On The Effects of Normalization in Adaptive MRF Hierarchies

Albert Y. C. Chen and Jason J. Corso

Dept. of Computer Science and Engineering
University at Buffalo, SUNY
aychen@buffalo.edu

Abstract. In this paper, we analyze the effects of energy normalization in adaptive-hierarchy-based energy minimization methods. Adaptive hierarchies provide a convenient multi-level abstraction of the underlying MRF. They have been shown to both accelerate computation and help avoid local minima. However, the standard recursive way of accumulating energy throughout the hierarchy causes energy terms to grow at different rates. Consequently, the faster-growing term, typically the unary term, dominates the overall energy at coarser level nodes, which hinders larger-scale energy/label change from happening. To solve the problem, we first investigate the theory and construction of adaptive hierarchies, then we analyze the theoretical bounds and expected values of its energy terms. Based on these analyses, we design and experimentally analyze three different energy-normalizing schemes. Our experiments show that properly normalized energies facilitate better use of the hierarchies during optimization: we observe an average improvement in the speed by 15% with the same accuracy.

1 Introduction

Markov random fields (MRF) provide a convenient and consistent way of modeling contextual constraints and stochastic interaction among variables, and have been shown useful in solving both low-level vision (e.g. restoration, stereo matching) and high-level vision problems (e.g. semantic image labeling). These problems are formulated as *labeling* tasks, where each site μ in the lattice G^0 is assigned a label $\mathcal{L}$ that can represent either low- or high-level features. The optimal solution is pursued by searching for the labeling configuration with the highest posterior probability, or equivalently, by minimizing the corresponding Gibbs energy function. Achieving the global optimum, however, is typically intractable, since the solution space is combinatorial in the number of labels. Therefore, a variety of approximation algorithms have been proposed to accelerate computation [1–4]; hierarchical approaches [5–9], such as multiscale/multi-resolution methods, multigrid relaxation, or renormalization group theory/transformation, are among the popularly studied ones.

Traditional hierarchies, where finer-level nodes are grouped into coarser-level nodes by their spatial coordinates instead of their intrinsic similarity, contain incoherent nodes around boundary regions at coarser levels. An incoherent node at layer $n+1$ (e.g., μ^{n+1} in Fig. 1) is a mixture of multiple intrinsically different nodes at level n, denoted $\{\mu_j^n\}$. Incoherent nodes not only blur the boundaries of the finer-level graph (e.g., the edges

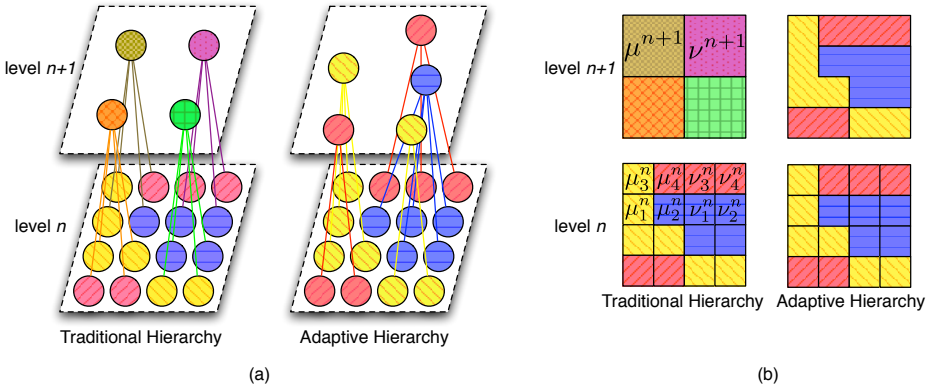

Fig. 1. Traditional versus Adaptive Hierarchies. The hierarchies in (a) are projected onto 4×4 images (b) for the ease of visualization. The coherent node at level $n+1$ of the adaptive hierarchy can accumulate the energies of belonging to the red, yellow, or blue labels (textured with slashes, backslashes, and horizontal lines respectively for monochrome prints) directly from their child nodes, whereas the incoherent nodes in the traditional hierarchy requires a re-computation.

between μ_3^n, μ_4^n, and μ_2^n, μ_4^n) but also make the direct reutilization of energies from the original graph difficult. For example, the energy between μ_2^n, ν_1^n and μ_4^n, ν_3^n are 0 while using the Potts model, yet the energy between μ^{n+1} and ν^{n+1} is 1, which cannot be derived from its children μ_j^n's directly and hence must be recalculated at level $n+1$. Conversely, it is difficult for a label/energy change at μ^{n+1} to propagate effectively to its children on level n: $\{\mu_j^n\}$. Therefore, traditional hierarchy-based approaches require an costly overhead of recalculating the energies and weights for every coarser level. Adaptive hierarchies, such as [10–12], produce coherent nodes because the methods coarsen based on the similarity of the node attributes (e.g., color) rather than spatial coordinates alone. These coherent nodes can be treated as a union of their underlying finer-level nodes, and the energy can be directly accumulated from level n to $n+1$. A label change at a coarser-level node is equivalent to altering all its underlying finest-level nodes, which triggers a large yet reasonable change in the overall energy.

Adaptive-hierarchy based energy minimization algorithms, such as [10, 13], modify the hierarchy dynamically to reflect the change in labels and energies. Experiments on a wide range of vision problems have demonstrated the aptness of adaptive hierarchy-based energy minimization methods, and have consistently reported convergence time within a few seconds on typically-sized images. These methods, although efficient, give rise to a new class of *normalization* problems that were nonexistent or insignificant on flat MRFs and traditional hierarchies. Nodes on any given level of a traditional hierarchy/flat MRF are of the same size, therefore, a constant set of weights (at each level) suffices to balance the influence of different energy terms. However, nodes in adaptive hierarchies grow at different speeds as the hierarchy is coarsened; therefore, different weights need to be learned for nodes of different sizes/shapes. Directly learning all of these weights is not possible. In this paper, we theoretically analyze the effects of these

normalization problems. To overcome them, we experimentally design three normalization schemes and analyze them. Our analysis indicates that a further speedup of 15% on average is possible when properly normalized hierarchies are used.

The remainder of the paper is organized as follows. We discuss the mechanism of an adaptive hierarchy-based energy minimization method—Graph-Shifts—in Section 2. We then analyze the theoretical upper and lower bounds and expected value of the energy terms, investigate the effects of having energy terms growing at different rates, and propose our normalization solution in Section 3. Experiments and results are presented in Section 4, and we conclude in Section 5.

2 Review of Adaptive-hierarchy-based Energy Minimization

Our discussion on adaptive hierarchy-based energy minimization is focused on the recently proposed graph-shifts algorithm [10], due to its efficiency and its inseparable relation with adaptive-hierarchies. In short, the graph-shifts algorithm is composed of two steps: (a) *coarsening*, i.e. the construction of the adaptive hierarchy, and (b) *shifting*, i.e. the dynamic modification of the hierarchical structure for energy minimization.

2.1 The Energy Model

Given an input image $\mathbf{I}$, each pixel corresponds to a node μ^0 in a regular lattice G^0. The superscript 0 indicates that we are at the lowest level of a to-be-defined hierarchy. Associate with each node a label variable m_{μ^0} that takes values from a fixed set of labels $\{\mathcal{L}_1, \mathcal{L}_2, ..., \mathcal{L}_K\}$. The task is to find the assignment of all label variables $\{m_{\mu^0}\}$ that yields the highest posterior probability (MAP-MRF), or equivalently, minimizes the following energy function:

$$E\left[\left\{m_{\mu^0} : \mu^0 \in G^0\right\}\right] = \lambda_1 \sum_{\mu^0 \in G^0} E_1\left(\mathbf{I}\left(S\left[\mu^0\right]\right), m_{\mu^0}\right) + \lambda_2 \sum_{\langle \mu^0, \nu^0 \rangle} E_2\left(m_{\mu^0}, m_{\nu^0}\right) \quad . \tag{1}$$

E_1 (unary energy) is the potential of each node being assigned a certain label, defined on the local sub-image $S\left[\mu^0\right]$ surrounding μ^0. E_2 (binary energy) is induced by the interaction between neighboring nodes, where $\langle \mu^0, \nu^0 \rangle$ denotes all neighbor pairs μ^0 and ν^0; λ_i's are the weights of the different energy terms, where $\sum_i \lambda_i = 1$.

We use the same energy model for E_1 as in [10] to ease direct comparison between unnormalized and normalized energies in adaptive hierarchies. In short, the E_1 term is the negative log probability on the local subimage surrounding μ trained via boosting in a supervised manner:

$$E_1\left(\mathbf{I}\left(S\left[\mu^0\right]\right), m_{\mu^0}\right) = -\log \Pr\left(m_{\mu^0} \mid \mathbf{I}\left(S\left[\mu^0\right]\right)\right) \quad . \tag{2}$$

A pairwise smoothness measurement is used for the E_2 term:

$$E_2\left(m_{\mu^0}, m_{\nu^0}\right) = 1 - \delta\left(m_{\mu^0}, m_{\nu^0}\right) \quad . \tag{3}$$

For evaluating normalization effects, these choices for the energy terms are largely arbitrary. As we will show in the remainder of the paper, the normalization problems are a function of the graph structure and not the form of the energy terms. For this paper, we choose high-level semantic image labeling as the main comparative problem.

2.2 Coarsening the Adaptive Hierarchy

The adaptive hierarchy is defined as a graph G with a set of nodes and a set of edges stratified on multiple levels of an hierarchy. All nodes and edges in level n can be viewed as a separate graph, denoted as G^n, and a node at level n is denoted as μ^n. As above, the lowest level of the hierarchy is essentially a lattice G^0 of regular sites (nodes) μ^0, and two nodes are linked with an edge if they are neighbors on the lattice.

Coarser-level nodes are computed recursively and stochastically as follows. Edges on G^0 are randomly turned on or off based on the local affinity. The *on* edges induce a connected components clustering; the clusters become nodes in the next coarse layer in the hierarchy. The structure of the coarsened adaptive hierarchy is constrained by a coarsening threshold τ_1 that limits the maximum number of nodes in a group, and an affinity threshold τ_2 that restricts un-similar nodes from joining. Any two nodes μ^{n+1}, ν^{n+1} (at level $n+1$) are connected by an edge if any two of their children (at level n) are connected. The nodes are recursively coarsened until the size of graph G^n at level n is within a pre-specified range of the number of labels.

A label layer $G^{\mathcal{L}}$ that contains a single node per label is attached on top of the highest layer of the current hierarchy G^T. Each node in G^T becomes a child of a node in $G^{\mathcal{L}}$ which it best fits (based on E_1), then takes the label it represents; each node in $G^{\mathcal{L}}$ has at least one child in G^T. The nodes in G are constrained to have a single parent except the nodes in $G^{\mathcal{L}}$ (which have no parents), and to have at least one child except for the ones in G^0 (which have no children). Since each node in $G \setminus G^{\mathcal{L}}$ has only one parent and can trace its ancestry back to a single node in $G^{\mathcal{L}}$, it will take the same label as its parent and ancestors (a.k.a. *parent-label constraint*), and an instance of the graph G is equivalent to a labeled segmentation $\left\{m_{\mu^0} : \mu^0 \in G^0\right\}$ of the image.

2.3 Recursive Computation of the Energy

For any node μ^n, let $P(\mu^n)$ be its parent, $C(\mu^n)$ be the set of its children, $A^l(\mu^n)$ denote its ancestor at level l, and $D^l(\mu^n)$ be the set of its descendants at level l. By construction, any coarser-level node μ^n's pixel-level descendants all belong to the same label. Therefore, μ^n is treated as a union of $\mu_i^0 \in D^0(\mu^n)$, and the overall energy is accumulated recursively throughout the hierarchy.

The unary term for assigning a label m_μ to a node μ is defined recursively as

$$E_1\left(\mu^n, m_{\mu^n}\right) = \begin{cases} E_1\left(\mathbf{I}\left(S\left[\mu^n\right]\right), m_{\mu^n}\right) & \text{if } n = 0 \\ \sum_{\mu^{n-1} \in C(\mu^n)} E_1\left(\mu^{n-1}, m_{\mu^{n-1}}\right) & \text{otherwise} \end{cases} \tag{4}$$

where $E_1\left(\mathbf{I}\left(S\left[\mu^n\right]\right), m_{\mu^n}\right)$ is defined in (2). The binary energy between two neighboring nodes μ, ν at the same level n, with labels m_μ and m_ν is defined as:

$$E_2\left(\mu^n, \nu^n, m_{\mu^n}, m_{\nu^n}\right) = \begin{cases} E_2\left(m_{\mu^n}, m_{\nu^n}\right) & \text{if } n = 0 \\ \sum_{\substack{\mu^{n-1} \in C(\mu^n) \\ \nu^{n-1} \in C(\nu^n) \\ \langle \mu^{n-1}, \nu^{n-1} \rangle}} E_2\left(\mu^{n-1}, \nu^{n-1}, m_{\mu^{n-1}}, m_{\nu^{n-1}}\right) & \text{otherwise} \end{cases} \tag{5}$$

where $E_2\left(m_{\mu^n}, m_{\nu^n}\right)$ is defined in (3). The overall energy specified in (1) is for level 0 of the hierarchy, however, it can be computed at any level n as:

$$E\left[\{m_{\mu^n} : \mu^n \in G^n\}\right] = \sum_{\mu^n \in G^n} E_1\left(\mu^n, m_{\mu^n}\right) + \sum_{\langle \mu^n, \nu^n \rangle} E_2\left(\mu^n, \nu^n, m_{\mu^n}, m_{\nu^n}\right) \quad . \tag{6}$$

Note that we've intentionally dropped the weights on the terms to avoid confusion since they will take different forms depending on our normalization scheme (Section 3).

2.4 Graph-Shifts

Graph-Shifts minimizes the energy by using a mechanism called a *shift*, which dynamically alters the adaptive hierarchy during energy minimization. A basic *shift* is the process of a node μ^n changing its parent to ν^n's parent, where $\langle \mu^n, \nu^n \rangle$. Due to the *parent-label constraint*, μ^n and all its descendants $D^{l<n}(\mu^n)$ will change their label to $P(\nu^n)$'s, thus cause a relabeling at G^0 and a change in total energy. Refer to [10] for more details regarding all types of *shifts* a node can perform.

Potential shifts are evaluated by their *shift-gradients*, which are computed efficiently using the recursive formulae in (4), (5). For a node μ^n shifting from label m_{μ^n} to label $\hat{m}_{\mu^n}$, the shift-gradient is

$$\begin{aligned} \Delta E\left(m_{\mu^n} \rightarrow \hat{m}_{\mu^n}\right) &= E_1\left(\mu^n, \hat{m}_{\mu^n}\right) - E_1\left(\mu^n, m_{\mu^n}\right) \\ &+ \sum_{\langle \mu^n, \nu^n \rangle} \left[E_2\left(\mu^n, \nu^n, \hat{m}_{\mu^n}, m_{\nu^n}\right) - E_2\left(\mu^n, \nu^n, m_{\mu^n}, m_{\nu^n}\right)\right] \quad . \end{aligned} \tag{7}$$

We go through all nodes in $G \setminus G^{\mathcal{L}}$, calculate possible shifts, and only those with $\Delta E < 0$ are added to our list of potential shifts $\mathcal{S}$.

At each round, Graph-Shifts chooses the shift with the steepest shift-gradient in $\mathcal{S}$ and makes the corresponding shift in the hierarchy. For the nodes affected by the shift, not only are their labels changed, but also their energies re-computed, possible shifts and shift-gradients re-calculated, and $\mathcal{S}$ updated. The algorithm is repeated until convergence, when no further shift will reduce the overall energy anymore (i.e. $\mathcal{S}$ becomes empty). Notice that, although higher-level shifts tend to induce larger energy changes, lower-level shifts might as well cause large energy changes and be executed before higher-level shifts.

3 Proper Energy Normalization in Adaptive Hierarchies

Usually, in MRFs, there is no need to normalize E_1 and E_2 individually in (1), since the weights λ_1, λ_2 are learned to optimize the labeling result. Traditional hierarchy's coarser-level nodes are also exempt from this concern, because the nodes are incoherent and requires a recalculation of E_1 and E_2 (i.e. they cannot accumulate finer-level energies for their own use). Adaptive hierarchies overcome the need of energy recalculation at coarser levels, yet the way it accumulates energies from finer-level nodes gives rise to a normalization problem caused by the different growth rates of E_1 and E_2. In

the upcoming subsections, we prove the theoretical bounds and expected values of the two energy terms, discuss the effect of having unnormalized energies during the energy minimization process, and describe our proposed ways of normalizing the energy terms in the adaptive hierarchy.

3.1 E_1 Term Growth Rate Analysis

Due to the way energies are accumulated throughout the adaptive hierarchy (as in (4)), a coarser-level node μ^n's E_1 term is primarily determined by the number of pixel-level (level 0) nodes it represents; let $\mathcal{M}(\mu^n) = |D^0(\mu^n)|$ denote this number. The theoretical upper/lower bound and the expected value of $\mathcal{M}(\mu^n)$, and therefore its E_1, is mainly decided by the coarsening threshold τ_1. Let $\Psi_1^n$and Φ_1^n denote the maximum and minimum possible E_1 energy μ^n can possess, and let $\mathcal{E}_1^0$ be an unknown constant that represents the average E_1 energy at $\mu^0 \in D^0(\mu^n)$. $\mathcal{E}_1^0$ is constrained on the energy model we use, e.g. the implementation of (2) using a discrete set of probability values $P(\cdot) = \{0, \frac{1}{255}, ..., \frac{254}{255}, 1\}$ with $\log(0)$ set to a finite number larger than $\log(\frac{1}{255})$ defines $\Psi_1^0 = 0 \leq \mathcal{E}_1^0 \leq 5.6 = \Phi_1^0$. For levels other than the lowest one, the upper bound of μ^n's E_1 energy, Ψ_1^n, grows exponentially as we go up the hierarchy:

$$\Psi_1^n = \max_{\mu^n} E_1(\mu^n, m_{\mu^n}) = \max_{\mu^n} \Psi_1^0 \cdot \mathcal{M}(\mu^n) = \Psi_1^0 \cdot (\tau_1)^n \quad . \tag{8}$$

The lower bound of μ^n's E_1 energy, Φ_1, is

$$\Phi_1^n = \min_{\mu^n} E_1(\mu^n, m_{\mu^n}) = \min_{\mu^n} \Phi_1^0 \cdot \mathcal{M}(\mu^n) = \Phi_1^0 \cdot (1)^n = \Phi_1^0 \quad . \tag{9}$$

As for the expected value of E_1 (denoted as $\mathbb{E}(\cdot)$), we treat the number of finer-level nodes that form a coarser-level node as a random variable $X \in \{1, 2, ..., \tau_1\}$ with probability $P_X(x) = 1/\tau_1$ for all x:

$$\mathbb{E}(X) = \sum_{X=1}^{\tau_1} \frac{1}{\tau_1} X \quad = \frac{1}{\tau_1} \sum_{X=1}^{\tau_1} X = \frac{1}{\tau_1} \cdot \frac{(\tau_1 + 1)\,\tau_1}{2} = \frac{\tau_1 + 1}{2} \quad . \tag{10}$$

Therefore, for any node μ^n, its expected E_1 energy is:

$$\mathbb{E}\left(E_1(\mu^n, m_{\mu^n})\right) = \mathbb{E}\left(\mathcal{E}_1^0 \cdot \mathcal{M}(\mu^n)\right) = \mathcal{E}_1^0 \cdot \left(\frac{\tau_1 + 1}{2}\right)^n \quad , \tag{11}$$

which still grows exponentially as we go up the hierarchy.

3.2 E_2 Term Growth Rate Analysis

The upper/lower bound and expected value of a node's *local* E_2 *energy* (sum of all E_2 on μ^n's edges, denoted as $E_2'(\mu^n)$), are conditioned not only on the number of pixel-layer nodes $\mathcal{M}$, but also on how the nodes are distributed. For any two neighboring nodes μ^0 and ν^0, the energy on their common edge is neglected after they are grouped into a coarser-level node μ^n, as defined in (5). Thus, $E_2'(\mu^n)$ is not determined by all

edges of all $\mu^0 \in D^0(\mu^n)$, but only on edges of all $\mu^0 \in D^0(\mu^n)$ where $\langle \mu^0, \nu^0 \rangle$, $\nu^0 \notin D^0(\mu^n)$. We define the size of this set as $\mathcal{N}$:

$$\mathcal{N}(\mu^n) = \sum_{\substack{\mu^0 \in D^0(\mu^n) \\ s.t.\ \langle \mu^0, \nu^0 \rangle,\ \nu^0 \notin D^0(\mu^n)}} \left(1 - \delta(m_{\mu^0}, m_{\nu^0})\right) . \tag{12}$$

$\mathcal{N}(\mu^n)$ is analogous to the perimeter of the object formed by all $\mu^0 \in D^0(\mu^n)$, while $\mathcal{M}(\mu^n)$ is analogous to the area it occupies (on a 2D lattice). Therefore, $E_2'(\mu^n)$ is constrained on $\mathcal{N}(\mu^n)$, while the relationship between $\mathcal{N}(\mu^n)$ and $\mathcal{M}(\mu^n)$ is dependent on the object's shape. This definition allows us to discover, when given a coarser-level node μ^n with a fixed $\mathcal{M}(\mu^n)$, the possible values of $\mathcal{N}(\mu^n)$ and therefore derive the upper/lower bound and expected value of a node's E_2' from $\mathcal{M}(\mu^n)$.

Let Ψ_2^n and Φ_2^n denote the upper and lower bound of $E_2'(\mu^n)$ at level n, and let $\mathcal{E}_2^0$ be the average E_2 energy of a single edge at level 0: $\langle \mu^0, \nu^0 \rangle$, where $\mu^0 \in D^0(\mu^n)$ and $\nu^0 \notin D^0(\mu^n)$. The bounds of $\mathcal{E}_2^0$ are dependent on the energy model used, e.g. (3) would yield $\mathcal{E}_2^0 = \{0, 1\}$, and are again denoted Ψ_2^0 and Φ_2^0. From basic geometry, given any $\mathcal{M}(\mu^n)$, $\mathcal{N}(\mu^n)$ is largest when its $D^0(\mu^n)$ are aligned linearly as in Fig. 2.(a), where $\mathcal{N}(\mu^n) = 2\mathcal{M}(\mu^n) + 2$. Therefore, in theory, Ψ_2^n can increase exponentially at the same speed of Ψ_1^n.

$$\begin{aligned}\Psi_2^n &= \max_{\mu^n} E_2'(\mu^n) = \max_{\mu^n} \sum_{\langle \mu^n, \nu^n \rangle} E_2(\mu^n, \nu^n, m_{\mu^n}, m_{\nu^n}) \\ &= \max_{\mu^n} \Psi_2^0 \cdot \mathcal{N}(\mu^n) = \max_{\mu^n} \Psi_2^0 \cdot 2\mathcal{M}(\mu^n) + 2 = \Psi_2^0 \cdot 2(\tau_1)^n + 2 .\end{aligned} \tag{13}$$

$\mathcal{N}(\mu^n)$ is smallest when the shape formed by all its $D^0(\mu^n)$ is near circle/square (depending on the neighborhood system used), as in Fig. 2.(b)-(d). Although $\mathcal{N}(\mu^n) = 4(\mathcal{M}(\mu^n))^{1/2}$, i.e. $\mathcal{N}$ grows only at a fraction of $\mathcal{M}$'s speed, notice that $\mathcal{N}(\mu^n) > \mathcal{M}(\mu^n)$ while $\mathcal{M}(\mu^n) < 16$.

$$\begin{aligned}\Phi_2^n &= \min_{\mu^n} E_2'(\mu^n) = \min_{\mu^n} \sum_{\langle \mu^n, \nu^n \rangle} E_2(\mu^n, \nu^n, m_{\mu^n}, m_{\nu^n}) \\ &= \min_{\mu^n} \Phi_2^0 \cdot \mathcal{N}(\mu^n) = \min_{\mu^n} \Phi_2^0 \cdot 4(\mathcal{M}(\mu^n))^{\frac{1}{2}} = \Phi_2^0 \cdot 4(1)^{\frac{n}{2}} = 4\Phi_2^0 .\end{aligned} \tag{14}$$

The expected value of $E_2'(\mu^n)$ requires the knowledge of the mean shape of all nodes at G^n. We estimate E_2' by using a rectangular-shaped node with variable length l and width αl ($\alpha \in \mathbb{R}^+$) to approximate any shape a nodes can possess. Therefore, for any node, $\mathcal{M} = \alpha l^2$, $\mathcal{N} = 2(\alpha l + l)$, $\mathcal{N} = 2(\alpha + 1)(\mathcal{M}/2)^{1/2}$, and its expected E_2':

$$\begin{aligned}\mathbb{E}(E_2'(\mu^n)) &= \mathbb{E}\left(\sum_{\langle \mu^n, \nu^n \rangle} E_2(\mu^n, \nu^n, m_{\mu^n}, m_{\nu^n})\right) = \mathbb{E}\left(\mathcal{E}_2^0 \cdot \mathcal{N}\right) \\ &= \mathcal{E}_2^0 \cdot 2(\alpha + 1)\left(\frac{\mathcal{M}}{2}\right)^{\frac{1}{2}} = \mathcal{E}_2^0 \cdot \sqrt{2}(\alpha + 1)\left(\frac{\tau_1 + 1}{2}\right)^{\frac{n}{2}} .\end{aligned} \tag{15}$$

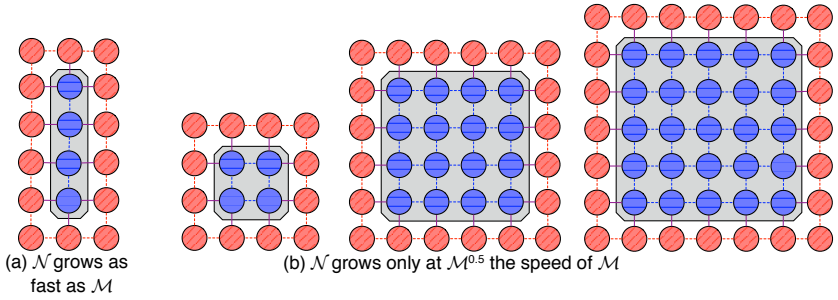

Fig. 2. $\mathcal{N}(\mu^n)$ versus $\mathcal{M}(\mu^n)$ under different shapes. Each node is a pixel-level node, the gray bevelled rectangle represents the range of the coarser-level node μ^n, and blue nodes (textured with horizontal lines for monochrome prints) indicate $\mu^0 \in D^0(\mu^n)$. Solid lines are the edges between nodes of different labels.

Empirically, α is a relatively small constant. Therefore, $\mathbb{E}(E_2'(\cdot))$ is only a fraction of $\mathbb{E}(E_1(\cdot))$ at the same level of the adaptive hierarchy.

3.3 The Effects of Un-Normalized Energies in Adaptive Hierarchies

As we have shown in 3.1 and 3.2, for any node μ^n, the relationship between its expected E_1 and E_2' is

$$\mathbb{E}\left(E_1\left(\mu^n, m_{\mu^n}\right)\right) \approx \mathbb{E}\left(E_2'\left(\mu^n\right)\right)^2 \approx \left(\left(\tau_1+1\right)/2\right)^n \quad . \tag{16}$$

In other words, μ^n's E_1 is expected to be $\left(\left(\tau_1+1\right)/2\right)^{n/2}$ times as large as its E_2' energy. Let $E'(\mu^n)$ be the local energy cached at node: μ^n, $E'(\mu^n) = E_1(\mu^n, m_{\mu^n}) + E_2'(\mu^n)$). The difference between $\mathbb{E}(E_1(\mu^n, m_{\mu^n}))$ and $\mathbb{E}(E_2'(\mu^n))$ becomes much more significant at coarser-level nodes, which would eventually cause the E_1 term to dominate $E'(\mu^n)$. While considered with their respected weights λ_1, λ_2 learned from the pixel-layer standard MRF

$$\begin{aligned}\mathbb{E}\left(E'\left(\mu^n\right)\right) &= \lambda_1 \mathbb{E}\left(E_1\left(\mu^n, m_{\mu^n}\right)\right) + \lambda_2 \mathbb{E}\left(E_2'\left(\mu^n\right)\right) \\ &\approx \lambda_1 \mathbb{E}\left(E_1\left(\mu^n, m_{\mu^n}\right)\right) + \lambda_2 \mathbb{E}\left(E_1\left(\mu^n, m_{\mu^n}\right)\right)^{1/2}\end{aligned} \tag{17}$$

$\lambda_2 \mathbb{E}(E_2'(\mu^n))$ becomes negligible as n increase. In other words, no matter what value λ_2 is assigned while it is learned, it has little effect on the final local energy of a coarser-level node.

The effects of this phenomenon are two-fold: (a) coarser-level nodes in the hierarchy are less likely to change their labels even when they are spatially inconsistent with their neighboring nodes, and therefore (b) *shifts* are more likely to happen at finer-level nodes. One might argue that the first effect is desirable, due to the intuition that coarser-level nodes are more likely to represent individual objects such as a ball or a box, thus being able to easily change their labels would be unreasonable. However, there is

no guarantee that the likelihood term is reliable (i.e. we might have bad coarser-label nodes at the beginning). Furthermore, even if we were to design an adaptive hierarchy with different weights λ_1, λ_2 for different levels, it should be learned instead of being adjusted by the coarsening factor. One notable drawback of the first effect is that the algorithm is more likely to be trapped in local minima, which contradicts with one of the original design goal of the Graph-Shifts algorithm. The second effect tends to increase the total number of shifts, since more finer-level shifts are needed to accomplish the same energy-change as one coarser-level shift.

3.4 Normalizing the Energy Terms

We experimentally design three strategies for normalizing E_1 and E_2' for nodes in the Graph-Shifts hierarchy: (a) energies normalized only at level 0 of the hierarchy, denoted as EN0, (b) energies normalized by a constant, denoted as ENC, and (c) energies normalized with the node mass ($\mathcal{M}$), denoted as ENM. We will experimentally analyze them, along with the unnormalized version, denoted as UNE, in 4.

The only difference between version (a) (EN0) and unnormalized energies (UNE) is that: at level 0 of the hierarchy, the output of E_1 and E_2 are normalized to the interval $[0, 1]$

$$\begin{aligned} \Omega_1\left(\mathbf{I}\left(S\left[\mu^0\right]\right), m_{\mu^0}\right) &= E_1\left(\mathbf{I}\left(S\left[\mu^0\right]\right), m_{\mu^0}\right)/\Psi_1^0 \ , \\ \Omega_2\left(m_{\mu^0}, m_{\nu^0}\right) &= E_2(m_{\mu^0}, m_{\nu^0})/\Psi_2^0 \ . \end{aligned} \tag{18}$$

For comparison, high-level energies defined in (2), (3) causes E_1 to output discrete values in the interval $[0, 5.6]$, and $E_2 \in \{0, 1\}$. Note that coarser-level nodes are still unnormalized. Since the optimum values for λ_1, λ_2 are learned for different energies, we show in 4 that this normalization have no effect on convergence speed or labeling accuracy. Proving this equivalence facilitates the normalization of coarser-level energies.

Version (b) (ENC) normalizes all nodes' E_1 and E_2' to the interval $[0, 1]$ as

$$\begin{aligned} \Omega_1\left(\mu^n, m_{\mu^n}\right) &= E_1\left(\mu^n, m_{\mu^n}\right) / \left(\Psi_1^0 \cdot \mathcal{M}\left(\mu^n\right)\right) \ , \\ \Omega_2\left(\mu^n\right) &= E_2'\left(\mu^n\right) / \left(\Psi_2^0 \cdot \mathcal{N}\left(\mu^n\right)\right) \ . \end{aligned} \tag{19}$$

The unnormalized energies E_1, E_2 are still used for the recursive computation of coarser-level energies. This is because normalizing the terms to $[0, 1]$ is just for balancing their effects on the final E, not for reweighting all finer-level nodes when they are grouped into a coarser-level one. Notice that, since all nodes in the hierarchy are normalized to the same interval $[0, 1]$, a coarser-level label-change no longer tends to cause a larger energy change. In other words, instead of favoring coarser-level shifts at the early rounds of the Graph-Shifts algorithm, it will give equal preference to finer- and coarser-level shifts. This characteristic is both an advantage and disadvantage, in the sense that local minima have a higher chance of being avoided, yet it is expected to take a much longer time to converge.

The third and final normalizing scheme, (c), (ENM) overcomes the unreasonable fairness between finer- and coarser-level shifts induced by version (b). The energies

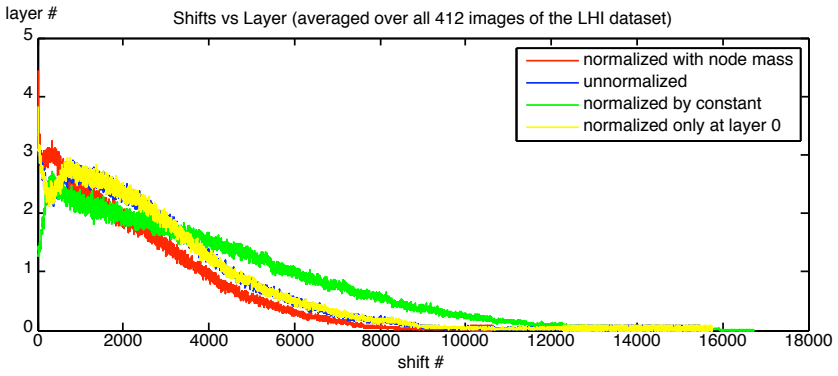

Fig. 3. Shifts versus the layer it occurred. For every labeling task, if it takes z shifts to converge, the layer at which the shift number (shift #) $y \leq z$ takes place is averaged over all labeling tasks where y occurred.

Ω_1 and Ω_2' are normalized to the same interval, so that the final energy term Ω is not dominated by a single energy term, yet higher-level nodes preserve their original tendency of causing a larger energy change.

$$\begin{aligned} \Omega_1\left(\mu^n, m_{\mu^n}\right) &= E_1\left(\mu^n, m_{\mu^n}\right) / \Psi_1^0 \ , \\ \Omega_2\left(\mu^n\right) &= \frac{E_2'\left(\mu^n\right)}{\Psi_2^0 \cdot \left(\mathcal{M}\left(\mu^n\right) / \mathcal{N}\left(\mu^n\right)\right)} \ . \end{aligned} \tag{20}$$

4 Experiments and Results

Our experiments are conducted on an 11-label, 412-image subset of the LHI dataset [14]. We randomly split the data into training and testing sets, where 170 of them were used for training and 242 of them for testing. We trained the PBT classifier [15] to select and fuse features from a set of 10^5 features, consisting of color, intensity, position, histograms, and Gabor filter responses. The pair-wise contextual relationships between different labels are also learned to construct the PBT classifier. For a test image, the probability of each pixel belonging to one of the 11 labels are computed from the PBT classifier, then formulated into the E_1 term using (2). We then compare the energy-minimization effects of the Graph-Shifts algorithm using unnormlized energies (UNE) versus the three versions of normalized energies (EN0, ENC, ENM defined in (18), (19), and (20) respectively). The hierarchy coarsening parameters are empirically set to $\tau_1 = 20$ and $\tau_2 = 0.5$ in all experiments. The optimum weights are learned, where UNE's $\lambda_1 = 0.1, \lambda_1 = 0.9$, EN0's $\lambda_1 = 0.3, \lambda_2 = 0.7$, ENC and ENM's $\lambda_1 = 0.6, \lambda_2 = 0.4$.

Our results show a significant improvement in convergence speed when coarser-level energies are normalized properly (using ENM). The average number of *shifts* required while using ENM is 3774, versus 4110 shifts for UNE; the average convergence time is 3.17 seconds versus 3.71 seconds, which is a 15% speedup. This result is expected, because unnormalized energy causes the E_1 term to dominate E, which

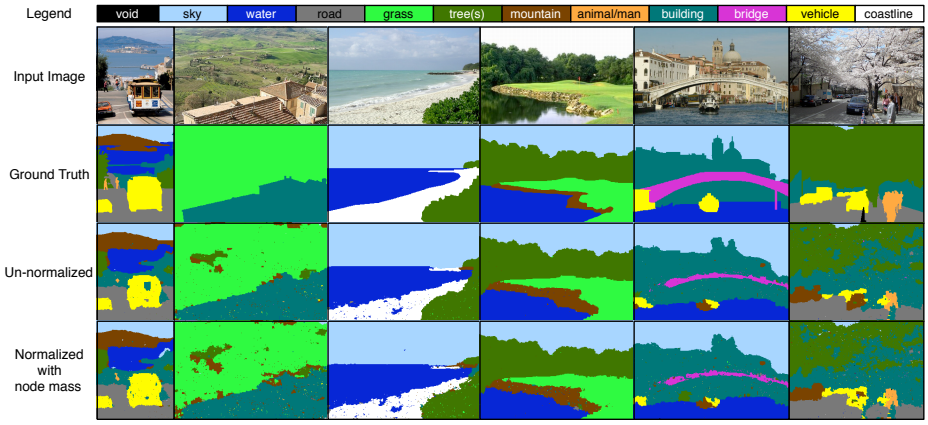

Fig. 4. Randomly selected labeling results. ENM converges to results similar to those of UNE in only 85% of time. The details are slightly different due to the different orders *shifts* take place while approximating the global minimum.

discourages higher-level shifts from taking place. Our analysis in Fig. 3 shows that, the properly normalized version (red line) not only converges faster, but also induces more coarser-level shifts (which tend to cause larger energy change) at earlier stages of the energy minimization process. Energies normalized only at level 0 (EN0, yellow line), exhibits almost the same characteristic as the unnormalized version (UNE, blue line). One interesting point is, while equally weighting all nodes at different levels of the hierarchy (ENC, green line), finer-level shifts are more likely to happen early during the energy minimization process, causing the algorithm to require more shifts (5740) and take 1.5 times as long (4.31 seconds) to converge.

Interestingly, however, UNE, EN0, ENC, ENM achieves almost the same labeling accuracy of $70\% \pm 0.5\%$. This is because the pixel-level MRF energy models are essentially the same for UNE, EN0, ENC, and ENM, as shown in 3.4, therefore should converge to similar energy-minimized results. Their difference in coarser-level energy accumulation only affects the level and order at which *shifts* takes place (Fig. 4). In ENM, coarser-level shifts happen more frequently at earlier stages of the energy minimization process, thus in some sense optimizes the shifting sequence. In UNE, since coarser-level shifts are unreasonably oppressed, groups of finer-level shifts have to be performed to achieve the same pixel-level label change.

5 Conclusion

In summary, this paper has investigated the theory and construction of adaptive hierarchies, then has examined the potential problems of using it to perform energy minimization without proper normalization. The recursive energy accumulation of adaptive hierarchies causes unnormalized energy terms to grow at different speeds, thus resulting in the faster growing terms to dominate the final energy in coarser-level nodes. Empirically, the unary energy outweighs the binary energy at coarser-level nodes, which makes

coarser-level shifts less likely to occur, therefore increasing the total number of shifts required for minimization. We designed three different methods for normalizing coarser-level energies, and experimentally confirmed that the best results are achieved when *the different energy terms of a node are normalized to the same interval, while coarser-level nodes still possess relatively larger energies compared to finer-level nodes*. Properly normalized energies triggers a 15% speedup in convergence time while maintaining the same accuracy rate. We plan to further justify our findings in the future by experimenting on other types of energy models, along with looking into the effects of proper normalization in other types of hierarchical algorithms.

Acknowledgements This work has been partly supported by NSF CAREER grant IIS 0845282 and DARPA grant HR0011-09-1-0022.

References

1. Besag, J.: Spatial interaction and the statistical analysis of lattice systems (with discussion). Journal of the Royal Statistical Society, B **36** (1974) 192–236
2. Boykov, Y., Veksler, O., Zabih, R.: Fast Approximate Energy Minimization via Graph Cuts. IEEE Trans. Pattern Analysis and Machine Learning (2001) 1222–1239
3. Geman, S., Geman, D.: Stochastic Relaxation, Gibbs Distributions, and Bayesian Restoration of Images. IEEE Trans. on Pattern Analysis and Machine Intelligence **6** (1984) 721–741
4. Yedidia, J., Freeman, W., Weiss, Y.: Generalized Belief Propagation. Advanced in Neural Information Processing Systems (2001) 689–695
5. Bouman, C., Liu, B.: Multiple resolution segmentation of textured images. IEEE Trans. on Pattern Analysis and Machine Intelligence **13**(2) (1991) 99–113
6. Bouman, C., Shapiro, M.: A multiscale random field model for Bayesian image segmentation. IEEE Trans. on Image Processing **3**(2) (1994) 162–177
7. Gidas, B.: A renormalization group approach to image processing problems. IEEE Trans. on Pattern Analysis and Machine Intelligence **11**(2) (1989) 164–180
8. Kato, Z., Berthod, M., Zerubia, J.: Multiscale Markov random field models for parallel imageclassification. In: Proc. of Fourth International Conference on Computer Vision. (1993)
9. Terzopoulos, D.: Image analysis using multigrid relaxation methods. IEEE Trans. on Pattern Analysis and Machine Intelligence **8**(2) (1986) 129–139
10. Corso, J.J., Yuille, A., Tu, Z.: Graph-Shifts: Natural Image Labeling by Dynamic Hierarchical Computing. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition. (2008)
11. Sharon, E., Brandt, A., Basri, R.: Fast Multiscale Image Segmentation. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition. Volume I. (2000) 70–77
12. Ahuja, N.: A transform for multiscale image segmentation by integrated edge and region detection. IEEE Trans. on Pattern Analysis and Machine Intelligence **18**(12) (1996) 1211–1235
13. Chen, A.Y.C., Corso, J.J., Wang, L.: HOPS: Efficient region labeling using higher order proxy neighborhoods. In: Proc. of International Conference on Pattern Recognition. (2008)
14. Yao, Z., Yang, X., Zhu, S.C.: Introduction to a Large Scale General Purpose Ground Truth Dataset: Methodology, Annotation Tool, and Benchmarks. In: Proc. of Int'l Conf. on Energy Minimization Methods in Computer Vision and Pattern Recognition. (2007)
15. Tu, Z.: Probabilistic Boosting-Tree: Learning Discriminative Models for Classification, Recognition, and Clustering. In: Proc. of the Tenth IEEE Int'l Conf. on Computer Vision, 2005. Volume 2. (2005)