

Move-based Algorithms for the Optimization of an Isotropic Gradient MRF Model

Behrooz Nasihatkon and Richard Hartley
Australian National University
National ICT Australia (NICTA)

Abstract—We consider an isotropic gradient model for the regularization terms in a multi-label MRF lattice. The isotropic gradient is modeled by considering 3-cliques in an 8-connected lattice. Of interest here are iterative move algorithms like alpha-expansion and alpha-beta swap, which try to minimize the energy function by solving a series of binary labeling problems. Such algorithms mainly differ in their update policy for the binary move at each iteration. Here, the aim is to study the submodularity of the binary move at each stage for a general update policy. We give the necessary and sufficient condition for the submodularity of a general update policy for the two major types of labels, namely *ordered labels* and *unordered labels*.

I. INTRODUCTION

In most of the applications of an MRF model, the regularization terms are defined as functions of labels at the horizontal and vertical edges (2-cliques) in a lattice. This approach suffers from not being isotropic: if the image is rotated by some angle we can get different results. In this paper, we try to mitigate this problem by considering functions on 3-cliques in an 8-connected lattice. Here, we deal with multi-label MRF models. To minimize the proposed energy function, one can use the so called *move-based* algorithms [1], in which the multi-label energy function is iteratively minimized by solving a sequence of binary optimization problems. Although they do not guarantee to find the global minimum to the multi-label problem, they perform quite well in practice.

Here, a general move algorithm is considered for which algorithms like alpha-expansion are special cases. Different move algorithms can be obtained from the general move algorithm by the choice of different update policies. The object of this paper is to completely specify the conditions for an update policy to give a submodular binary problem at each iteration of the move algorithm. Using the given conditions one can easily check the submodularity for different kinds of move functions and different types of labels. In this paper we only consider the theoretical aspects of the problem. The design and implementation of efficient algorithms based on the obtained results is left for further research.

As mentioned before, the 2-clique models are the most widely used regularizers in an MRF lattice. The *Potts model* is

one of the simplest models dealing with the multi-label case:

$$E(\mathbf{x}) = \sum_{(i,j) \in \mathcal{C}^2} 1(x_i \neq x_j),$$

where \mathcal{C}^2 is the set of edges (2-cliques) of the graph and $1(x_i \neq x_j)$ is equal to 1 if $x_i \neq x_j$ and is zero if $x_i = x_j$. The above model suits the case of unordered labels, where for a pair of labels it only matters whether or not they are equal. As a regularizer which can be applied to other types of labels as well, we can consider the more general model

$$E(\mathbf{x}) = \sum_{(i,j) \in \mathcal{C}^2} d(x_i, x_j), \quad (1)$$

where d can be any metric¹. A major example here is the case of *ordered labels*, that is when the label set is in the form of $\mathcal{L} = \{0, 1, \dots, M-1\}$ and the distance between two labels is defined as $d(x, y) = |x - y|$.

A natural regularization measure in a grid of labels is the magnitude of the gradient. In a continuous image domain, the integral of the magnitude of gradient over the domain is called *Total Variation*. Total Variation has interesting properties as a regularizer. Especially, it is known for its ability to handle discontinuities in an image in the sense that for a monotonic transition between two states it gives the same cost no matter whether the transition is sharp or gradual (see [2], [3]). Graphical Models have been suggested in the literature for the approximation and optimization of Total Variation (see [4], [5] for example). To see how the energy function (1) can be seen as an approximation of Total Variation, consider a node i in the lattice. Show its right (or left) neighbour by j and its upper (or lower) neighbour by k . Now consider the magnitude of gradient at node i approximated by

$$T_1(x_i, x_j, x_k) = d(x_i, x_j) + d(x_i, x_k).$$

Then, neglecting the nodes at the boundary, the energy function (1), up to a global scaling factor, can be seen as the sum of the gradients at different nodes. The above formula is not a proper choice as an isotropic approximator of the magnitude of the gradient. It gives more penalty to the diagonal variations compared to the vertical or horizontal ones. Therefore, by rotating the image we can get quite different results.

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

¹In fact, our results hold even if d is a semi-metric, having all properties of the metric, but not necessarily the triangular inequality.

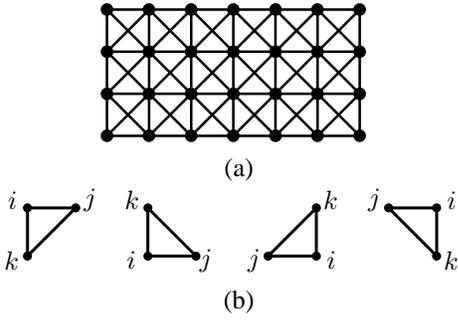


Fig. 1. (a) An 8-connected lattice, (b) possible shapes for the 3-clique (i, j, k) in an 8-connected lattice. Notice that the order of elements of (i, j, k) does matter: the second and third elements j and k are always respectively the vertical and horizontal neighbours of the first element i .

A better choice to model the magnitude of gradient can be

$$T(x_i, x_j, x_k) = \sqrt{d(x_i, x_j)^2 + d(x_i, x_k)^2} \quad (2)$$

where j is the left (or right) neighbour and k is the upper (or lower) neighbour of i . This model however is more complicated as the energy involves the summation of functions of cliques of size 3 rather than cliques of size 2. This results in an 8-connected network (Fig. 1).

When it comes to inference, it is important that the energy minimization problem can be solved efficiently. In the case of binary variables, that is $x_i \in \{0, 1\}$, the concept of submodularity plays a role. A pseudo-Boolean function $f: \{0, 1\} \times \{0, 1\} \rightarrow \mathbb{R}$ of two variables is *submodular* if

$$f(0, 1) + f(1, 0) \geq f(0, 0) + f(1, 1). \quad (3)$$

A pseudo-Boolean function of two variables can be written as a sum of quadratic and linear terms. From (3) it is easy to see that the linear terms do not play a role in submodularity. A pseudo-Boolean function of n variables is called submodular if any restriction to any pair of variables is submodular. In [6], it has been proved that an energy function consisting of a sum of two and three variable potentials is graph representable (and hence can be optimized using the polynomial time graph cut algorithms) if and only if it is submodular.

It is easy to check that on a binary domain, where $u_i, u_j, u_k \in \{0, 1\}$, the potential function $T(u_i, u_j, u_k) = \sqrt{(u_i - u_j)^2 + (u_i - u_k)^2}$ is submodular. This follows from the submodularity of the restrictions to any two variables. As the function is symmetric in its second and third arguments (u_j and u_k) we just need to check for the following restrictions

$$\begin{aligned} T_1(u_j, u_k) &= T(0, u_j, u_k) = \sqrt{u_j + u_k}, \\ T_2(u_j, u_k) &= T(1, u_j, u_k) = \sqrt{2 - (u_j + u_k)}, \\ T_3(u_i, u_j) &= T(u_i, u_j, 0) = \sqrt{2u_i - 2u_i u_j + u_j}, \\ T_4(u_i, u_j) &= T(u_i, u_j, 1) = \sqrt{1 - 2u_i u_j + u_j}, \end{aligned}$$

all of which can be easily checked for submodularity, that is $T_i(1, 0) + T_i(0, 1) \geq T_i(1, 1) + T_i(0, 0)$.

For the multi-label case, obtaining an optimal solution is not generally tractable. However, iterative move-based algorithms

like alpha-expansion can give good results. In such algorithms, at each iteration a binary problem is solved. A sequence of binary problems lowering the energy at every stage gives a locally optimal solution. The object of this paper is to study the submodularity of the binary sub-problems at each iteration.

In this paper, first a general move algorithm for the optimization of multi-label energy functions is described for the isotropic gradient model introduced above. Specific strategies like alpha-expansion can be obtained from this general algorithm by choosing different label update policies. Given this general algorithm, we study the submodularity of the energy function for the two major cases of ordered and unordered labels. In each case, a necessary and sufficient condition is given for a general update policy to result in submodular binary move sub-problems.

II. MOTIVATION

In this section we study the reason why our suggested isotropic gradient model (2) works better than the 2-clique model (1). Here, we consider two different cases separately: the *ordered labels* and the *unordered labels*.

A. Ordered Labels

In this case the label set \mathcal{L} is in the form of $\{0, 1, \dots, M-1\}$ and the distance between two labels x and y is simply defined as $d(x, y) = |x - y|$. Usually, this case is used as an approximation to the continuous labels, especially when M is large enough. Therefore, to verify our isotropic model for this case, we simply consider the case of continuous labels.

Consider a single square cell of the lattice, as shown in Fig. 2. We can approximate the gradient using (1) or (2) in the middle of this square. Now, assume a uniform gradient of unit magnitude on a neighbourhood around this square, whose direction makes an angle of θ with the horizontal axis. Assuming a label value of 0 for the top left node, the label values for the other nodes will be as shown in Fig. 2. According to these values, for the 2-clique model defined in (1) the *normalized cost* will be:

$$|\cos \theta| + |\sin \theta|,$$

where normalized means that the cost is scaled such that it has a value of 1 at $\theta = 0$. The normalized cost using the suggested isotropic model (2) will be

$$4\sqrt{|\cos \theta|^2 + |\sin \theta|^2}/4 = 1,$$

which does not depend on θ . This shows that in the case of continuous labels, the suggested isotropic model gives an exact approximation of the magnitude of the gradient. Notice that the 2-clique model (1), gives a value of $\sqrt{2}$ for a diagonal edge with $\theta = \pi/4$. Therefore, it can deviate from the true gradient by a relative factor of $\sqrt{2}$.

B. Unordered Labels

In this case, for any pair of labels we can just say whether or not they are equal. The distance between the two labels x and y is simply defined as $d(x, y) = 1_{(x \neq y)}$ which is equal

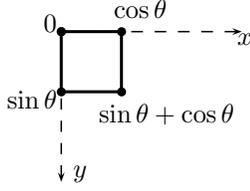


Fig. 2. Suppose each side of the above square cell has a unit length. We have applied a gradient of magnitude 1 and with an angle of θ with the horizontal axis. We have assumed a label value of 0 for the top left node and calculated the label of the other nodes as a function of θ .

to one if $x \neq y$ and equal to zero if $x = y$. As a special case, we can consider the basic binary labels where $\mathcal{L} = \{0, 1\}$.

In this case, the verification cannot be done by approximating using the continuous labels. Instead, here we consider the edge cost per length. We look at a small section of an edge with an angle of $\theta = \arctan \frac{1}{m}$ for some positive integer m . If the grid is fine enough, the above edge appears as a set of stairs where every single stair goes m units to the right and then one unit down as shown in Fig. 3. This only covers an angle range of $0 < \theta \leq \pi/4$, but other ranges of angles can be modeled accordingly. We compare the cost of the edge per length unit for the 2-clique model (1) and the suggested isotropic gradient model (2).

In Fig. 3 an edge is made by the transition from the white labeled area to the grey labeled area. We know that for each step of the edge we have m vertical active edges, namely $(i_1, j_1), \dots, (i_m, j_m)$ and one horizontal active edge (j_0, j_1) . By active edge we mean an edge with a nonzero cost. Notice that the horizontal edge (i_m, i_{m+1}) is the same as (j_0, j_1) for the next stair and hence does not count. Therefore, the cost for each stair using the 2-clique model (1) would be $m + 1$. As the length of each stair is $\sqrt{m^2 + 1}$, the edge cost per length unit is

$$\frac{m + 1}{\sqrt{m^2 + 1}} = \frac{\cot \theta + 1}{\sqrt{\cot^2 \theta + 1}} = \sin \theta + \cos \theta$$

where $0 \leq \theta \leq \pi/4$.

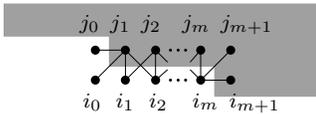


Fig. 3. An edge in the grid made by a transition from a white labeled region to a grey labeled region. The edge has an angle $\theta = \arctan \frac{1}{m}$, that is for every m unit going right it goes one unit down. Only the active edges (edges with a nonzero cost) have been depicted in the picture.

Now, consider the suggested isotropic model (2). In this case, for each step, there are 2 active 3-cliques (j_1, j_0, i_1) and (i_m, i_{m+1}, j_m) with a cost of $\sqrt{1+1} = \sqrt{2}$ and $4m$ active 3-cliques with a cost of $\sqrt{1+0} = 1$. As the length of each stair is $\sqrt{m^2 + 1}$, the *normalized* cost of the edge per length unit would be

$$\frac{1}{4} \cdot \frac{4m + 2\sqrt{2}}{\sqrt{m^2 + 1}} = \frac{1}{4} \cdot \frac{4 \cot \theta + 2\sqrt{2}}{\sqrt{\cot^2 \theta + 1}} = \cos \theta + \frac{\sqrt{2}}{2} \sin \theta$$

where $0 \leq \theta \leq \pi/4$. A normalized cost means that the cost is scaled such that it is equal to 1 for a horizontal edge ($\theta = 0$).

A comparison of edge costs as a function of θ for the two models (1) and (2) is shown in Fig. 4. The 2-clique model (1) has a maximum of $\sqrt{2} \approx 1.4$ while the suggested model (2) reaches $\sqrt{6}/2 \approx 1.2$ at its maximum. Therefore, in this case the suggested isotropic model does not exactly give a uniform cost function for all edge orientations, however, it still works better than the 2-clique model.

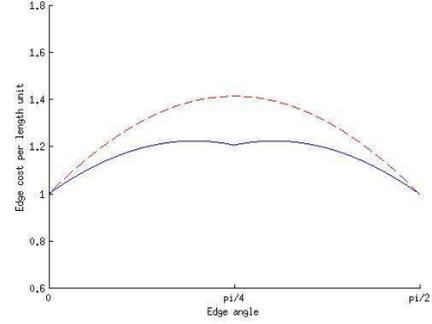


Fig. 4. A comparison of the edge costs between the 2-clique model (dashed line) and the suggested isotropic model (solid line) for $0 \leq \theta \leq \pi/2$. The suggested model here is not quite rotation-invariant, however, it still performs better than the 2-clique model.

C. Conclusion

As a conclusion, we can say that the suggested isotropic model (2) well suits the case of *ordered labels*. Especially when the number of labels M is large, this model has a considerable advantage over the 2-clique model (1). As for the case of *unordered labels* or binary labels, the suggested model acts fairly better than the 2-clique model, however, it cannot be considered rotation invariant.

III. THE GENERAL MOVE ALGORITHM

Consider a general energy function $E(\mathbf{x})$ of the labels $\mathbf{x} = [x_1, x_2, \dots, x_n]$, with $x_i \in \mathcal{L}$. In a multi-label scenario the size of the set \mathcal{L} is normally bigger than two. To minimize this energy function, we focus on move algorithms in which at every iteration a binary problem is solved, lowering the energy throughout iterations. Here, after describing two of the known move algorithms, we present a typical move algorithm in its general form.

a) *Alpha-expansion*: Perhaps the most popular example of this type is the alpha-expansion algorithm. In this algorithm, a binary variable u_i is assigned to each node i . A parameter α iterates through the different values in the label set \mathcal{L} . At each iteration the variable x_i is updated according to

$$x'_i = l_\alpha^{u_i}(x_i) = x_i \bar{u}_i + \alpha u_i, \quad (4)$$

where $x'_i = l_\alpha^{u_i}(x_i)$ shows the updated label of node i . The above means that the variable x_i remains unchanged if $u_i=0$ and is changed to α if $u_i=1$. The vector of labels $\mathbf{x} \in \mathcal{L}^n$ at the

next iteration is therefore a function of $\mathbf{u} = [u_1, u_2, \dots, u_n]$. This updated vector of labels is shown here by

$$\mathbf{x}' = \mathbf{l}_\alpha^{\mathbf{u}}(\mathbf{x}) = [l_\alpha^{u_1}(x_1), l_\alpha^{u_2}(x_2), \dots, l_\alpha^{u_n}(x_n)].$$

A general energy function $E(\mathbf{x})$ evaluated at the new updated labels gives $E(\mathbf{l}_\alpha^{\mathbf{u}}(\mathbf{x}))$ which is a function of \mathbf{u} . Therefore, at each iteration we choose the \mathbf{u} that minimizes $E(\mathbf{l}_\alpha^{\mathbf{u}}(\mathbf{x}))$ and then update the label values as $\mathbf{l}_\alpha^{\mathbf{u}}(\mathbf{x}) \in \mathcal{L}$. Fig. 5 shows an outline of the algorithm.

```

procedure ALPHA-EXPANSION( $\mathbf{x}, \mathcal{L}$ )
  repeat
    for each  $\alpha \in \mathcal{L}$  do
       $\mathbf{u}^* \leftarrow \operatorname{argmin}_{\mathbf{u}} E(\mathbf{l}_\alpha^{\mathbf{u}}(\mathbf{x}))$ 
       $\mathbf{x} \leftarrow \mathbf{l}_\alpha^{\mathbf{u}^*}(\mathbf{x})$ 
    end for
  until convergence
  return  $\mathbf{x}$ 
end procedure

```

Fig. 5. The alpha-expansion algorithm. The inputs to the algorithm are the initial label values \mathbf{x} and the set of labels \mathcal{L} .

b) *Alpha-beta swap*: Another algorithm of this family is called alpha-beta swap. In this algorithm there are two parameters α and β iterating through all possible distinct pairs of label values. At every iteration only the nodes with labels α or β are updated. The update equation is

$$x'_i = l_{\alpha,\beta}^{u_i}(x_i) = \begin{cases} \alpha \bar{u}_i + \beta u_i & \text{if } x_i \in \{\alpha, \beta\} \\ x_i & \text{otherwise} \end{cases} \quad (5)$$

In other words, the nodes whose label value is not α or β stay unchanged and other nodes can switch to α or β .

Our results in this research does not depend on a specific update criterion. Instead, we consider a general move algorithm. Here, there is a parameter vector θ that varies over iterations. At each iteration, given the binary variable u_i the label of each node i is updated to either of the two possible states $l_\theta^0(x_i)$ or $l_\theta^1(x_i)$:

$$x'_i = l_\theta^0(x_i) \bar{u}_i + l_\theta^1(x_i) u_i = l_\theta^{u_i}(x_i) \quad (6)$$

where x_i shows the current label of the node i . Therefore, different choices of l_θ^0 and l_θ^1 give different update criteria. Here, the pair of functions (l_θ^0, l_θ^1) is called the *update policy* for the parameter θ . Notice that using this update policy, the updated value for a node i only depends on θ , the current label value x_i and the binary variable u_i . It does not depend on the location of node i or the label values of its neighbours. A sketch of the general move algorithm can be seen in Fig. 6.

It is clear that an update policy must possess certain properties to act reasonably. For example, as a trivial case, if for all label values $x \in \mathcal{L}$ we have $l_\theta^0(x) = l_\theta^1(x)$, then the policy is not sensible as it cannot make any updates. Other properties can be mentioned for a sensible policy. Here, we just mention one property, namely the *state preservation property*, which means that each label must be able to preserve its current state after each iteration, in other words, for each

```

procedure GENERAL-MOVE-ALGORITHM( $\mathbf{x}, \mathcal{P}$ )
  repeat
    for each  $\theta \in \mathcal{P}$  do
       $\mathbf{u}^* \leftarrow \operatorname{argmin}_{\mathbf{u}} E(\mathbf{l}_\theta^{\mathbf{u}}(\mathbf{x}))$ 
       $\mathbf{x} \leftarrow \mathbf{l}_\theta^{\mathbf{u}^*}(\mathbf{x})$ 
    end for
  until convergence
  return  $\mathbf{x}$ 
end procedure

```

Fig. 6. The general move algorithm for multi-label energy minimization. The inputs are the initial labels \mathbf{x} and the set of parameter values \mathcal{P} .

label value $x \in \mathcal{L}$ and each θ we must either have $l_\theta^0(x) = x$ or $l_\theta^1(x) = x$. It can be easily seen that the alpha-expansion and alpha-beta swap algorithms possess this property. Notice that in the state preservation property, what value of $u \in \{0, 1\}$ gives $l_\theta^u(x) = x$ depends on x and θ . The important fact about the state preservation property is that for an update policy with this property, for every θ , $l_\theta^u(x)$ is *surjective* as a function of $u \in \{0, 1\}$ and $x \in \mathcal{L}$. Throughout the paper the state preservation property (and hence the surjectivity) is presumed for all update policies.

As mentioned before, in this paper we are concerned with the binary move problem at each iteration of the general move algorithm sketched in Fig. 6. We are looking for the update policies for which the energy function $E(\mathbf{l}_\theta^{\mathbf{u}}(\mathbf{x}))$ is submodular as a function of the binary labels \mathbf{u} , for which case there exist polynomial time algorithms to solve the optimization problem.

IV. SUBMODULARITY IN THE ISOTROPIC GRADIENT MODEL

As stated earlier, we deal with energy functions of the form

$$E(\mathbf{x}) = \sum_{(i,j,k) \in \mathcal{C}^3} T(x_i, x_j, x_k) + L(\mathbf{x}), \quad (7)$$

for an 8-connected lattice where \mathcal{C}^3 denotes the set of all 3-cliques, $L(\mathbf{x})$ shows the linear terms and $T(x_i, x_j, x_k) = \sqrt{d(x_i, x_j)^2 + d(x_i, x_k)^2}$, as defined in (2). The variables x_i, x_j and x_k take values in the set of labels \mathcal{L} whose size is usually bigger than two. Here, a 3-clique is shown by a 3-tuple (i, j, k) whose second element j always shows a horizontal (left or right) neighbour of the first element i and the third element k is a vertical neighbour of i , as shown in Fig. 1(b).

Obviously, we require the binary optimization problem at every iteration of the general move algorithm (Fig. 6) to be solved efficiently. If we are after the globally optimal solution for the binary problem, this means that $E(\mathbf{l}_\theta^{\mathbf{u}}(\mathbf{x}))$ in the general move algorithm must be *submodular* as a function of the binary labels \mathbf{u} . Here, we study the submodularity of this function and see by the choice of what update policy (functions l_θ^0 and l_θ^1) the submodularity holds. This is equivalent to studying submodularity for every restriction of the energy function to two variables. Here, for simplicity, we drop the parameter θ in all functions as we are concentrating on a

single iteration. Therefore, $l_{\theta}^u(x_i)$ and $\mathbf{I}_{\theta}^u(\mathbf{x})$ become $l^u(x_i)$ and $\mathbf{I}^u(\mathbf{x})$. Using (6) and (7) we get

$$E'_{\mathbf{x}}(\mathbf{u}) = E(\mathbf{I}^u(\mathbf{x})) = \sum_{(i,j,k) \in \mathcal{C}^3} T'_{x_i, x_j, x_k}(u_i, u_j, u_k) + L'_{\mathbf{x}}(\mathbf{u}), \quad (8)$$

where $L'_{\mathbf{x}}(\mathbf{u})$ is a linear function of \mathbf{u} and hence does not matter in the study of submodularity, and the term $T'_{x_i, x_j, x_k}(u_i, u_j, u_k)$ is defined as

$$T'_{x_i, x_j, x_k}(u_i, u_j, u_k) = T(l^{u_i}(x_i), l^{u_j}(x_j), l^{u_k}(x_k)). \quad (9)$$

where $l^{u_i}(x_i) = \bar{u}_i l^0(x_i) + u_i l^1(x_i)$, as defined in (6). We proceed by the study of submodularity of the potential function T'_{x_i, x_j, x_k} on a single 3-clique. Afterwards, we show that the submodularity of T'_{x_i, x_j, x_k} is necessary and sufficient for the submodularity of $E'_{\mathbf{x}}$ in (8).

A. Submodularity on a Single 3-Clique

Our main result in this section is that the submodularity of T'_{x_i, x_j, x_k} defined in (9) as a function of \mathbf{u} is reduced to the submodularity of its restriction to the diagonally neighbouring variables. For the sake of compactness, from here on, for every label $x \in \mathcal{L}$ we may use x^0 and x^1 to represent $l^0(x)$ and $l^1(x)$ respectively. The main theorem is

Theorem 1. *With the potential function T'_{x_i, x_j, x_k} defined in (9) and assuming that the update policy has the state preservation property defined in sec. III, the following are equivalent*

- (i) *The potential function T'_{x_i, x_j, x_k} defined in (9) is submodular for all values of x_i, x_j and x_k .*
- (ii) *any restriction of $T'_{x_i, x_j, x_k}(u_i, u_j, u_k)$ to the diagonally neighbouring variables u_j and u_k is submodular for all values of x_i, x_j and x_k .*
- (iii) *For any three labels $x, y, z \in \mathcal{L}$ we have*

$$(d(x, y^1) - d(x, y^0)) (d(x, z^1) - d(x, z^0)) \geq 0, \quad (10)$$

where, as mentioned before, x^u is a compact form for $l^u(x)$.

Proof: To prove the theorem, first we show that (i) implies (ii), then we show that (ii) and (iii) are equivalent, then we show that (ii) and (iii) together imply (i).

The first part of the proof, that is (i) \Rightarrow (ii), easily follows from our definition of the submodularity for functions of more than two variables.

To show that (i) \Leftrightarrow (ii), assume the potential function T'_{x_i, x_j, x_k} defined in (9) is submodular for any values of x_i, x_j and x_k in \mathcal{L} . Then any restriction of $T'_{x_i, x_j, x_k}(u_i, u_j, u_k)$ to the diagonally neighbouring variables u_j and u_k is submodular, that is to say the relation

$$T(l^{u_i}(x_i), l^0(x_j), l^1(x_k)) + T(l^{u_i}(x_i), l^1(x_j), l^0(x_k)) \geq T(l^{u_i}(x_i), l^0(x_j), l^0(x_k)) + T(l^{u_i}(x_i), l^1(x_j), l^1(x_k)) \quad (11)$$

must hold for any values of x_i, x_j and x_k and either values of u_i . Now, notice that x_i, x_j and x_k can get any values in \mathcal{L} and because of the surjectivity of $l^u(x)$, coming from the

state preservation property of the update policy, $l^{u_i}(x_i)$ can also get any value in \mathcal{L} . Hence, (11) is equivalent to

$$T(x, y^0, z^1) + T(x, y^1, z^0) \geq T(x, y^0, z^0) + T(x, y^1, z^1),$$

holding for any $x, y, z \in \mathcal{L}$, with y^0 and y^1 being compact ways to show $l^0(y)$ and $l^1(y)$ for any $y \in \mathcal{L}$. The above is equivalent to

$$\sqrt{d(x, y^0)^2 + d(x, z^1)^2} + \sqrt{d(x, y^1)^2 + d(x, z^0)^2} \geq \sqrt{d(x, y^0)^2 + d(x, z^0)^2} + \sqrt{d(x, y^1)^2 + d(x, z^1)^2}.$$

By squaring both sides of the above, canceling equal terms from both sides and squaring again we get

$$(d(x, y^1)^2 - d(x, y^0)^2) (d(x, z^1)^2 - d(x, z^0)^2) \geq 0,$$

By factoring out the terms $(d(x, y^1) + d(x, y^0))$ and $(d(x, z^1) + d(x, z^0))$ from the left hand side of the above we get (10)². From the above discussion it follows that (ii) \Rightarrow (iii). But, as the steps taken from (ii) to (iii) are reversible, we can say (ii) \Leftrightarrow (iii).

The last step is to show that (ii) and (iii) together imply (i). Assume that (ii) and (iii) hold. We have to show that any restriction of $T'_{x_i, x_j, x_k}(u_i, u_j, u_k)$ to horizontally, vertically and diagonally neighbouring variables is submodular. From (ii) we know that the restriction of $T'_{x_i, x_j, x_k}(u_i, u_j, u_k)$ to the diagonally neighbouring variables u_j and u_k is submodular. As the proofs for horizontal and vertical neighbours are similar, we just need to prove the submodularity of the restriction of $T'_{x_i, x_j, x_k}(u_i, u_j, u_k)$ to the horizontally neighbouring variables u_i and u_j . This means that we have to prove that

$$T(l^0(x_i), l^1(x_j), l^{u_k}(x_k)) + T(l^1(x_i), l^0(x_j), l^{u_k}(x_k)) \geq T(l^0(x_i), l^0(x_j), l^{u_k}(x_k)) + T(l^0(x_i), l^0(x_j), l^{u_k}(x_k)) \quad (12)$$

holds for any values of x_i, x_j, x_k and u_k . As x_i, x_j and x_k can get any values in \mathcal{L} and as $l^{u_k}(x_k)$ can also have any value in \mathcal{L} (as a result of the state preservation property of the update policy), (12) is equivalent to

$$T(y^0, z^1, t) + T(y^1, z^0, t) \geq T(y^0, z^0, t) + T(y^0, z^0, t),$$

holding for any $y, z, t \in \mathcal{L}$. Therefore, all we have to prove is

$$\sqrt{d(y^0, z^1)^2 + d(y^0, t)^2} + \sqrt{d(y^1, z^0)^2 + d(y^1, t)^2} \geq \sqrt{d(y^0, z^0)^2 + d(y^0, t)^2} + \sqrt{d(y^1, z^1)^2 + d(y^1, t)^2}. \quad (13)$$

Now, it is obvious that if $z^0 = z^1$ the above holds as an equality relation. If $z^0 \neq z^1$, as (iii) holds, by setting y and z in (10) equal to y and z in (13) and setting x in (10) equal to z^0 we get

$$(d(z^0, y^1) - d(z^0, y^0)) (d(z^0, z^1) - d(z^0, z^0)) \leq 0.$$

As d is a metric and $z^0 \neq z^1$ we can say that $d(z^0, z^0) = 0$ and $d(z^0, z^1) > 0$. The above relation then gives

$$d(z^0, y^1) \geq d(z^0, y^0). \quad (14)$$

²Notice that as $d(x, y) \geq 0$, even when $(d(x, y^1) + d(x, y^0))$ is equal to zero we can factor it out, as in this case $(d(x, y^1) - d(x, y^0))$ would be also equal to zero.

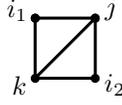


Fig. 7. Two cliques sharing the diagonal neighbours j and k .

Similarly, by setting $x = z^1$ in (10) we get

$$d(z^1, y^0) \geq d(z^1, y^1). \quad (15)$$

The relation (13) follows from (14) and (15). ■

B. Submodularity of the Energy Function

Now, we turn to finding the submodularity condition for the energy function $E'_x(\mathbf{u})$ defined in (8). It turns out that with the potential functions defined in (2), the submodularity of $E'_x(\mathbf{u})$ reduces to the submodularity of a single 3-clique potential $T'_{x_i, x_j, x_k}(u_i, u_j, u_k)$, as stated in the next proposition.

Proposition 1. *Assuming the state preservation property defined in sec. III for an update policy, the energy function $E'_x(\mathbf{u})$ defined in (8) is submodular for any value of $\mathbf{x} \in \mathcal{L}^n$ if and only if the potential function T'_{x_i, x_j, x_k} defined in (9) is submodular for any values of x_i, x_j and $x_k \in \mathcal{L}$.*

Proof: The backward direction of the proof is immediate because if the 3-clique potential $T'_{x_i, x_j, x_k}(u_i, u_j, u_k)$ is submodular for any values of x_i, x_j and x_k , so is $E'_x(\mathbf{u})$ as a sum of 3-clique potential functions plus some linear terms.

To prove the forward direction, assume that $E'_x(\mathbf{u})$ is submodular for any value of the vector of labels \mathbf{x} . Then its restriction to any diagonally neighbouring variables must be submodular. Suppose $E'_x(\mathbf{u})$ defined in (8) is restricted to u_j and u_k where j and k are diagonal neighbours. As there are two cliques sharing the nodes j and k (Fig. 7), call them (i_1, j, k) and (i_2, k, j) , the restriction will be

$$\begin{aligned} E'_{j,k}(u_j, u_k) = & T(x'_{i_1}(u_{i_1}), x'_j(u_j), x'_k(u_k)) \\ & + T(x'_{i_2}(u_{i_2}), x'_k(u_k), x'_j(u_j)) \\ & + L''(u_j) + L''(u_k) \end{aligned} \quad (16)$$

where $L''(u_j)$ and $L''(u_k)$ are linear terms and do not play a role in submodularity. From the submodularity of $E'_x(\mathbf{u})$, we know that the restriction $E'_{j,k}(u_j, u_k)$ is submodular as a function of u_j and u_k for any values given to $x_{i_1}, x_{i_2}, u_{i_1}$ and u_{i_2} in (16). Therefore, it is submodular for the cases where $x_{i_1} = x_{i_2}$ and $u_{i_1} = u_{i_2}$. By replacing x_{i_2} by x_{i_1} and u_{i_2} by u_{i_1} in (16) and considering the fact that T , defined in (2), is symmetric in its last two arguments, we can conclude that

$$2T(x'_{i_1}(u_{i_1}), x'_j(u_j), x'_k(u_k)) + L''(u_j) + L''(u_k)$$

is submodular. The above being submodular is equivalent to the submodularity of $T(x'_{i_1}(u_{i_1}), x'_j(u_j), x'_k(u_k))$. As x_{i_1}, x_j, x_k and u_{i_1} are considered arbitrary, this means that the restriction of the function $T(x_i(u_i), x_j(u_j), x_k(u_k))$ to the variables u_j and u_k is submodular for all possible values of x_{i_1}, x_j, x_k and either values of u_i . According to Theorem 1

this is equivalent to the submodularity of $T'_{x_i, x_j, x_k}(u_i, u_j, u_k)$ defined in (9) for every $x_i, x_j, x_k \in \mathcal{L}$. ■

Considering the above proposition along with Theorem 1, we can say that a necessary and sufficient condition for the energy function $E'_x(\mathbf{u})$ to be submodular is the condition (iii) mentioned in Theorem 1, that is the relation (10) must hold for every values of x, y and z in \mathcal{L} . This condition means that, for any values of x, y and z , the real numbers $d(x, y^1) - d(x, y^0)$ and $d(x, z^1) - d(x, z^0)$ must have the same sign (or otherwise either of them must be zero).

Before we go on with checking the submodularity for different types of labels, we mention a simple yet useful lemma. Define the set of *active* labels as

$$\mathcal{A} = \{x \in \mathcal{L} \mid l^0(x) \neq l^1(x)\}. \quad (17)$$

Because of the state preservation property, we know that if $x \notin \mathcal{A}$ then we have $l^0(x) = l^1(x) = x$. Therefore, the active labels are those that have the potential to change. It is obvious that for a sensible update \mathcal{A} must be nonempty.

As an example, in the alpha-beta swap algorithm, at each iteration we have $\mathcal{A} = \{\alpha, \beta\}$. Variables with the labels other than α and β cannot change. For the alpha-expansion algorithm this set is $\mathcal{A} = \mathcal{L} \setminus \{\alpha\}$, as all the variables can change except those with label α . Intuitively, it seems that the update policies with a smaller set of active labels are more likely to be submodular, as only a few labels have the potential to change. On the other hand, update policies with a larger set of active labels are supposed to converge after fewer iterations and give better results as more labels are involved in the optimization of the binary move sub-problem, for which the global optimum is found provided that it is submodular.

To see how the submodularity relates to the active sets, observe that the submodularity condition (10) holds if either y or z is not an active set, that is $y^0 = y^1$ or $z^0 = z^1$. This means that to study the submodularity we only need to check for the cases where $y, z \in \mathcal{A}$. We state this as a lemma whose proof easily follows from the discussion above:

Lemma 1. *The energy function E'_x defined in (8) is submodular for any $\mathbf{x} \in \mathcal{L}^n$ if and only if for any $y, z \in \mathcal{A}$ and any $x \in \mathcal{L}$ relation (10) holds.*

The above lemma tells us that we only need to check (10) for the case of $y, z \in \mathcal{A}$ rather than for all $y, z \in \mathcal{L}$, however, we still have to check for all values of $x \in \mathcal{L}$. Now, we investigate the submodularity for two cases of *ordered* and *unordered* labels, using the submodularity condition (10).

1) *Ordered Labels:* In this case labels are in the form of $\mathcal{L} = \{0, 1, \dots, M-1\}$ and the distance function is simply the Euclidean distance: $d(x, y) = |x - y|$. The submodularity condition (10) will then become

$$(|x - y^1| - |x - y^0|) (|x - z^1| - |x - z^0|) \geq 0. \quad (18)$$

According to Lemma 1, we just need to check (18) for the cases where $y, z \in \mathcal{A}$. Here, (18) says that for any label $x \in \mathcal{L}$, if there exists a label $y \in \mathcal{A}$ such that x is closer to $y^0 = l^0(y)$ than $y^1 = l^1(y)$, then for any other label $z \in \mathcal{A}$, x must be

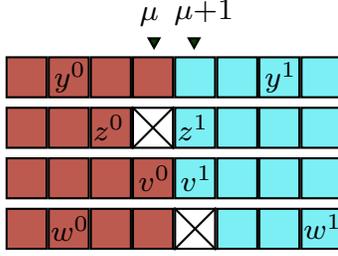


Fig. 8. An example of a mirrored update policy. The set of active labels is $\mathcal{A} = \{y, z, v, w\}$. Notice that for every $t \in \mathcal{A}$ we have $t^0 < t^1$ and $(t^0 + t^1)/2 \in \{\mu, \mu + \frac{1}{2}, \mu + 1\}$. For every $t \in \{y, z, v, w\}$ there is a row of blocks in the image in which the red (dark grey) blocks show the labels x such that $|x - t^0| < |x - t^1|$ and the blue (light grey) blocks are those labels x such that $|x - t^0| > |x - t^1|$. Essentially, the submodularity condition (18) says that the red region of one row must never intersect with the blue region of any other row.

closer to z^0 than z^1 or have an equal distance from z^0 and z^1 . As we will show, for the ordered labels this is equivalent to the case where the labels $l^1(x)$ for all $x \in \mathcal{A}$ are roughly mirrored images of the labels $l^0(x)$ for some centre of reflection. This concept is defined below.

Definition 1. An update policy (l^0, l^1) is called mirrored if

- (i) Either for all $x \in \mathcal{A}$ we have $l^0(x) < l^1(x)$ or for all $x \in \mathcal{A}$ we have $l^0(x) > l^1(x)$, and
- (ii) there exists $\mu \in \mathcal{L}$ such that for all $x \in \mathcal{A}$ we have

$$(l^0(x) + l^1(x))/2 \in \{\mu, \mu + \frac{1}{2}, \mu + 1\}.$$

It is called mirrored as for all the labels $x \in \mathcal{A}$, $l^1(x)$ is the mirrored image of $l^0(x)$, however, the centre of symmetry can be μ , $\mu + \frac{1}{2}$ or $\mu + 1$ for different labels. This concept is illustrated in Fig. 8.

The next proposition says that being a mirrored policy is the necessary and sufficient condition for submodularity.

Proposition 2. With the set of labels \mathcal{L} defined as $\{0, 1, \dots, M-1\}$ and the distance function on $\mathcal{L} \times \mathcal{L}$ defined as $d(x, y) = |x - y|$ (ordered labels), and having state preservation property for the update policy, the energy function E'_x defined in (8) is submodular for all values of \mathbf{x} if and only if (l^0, l^1) is a mirrored update policy.

Proof: According to Lemma 1, we only have to check the submodularity condition (10) for all $x \in \mathcal{L}$ and all $y, z \in \mathcal{A}$.

To prove the backward direction, assume that conditions (i) and (ii) of a mirrored policy hold. From condition (i), we just consider the first case where for all $x \in \mathcal{A}$ we have $l^0(x) < l^1(x)$ as the proofs by assuming the other case is similar. Therefore, for any $y, z \in \mathcal{A}$ we have $y^1 > y^0$ and $z^1 > z^0$. Now assume that in (18) the left term in parentheses is negative, that is

$$|x - y^1| - |x - y^0| < 0, \quad (19)$$

we will show that in this case the other term $(|x - z^1| - |x - z^0|)$ is nonpositive and hence the relation (18) holds. First, notice

that, as $y^1 > y^0$, we have $x - y^1 < x - y^0$. This along with (19) gives $-(x - y^0) < x - y^1$. It follows that

$$x > (y^0 + y^1)/2 = \mu + \delta_y,$$

for some $\delta_y \in \{0, \frac{1}{2}, 1\}$, where the equality comes from the property (ii) of a mirrored policy and the fact that $y \in \mathcal{A}$. As x and μ are integers and $\delta_y \geq 0$, we can say $x \geq \mu + 1$, which gives $x \geq (z^0 + z^1)/2$ as for a mirrored policy we have $(z^0 + z^1)/2 \in \{\mu, \mu + 1/2, \mu + 1\}$ for the active label $z \in \mathcal{A}$. It then follows that

$$-(x - z^0) \leq x - z^1. \quad (20)$$

From $z^0 < z^1$ we have $x - z^1 < x - z^0$ which along with (20) gives $-(x - z^0) \leq x - z^1 < x - z^0$, that is $|x - z^1| \leq |x - z^0|$. Thus, the left term in the parentheses in (18) is nonpositive. A similar discussion can also prove that if $(|x - y^1| - |x - y^0|)$ is positive in (18), then $(|x - z^1| - |x - z^0|)$ is nonnegative and hence (18) always holds.

To prove the forward direction, assume that (18) obtains. By setting $x = 0$ (the smallest label value) in (18) we get

$$(y^1 - y^0)(z^1 - z^0) \geq 0.$$

As y and z can have any values, the above gives the condition (i) of a mirrored update policy.

From here on we assume that in condition (i) proved above, the first possibility is true, that is for all $x \in \mathcal{A}$ we have $x^0 < x^1$. The proof by assuming the second possibility, that is for all $x \in \mathcal{A}$ one has $x^0 > x^1$, is similar.

To get condition (ii) of a mirrored policy, choose μ as

$$\mu = \min_{x \in \mathcal{A}} \left\lfloor \frac{l^0(x) + l^1(x)}{2} \right\rfloor, \quad (21)$$

and, as \mathcal{A} is nonempty, we can set a variable z equal to an optimal x for (21), that is $\mu = \lfloor (z^0 + z^1)/2 \rfloor$. Now we take an arbitrary active label $y \in \mathcal{A}$ and show that the condition (ii) of a mirrored policy holds for y . From the definition of μ in (21) we know that

$$(y^0 + y^1)/2 \geq \mu. \quad (22)$$

Now, as $z^1 > z^0$, it is obvious that $\mu + 1 = \lfloor (z^0 + z^1)/2 \rfloor + 1$ is closer to z^1 than z^0 and hence $|z^1 - (\mu + 1)| - |z^0 - (\mu + 1)| < 0$. Therefore, by setting $x = \mu + 1$ in (18) (with y and z as defined above), the submodularity condition (18) gives

$$|y^1 - (\mu + 1)| - |y^0 - (\mu + 1)| \leq 0.$$

This means that y^0 is not closer to $\mu + 1$ than y^1 . Therefore

$$(y^0 + y^1)/2 \leq \mu + 1. \quad (23)$$

The constraints (22) and (23) leaves three possibilities for $(y^0 + y^1)/2$, that is $(y^0 + y^1)/2 \in \{\mu, \mu + \frac{1}{2}, \mu + 1\}$. This is the condition (ii) for being a mirrored policy. ■

From the above proposition, one can check that alpha-expansion does not give a submodular function for ordered labels when the size of the label set \mathcal{L} is bigger than two. This can be seen for example by setting $\alpha = 0$ and observing

that the update policy is not mirrored. However, alpha-beta swap does give a submodular function, as the update policy is mirrored for any choice of α and β , with $\mu = \lfloor(\alpha + \beta)/2\rfloor$. As mentioned before, the disadvantage of alpha-beta swap is that only two labels have the potential to change at each iteration. However, according to the above theorem, submodular algorithms can be designed in which more than two labels are changed. For example, at the same iteration where α and β are swapped, $\alpha-1$ and $\beta+1$ can also be swapped (given $0 < \alpha < \beta < M-1$) without affecting the submodularity.

2) *Unordered Labels*: In this case, the only thing we know about the labels is whether or not they are equal. The distance function is defined as $d(x, y) = 1(x \neq y) \in \{0, 1\}$, that gives 0 if $x = y$ and 1 otherwise. Because of the binary nature of the distance function, compared to the case of ordered labels, it appears that here the submodularity holds for a vaster range of update policies. We will shortly show that this statement is true in some sense.

Here, the submodularity condition (10) becomes

$$(1(x \neq y^1) - 1(x \neq y^0)) (1(x \neq z^1) - 1(x \neq z^0)) \geq 0. \quad (24)$$

The next proposition gives a necessary and sufficient submodularity condition for the unordered labels.

Proposition 3. *With the distance function defined as $d(x, y) = 1(x \neq y)$ (unordered labels), and by assuming the state preservation property for the update policy, the energy function E'_x defined in (8) is submodular for all $\mathbf{x} \in \mathcal{L}^n$ if and only if for any pair of active labels $y, z \in \mathcal{A}$, we have $l^0(y) \neq l^1(z)$.*

Proof: According to Lemma 1, we only have to check (10) for the case where $y, z \in \mathcal{A}$, that is $y^0 \neq y^1$ and $z^0 \neq z^1$.

We start with the proof of the backward direction. Assume that for any $y, z \in \mathcal{A}$ we have $l^0(y) \neq l^1(z)$, it gives $y^0 \neq z^1$ and $y^1 \neq z^0$ for $x, y \in \mathcal{A}$. Therefore, in (24), both $1(x \neq y^1)$ and $1(x \neq z^0)$ cannot be equal to 0 at the same time nor can be $1(x \neq y^0)$ and $1(x \neq z^1)$. It follows that (24) holds for all $y, z \in \mathcal{A}$ and any value of $x \in \mathcal{L}$.

To prove the other direction, assume that there exist $y, z \in \mathcal{A}$ such that $y^0 = z^1$. Now, in (24) take $x = y^0 = z^1$, as $y, z \in \mathcal{A}$, it follows that $x \neq y^1$ and $x \neq z^0$, and therefore (24) does not hold. This proves the proposition. ■

From the above theorem it can be checked that alpha-expansion as well as alpha-beta swap gives a submodular binary optimization in an unordered scheme, as it never happens for any $y, z \in \mathcal{A}$ that $l^0(y) = l^1(z)$.

Now consider the case where, like the case of ordered labels, the set \mathcal{L} is in the form of $\mathcal{L} = \{0, 1, \dots, M-1\}$, but unlike the ordered label case, we use the distance function $d(x, y) = 1(x \neq y)$. If the policy is mirrored, as defined in Definition 1, then we have $x^0 \leq \mu < y^1$ or $x^0 > \mu \geq y^1$ for all $x, y \in \mathcal{A}$ and hence it never happens that $x^0 = y^1$. This means that the binary problem would be also submodular in an unordered scheme, according to proposition 3. The reverse, however, is not true in general as there exist update policies like in alpha-expansion that are not generally submodular in the ordered case, while being always submodular in an unordered system.

V. CONCLUSION

In this paper we suggested an isotropic gradient MRF model and studied the submodularity of the binary move problem in the move-based optimization algorithms. We considered two cases of *ordered labels* and *unordered labels*. For the ordered labels, our findings were as follows:

- The suggested model well suits the ordered case, especially if the number of labels is large.
- The necessary and sufficient condition for the binary move update policy to give a submodular binary energy function is that the two update possibilities for each label are (roughly) mirrored reflections of each other.
- As a result, the application of alpha-expansion for the suggested multi-label isotropic model is ruled out for the ordered labels.
- Alpha-beta swap can be applied in the ordered labels scenario as it gives a submodular binary move problem.
- Better algorithms than alpha-beta swap can be suggested in terms of having more number of labels involved in the binary move at each iteration.

For the unordered case we can say:

- The suggested model is not quite isotropic in terms of giving the same cost as the edges are rotated, yet it works better than the horizontal and vertical 2-clique model.
- For the unordered case, the submodularity holds for vaster types of binary moves, including alpha-expansion and alpha-beta swap.

According to the above, we can see the need for the development of new algorithms to deal with the case of ordered labels. This paper mainly addressed the theoretical aspects of this topic, yet it serves as a basis for the design of efficient optimization algorithms for the isotropic gradient MRF model.

REFERENCES

- [1] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.
- [2] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.
- [3] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, "An introduction to total variation for image analysis," in *Theoretical Foundations and Numerical Methods for Sparse Recovery*. De Gruyter, 2010.
- [4] D. S. Hochbaum, "An efficient algorithm for image segmentation, Markov random fields and related problems," *J. ACM*, vol. 48, no. 4, pp. 686–701, Jul. 2001.
- [5] A. Chambolle and J. Darbon, "On total variation minimization and surface evolution using parametric maximum flows," *Int. J. Comput. Vision*, vol. 84, no. 3, pp. 288–307, Sep. 2009.
- [6] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 2, pp. 147–159, feb. 2004.