Extended Pairwise Potentials

Steve Gu
Duke University
steve@cs.duke.edu

Ying Zheng
Duke University
yuanqi@cs.duke.edu

Carlo Tomasi
Duke University
tomasi@cs.duke.edu

Abstract

Inference in Markov Random Fields can be cast as the minimization of a potential function that is typically composed of unary and pairwise terms. The pairwise potentials measure the cost of assigning labels to two neighboring pixels and are often in the form of differences between labels, rather than of their separate values. We generalize this formulation to allow pairwise potentials to depend on both label differences AND their separate values. We also show that the minimization can be computed efficiently by using an extended version of the generalized distance transform in the belief propagation algorithm. We show that the generalized potential function may be useful in applications such as image restoration and labeling where fine grained control is desirable.

1. Introduction

Markov Random Field (MRF) models have been quite popular in computer vision due to their excellent performance [2] and efficient approximate optimization techniques based on graph cuts [1] or belief propagation [3].

The general problem is one of label assignment: Let $\mathcal{P}$ be the image pixel grid and $\mathcal{L}$ be a finite set of labels. The cost of a labeling $f : \mathcal{P} \to \mathcal{L}$ is:

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{(p,q) \in \mathcal{N}} \Phi(f_p, f_q) \tag{1}$$

where f_p is the label of pixel p and $\mathcal{N}$ is a neighborhood system such as the four-connected image grid. $D_p(f_p)$ is the data cost for assigning label f_p to pixel p and $\Phi(f_p, f_q)$ is what we call the pairwise potential, which measures the cost of assigning label f_p and f_q to neighboring pixels p and q. The goal is to find the labeling with minimal costs.

In low-level vision problems such as image de-noising, restoration and stereo, the pairwise potential $\Phi(f_p, f_q)$ depends typically only on the difference between labels rather than their separate values. For instance, the typical design for $\Phi(f_p, f_q)$ is the squared Euclidean distance:

$$\Phi(f_p, f_q) = (f_p - f_q)^2 \tag{2}$$

or the L_1 distance:

$$\Phi(f_p, f_q) = |f_p - f_q| \ . \tag{3}$$

The former induces a cost that grows more rapidly than the latter does with the difference between the two labels. Another useful variant is the truncated pairwise potential which imposes a ceiling on the cost of a labeling:

$$\Phi(f_p, f_q) = \min\{|f_p - f_q|, d\} \tag{4}$$

where d is a constant positive number.

Although these designs differ in rationales, their common feature is that they depend only on the difference of labels rather than on their separate values. This is quite natural, as it captures a preference for piecewise smooth images, in which nearby pixels tend to have similar values.

The restriction to differences has led to efficient algorithms based on the generalized difference transform [2]. However, this restriction is burdensome at times. For instance, if the frequency of occurrence of different labels is different, a pairwise potential that knows only the difference between labels cannot use this extra information in a flexible way. In another scenario, suppose that we know some labels coexist easily with others nearby and some do not. How to incorporate this prior into the pairwise potential?

The contributions of this paper are twofold: First, we extend the pairwise potential so that it depends on both the difference of labels and their separate values. Second, we give an efficient message passing algorithm to compute the optimal labeling. More specifically, we show that the propagation of messages among nearby nodes takes $O(k \log k)$ time instead of the otherwise quadratic time, where k is the number of labels.

2. Extended Pairwise Potentials

We propose to study pairwise potentials of the form:

$$\Phi(f_p, f_q) = \frac{1}{2}\left[\lambda_p(f_p) + \lambda_q(f_q)\right](f_p - f_q)^2 \tag{5}$$

where $\lambda : \mathcal{P} \times \mathcal{L} \to \mathbb{R}^+$ is a general function defined jointly in the product of image grid and label space. The expression

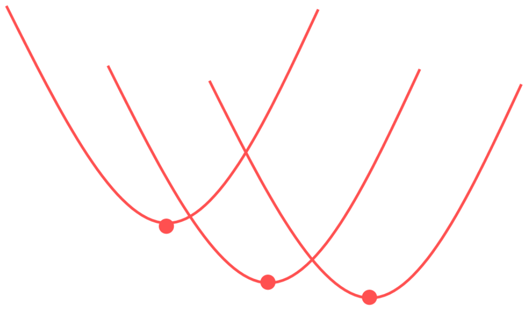

Figure 1. The lower envelope of parabolas with equal eccentricity.

$\lambda_p(f_p)$ is therefore a simplified notation for $\lambda(p, f_p)$. Our discussion is not affected even if we replace $(f_p - f_q)^2$ with $|f_p - f_q|$ or a truncated version similar to Equation (4).

Clearly, the new definition reduces to the original pairwise potential by setting $\lambda_p = 1$ for all p and f_p. The added flexibility is that λ is arbitrarily defined on both pixel and label space rather than a single constant. From a regularization point of view, instead of using a single fixed number to balance the data cost and the discontinuity cost, the new regularization is completely adaptive and allows a fine grained incorporation of extra information about the task at hand.

Several applications can benefit from this generalization. For instance, in the task of image restoration, suppose we estimate from images that certain colors appear more frequently than others. We can then set $\lambda_p(f_p)$ to be inversely proportional to the probability of occurrence of f_p. This implicitly encourages using certain color models to restore the image. In the extreme, we can set $\lambda_p(f_p)$ to infinity for irrelevant colors in f_p and this results in color compression.

In object labeling, let us create the hypothetical label space $\mathcal{L} = \{\text{sky}, \text{grass}, \text{horse}, \text{dinosaur}, \cdots\}$. If we know that the occurrence of a dinosaur is less likely compared to sky or horse, we then can set $\lambda_p(\text{dinosaur})$ to be a large number in order to discourage the labeling of any object to be a dinosaur in a soft manner. Note that this does not rule out the possibility that a dinosaur is actually detected.

3. Efficient Computation

We discuss how to do the optimization efficiently with extended pairwise potentials. Felzenszwalb and Huttenlocher [2] showed that computing the optimal labeling in Equation (1) using the message passing algorithm can be accelerated significantly when the pairwise potential depends only on the difference of labels. More specifically, they developed a linear time algorithm to compute the propagation of messages using the generalized distance transform, which amounts to computing the lower envelope of a set of parabolas (or cones) with equal eccentricity (or slope) (Figure 1). Compared to that algorithm, our study leads to a further generalization in one dimensional cases for both the squared Euclidean and L_1 distances. Geometrically we compute the lower envelope of a set of parabolas with unequal eccentricity (Figure 2).

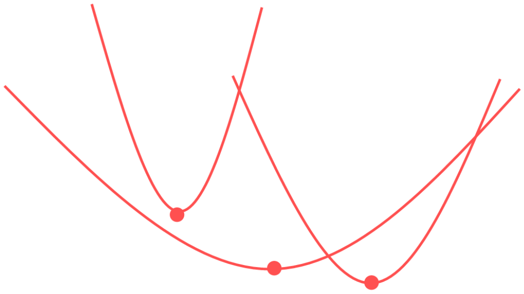

Figure 2. Parabolas with unequal eccentricity.

We first revisit the belief propagation algorithm. The method works by aggregating and passing messages in parallel and in an iterative way. Each message is a vector of dimension equal to the number of labels. Let $m^t_{p\to q}$ be the message sent from node p to node q at iteration t. This message is computed in the following way:

$$m^t_{p\to q}(f_q) = \min_{f_p}\left(\Phi(f_p, f_q) + \underbrace{D_p(f_p) + \sum_{s\in\mathcal{N}_p\backslash q} m^{t-1}_{s\to p}(f_p)}_{h(f_p)}\right) \quad (6)$$

$$= \min_{f_p}\left[\Phi(f_p, f_q) + h(f_p)\right] \quad (7)$$

where $h(f_p)$ aggregates the local messages at iteration $t-1$ and the data cost for pixel p. After a number of iterations a belief vector results from each pixel, and a label that minimizes the belief vector is selected individually at each node. How to compute the messages defined in Equation (7) efficiently is a challenging problem.

Clearly a naive algorithm takes $O(k^2)$ time for each piece of message where k is the number of labels. On the other hand, when the pairwise potential $\Phi(f_p, f_q)$ takes the form $|f_p - f_q|$ or $(f_p - f_q)^2$, the computation is known to be $O(k)$. We show that for the extended pairwise potentials, the messages can be computed in $O(k \log k)$ time. First we split the potential $\Phi(f_p, f_q)$ into two pieces:

$$\Phi(f_p, f_q) = \Phi_{p\to q}(f_p, f_q) + \Phi_{q\to p}(f_p, f_q) \quad (8)$$

where $\Phi_{p\to q} = \frac{1}{2}\lambda_p(f_p)(f_p - f_q)^2$ and $\Phi_{q\to p}(f_p, f_q)$ is the remaining term. Because each pairwise potential is computed twice for each incident node, we are mainly interested in computing the following objective (ignoring constant fac-

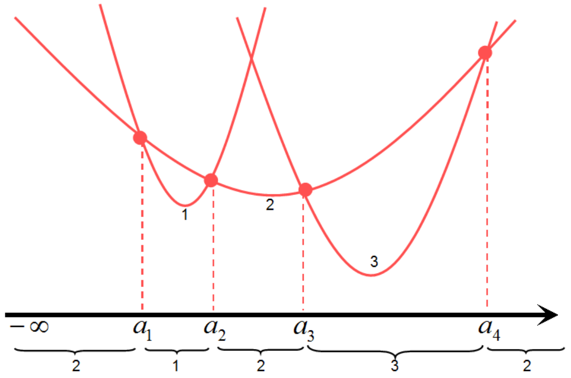

Figure 3. Representation of the lower envelope. Red dots are the transition points and each interval memorizes the index of the parabola that forms the lower envelope within that interval.

tors or truncated variations):

$$m^t_{p\to q}(f_q) = \min_{f_p}\left[\lambda_p(f_p)(f_p - f_q)^2 + h(f_p)\right] \quad (9)$$

Equation (9) is an extension of the generalized distance transform. To put it in a more clear form, we want to compute the following distance transform:

$$g(x) = \min_{y}\left[\lambda(y)(x - y)^2 + h(y)\right] \quad (10)$$

where $x, y \in \mathcal{L}$ and g, λ, h are simplified notation for the above messages and functions. Geometrically this is equivalent to computing the lower envelope of a set of parabolas with different eccentricity (Figure 2).

4. The Algorithm

We give a simple $O(k \log k)$ algorithm using divide and conquer. At a high level, the algorithm partitions the k-dimensional array (the message) into two disjoint subsets, each of size at most $\lceil \frac{k}{2} \rceil$. Lower envelopes are first computed on each subset and then merged together.

We use a list of transition points to represent the lower envelope. More formally, the lower envelope $\mathcal{L}$ is represented by a sequence $(-\infty = a_0, a_1, \cdots, a_n)$ where $a_0 < a_1 < \cdots < a_n$ are real numbers that form the transition points (Figure 3). To each transition point we also associate the index of the parabola that forms the lower envelop immediately to the right of that point.

We show that merging two lower envelopes $\mathcal{L}_1 = (a_0, \cdots, a_m)$ and $\mathcal{L}_2 = (b_0, \cdots, b_n)$ takes time $O(m + n)$. This is achieved by the following:

Step 1. Sequences merge: Unite $\mathcal{L}_1$ and $\mathcal{L}_2$ to $(c_0, \cdots, c_k)$ with $c_0 < \cdots < c_k$ and $c_i \in \mathcal{L}_1$ or $c_i \in \mathcal{L}_2$. Identical transition points are merged into single points.

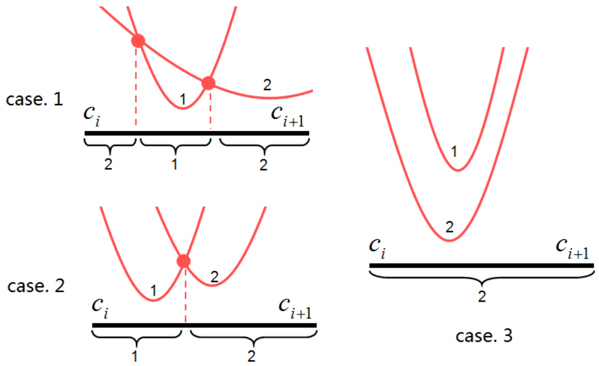

Figure 4. Three cases of interval splitting corresponding to two, one, or no intersection between two parabolas that form the lower envelope in the interval (c_i, c_{i+1}).

Step 2. Intersection of parabolas: For each interval (c_i, c_{i+1}), let v_1 and v_2 be the two parabolas that form the lower envelope in that interval in $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively. Split (c_i, c_{i+1}) into sub-intervals according to three different cases of intersections (Figure 4).

Step 3. Index update: Update the index of the parabola that lies on the lower envelope immediately to the right side of each transition point. Merge adjacent intervals together if they share the same parabola index.

4.1. Complexity Analysis

Both Step 1 and Step 3 clearly take linear time in m and n. For Step 2, since each interval can split into at most three intervals, the resulting number of intervals is still linear in m and n. Let $T(k)$ be the time for computing the lower envelope of k parabolas. The following recursion holds:

$$T(k) = \begin{cases} 2T(\frac{k}{2}) + O(k) & k > 1, \\ O(1) & k = 1, \end{cases} \quad (11)$$

The running time is therefore $T(k) = O(k \log k)$.

5. Experiments

We apply the extended pairwise potentials to the task of image restoration. The goal is to test its ability in incorporating certain color distributions for the fine-grained control of image restoration. This might be useful for compression where very few colors are used to encode an image.

Figure 5 shows our experiment results on synthetic data. In Figure 6 and Figure 7 we show the results on real images. In all the cases, λ takes roughly the shape of the inverse of the intensity histogram of images. The resulting image restorations therefore use only very few colors to encode an image. We expect that our method may also be useful for image segmentation.

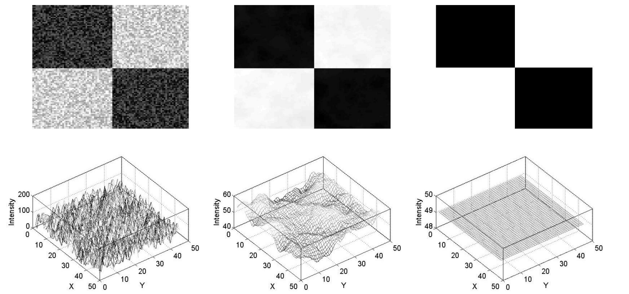

Figure 5. Top row: Left is the input image. Middle is the image restoration obtained by [2]. Right is our result. The bottom row shows the zoomed in surface plot of the upper left black cell in the top row. The weights λ are set low for black and white and high for other intensity values. Best viewed when enlarged.

6. Conclusions

We extended pairwise potentials in MRFs so that they depend on both the difference of labels and their separate values. The computation is made efficient by reducing the message passing to computing the lower envelope of set of parabolas with unequal eccentricity. We show that this step can be computed in $O(k \log k)$ time using divide and conquer. Experiments show that the pairwise potential may be useful in applications like image restoration where a fine-grained integration of prior knowledge is preferred.

Acknowledgement: This material is based upon work supported by the National Science Foundation under Grant No. IIS-1017017 and by the Army Research Office under Grant No. W911NF-10-1-0387.

References

[1] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE PAMI*, 23(11):1222–1239, 2001. 1

[2] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1):41–54, 2006. 1, 2, 4

[3] Y. Weiss and W. Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):736–744, 2001. 1

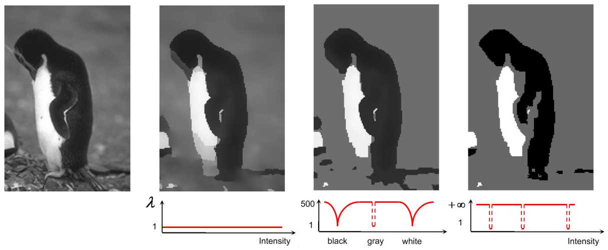

Figure 6. The left two images are the input image and the restoration by [2]. The right two images are the image restoration results by our chosen potentials. Red curves illustrate the choice of λ.

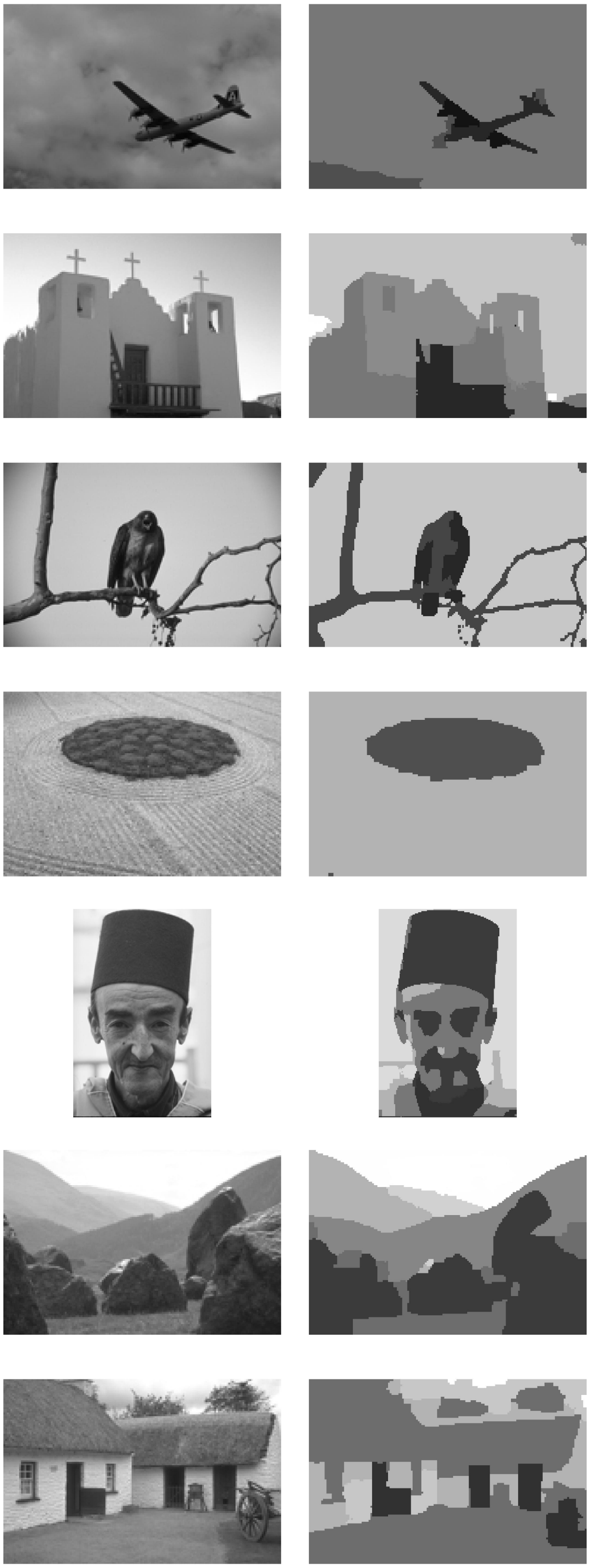

Figure 7. Left: input; Right: restored images with color models.