HOPS: Efficient Region Labeling using Higher Order Proxy Neighborhoods

Albert Y. C. Chen[1], Jason J. Corso[1], and Le Wang[2]
[1]*Dept. of Computer Science and Engineering*
[2]*Dept. of Geography*
University at Buffalo SUNY
aychen@cse.buffalo.edu

Abstract

We present the Higher Order Proxy Neighborhoods (HOPS) approach to modeling higher order neighborhoods in Markov Random Fields (MRFs). HOPS incorporates more context information into the energy function in a recursive and cached manner. It induces little or no additional computational cost in the overall minimization process, and can better represent the underlying energy leading to fewer total computations. Indeed, when integrated with the Graph-Shifts energy minimization algorithm we observe a 30% average decrease to the convergence time. We apply HOPS to high-level labeling of natural and geospatial images; our results show that HOPS leads to smoother labelings that better follow object boundaries. HOPS can label an image with an average 75% accuracy in a couple of seconds.

1. Introduction

Energy minimization is the standard approach for solving image labeling problems, such as assigning an object class with each pixel. During energy minimization, achieving the global optimum can be very costly in terms of computational complexity (the solution space is combinatorial in the number of labels). Thus, a variety of approximation methods have been proposed. Previous algorithms, however, either suffer from low convergence speeds [6, 5] and local minima [1], or are able to perform large changes in the energy space but do not guarantee convergence on some energies [2].

The recently proposed Graph-Shifts algorithm [3, 4] exploits an adaptive hierarchical representation of an image to rapidly decrease the energy and converge efficiently. Graph-Shifts does minimization by dynamically manipulating the hierarchical representation to relabel, or *shift*, large groups of pixels at the same time. Over 75% of the total energy reduction is performed within the first 5% of all shifts; its average convergence time on a modern PC is only several seconds, which is substantially faster than many energy minimization algorithms. Although it was originally invented for segmenting sub-cortical structures in Magnetic Resonance images [3], it has been shown to work well in other applications, such as natural image restoration [4]. However, due to the first order Markov Random Field (MRF) energy model it adopted, its labeling results on natural images can be noisy and do not always follow local boundary structure.

Our proposed Higher Order Proxy Neighborhoods (HOPS), which is inspired by the Belief Propagation algorithm [5], extends the MRF energy to third order neighbors in a recursive and cached manner with little additional computational cost. With more contextual information taken into consideration, HOPS further optimizes the energy minimization process, resulting in a 30% speedup of the already efficient Graph-Shifts algorithm. The number of noisily labeled regions are greatly reduced and object boundaries are better preserved, thus producing more meaningful labels when HOPS is applied to natural images.

2. The Graph-Shifts Algorithm

In short, the Graph-Shifts algorithm is composed of two stages: (1) Coarsening: an adaptive hierarchy is built on top of the image by recursively grouping the nodes (pixels) with similar values (color, intensity, depth) together, and (2) Shifting: nodes *shift* iteratively by taking the label of their neighbor, which would cause all of its descendants to take the new label.

2.1 The Adaptive Dynamic Hierarchy

The hierarchical structure differs from traditional pyramidal structures of the image in the following ways: (1) The hierarchy is built according to the data

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

instead of pixel coordinates. (2) The hierarchy is dynamic, so that the structure of nodes (parent-child relationship) will be changed iteratively while the energy is being minimized.

A multiple-layer hierarchical graph G consists a set of nodes $\mu \in U$ and edges. The nodes at the lowest layer, which represent the image pixels, are joined by an edge if and only if they are neighboring pixels (first order). Nodes at coarser layers are connected if any of their children are connected, and this connectivity construction is applied recursively to each layer from bottom to top. Each node μ could only have one parent node, except the nodes at the top layer, while each node except the nodes at the bottom layer must have at least one child node. The top layer of the hierarchy is defined as the special $root$ layer of nodes. Each single node in this layer represents one of the K labels, and all its descending nodes are assigned the same label. Since every single node at the bottom layer would be assigned a label, an instance of the graph G is the same as an image labeled into K classes.

2.2 The Energy Models

Starting from the lowest level of G, the energy function is defined as:

$$E[\{m_\omega : \omega \in D\}] = \sum_{\mu \in D} E_1(\mathbf{I}(S[\mu]), m_\mu) + \frac{1}{2} \sum_{\langle \mu, \nu \rangle} E_2(m_\mu, m_\nu) \tag{1}$$

where D is the image lattice, and $S\,[\nu]$ is the local subimage surrounding ν. When applied to high-level vision problems such as image labeling, E_1 is the local evidence that pixel μ takes label m_μ, where the probability $P(m_\mu|\phi(\mathbf{I})(\mu))$ is conditioned on a local sub-image of μ; it is a context-sensitive discriminative model (explained in section 4).

$$E_1(\mu, m_\mu) = -\log P\big(m_\mu|\mathbf{I}(S[\mu])\big) \tag{2}$$

E_2 can be viewed as the penalty applied to neighboring nodes possessing different class labels, which is

$$E_2(m_\mu, m_\nu) = 1 - \delta_{m_{\mu,\nu}} \quad . \tag{3}$$

The energy is accumulated recursively from bottom to top throughout the hierarchy. Leaf nodes calculate their energy directly from (1) - (3), while non-leaf nodes sum the energy from all their child nodes. Refer to [4] for details on the recursion.

2.3 Graph-Shifts

When a node μ decides to change its class label from m_μ to its neighboring node's class $\hat{m}_\mu$, all of μ's descendants would also follow the change. This process is called a *shift*, and it is denoted as $m_\mu \to \hat{m}_\mu$. The *shift-gradient* $\Delta E(m_\mu \to \nu)$ is defined as:

$$\Delta E(m_\mu \to \hat{m}_\mu) = E_1(\mu, \hat{m}_\mu) - E_1(\mu, m_\mu) + \frac{1}{2} \sum_{\eta : \langle \mu, \eta \rangle} [E_2(\hat{m}_\mu, m_\eta) - E_2(m_\mu, m_\eta)] \tag{4}$$

For each node in the hierarchy, shift-gradients among all neighbors are calculated and only those that are negative (which would cause an energy decrease) will be stored in the list of potential shifts S. The shift with the steepest shift-gradient is chosen at each iteration, causing the node and all its descendants' labels to be changed. The nodes and edges affected by the shift will be updated, along with the list S with recomputed energies. The process is iterated until convergence when S becomes empty, which means that any further shift will not decrease the energy.

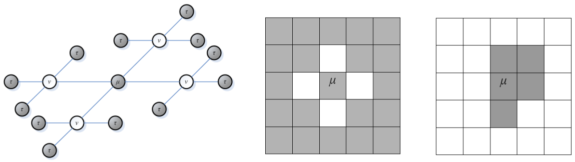

Figure 1. Two examples of when the traditional first order pairwise neighborhood is insufficient.

3. Extending Graph-Shifts

With the original first order neighborhood, the algorithm may perform unnecessary shifts. Two simple examples are illustrated in figure 1. Suppose in the original pairwise first order neighborhood system, we have neighboring nodes on the same layer that are assigned to two classes $+1$ (white) and -1 (gray) as shown on the leftmost graph. To facilitate visualization, an equivalent pixel layer representation is shown in the middle. Since the visiting order is data dependent, if node μ is visited before its neighbors, it would change its class from -1 to $+1$, but the neighboring nodes would possibly change from $+1$ to -1 sometime later due to the strong response of the E_2 term. If μ is changed back to -1 again due to label change of its ancestors, the effort of the previous shift would be redundant; if μ retains label $+1$, then it would be a single node with $+1$ in the middle of a field of -1, which is unlikely to be a reasonable label, and should be treated as noise to be removed.

3.1 The Higher Order Proxy Neighborhoods

In order to avoid being confused in such local situations thus performing unnecessary shifts, one could completely overhaul the E_2 term and build a local histogram surrounding μ of size $x \times x$ to justify the distribution of different classes, or to extend the neighborhood relationship to one that covers more ground (e.g. a pairwise third order neighborhood). The first approach would increase the computational time by $x^2/|N_1|$, where x is the size of the local window and $|N_i|$ is the size of the original ith order neighborhood. The second approach is even more complicated, since connecting a node with all its second and third order neighbors would increase the number of connected neighbors for any node μ by $|N_1| \cdot |N_2| \cdot |N_3|$. Such greater neighborhood complexity would increase the computational cost of the minimization.

We propose HOPS, which incorporates energy from second and third order neighbors yet still maintains the standard first order neighborhood system. Inspired by Belief Propagation [5], the connectivity of the nodes and hierarchy is not changed. Rather, information from higher order neighbors τ_j is received indirectly through first order neighbor proxies ν_i in a process like message passing (as shown in figure 2). The energy function is revised as:

$$E[\{m_\omega : \omega \in D\}] = \sum_{\mu \in D} E_1(\mathbf{I}(S[\mu]), m_\mu) + \frac{1}{2}\Bigg(\sum_{\langle \mu, \nu_i \rangle} E_{2-1}(m_\mu, m_{\nu_i}) + \frac{1}{\alpha} \sum_{\langle \nu_i, \tau_i \rangle} E_{2-2}(m_{\nu_i}, m_{\tau_j}) \Bigg) \tag{5}$$

where E_{2-1} and E_{2-2} can be implemented using equation (3), and α depends on the ν_i's number of neighbors.

If the energy is calculated directly at each node, the computation time is expected to increase by an order of 2 at each iteration. However, since the E_{2-2} term calculated at node μ would be the same as the E_{2-1} term of node ν, we could get an algorithm with only linear increase in time complexity, if we cache and update the E_{2-1} term in each node ν. By doing so, whenever an arbitrary node μ is calculating its energy and needs the E_{2-2} term, it can simply sum up all the E_{2-2} term passed back from its first order neighbors ν.

With HOPS, redundant shifts are avoided in situations such as the ones illustrated in figure 1. Node μ in the first graph will possibly change its value from -1 to $+1$ even if it is selected before its neighbors. In the rightmost graph, the region with label value -1 will have a higher probability of being changed into a smooth field of value $+1$, since the energy function is now taking more neighborhood/context information into consideration when the shift is performed.

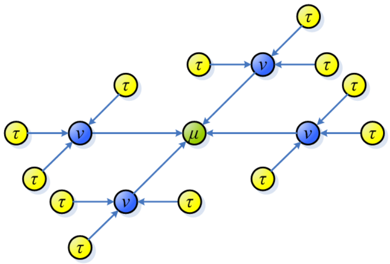

Figure 2. The Higher Order Proxy Neighborhoods.

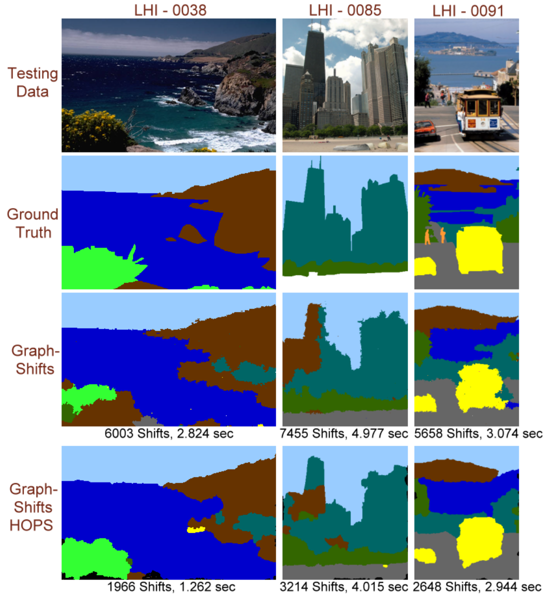

Figure 3. HOPS v.s. First-Order.

4. Experiment Setup and Results

Our experiments have been conducted on a subset of the Lotus-Hill Institute (LHI) Dataset [8], which consists of 400 10-class images, and 2-class aerial images. We randomly split the data into 200 training sets and 200 testing sets, and the probability maps $P(m_\mu|\phi(\mathbf{I})(\mu))$ for the unary energy term E_1 are generated by the Probability Boosting Tree (PBT) algorithm [7]. All processing time is calculated on a Centrino 1.6GHz laptop with 1GB DDR-266 RAM using a Java implementation of the algorithm, and all parameters are set the same as the original Graph-Shifts algorithm [3, 4].

The computational complexity for each iteration of our method is $O(kN)$ if energies are cached as described, and $O(N^2)$ if the E_{2-2} term is recalculated

	🟦	🟦	⬛	🟩	🟩	🟫	🟧	🟦	🟪	🟨
sky	**88.2**	8.6			0.3	1.9		0.9		
water	2.8	**65.3**	4.6	2.1	5.9	16.0		3.0		0.3
road	0.1	16.1	**72.6**	0.4	0.8	5.3		3.6		0.9
grass	0.5	1.0	0.6	**69.1**	7.8	19.4		1.5		0.1
tree	1.1	1.8	0.5	4.3	**67.1**	17.3		7.6		0.2
mountain	1.9	3.4	0.6	4.0	9.8	**74.6**		5.4		0.1
creature		5.1	6.1	3.0	4.1	30.1	**8.6**	40.2		2.8
building	1.8	1.1	1.9	1.1	3.7	9.7		**80.1**		0.5
bridge	0.2	2.2	6.2	6.5	4.2	13.2		57.2	**8.2**	2.2
vehicle		5.2	11.0	0.1	6.5	20.5	0.1	21.6		**34.8**

Table 1. Confusion matrix for the HOPS extension on the LHI dataset.

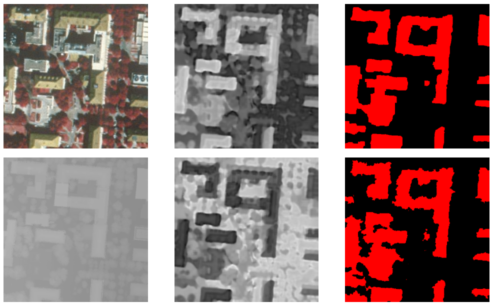

Figure 4. Results of aerial photo labeling.

instead of being cached. Comparing this with $O(N)$ for systems using the first order pairwise neighborhood, a slight increase in total computation time seems inevitable. However, since HOPS allows more contextual information to be considered into the energy function, potential redundant shifts such as the ones illustrated in Figure 1 would not be chosen, thus greatly reducing the total shifts needed for convergence. In our experiments, the total number of iterations until convergence is decreased by 30 – 75%, and the actual convergence time is decreased by 30 – 50%. For all 200 testing images, the overall pixel accuracy of our HOPS extension is 75% (refer to Table 1), which is roughly the same as the original Graph-Shifts algorithm. However, as we can see from figure 3, the proposed HOPS extension of the Graph-Shifts algorithm produces much more "reasonable" labels, and the object boundaries are better followed.

We also tested HOPS on color-infrared aerial photos and airborne LiDAR (Light Detection and Ranging) data, to prove that the extended neighborhood would also speed up the whole convergence progress, while performing as well as the original first order neighbor. The columns from left to right of Figure 4 are: a pseudo-color composition of the color-infrared photo and first return of LiDAR, histogram based probability maps (first row is the building class and second is the background class), labeling results of first-order and HOPS. Both HOPS and first-order Graph-Shifts are able to achieve good labeling results, yet HOPS constantly converges 30% faster than the first-order Graph-Shifts algorithm.

5. Conclusion

In summary, the proposed Higher Order Proxy Neighborhoods (HOPS) successfully improves the convergence speed of the already efficient Graph-Shifts algorithm by an average of 30% and produces more refined labeling results. Redundant iterations are effectively avoided by incorporating more neighboring context information into the energy function. Thus the convergence speed is increased while outputing smoother labels which follow local boundaries better. Unlike traditional higher order nieghborhood systems which would increase the complexity of node connectivity by orders of $|N_i|$, HOPS retrieves higher order neighbor information in a recursive and cached manner, which induces little additional computational cost for each iteration. These time-efficient characteristics makes HOPS ideal for real-time labeling tasks, and we plan to challenge accurate real-time video object labeling problems in the future, using MRF-based dynamic hierarchicical models with HOPS.

References

[1] J. Besag. Spatial interaction and the statistical analysis of lattice systems (with discussion). *Journal of the Royal Statistical Society, B*, 36:192–236, 1974.

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 23(11):1222–1239, 2001.

[3] J. J. Corso, Z. Tu, A. Yuille, and A. Toga. Segmentation of Sub-cortical Structures by the Graph-Shifts Algorithm. *In Proceedings of Information Processing in Medical Imaging (IPMI)*, pages 183–197, 2007.

[4] J. J. Corso, A. Yuille, and Z. Tu. Graph-Shifts: Natural Image Labeling by Dynamic Hierarchical Computing. *In Proceedings of IEEE Conference Computer Vision and Pattern Recognition (CVPR)*, 2008.

[5] B. J. Frey and D. MacKay. A Revolution: Belief Propagation in Graphs with Cycles. *In Proceedings of Neural Information Processing Systems (NIPS)*, 1997.

[6] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and Bayesian Restoration of Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 6:721–741, 1984.

[7] Z. Tu. Probabilistic Boosting-Tree: Learning Discriminative Models for Classification, Recognition, and Clustering. *In Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2005.

[8] Z. Yao, X. Yang, and S. C. Zhu. Introduction to a Large Scale General Purpose Ground Truth Dataset: Methodology, Annotation Tool, and Benchmarks. *In Proceedings of Int'l Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, 2007.