discrete Energy Minimization, Markov and Conditional Random Fields, Object Recognition and Segmentation.

I. 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 unsupervised¹ segmentation algorithms [5]–[7]. 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 classification². 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 [5], i.e. segments do not contain multiple objects. As observed by [8] and [9] this is not always the case and segments obtained using unsupervised segmentation methods are often wrong. To overcome these problems [8] and [9] 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 figure 1).

A. *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

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

²In 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.

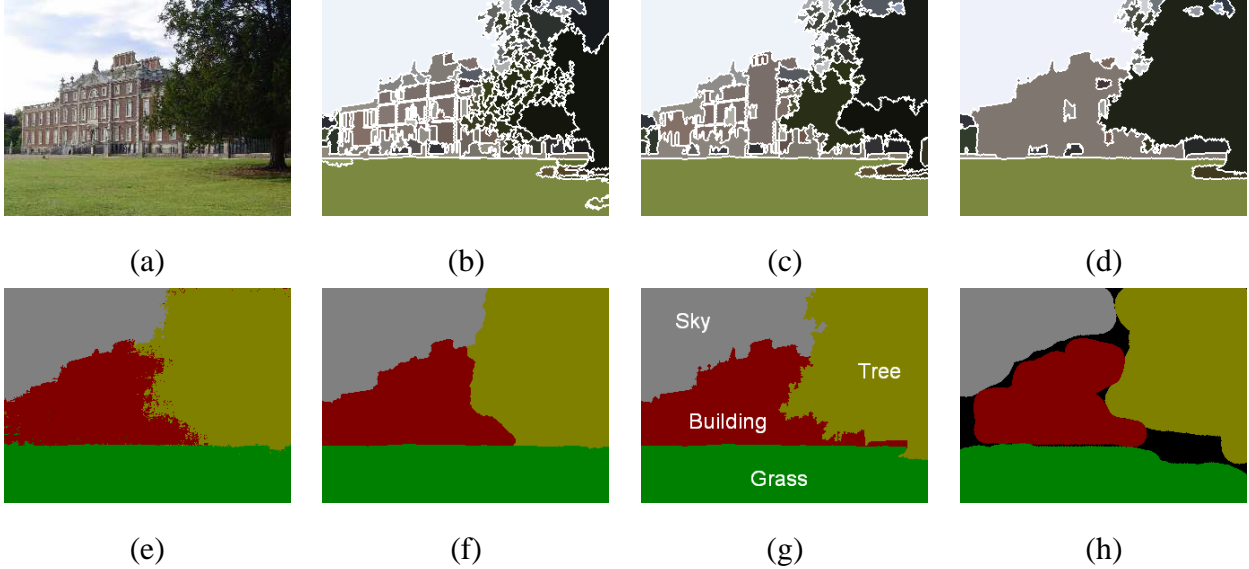


Fig. 1. Incorporating higher order potentials for object segmentation. (a) An image from the MSRC-21 dataset. (b),(c) and (d) Unsupervised image segmentation results generated by using different parameters values in the mean-shift segmentation algorithm [10]. (e) The object segmentation obtained using the unary likelihood potentials from Textonboost. (f) The result of performing inference in the pairwise CRF defined in section II. (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.

algorithms which integrate *object specific* top-down information with *image based* low-level features [11]–[15]. 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.* [16] 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 condition 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

1
2
3
4 order potentials defined on superpixels dramatically improves the object segmentation result. In
5 particular, it leads to segmentations with much better definition of object boundaries as shown
6 in figure 1.
7
8
9

10 11 *B. Higher Order CRFs*

12
13 Higher order random fields are not new to computer vision. They have been long used to
14 model image textures [17]–[19]. The initial work in this regard has been quite promising and
15 higher order CRFs have been shown to improve results for problems such as image denoising
16 and restoration [19], and texture segmentation [3]. However their use has been quite limited due
17 to lack of efficient algorithms for performing inference in these models.
18
19
20
21
22

23 Traditional inference algorithms such as BP are quite computationally expensive for higher
24 order cliques although recent work has improved their performance for certain classes of potential
25 functions. Lan *et al.* [17] proposed approximation methods for BP to make efficient inference
26 possible in higher order MRFs. This was followed by the recent work of Potetz [20] in which
27 he showed how belief propagation can be efficiently performed in graphical models containing
28 moderately large cliques. However, as these methods were based on BP, they were quite slow
29 and took minutes or hours to converge.
30
31
32
33
34

35 Kohli *et al.* [3] recently showed how certain higher order clique potentials can be minimized
36 using move making algorithms for approximate energy minimization such as α -expansion and
37 $\alpha\beta$ -swap [21]. They introduced a class of higher order potentials called the P^n Potts model
38 and showed how the optimal *expansion* and *swap* moves for energy functions containing these
39 potentials can be computed in polynomial time by solving a st-mincut problem. The complexity
40 of their algorithm increased linearly with the size of the clique and thus it was able to handle
41 cliques composed of thousands of latent variables.
42
43
44
45
46
47

48 The higher order energy functions characterizing the higher order CRFs arising from our work
49 are more general in form than the P^n Potts model and thus cannot be minimized efficiently
50 using the algorithm of [3]. We introduce a new family of higher order potentials which is
51 a generalization of the P^n Potts class. The potential functions belonging to this family are
52 parameterized with a truncation parameter Q which controls their *robustness*. We will show how
53 energy functions composed of these *robust* potentials can be minimized using move making
54 algorithms such as α -expansion and $\alpha\beta$ -swap. Specifically, we show how the optimal swap and
55
56
57
58
59
60
61
62
63
64
65

1
2
3
4 expansion moves for such potentials can be found using algorithms for computing the st-mincut.
5
6

7 8 *C. Organization of the Paper* 9

10 This paper proposes a general framework for solving labelling problems which has the ability
11 to utilize higher order potentials defined on segments³. We test this framework on the problem
12 of object segmentation and recognition by integrating label consistency and shape based terms
13 defined on segments with conventional unary and pairwise potentials. We show how inference in
14 this framework can be efficiently performed by extending the recent work on minimizing energy
15 function with higher order cliques [3]. To summarize, the novelties of our approach include:
16
17
18
19
20

- 21 1) The method for efficiently solving a new family of higher order potentials which we call
22 the robust P^n model, and is a generalization of the P^n Potts model.
23
- 24 2) A novel higher order region consistency potential which is a strict generalization of the
25 commonly used pairwise contrast sensitive smoothness potential.
26
27
- 28 3) The application of higher order CRFs for object segmentation and recognition which
29 integrate the above mentioned higher order potentials with conventional unary and pairwise
30 potentials based on colour, location, texture, and smoothness.
31
32
33

34 An outline of the paper follows. In section II we discuss the basic theory of conditional random
35 fields. We then show how pairwise CRFs can be used to model labelling problems like object
36 segmentation. In section III we augment the pairwise CRF model by incorporating novel higher
37 order potentials based on super-pixel segmentations. In section IV we review the work on move
38 making algorithms for solving higher order energy functions. The potential functions which can
39 be solved using our method are described in section V. Finally, in section VI we show how the
40 optimal expansion and swap moves for energy functions composed of such potentials can be
41 computed by solving a st-mincut problem. The experimental results of our method are given in
42 section VII. These include qualitative and quantitative results on well known and challenging
43 data sets for object segmentation and recognition. The conclusions and directions for future
44 work are listed in section VIII. The proofs of the theorems stated in the paper are given in the
45 Appendix.
46
47
48
49
50
51
52
53
54
55

56
57
58 ³An earlier version of this paper appeared as [22].
59
60
61
62
63
64
65

II. PRELIMINARIES

We start by providing the basic notation used in the paper. Consider a discrete random field \mathbf{X} defined over a lattice $\mathcal{V} = \{1, 2, \dots, N\}$ 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 $\mathcal{L} = \{l_1, l_2, \dots, l_k\}$. 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 $\Pr(\mathbf{x}|\mathbf{D})$ over the labellings of a Markov Random Field (MRF) is a *Gibbs* distribution and can be written as

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

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 [4]. The corresponding Gibbs energy is defined as

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

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

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

A conditional random field (CRF) may be viewed as an MRF globally conditioned on the data [23]. 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.

A. 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:

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

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 [14], [16], [24]–[26]. The unary potential used by us can be written as:

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

where θ_T , θ_{col} , and θ_l are parameters weighting the potentials obtained from TextonBoost(ψ_T) [16], 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:

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

where the function $g(i, j)$ is an edge feature based on the difference in colors of neighboring pixels [27]. It is typically defined as:

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

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 [16], [26], [27] for more details.

1) 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 [21]. The resulting segmentation can be seen in figure 1. We also tried other energy minimization algorithms such as sequential tree-reweighted message passing (TRW-S) [28], [29]. The α -expansion algorithm was preferred because it was faster and gave a solution with lower energy compared to TRW-S.

1
2
3
4 2) *Need for Higher Order CRFs:* The use of Potts model [21] potentials in the CRF model
5 makes it favour smooth object boundaries. Although this improves results in most cases it
6 also introduces an undesirable side effect. Smoothness potentials make the model incapable
7 of extracting the fine contours of certain object classes such as trees and bushes. As seen in
8 the results, segmentations obtained using pairwise CRFs tend to be oversmooth and quite often
9 do not match the actual object contour. In the next section we show how these results can
10 be significantly improved by using higher order potentials derived from multiple segmentations
11 obtained from an unsupervised image segmentation method.
12
13
14
15
16
17
18
19

20 III. INCORPORATING HIGHER ORDER POTENTIALS

21
22 Methods based on grouping regions for segmentation generally make the assumption that all
23 pixels constituting a particular segment (or region) belong to the same object [5]. This is not
24 always the case, and image segments quite often contain pixels belonging to multiple object
25 classes. For instance, in the segmentations shown in figure 2 the bottom image segment contains
26 some ‘building’ pixels in addition to all the grass pixels.
27
28
29
30

31 Unlike other object segmentation algorithms which use the label consistency in segments
32 as a hard constraint, our method uses it as a *soft constraint*. This is done by using higher
33 order potentials defined on the image segments generated using unsupervised segmentation
34 algorithms. Specifically, we augment the pairwise CRF model explained in the previous section
35 by incorporating higher order potentials defined on sets or regions of pixels. The Gibbs energy
36 of this higher order CRF can now be written as:
37
38
39
40
41

$$42 E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j) + \sum_{c \in \mathcal{S}} \psi_c(\mathbf{x}_c), \quad (8)$$

43 where \mathcal{S} refers to a set of image segments (or super-pixels), and ψ_c are higher order potentials
44 defined on them. In our experiments, the set \mathcal{S} consisted of all segments of multiple segmentations
45 of an image obtained using an unsupervised image segmentation algorithm such as mean-
46 shift [10] (see section III-D for more details). We will now describe in detail how these higher
47 order potentials are defined.
48
49
50
51
52
53

54 A. Region based consistency potential

55 The *region consistency potential* is similar to the smoothness prior present in pairwise CRFs [27].
56 It favours all pixels belonging to a segment taking the same label, and as will be shown later is
57
58
59
60
61
62
63
64
65

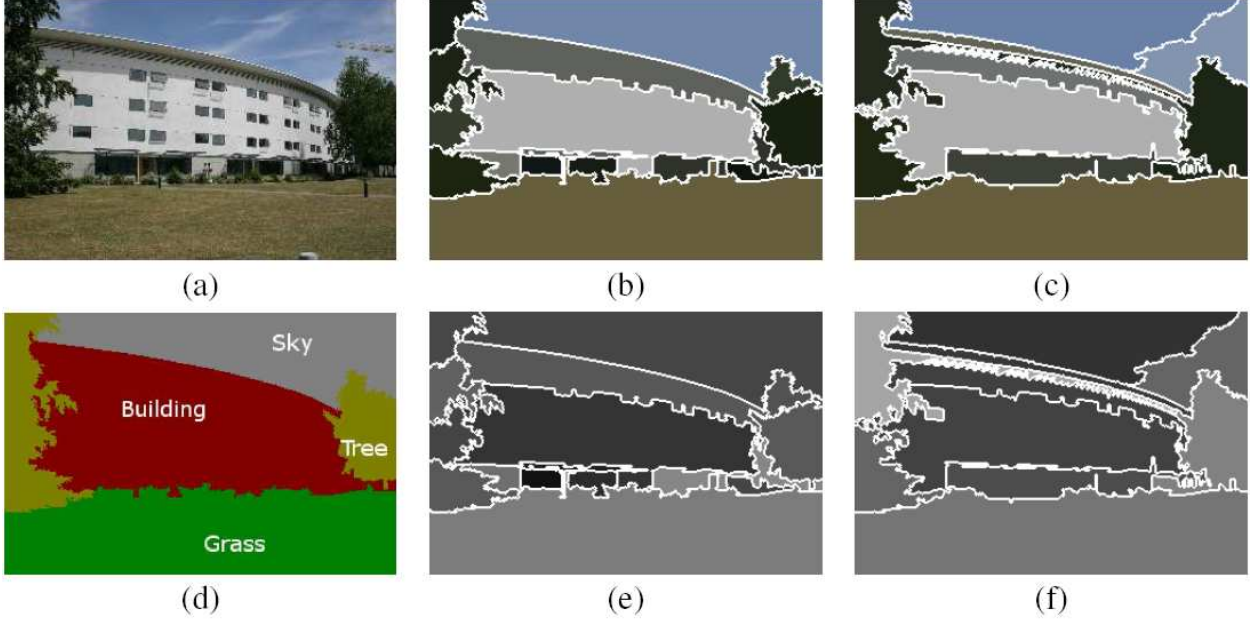


Fig. 2. *Quality sensitive region consistency prior. (a) An image from the MSRC data set. (b) and (c) Two different segmentations of the image obtained using different parameter values for the mean-shift algorithm. (d) A hand labelled object segmentation of the image. (e) and (f) The value of the variance based quality function $G(c)$ (see equation 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.*

particularly useful in obtaining object segmentations with fine boundaries. It takes the form of a \mathcal{P}^n Potts model (see equation 16) [3]:

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

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 section VII. The reader should note that this potential takes multiple variables as argument and thus cannot be expressed in the conventional pairwise CRF model.

B. 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 figure 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 [30] which uses inter and intra region similarity to measure the quality or goodness of a segment. Formally, the potential function is written as:

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

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.

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

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

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

This is the same as the pairwise potential (6) commonly used in pairwise CRFs for different image labelling problems [26], [27]. 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 figure 2.

C. 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

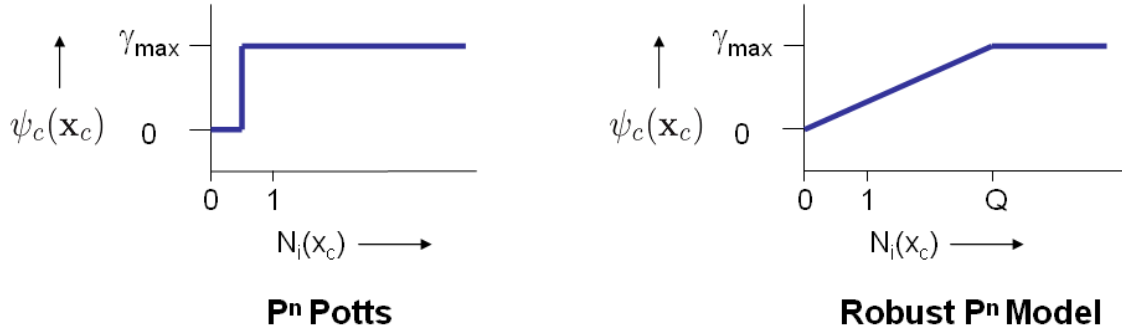


Fig. 3. Behaviour of the rigid P^n Potts potential and the Robust P^n model potential. The figure shows how the cost enforced by the two higher order potentials changes with the number of variables in the clique not taking the dominant label i.e. $N_i(\mathbf{x}_c) = \min_k(|c| - n_k(\mathbf{x}_c))$.

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 figure 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:

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

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 the rigidity of the higher order clique potential. We will show in section IV 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 figure 3). This enables the robust potential to allow some variables in the clique to take different labels. In the image shown in figure 4, the robust P^n potentials allows some pixels of the ‘sky’ segment to take the label ‘bird’ thus producing a much better segmentation. Experiment results are shown for multiple values of the truncation parameter Q . More qualitative results can be seen in figure 13.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

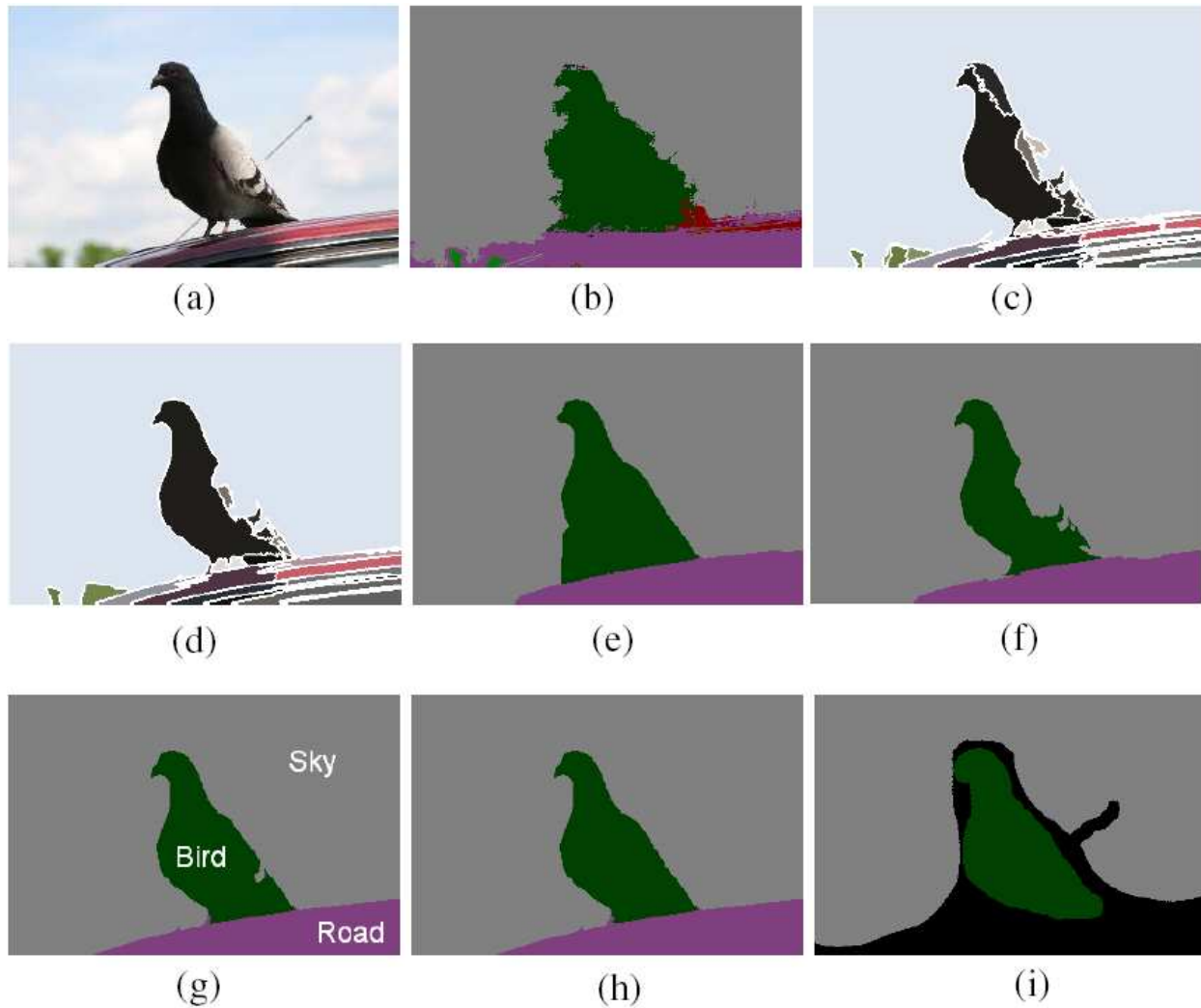


Fig. 4. Object segmentation and recognition using the Robust P^n higher order potentials (13). (a) Original Image. (b) Labelling from unary likelihood potentials from Textonboost [16]. (c) and (d) Segmentations obtained by varying the parameters of the Mean shift algorithm for unsupervised image segmentation [10]. (e) Result obtained using pairwise potential functions as described in [16]. (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.

D. 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



Fig. 5. *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.*

non-overlapping segments. The computer vision literature contains algorithms for sampling the likely segmentations of an image [31] or for generating multi-scale segmentations [32]. However, following in the footsteps of [9] we choose to generate multiple segmentations by varying the parameters of the mean shift segmentation algorithm [10]. This method belongs to the class of unsupervised segmentation algorithms which work by clustering pixels on the basis of low level image features [10], [33], [34]. They have been shown to give decent results which have proved to be useful for many applications [6], [8], [35].

The kernel used in the mean shift algorithm is defined as the product of spatial and range kernels. The spatial domain 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 figure 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 $(h_s, h_r) = \{(7, 6.5), (7, 9.5), (7, 15)\}$.

IV. 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 [36]. 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 [28], [29], [37], and move making algorithms such as the graph cut based α -expansion and $\alpha\beta$ -swap [21]. 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.

A. 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 [21] whose move space size increases exponentially with the number of variables involved in the energy function. We will use the notation of [3] 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 $T_\alpha(\cdot)$ for an α -expansion move transforms the label of a random variable X_i as

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

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 $T_{\alpha\beta}(\cdot)$ for an $\alpha\beta$ -swap transforms the label of a random variable x_i as

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

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.

B. Recent Developments

The last few years have seen a lot of interest in graph cut based move algorithms for energy minimization. Komodakis *et al.* [38], [39] 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.* [2] 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 [40] proposed a move algorithm in which variables

1
2
3
4 can choose any label from a range of labels. They showed that this move space allowed them
5 to obtain better minima of energy functions with truncated convex pairwise terms. Kumar and
6 Torr [1] have since shown that the range move algorithm achieves the same guarantees as the
7 ones obtained by methods based on the standard linear programming relaxation. More recently,
8 Lempitsky, Rother and Blake [41] proposed an algorithm which encoded labels by a binary
9 string. During each move, the variables were allowed to change a particular bit of the binary
10 string. They showed that this particular move encoding strategy results in a substantial speedup
11 when minimizing energy functions with large label sets.
12
13
14
15
16
17
18
19

20 C. Computing Moves using Graph Cuts

21 We had discussed in section IV-A that the optimal expansion and swap moves can be computed
22 by minimizing a (move) function of binary variables. Functions of binary variables ($F : \{0, 1\} \rightarrow \mathbb{R}$)
23 are usually referred to as *Pseudo-boolean* functions. It is known that a move function can
24 be minimized in polynomial time if it is submodular [45] (See Appendix A). Submodular set
25 functions are encountered in many areas of research and are particularly useful in combinatorial
26 optimization, probability and geometry [42], [43]. Many optimization problems relating to sub-
27 modular functions can be solved efficiently. In this respect they are similar to convex/concave
28 functions encountered in continuous optimization.
29
30
31
32
33
34
35
36

37 Algorithms for submodular function minimization have high runtime complexity. Although
38 recent work has been successful in reducing the runtime complexity of these algorithms, they
39 are still quite computationally expensive and cannot be used to minimize large functions. For
40 instance, the complexity of the current best algorithm for general submodular function mini-
41 mization is $O(n^5\mathcal{T} + n^6)$ where \mathcal{T} is the time taken to evaluate the function [45]. This algorithm
42 improved upon the previous best algorithm by a factor of $n \log n$.
43
44
45
46
47

48 Some submodular functions can be minimized much more efficiently by solving an st-mincut
49 problem [44]. Specifically, all submodular functions of binary variables of order at most 3 can be
50 minimized in this manner [36], [44]. Researchers have shown that certain higher order functions
51 can be transformed into submodular functions of order 2, and thus can also be minimized [44],
52 [46]. The same transformation technique can be used to minimize some functions of multi-valued
53 variables [47]–[49].
54
55
56
57
58

59 Solving pairwise CRFs using move making algorithms involves computing optimal moves
60
61
62
63
64
65

1
2
3
4 by minimizing quadratic pseudo-boolean move functions. Boykov *et al.* [21] showed that all
5 expansion move functions encountered while minimizing an energy function composed of metric
6 potential functions are submodular. As these functions are quadratic, they can be efficiently
7 minimized by solving an equivalent st-mincut problem. They also showed that all $\alpha\beta$ -swap
8 move functions can be exactly minimized if the pairwise potentials of the CRF are semi-metric.
9

14 V. ROBUST HIGHER ORDER POTENTIALS

15
16
17 Kohli *et al.* [3] recently characterized a class of higher order clique potentials for which
18 the optimal expansion and swap moves can be computed by minimizing a submodular function.
19 However, as discussed earlier, minimizing a general submodular function is quite computationally
20 expensive and it is infeasible to apply this procedure to minimize energy functions encountered
21 in computer vision problems, which generally involve millions of random variables. In the same
22 work, they also introduced a class of higher order clique potentials called the P^n Potts model
23 and showed that the optimal *expansion* and *swap* moves for energy functions containing these
24 potentials can be computed in polynomial time by solving a st-mincut problem. The P^n Potts
25 model was defined as:
26
27

$$28 \psi_c(\mathbf{x}_c) = \begin{cases} \gamma_k & \text{if } x_i = l_k, \forall i \in c, \\ \gamma_{\max} & \text{otherwise.} \end{cases} \quad (16)$$

29
30 where $\gamma_{\max} \geq \gamma_k, \forall l_k \in \mathcal{L}$. This potential is a higher order generalization of the widely used
31 Potts model potential which is defined over cliques of size two as $\psi_{ij}(a, b) = \gamma_k$ if $a = b = l_k$
32 and γ_{\max} otherwise.
33

34
35 In this paper we introduce a novel family of higher order potentials which we call the Robust
36 P^n model. This family contains the P^n Potts model as well as its *robust* variants, and can
37 be used for modelling many computer vision problems. We show that the optimal swap and
38 expansion move energy functions for any Robust P^n model potential can be transformed into a
39 second order submodular function by the addition of at most two binary auxiliary variables. This
40 transformation enables us to find the optimal swap and expansion moves in polynomial time⁴.
41
42

43
44
45
46
47
48
49
50
51
52
53
54
55
56 ⁴All second order submodular functions of binary variables can be minimized exactly in polynomial time by solving an
57 st-mincut problem [36], [44].
58
59
60
61
62
63
64
65

The Robust P^n model potentials take the form:

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

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:

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

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 $\mathbf{X} = \{X_1, X_2, \dots, X_7\}$ where each $X_i, i \in \{1, 2, \dots, 7\}$ 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 γ_{\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 γ_{\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 figure 3). This enables the robust potential to allow some variables in the clique to take different labels.

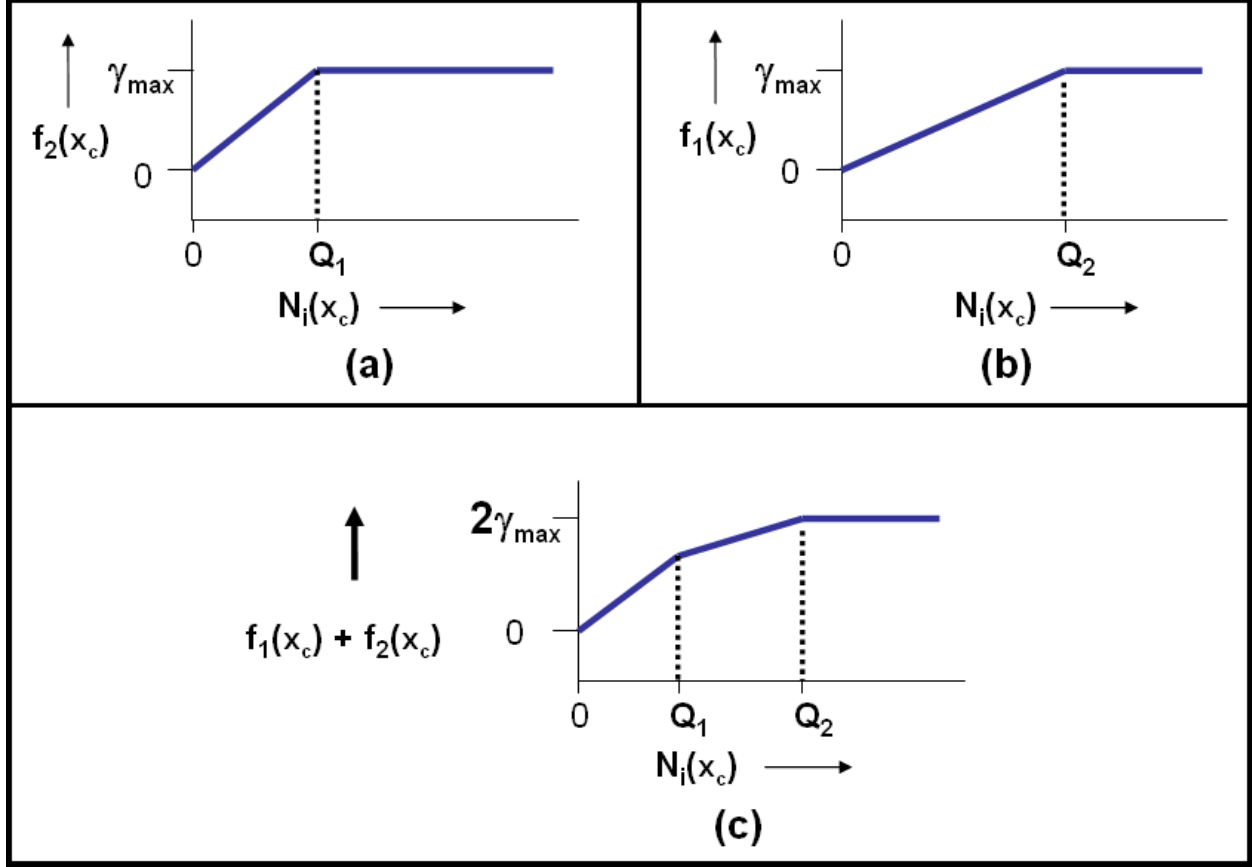


Fig. 6. Approximating Concave Consistency Potentials. The figure shows the result of combining two robust higher order potentials (a) and (b). The resulting potential function is shown in (c).

A. 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:

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

where \mathcal{F}_c is a non-decreasing concave function⁵. This is illustrated in figure 6.

B. 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 to transform the higher order swap and expansion move energy functions

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

to second order functions. The potentials belonging to this new family have the form:

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

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

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

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

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

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:

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

$Q_k, k \in \mathcal{L}$ 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 energy⁶.

VI. 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 minimizing them using graph cuts.

⁶Higher 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.

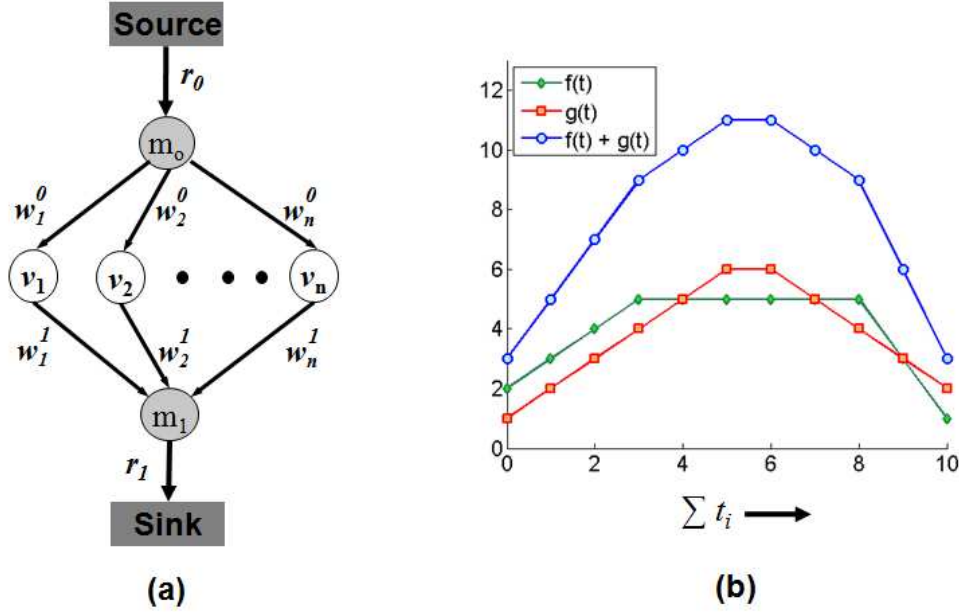


Fig. 7. (a) Graph construction for minimizing higher order quadratic functions of the form (25) using the transformation given in theorem 1. For every binary variable t_i in the energy function there is a corresponding graph node v_i . The minimum cost source sink cut (st-mincut) divides the graph into two sets: the source set (S) and the sink set (T). $v_i \in (S)$ implies $t_i = 1$ while $v_i \in (T)$ 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 ($c = \{1, 2, \dots, 10\}$). 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 $g(\mathbf{t}) = \min(2 + \sum_{i \in c} (1 - t_i), 1 + \sum_{i \in c} t_i, 6)$.

A. Transforming Higher Order Move Energies

The problem of transforming a general submodular higher order function to a second order one has been well studied [44]. 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:

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

can be transformed to submodular quadratic pseudo-boolean functions, and hence can be minimized using graph cuts. 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 , θ_{\max} are parameters of the potential satisfying the constraints $\theta_{\max} \geq \theta_0$, $\theta_{\max} \geq \theta_1$, and

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

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:

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

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

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

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.

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

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

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

where $F_c : \mathbb{R} \rightarrow \mathbb{R}$ is any concave function. See figure 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 $W(s)$, $s \subseteq c$:

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

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

B. 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:

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

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

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

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:

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

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:

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

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

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:

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

are satisfied, then the expression:

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

is greater than γ_{\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:

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

which can be ignored while computing the swap moves. However, if constraints (36) are not satisfied, the move energy becomes: $\psi_c^m(\mathbf{x}_{c_a}^p, \mathbf{t}_{c_a}) =$

$$\min\{(W(c_a) - f_0^m(\mathbf{t}_{c_a}))\theta_\alpha + \lambda_\alpha, (W(c_a) - f_1^m(\mathbf{t}_{c_a}))\theta_\beta + \lambda_\beta, \lambda_{\max}\} \quad (39)$$

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 equation (27), and can be transformed to a pairwise function by introducing binary auxiliary variables m_0 and m_1 as:

$$\begin{aligned} \psi_c^m(\mathbf{t}_c) = & \min_{m_0, m_1} \left(r_0(1 - m_0) + \theta_\beta m_0 \sum_{i \in c_a} w_i(1 - t_i) \right. \\ & \left. + r_1 m_1 + \theta_\alpha(1 - m_1) \sum_{i \in c_a} w_i t_i - \delta, \right) \end{aligned} \quad (40)$$

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 [36]. The graph construction for minimizing the energy function (40) is shown in figure (8a). 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.

C. 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: $\psi_c^m(\mathbf{t}_c) = \psi_c(T_\alpha(\mathbf{x}_c^p, \mathbf{t}_c))$ or, $\psi_c^m(\mathbf{t}_c) =$

$$\min\{\lambda_\alpha + \theta_\alpha \sum_{i \in c} w_i t_i, \lambda_d + \theta_d \sum_{i \in c_d} w_i(1 - t_i), \lambda_{\max}\} \quad (41)$$

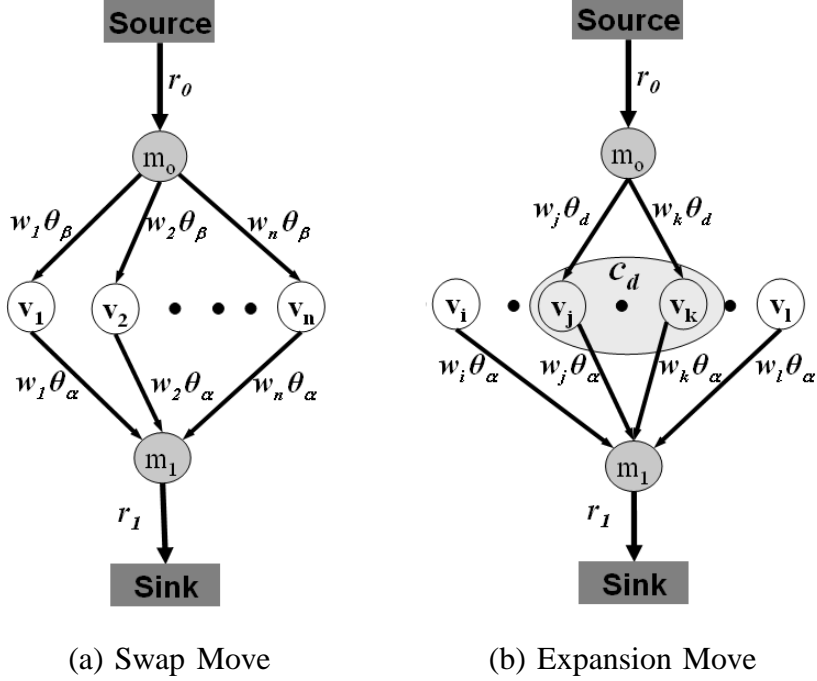


Fig. 8. (a) Graph construction for minimizing the swap energy function (40). (b) Graph construction for minimizing the expansion move energy function (43). For every binary move variable t_i in the energy function there is a corresponding graph node v_i . The minimum cost source sink cut (st-mincut)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

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:

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

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 .

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

$$\begin{aligned} \psi_c^m(\mathbf{t}_c) = \min_{m_0, m_1} & \left(r_0(1 - m_0) + \theta_d m_0 \sum_{i \in c_d} w_i(1 - t_i) \right. \\ & \left. + r_1 m_1 + \theta_\alpha(1 - m_1) \sum_{i \in c} w_i t_i - \delta. \right) \end{aligned} \quad (43)$$

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

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

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

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

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

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

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

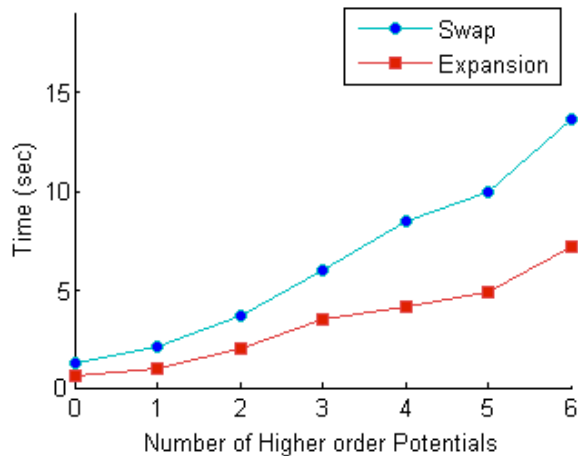
VII. EXPERIMENTS

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.

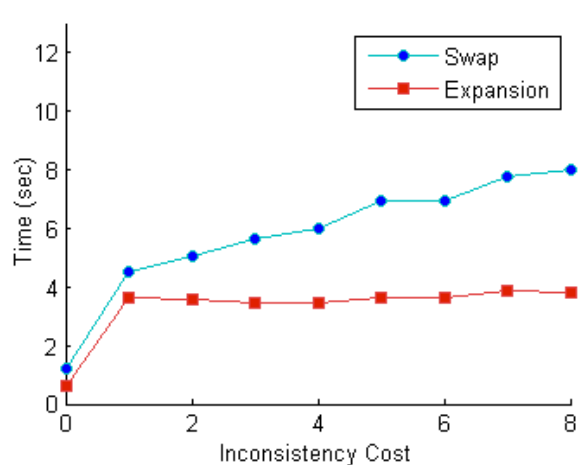
A. Computational Performance

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 [17] 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 tests⁷. 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.

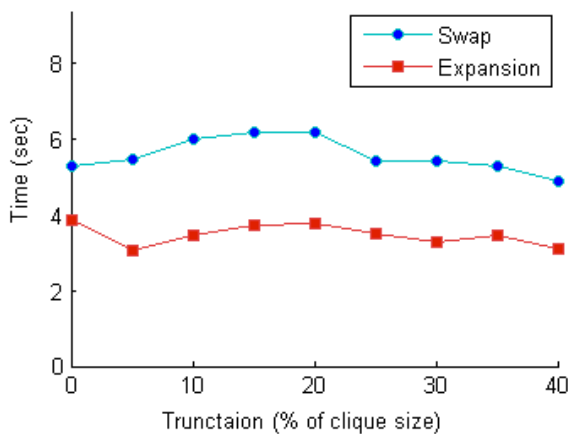
⁷It should be noted that the Robust P^n model potentials can be transformed into pairwise potentials by the addition of multi-label auxiliary variables. This enables the use of message passing algorithms for minimizing energy functions composed of them. However, the analysis of these transformations,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



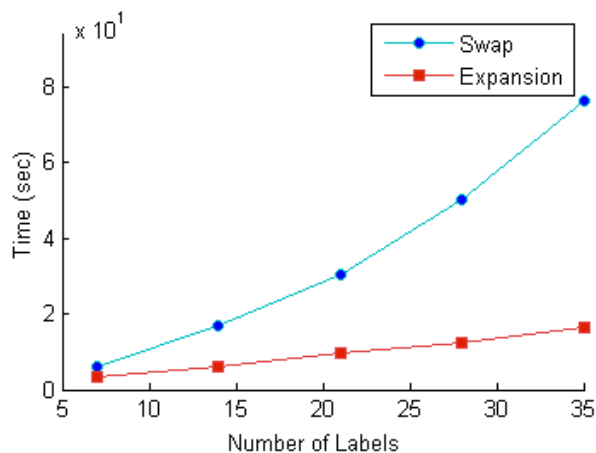
(a) No. of Potentials



(b) Inconsistency Cost



(c) Truncaion



(d) No. of Labels

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 [16]. 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 figure 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

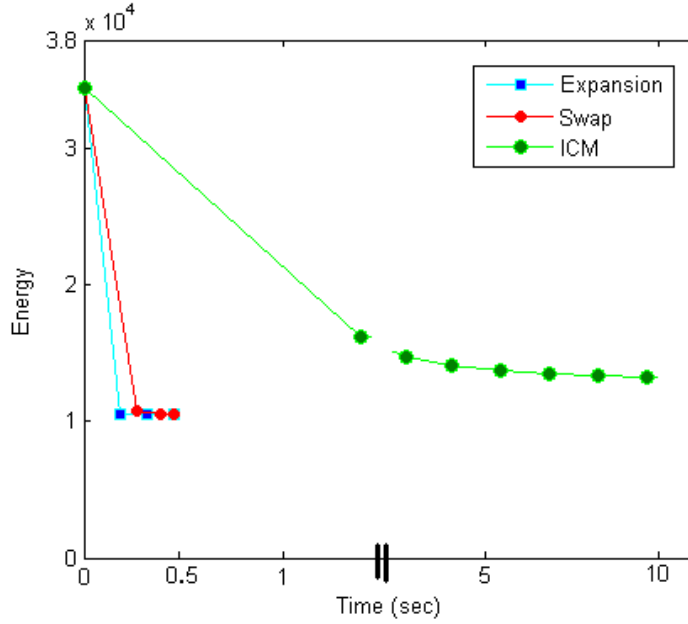


Fig. 10. Comparison of solution energy with respect to runtime. For the experiment, we used a CRF defined over a rectangular grid of 6000 random variables with a label set of size 7. The pairwise terms of the random field enforced 8 connectivity. The energy function used in the experiment had higher 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.

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 figure 9(b) shows the effect of parameter γ_{\max} of the Robust P^n model potentials on the convergence time. The graph in figure 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 graph 9(d).

B. Object Segmentation Results

For comparative evaluation of our method we implemented the state of the art TextonBoost [16] algorithm which uses a pairwise CRF. We then augmented the CRF model by adding higher order

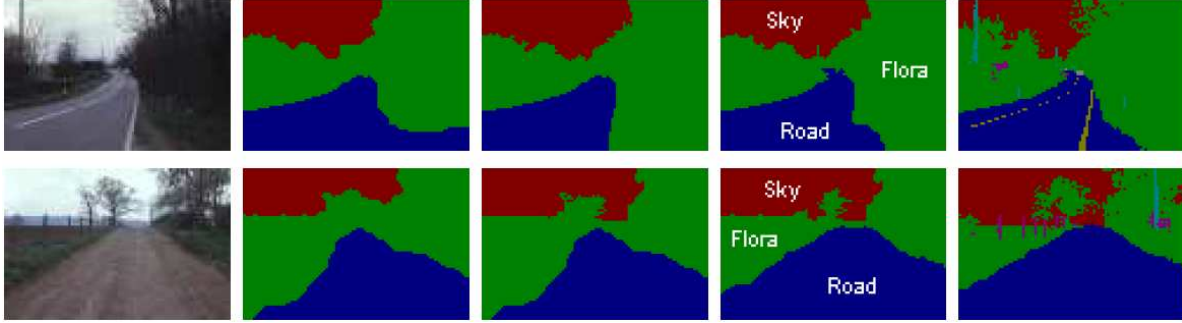


Fig. 11. *Qualitative object segmentation and recognition results. The first column shows the original image from the Sowerby-7 dataset. Column 2 shows the result of performing inference in the pairwise CRF model described in section II. The result obtained using the P^n Potts potential (10) is shown 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.*

potentials defined on segments obtained from mean-shift [10].

Datasets: We tested both the pairwise CRF and higher order CRF models on the MSRC-21 [16] and Sowerby-7 [5] 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.

C. 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 [16]. 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 kept these weights constant and learned the optimal parameters for pairwise potentials. Pairwise and higher order potentials have similar

1
2
3
4 functionality in the framework, thus learning of higher order parameters from the model with
5 optimal unary and pairwise parameters would lead to very low weights of higher order potentials.
6
7 Instead we learned optimal higher order parameters in CRF with only unary and higher order
8 potentials and in the last step the ratio between pairwise and higher order potentials. The final
9 trained coefficients for the MSRC dataset were $\theta_T = 0.52$, $\theta_{col} = 0.21$, $\theta_l = 0.27$, $\theta_p = 1.0$,
10 $\theta_v = 4.5$, $\theta_\beta = 16.0$, $\theta_\alpha = 0.8$, $\theta_p^h = 0.2$, $\theta_v^h = 0.5$, $\theta_\beta^h = 12.0$ ⁸. Parameter learning for higher
11 order CRFs is an ongoing topic of research.
12
13
14
15
16
17
18

19 *D. Quantitative Segmentation Results*

20
21 The results of our experiments show that integration of higher order P^n Potts model potentials
22 quantitatively and qualitatively improves segmentation results. The use of the robust potentials
23 lead to further improvements (see figure 4,11, 13 and 15). Inference on both the pairwise and
24 higher order CRF model was performed using the graph cut based expansion move algorithm.
25 The optimal expansion moves for the energy functions containing the Robust P^n potential (13)
26 were computed using the method described in the previous section .
27
28
29
30
31

32 **Effect of Multiple Segmentations:** The use of multiple segmentations allows us to obtain
33 accurate segmentations of objects with thin structures. For instance, consider the image shown in
34 figure 12 (a). Our method produces an accurate segmentation (figure 12 (g)) of the bird which,
35 unlike the solution of the pairwise CRF (figure 12 (f)), also contains the bird's leg. This result
36 does not require that many super-pixels contain both: a part of the bird's leg, and a part of the
37 bird's body. In fact, as shown in figures 12 (b) and (c), many super-pixels contain only the leg
38 and many other super-pixels contain only (a part of) the bird without the leg. As we explain
39 below, our method can work even if only one super-pixel contains both the bird body and leg
40 together.
41
42
43
44
45
46
47
48

49 The reader should observe that solution of the higher order CRF (figure 12 (g)) is roughly
50 *consistent*⁹ with all super-pixels present in the multiple segmentations (figures 12 (b), (c) and
51
52

53
54 ⁸The magnitude of the learned parameter values does not correctly reflect the relative strength (importance) of higher order
55 potentials vis a vis pairwise potentials. Higher order potential costs (see equations 9 and 10) are multiplied by a term dependent
56 on the size of the clique (segment). This is typically a large number and makes the cost of higher order potentials high compared
57 to that of the pairwise potentials.
58

59 ⁹The solution does not assign different labels to many pixels belonging to the the same super-pixel.
60
61
62
63
64
65

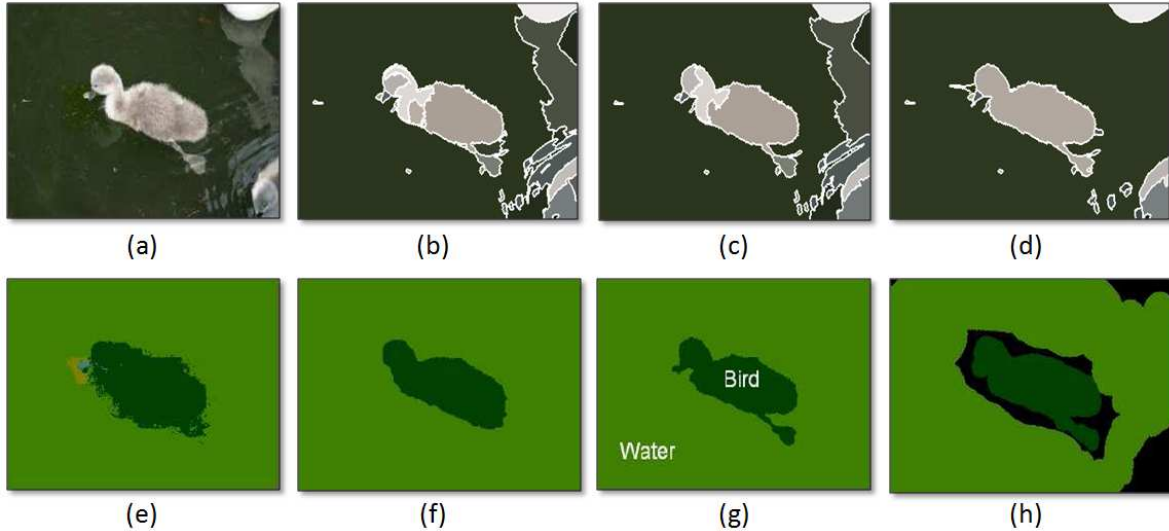


Fig. 12. *Segmenting objects with thin structures using multiple segmentations. (a) An images from the MSRC-21 dataset. (b), (c) and (d) Multiple segmentations of the image obtained by varying the parameters of the mean shift algorithm [10]. (e) Labelling from TextonBoost unary potentials (see section II). (f) Result of the pairwise CRF (see section II). (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.*

(d)). The solution is thus assigned a low cost by the higher order label consistency potentials. Now consider the solution of the pairwise CRF (figure 12 (f)). This labelling is consistent with super-pixels in two segmentations (figure 12 (b) and (c)), but is inconsistent with regards to the segmentation shown in figure 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.* [16] 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.* [26]) 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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65



Fig. 13. Some qualitative results. Please view in colour. **First Row:** Original Image. **Second Row:** Unary likelihood labelling from Textonboost [16]. **Third Row:** Result obtained using a pairwise contrast preserving smoothness potential as described in [16]. **Fourth Row:** Result obtained using the P^n Potts model potential [3]. **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.

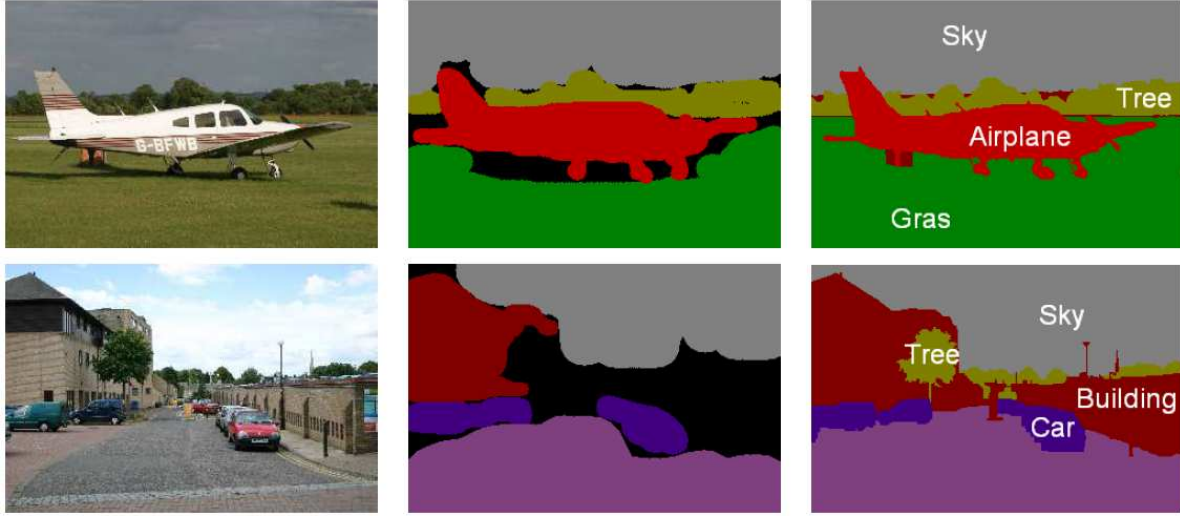


Fig. 14. Accurate hand labelled segmentations which were used as ground truth. The figure shows some images from the MSRC data set (column 1), the hand labelled segmentations that came with the data set (column 2), and the new segmentations hand labelled by us which were used as ground truth (column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

results.

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

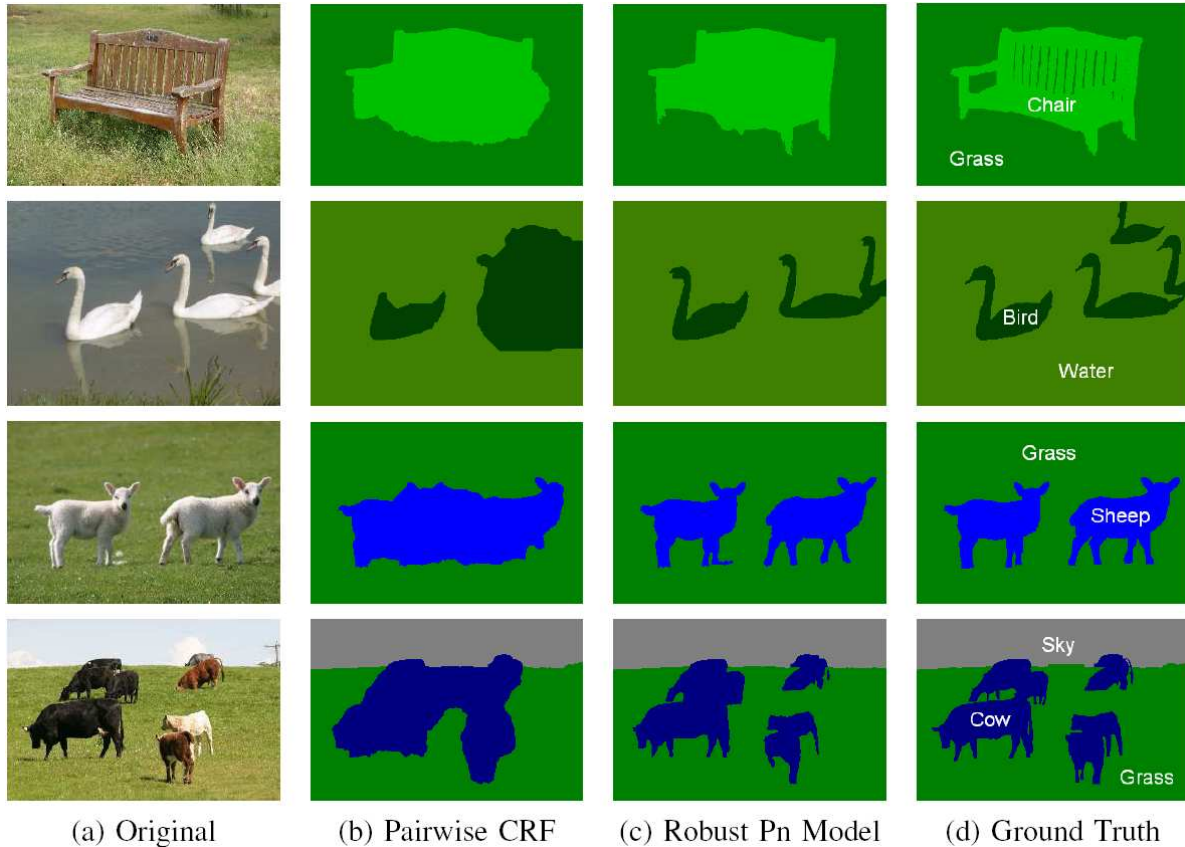


Fig. 15. *Qualitative results of our method. (a) Original Images. (b) Segmentation result obtained using the pairwise CRF (explained in section II). (c) Results obtained by incorporating the Robust P^n higher order potential (13) defined on segments. (d) Hand labelled result used as ground truth.*

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 percentage pixel-wise accuracy. This phenomenon is illustrated in figure 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 figure 17. The accuracy of different segmentation methods is plotted in the graph shown in figure 18.

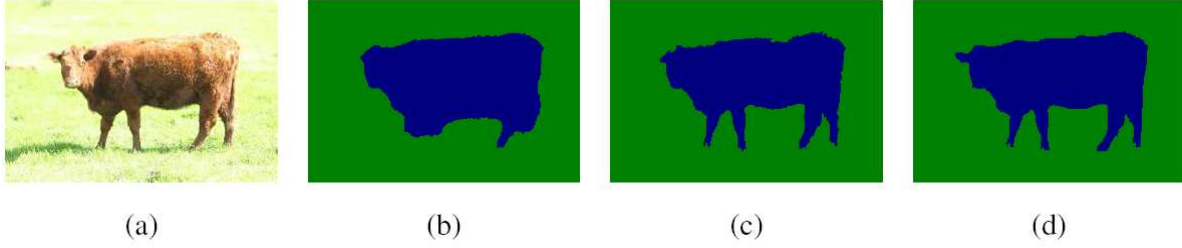


Fig. 16. *The relationship between qualitative and quantitative results. (a) Original Image. (b) Segmentation result obtained using the pairwise CRF (explained in section II). Overall pixelwise accuracy for the result is 95.8%. (c) Results obtained by incorporating the Robust P^n higher order potential (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.*

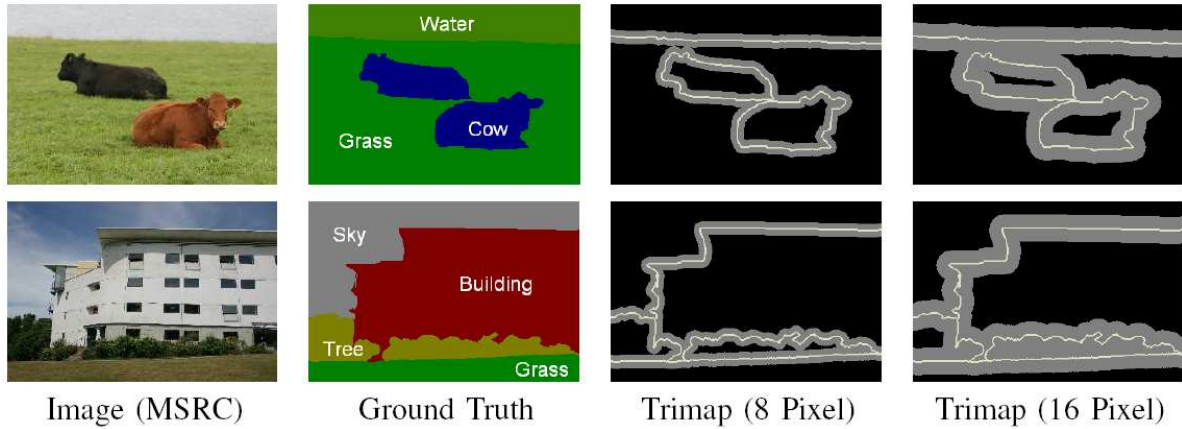


Fig. 17. *Boundary accuracy evaluation using trimap segmentations. The first column shows some images from the MSRC dataset [16]. The ground truth segmentations of these image are shown in column 2. Column 3 shows the trimaps used for measuring the pixel labelling accuracy. The evaluation region is coloured 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.*

VIII. 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

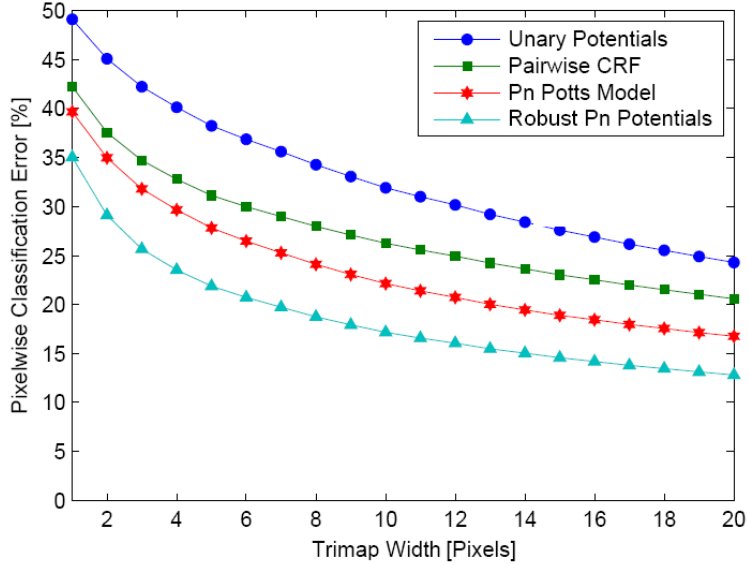


Fig. 18. Pixelwise classification error in our results. The graph shows how the overall pixelwise classification error varies as we increase the width of the evaluation region.

and swap move algorithms. Our methods for computing the optimal expansion and swap moves are extremely 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 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 [39]) for efficiently minimizing the class of higher order potentials proposed in this paper.

IX. 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.

REFERENCES

- [1] M. Kumar, and P. Torr, "Improved Moves for Truncated Convex Models," in *Proceedings of Advances in Neural Information Processing Systems*, 2008.
- [2] K. Alahari, P. Kohli, and P. Torr, "Reduce, Reuse and Recycle: Efficiently Solving Multi-Label MRFs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [3] P. Kohli, M. Kumar, and P. Torr, " P^3 and beyond: Solving energies with higher order cliques," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
- [4] S. Lauritzen, *Graphical Models*, Oxford University Press, 1996.
- [5] X. He, R. Zemel, and D. Ray, "Learning and incorporating top-down cues in image segmentation." in *European Conference on Computer Vision*, 2006, pp. 338–351.
- [6] D. Hoiem, A. Efros, and M. Hebert, "Automatic photo pop-up." *ACM Trans. Graph.*, vol. 24, no. 3, pp. 577–584, 2005.
- [7] A. Rabinovich, S. Belongie, T. Lange, and J. Buhmann, "Model order selection and cue combination for image segmentation." in *IEEE Conference on Computer Vision and Pattern Recognition (1)*, 2006, pp. 1130–1137.
- [8] D. Hoiem, A. Efros, and M. Hebert, "Geometric context from a single image." in *International Conference on Computer Vision*, 2005, pp. 654–661.
- [9] B. Russell, W. Freeman, A. Efros, J. Sivic, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections." in *IEEE Conference on Computer Vision and Pattern Recognition (2)*, 2006, pp. 1605–1614.
- [10] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.
- [11] E. Borenstein and J. Malik, "Shape guided object segmentation." in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 969–976.
- [12] X. He, R. Zemel, and M. Carreira-Perpiñán, "Multiscale conditional random fields for image labeling." in *IEEE Conference on Computer Vision and Pattern Recognition (2)*, 2004, pp. 695–702.
- [13] R. Huang, V. Pavlovic, and D. Metaxas, "A graphical model framework for coupling MRFs and deformable models," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. II, 2004, pp. 739–746.
- [14] M. Kumar, P. Torr, and A. Zisserman, "Obj cut." in *IEEE Conference on Computer Vision and Pattern Recognition (1)*, 2005, pp. 18–25.
- [15] A. Levin and Y. Weiss, "Learning to combine bottom-up and top-down segmentation." in *European Conference on Computer Vision*, 2006, pp. 581–594.
- [16] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "*TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation." in *European Conference on Computer Vision*, 2006, pp. 1–15.

- 1
2
3
4
5 [17] X. Lan, S. Roth, D. Huttenlocher, and M. Black, "Efficient belief propagation with learned higher-order markov random
6 fields." in *European Conference on Computer Vision*, 2006, pp. 269–282.
- 7 [18] R. Paget and I. Longstaff, "Texture synthesis via a noncausal nonparametric multiscale markov random field." *IEEE*
8 *Transactions on Image Processing*, vol. 7, no. 6, pp. 925–931, 1998.
- 9 [19] S. Roth and M. Black, "Fields of experts: A framework for learning image priors." in *IEEE Conference on Computer*
10 *Vision and Pattern Recognition*, 2005, pp. 860–867.
- 11 [20] B. Potetz, "Efficient belief propagation for vision using linear constraint nodes," in *IEEE Conference on Computer Vision*
12 *and Pattern Recognition*, 2007.
- 13 [21] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts." *IEEE Trans. Pattern Anal.*
14 *Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, 2001.
- 15 [22] P. Kohli, L. Ladicky, and P. Torr, "Robust higher order potentials for enforcing label consistency," in *CVPR*, 2008.
- 16 [23] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labelling
17 sequence data," in *International Conference on Machine Learning*, 2001, pp. 282–289.
- 18 [24] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive image segmentation using an adaptive GMMRF model,"
19 in *European Conference on Computer Vision*, 2004, pp. I: 428–441.
- 20 [25] M. Bray, P. Kohli, and P. Torr, "Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic
21 graph-cuts." in *European Conference on Computer Vision*, 2006, pp. 642–655.
- 22 [26] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: interactive foreground extraction using iterated graph cuts," in *ACM*
23 *Trans. Graph.*, 2004, pp. 309–314.
- 24 [27] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images,"
25 in *International Conference on Computer Vision*, 2001, pp. I: 105–112.
- 26 [28] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization." *IEEE Trans. Pattern Anal. Mach.*
27 *Intell.*, vol. 28, no. 10, pp. 1568–1583, 2006.
- 28 [29] M. Wainwright, T. Jaakkola, and A. Willsky, "Map estimation via agreement on trees: message-passing and linear
29 programming." *IEEE Transactions on Information Theory*, vol. 51, no. 11, pp. 3697–3717, 2005.
- 30 [30] X. Ren and J. Malik, "Learning a classification model for segmentation." in *International Conference on Computer Vision*,
31 2003, pp. 10–17.
- 32 [31] Z. Tu and S. Zhu, "Image segmentation by data-driven markov chain monte carlo." *IEEE Trans. Pattern Anal. Mach.*
33 *Intell.*, vol. 24, no. 5, pp. 657–673, 2002.
- 34 [32] E. Sharon, A. Brandt, and R. Basri, "Segmentation and boundary detection using multiscale intensity measurements." in
35 *IEEE Conference on Computer Vision and Pattern Recognition (1)*, 2001, pp. 469–476.
- 36 [33] J. Shi and J. Malik, "Normalized cuts and image segmentation." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8,
37 pp. 888–905, 2000.
- 38 [34] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation." *International Journal of Computer*
39 *Vision*, vol. 59, no. 2, pp. 167–181, 2004.
- 40 [35] J. Wang, P. Bhat, A. Colburn, M. Agrawala, and M. Cohen, "Interactive video cutout." *ACM Trans. Graph.*, vol. 24, no. 3,
41 pp. 585–594, 2005.
- 42 [36] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?." *IEEE Trans. Pattern Anal.*
43 *Mach. Intell.*, vol. 26, no. 2, pp. 147–159, 2004.
- 44 [37] J. Yedidia, W. Freeman, and Y. Weiss, "Generalized belief propagation." in *NIPS*, 2000, pp. 689–695.
- 45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

- 1
2
3
4
5 [38] N. Komodakis and G. Tziritas, “A new framework for approximate labeling via graph cuts.” in *International Conference on Computer Vision*, 2005, pp. 1018–1025.
6
7 [39] N. Komodakis, G. Tziritas, and N. Paragios, “Fast, approximately optimal solutions for single and dynamic MRFs,” in *CVPR*, 2007.
8
9 [40] O. Veksler, “Graph cut based optimization for MRFs with truncated convex priors,” in *CVPR*, 2007.
10
11 [41] V. Lempitsky, C. Rother, and A. Blake, “Logcut - efficient graph cut optimization for markov random fields,” in *ICCV*, 2007.
12
13 [42] S. Fujishige, *Submodular functions and optimization*. Amsterdam: North-Holland, 1991.
14
15 [43] L. Lovasz, “Submodular functions and convexity,” in *Mathematical Programming: The State of the Art*, 1983, pp. 235–257.
16
17 [44] E. Boros and P. Hammer, “Pseudo-boolean optimization.” *Discrete Applied Mathematics*, vol. 123, no. 1-3, pp. 155–225, 2002.
18
19 [45] J. Orlin, “A faster strongly polynomial time algorithm for submodular function minimization.” in *Proceedings of Integer Programming and Combinatorial Optimization*, 2007, pp. 240–251.
20
21 [46] D. Freedman and P. Drineas, “Energy minimization via graph cuts: Settling what is possible.” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 939–946.
22
23 [47] B. Flach, “Strukturelle bilderkennung,” Universit at Dresden, Tech. Rep., 2002.
24
25 [48] H. Ishikawa, “Exact optimization for markov random fields with convex priors,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, pp. 1333–1336, October 2003.
26
27 [49] D. Schlesinger and B. Flach, “Transforming an arbitrary minsum problem into a binary one,” Dresden University of Technology, Tech. Rep. TUD-FI06-01, April 2006.
28
29
30
31
32

33 APPENDIX A: SUBMODULAR FUNCTIONS

34
35 To provide a formal definition of a submodular function, we will first need to define the
36 concept of a projection of a function. A projection of a function $f : \mathcal{L}^n \rightarrow \mathbb{R}$ on s variables
37 is a function $f^p : \mathcal{L}^s \rightarrow \mathbb{R}$ which is obtained by fixing the values of $n - s$ arguments of $f(\cdot)$.
38 A function of one binary variable is always submodular. A function $f(x_1, x_2)$ of two binary
39 variables $x_1, x_2 \in \mathbb{B} = \{0, 1\}$ is submodular if and only if: $f(0, 0) + f(1, 1) \leq f(0, 1) + f(1, 0)$.
40 A function $f : \mathbb{B}^n \rightarrow R$ is submodular if and only if all its projections on 2 variables are
41 submodular [36], [44].
42
43
44
45
46
47
48

49 APPENDIX B: PROOFS

50
51 **Theorem 1** The higher order pseudo-boolean function:
52

$$53 \quad f(\mathbf{t}_c) = \min \left(\theta_0 + \sum_{i \in c} w_i^0 (1 - t_i), \theta_1 + \sum_{i \in c} w_i^1 t_i, \theta_{\max} \right) \quad (46)$$

54
55 can be transformed to the submodular quadratic pseudo-boolean function:
56

$$57 \quad f(\mathbf{t}_c) = \min_{m_0, m_1} \left(r_0 (1 - m_0) + m_0 \sum_{i \in c} w_i^0 (1 - t_i) + r_1 m_1 + (1 - m_1) \sum_{i \in c} w_i^1 t_i - K \right) \quad (47)$$

58
59
60
61
62
63
64
65

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 , θ_{\max} are function parameters that take values in \mathbb{R} .

Proof: We decompose the function (47) as:

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

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

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

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

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

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

Adding equations (50) and (51) and using the constraint 26, we get $\mathcal{F}^0(\mathbf{t}_{c_a}) + \mathcal{F}^1(\mathbf{t}_{c_a}) =$

$$\begin{cases} r_0 + \sum_{i \in c} w_i^1 t_i & \text{if } \sum_{i \in c} w_i^1 t_i \leq r_1 \\ r_1 + \sum_{i \in c} w_i^0(1 - t_i) & \text{if } \sum_{i \in c} w_i^0(1 - t_i) \leq r_0 \\ r_0 + r_1 & \text{otherwise.} \end{cases} \quad (52)$$

Substituting this in equation (48), we get

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

This equation can alternatively be written as:

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

■

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

$$\begin{aligned} \psi_c^m(\mathbf{t}_c) &= \min_{m_0, m_1} \left(r_0(1 - m_0) + \theta_d m_0 \sum_{i \in c_d} w_i(1 - t_i) \right. \\ &\quad \left. + r_1 m_1 + \theta_\alpha(1 - m_1) \sum_{i \in c} w_i t_i - \delta. \right) \end{aligned} \quad (55)$$

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

Proof: We decompose the move energy (55) as:

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

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

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

$$= \min_{m_0} (\gamma_{\max} - \gamma_d - R_d \theta_d)(1 - m_0)$$

$$+ \frac{\gamma_{\max} - \gamma_d}{Q_d} m_0 f_0^m(\mathbf{t}_{c_d}) \quad (\text{Recall } R_d = W(c - c_d))$$

$$= \min_{m_0} (\gamma_{\max} - \gamma_d)(1 - m_0)$$

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

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

Similarly,

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

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

$$= \min_{m_1} (\gamma_{\max} - \gamma_\alpha)m_1$$

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

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

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

Adding (59) and (64) and using the relations¹⁰

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

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

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

we get: $\mathcal{F}^0(\mathbf{t}_{c_a}) + \mathcal{F}^1(\mathbf{t}_{c_a}) =$

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

Substituting in (56) and simplifying we get $\psi_c^m(\mathbf{t}_c, \mathbf{t}_{c_d}) =:$

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

which is the same as (42). ■