International Journal of Computer Vision, vol. 100, no. 1, pp. 38–58, October 2012

Minimizing Energies with Hierarchical Costs

Andrew Delong · Lena Gorelick · Olga Veksler · Yuri Boykov

Received: October 15, 2011 / Accepted: March 20, 2012

Abstract Computer vision is full of problems elegantly expressed in terms of energy minimization. We characterize a class of energies with *hierarchical costs* and propose a novel *hierarchical fusion* algorithm. Hierarchical costs are natural for modeling an array of difficult problems. For example, in semantic segmentation one could rule out unlikely object combinations via hierarchical context. In geometric model estimation, one could penalize the number of unique model *families* in a solution, not just the number of models—a kind of hierarchical MDL criterion. Hierarchical fusion uses the well-known α-expansion algorithm as a subroutine, and offers a much better approximation bound in important cases.

Keywords Energy minimization · Hierarchical models · Graph cuts · Markov random fields (MRFs) · Segmentation.

1 Introduction

Energy minimization is of strong practical and theoretical importance to computer vision. An energy expresses our criteria for a good solution—low energies are good, high energies are bad—independent of any algorithm. Algorithms are however hugely important in practice. Even for low-level vision problems we are confronted by energies that are computationally hard (often NP-hard) to minimize. As a consequence, a significant portion of computer vision research is dedicated to identifying energies that are useful and yet reasonably tractable. Our work is of precisely this nature.

Computer vision is full of 'labeling' problems cast as energy minimization. For example, the data to be labeled could be pixels, interest points, point correspondences, or mesh data such as from a range scanner. Depending on the application, the labels could be either semantic (object classes, types of tissue) or describe geometry/appearance (depth, orientation, shape, texture).

There are many labeling problems for which the labels naturally form groups. In computer vision, a recent trend is the use of 'context' to resolve ambiguities in object recognition (*e.g.* Choi et al., 2010; Ladický et al., 2010a; Zhou et al., 2011). The idea is that certain groups of labels are self-consistent because they tend to appear together, *e.g.* the {*car*,*road*,*sky*} labels all belong to the "outdoors" context, while {*table*,*chair*,*wall*} all belong to the "indoors" context. In computer graphics, one may wish to automatically classify the faces of a 3D mesh into semantic parts for the benefit of artists and animators (Kalogerakis et al., 2010). The part labels *arm*, *tail*, and *wheel* naturally belong to different groups based on their context (humanoid, quadruped, vehicle). In operations research, *facility location* can be cast as a labeling problem, and hierarchical variants have been studied (Svitkina and Tardos, 2006; Sahin and Süral, 2007). All of these disparate labeling problems are similar from an optimization point of view.

When labels are explicitly grouped in a hierarchy, the costs in the energy are naturally structured. In this work, we characterize a class of energies as having *hierarchical costs*. If an energy satisfies our "h-metric" and "h-subset" conditions, then we can often minimize it much more effectively. We provide a novel *hierarchical fusion* (h-fusion) algorithm to minimize our class of energies. Our algorithm generalizes the well-known α-expansion algorithm (Boykov et al., 2001) yet we provide better empirical performance and a

A. Delong · L. Gorelick · O. Veksler · Y. Boykov
Department of Computer Science, University of Western Ontario, London, Ontario, Canada N6A 5B7
E-mail: andrew.delong@gmail.com

L. Gorelick
E-mail: lena.gorelick@gmail.com

O. Veksler
E-mail: olga@csd.uwo.ca

Y. Boykov
E-mail: yuri@csd.uwo.ca

Table 1 A selection of relevant energy-based problem formulations in computer vision. All are a special case of energy (1), with the exception of Ladický et al. (2010a) which in principle allows for a wider class of energies that encourage 'parsimony'.

paper	V	H	algorithms	applications
Zhu and Yuille (1996)	semi-metric	per-label	region merging	unsupervised segmentation
Torr (1998)	$\times$	per-label	expectation maximization + pruning	model selection, motion estimation
Boykov et al. (2001)	metric, semi-metric	$\times$	α-expansion, $\alpha\beta$-swap	stereo, denoising
Kolmogorov (2006)	arbitrary	$\times$	tree-reweighted message passing	stereo
Li (2007)	$\times$	per-label	LP relaxation + rounding	motion estimation
Lazic et al. (2009)	$\times$	per-label	belief propagation	motion estimation
Kumar and Koller (2009)	r-HST metric	$\times$	hierarchical graph cuts	denoising, scene registration
Delong et al. (2010)	metric, semi-metric	any subsets	α-expansion, $\alpha\beta$-swap, greedy FL	homography detection, motion estimation, unsupervised segmentation
Barinova et al. (2010)	$\times$	per-label	greedy facility location (FL)	object detection
Ladický et al. (2010a)	metric, semi-metric	parsimonious*	α-expansion, $\alpha\beta$-swap	object recognition
this work	h-metric	h-subsets	h-fusion w/ α-expansion	unsupervised segmentation

better approximation bound. The improved theoretical guarantees are important because, in practice, α-expansion can easily get stuck in poor local minima for this useful class of energies; to the best of our knowledge, our h-fusion algorithm is state of the art. Like the original fusion algorithm Lempitsky et al. (2010) ours is highly parallelizable.

With respect to our energy itself, the most relevant work is the "label costs" of Delong et al. (2012). Our notion of *hierarchical costs* is a special case of their energy, yet it is important to explicitly characterize this subclass because, as we show, it permits a much better minimization algorithm. With respect to our algorithm, by far the most closely related work is the r-HST metrics of Kumar and Koller (2009). We review both these works in some detail.

2 Review of Related Work

First we review energies with "label costs" as described in Delong et al. (2012). Let the set $\mathcal{P}$ index the data that needs to be labeled, and let $\mathcal{L}$ be the set of possible labels. A *labeling* is any complete assignment $f = (f_p)_{p\in\mathcal{P}}$ where variable $f_p \in \mathcal{L}$ designates the label assigned at index p. For example if $\mathcal{P} = \{p, q\}$ and $\mathcal{L} = \{\ell_1, \ell_2\}$ then a valid labeling might be $f = (\ell_2, \ell_1)$ where $f_p = \ell_2, f_q = \ell_1$.

We seek joint assignment f that minimizes an energy balancing three types of criteria

$$E(f) \;=\; D(f) \;+\; V(f) \;+\; H(f). \tag{1}$$

The D term encodes the individual preference of each variable f_p, whereas the V term encourages pairs of variables to take similar values. These are typically expressed as

$$D(f) = \sum_{p\in\mathcal{P}} D_p(f_p) \qquad \textit{(data costs)}$$

$$V(f) = \sum_{pq\in\mathcal{N}} w_{pq} V(f_p, f_q) \qquad \textit{(smoothness costs)}$$

where the neighbourhood set $\mathcal{N}$ identifies pairs of variables f_p and f_q that interact, and each weight $w_{pq} \geq 0$ scales the strength of V for that particular pair. Energies with these terms are quite standard in vision, particularly in models based on *Markov random fields* (MRFs) and when performing *maximum a posteriori* (MAP-MRF) inference (Li, 1994).

The H term encourages the labeling f to use as few unique labels as is necessary. For example, do not explain the data with six labels if five would suffice just as well. The general form that we use in this work can be expressed as

$$H(f) = \sum_{L\in\mathcal{H}} H(L)\delta_L(f) \qquad \textit{(label costs)}$$

where $\mathcal{H}$ is any collection of subsets of $\mathcal{L}$, each $H(L)$ is the *label cost* of subset $L \subseteq \mathcal{L}$, and the label cost indicator $\delta_L(\cdot)$ is defined as

$$\delta_L(f) = \begin{cases} 1 & \text{if } \exists p : f_p \in L \\ 0 & \text{otherwise.} \end{cases}$$

Intuitively, $H(L)$ is the shared cost to be paid if f uses any labels drawn from label group L. The cost is *shared* because it is paid at most once, regardless of how many labels from L appear in f. In many applications there is no reason to group the labels, and so setting $\mathcal{H} = \{\{\ell\}\}_{\ell\in\mathcal{L}}$ restricts us to individual *per-label* costs $H(\{\ell\})$ or, with slight abuse of notation, setting $\mathcal{H} = \mathcal{L}$ and writing each per-label cost as $H(\ell)$. Label costs are an important special case of 'high-arity' potentials recently studied in computer vision (*e.g.* Werner, 2008; Woodford et al., 2009).

From an energy standpoint there are a number of works that combine data costs D with V and/or H. Table 1 lists a small selection of such works in computer vision. As can be seen from the V column, many techniques are only applicable if $V(\cdot,\cdot)$ satisfies certain assumptions.

Definition 1 A smoothness cost V is *semi-metric* if it satisfies

$$V(\alpha,\alpha) = 0 \qquad \forall \alpha \in \mathcal{L}$$
$$V(\beta,\alpha) = V(\alpha,\beta) \geq 0 \qquad \forall \alpha,\beta \in \mathcal{L}$$

and V is a *metric* if it additionally satisfies

$$V(\alpha,\beta) \leq V(\alpha,\gamma) + V(\gamma,\beta) \qquad \forall \alpha,\beta,\gamma \in \mathcal{L}.$$

These assumptions were originally outlined by Boykov et al. (2001) as sufficient conditions for their α-expansion and $\alpha\beta$-swap algorithms. These conditions arise because of the inherent limitations of graph cut methods (Boykov and Jolly, 2001; Kolmogorov and Zabih, 2004). Because our algorithmic approach is also based on graph cuts, we shall see a similar kind of limitation arise in Section 4.1.

We note that, with the exception of Zhu and Yuille (1996), all the works listed in Table 1 are of a discrete nature where $\mathcal{P}$ is a finite set. A number of *variational* formulations of E have recently been developed with continuous analogues of $D+V$ (*e.g.* Pock et al., 2008, 2009; Olsson et al., 2009) and of $D+V+H$ (Yuan and Boykov, 2010). Our main ideas also apply to such continuous formulations, but we focus on the discrete setting.

Energies of the form (1) are NP-hard to minimize in all but a few special cases. Even $D+V$ is NP-hard to minimize for $|\mathcal{L}| \geq 3$ by reduction from 3-TERMINAL-CUT (Boykov et al., 2001); in fact this case is max-SNP-hard (Cunningham and Tang, 1999), meaning there is some $\epsilon > 0$ for which no polynomial-time $(1+\epsilon)$-approximation algorithm can exist (*i.e.* no polynomial-time approximation scheme (PTAS)). The $D+H$ case is NP-hard by straight-forward reduction from SET-COVER using only per-label costs $H(\ell)$. A hardness result for approximating SET-COVER by Feige (1998) implies that $D+H$ cannot be approximated within a ratio of $(1-\epsilon)\ln|\mathcal{P}|$ in polynomial time unless the complexity class NP $\subseteq$ DTIME$[n^{O(\log\log n)}]$, *i.e.* NP would have to be only slightly super-polynomial, which is currently deemed unlikely. This observation will help to put the approximation bound for our algorithm into perspective.

Observation 1 *Feige's hardness result is evidence that no polynomial-time algorithm can minimize energy* (1) *within a constant ratio of the optimum.*

From an algorithmic standpoint, the most similar work to ours is a recent paper by Kumar and Koller (2009). Their aim is to efficiently minimize energies of the form $D+V$. They consider the class of *r-hierarchically well-separated tree* (r-HST) metrics (Bartal, 1998) which are a special case of *metrics* defined above. We discuss r-HST metrics in Section 8, but for now it is enough to understand that they are a special case of metrics and that each has an associated constant $r > 1$. Kumar and Koller provide an algorithm that, for a particular r-HST metric, provides an $\frac{2r}{r-1}$-approximation to the globally optimal labeling. Although this coefficient is very large for $r \approx 1$, the approximation only depends on V (not on $|\mathcal{P}|$ or $|\mathcal{L}|$). In some cases this ratio is better than the well-known bound for the α-expansion algorithm (Boykov et al., 2001) and the $O(\lg|\mathcal{L}|\lg\lg|\mathcal{L}|)$ bound for linear programming relaxation (Kleinberg and Tardos, 2002).

Kumar and Koller describe their algorithmic process as *hierarchical graph cuts*. This does not refer to computing a graph cut in a hierarchical manner, but rather to minimizing an energy $D+V$ via a hierarchical *sequence* of standard graph cuts. They show that the r-HST metric assumption on V is sufficient to apply their algorithm. We aim to minimize energies of the form $D+V+H$ and, motivated by the difficult H term, have independently developed an algorithm we call *hierarchical fusion* (h-fusion). However, at the highest conceptual level our algorithm is the same as theirs—it is only our class of energies and our sequence of subproblems that is different, each of which we solve with the extended α-expansion of Delong et al. (2012). The h-fusion process will be explained in Section 5.

Also worth mentioning is a recent work by Felzenszwalb et al. (2010) concerning energies of the form $E = D+V$. By making the strong assumption that *both* D and V are *tree metrics*, they can compute a global optimum. However, most applications do not satisfy the metric assumption on data costs D. Note that our work makes no such assumption. We discuss tree metrics in Section 4.1.

Our contributions This work is about characterizing a class of energies that will be useful in vision, along with a fast and effective algorithm for dealing with large-scale problems. Specifically, for energies of the form $D+V+H$,

a) we define *h-metric* smoothness costs V, a wider class than tree metrics yet still sufficient for our h-fusion algorithm to apply,
b) we define *h-subset* label costs H, a sufficient condition to apply h-fusion with high-order label costs,
c) we prove that the approximation bound of h-fusion is much better than α-expansion in important cases, and
d) we provide worst-case examples to show that our theoretical bound is tight in some reasonable sense.

Contribution (a) is about a more general characterization of V than that used by Kumar and Koller (2009) and by Felzenszwalb et al. (2010), while (b–d) are completely novel results on how to handle label costs in this framework.

The remainder of this paper is organized as follows. Section 3 reviews the α-expansion algorithm in some detail. Section 4 then introduces our notion of *hierarchical costs*, a useful subclass of energy (1). Section 5 describes our h-fusion algorithm, and Section 6 derives its approximation bound. Section 7 gives some experiments to suggest how our energies and algorithm work. Finally, Section 8 discusses other applications, relations to facility location, and extensions.

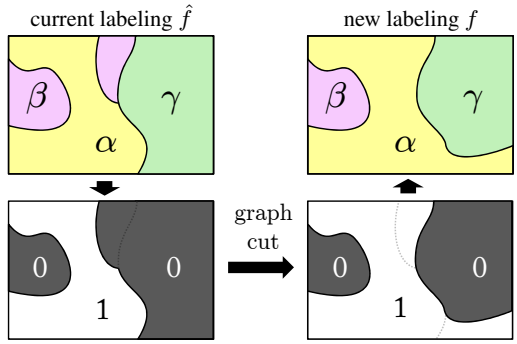


Fig. 1 In an α-expansion move, each location p is given a binary choice: keep its current label $\hat{f}_p$ or switch to label α. Here we use **1** to indicate a variable that decided to take label α. Because variables interact through V, the binary decisions are inter-dependent and so the optimal move is calculated by a graph cut.

3 Review of α-Expansion

The algorithm we introduce in this paper uses the well-known α-expansion algorithm (Boykov et al., 2001) as a key subroutine. The algorithm was designed for energies of the form $D + V$, though we employ an extension to $D + V + H$ by Delong et al. (2012). Our approximation bound is therefore intricately linked with α-expansion and its limitations, so we review the algorithm here. Readers familiar with α-expansion may skip ahead to Section 4.

3.1 How α-expansion works

The α-expansion algorithm performs local search using a powerful class of 'moves'. Given an initial labeling $\hat{f}$ and some particular label $\alpha \in \mathcal{L}$, an α-expansion move gives each variable the following binary choice: either keep the current label $\hat{f}_p$, or switch to label α. Let $\mathcal{M}^\alpha(\hat{f})$ denote the set of all moves (labelings) that can be generated this way, in other words

$$\mathcal{M}^\alpha(\hat{f}) = \big\{\, f \;:\; f_p \in \{\hat{f}_p\} \cup \{\alpha\} \,\big\}. \tag{2}$$

All variables are simultaneously allowed to keep their current label or to switch, so there are an exponential number of possible moves. For each choice of α we must efficiently find the best possible move. In practice, this sub-problem is solved by casting it as a *graph cut* (Greig et al., 1989) and using combinatorial algorithms to compute the optimal binary configuration (*e.g.* Goldberg and Tarjan, 1988; Boykov and Kolmogorov, 2004; Strandmark and Kahl, 2010). Figure 1 illustrates the steps behind a single expansion move. Because a graph cut finds the best move from an exponential number of possibilities, the α-expansion algorithm is a *very large-scale neighbourhood search* (VLSN) technique (Ahuja et al., 2002) and is very competitive in practice (Szeliski et al., 2006).

With respect to some current labeling $\hat{f}$, the full set of possible expansion moves is $\mathcal{M}(\hat{f}) = \bigcup_{\alpha\in\mathcal{L}} \mathcal{M}^\alpha(\hat{f})$. The α-expansion algorithm simply performs local search over the full search neighbourhood $\mathcal{M}(\hat{f})$. Perhaps surprisingly, local search with expansion moves will terminate with a labeling $\hat{f}$ that is within a constant factor from the globally optimal labeling f^* (see Section 3.3). The α-expansion algorithm is generally implemented as shown below.

α-EXPANSION(E) — local search with expansion moves

```
1 f̂ := arbitrary labeling
2 repeat
3     for each α ∈ L
4         f := argmin_{f ∈ M^α(f̂)} E(f)    // solve via graph cut
5         if E(f) < E(f̂)
6             f̂ := f
7 until converged
8 return f̂
```

3.2 Graph cuts and the limits of α-expansion

From Table 1 we see that α-expansion is applicable if V is a *metric* (Definition 1). We briefly review how this limitation arises, as it will be relevant to our new h-fusion algorithm.

The main subproblem for α-expansion (line **4**) is to find, for a particular label $\alpha \in \mathcal{L}$, the best move $f \in \mathcal{M}^\alpha(\hat{f})$. The best move is the one with minimal $E(f)$. Expansion moves are fundamentally binary so we can encode a move as a function $f(\mathbf{x})$ of binary vector $\mathbf{x} = (x_p)_{p\in\mathcal{P}}$ where

$$f(\mathbf{x})_p = \begin{cases} \hat{f}_p & \text{if } x_p = 0 \\ \alpha & \text{if } x_p = 1 \end{cases}$$

To solve the subproblem on line **4**, we can construct a binary energy $E'(\mathbf{x}) = D'(\mathbf{x}) + V'(\mathbf{x})$ but with specially constructed data terms D'_p and smoothness terms V'_{pq}:

$$\begin{aligned} D'_p(0) &:= D_p(\hat{f}_p) & V'_{pq}(0,0) &:= V(\hat{f}_p, \hat{f}_q) \\ D'_p(1) &:= D_p(\alpha) & V'_{pq}(0,1) &:= V(\hat{f}_p, \alpha) \\ & & V'_{pq}(1,0) &:= V(\alpha, \hat{f}_q) \\ & & V'_{pq}(1,1) &:= V(\alpha, \alpha) \end{aligned} \tag{3}$$

where D and V are the terms of the original multi-label energy $E(f)$. Since $E'(\mathbf{x}) = E(f(\mathbf{x}))$, minimizing E' finds an optimal expansion move w.r.t. E.

A binary energy of form $E' = D' + V'$ can be minimized efficiently by a graph cut if $E'(\mathbf{x})$ is a *submodular* function (Boros and Hammer, 2002; Kolmogorov and Zabih, 2004). Energy E' is submodular if and only if it satisfies

$$V'_{pq}(0,0) \;+ V'_{pq}(1,1) \;\leq\; V'_{pq}(0,1) + V'_{pq}(1,0) \tag{4}$$

for all $pq \in \mathcal{N}$. Now, *for which multi-label energies does the expansion energy* (3) *result in submodular* E'? The submodularity condition (4) holds if and only if

$$V(\hat{f}_p, \hat{f}_q) + V(\alpha, \alpha) \;\leq\; V(\hat{f}_p, \alpha) + V(\alpha, \hat{f}_q). \tag{5}$$

for all neighbours pq and all possible values of $\hat{f}_p$ and $\hat{f}_q$. Since we must assume $\hat{f}_p$ and $\hat{f}_q$ could take any label in $\mathcal{L}$ we arrive at the following due to Kolmogorov and Zabih (2004); Boykov and Veksler (2006).

Observation 2 *The α-expansion algorithm is applicable to energies of the form $D + V$ if and only if for all $\alpha, \beta, \gamma \in \mathcal{L}$*

$$V(\alpha,\alpha) + V(\beta,\gamma) \;\leq\; V(\alpha,\gamma) + V(\beta,\alpha). \tag{6}$$

The metric assumption is sufficient for (6) to hold, and so α-expansion is applicable if V is metric energy.

Rother et al. (2005) showed that, by assuming a non-arbitrary initial labeling, α-expansion can be applied to a wider class of energies: for each $\beta, \gamma \in \mathcal{L}$, either V must satisfy (6) for all $\alpha \in \mathcal{L}$, *or* $V(\beta,\gamma) = \infty$. Unfortunately, α-expansion offers no approximation guarantees for this extended class (Theorem 1 below). In this paper we define a class of smoothness costs V called h-metrics, and it too can be extended to non-metric infinities this way. However, we aim to quantify approximation bounds for our algorithm so, for simplicity, we will not include such infinities in our definition of h-metrics.

Delong et al. (2012) showed that α-expansion is also applicable to energies with label costs as long as $H(L) \geq 0$ for each label subset $L \subset \mathcal{L}$. The expansion step requires a binary energy of the form $E'(\mathbf{x}) = D'(\mathbf{x}) + V'(\mathbf{x}) + H'(\mathbf{x})$ where H' defines very high-order potentials over $\mathbf{x}$, unlike V' which defines only 'quadratic' (pairwise) potentials. We will use their construction in our main subroutine.

3.3 Approximation bounds of α-expansion

Local search with expansion moves is guaranteed to terminate at a local minimum $\hat{f}$ that is within a constant factor of the global optimum f^* (Veksler, 1999; Boykov et al., 2001). The actual bound is $E(\hat{f}) \leq 2cE(f^*)$ where $c \geq 1$ is some constant that depends on V. If c is small, then we can expect α-expansion to do at least a reasonable job. If c is large, the the bound is meaningless and we have even more reason to try other algorithms (*e.g.* Kolmogorov, 2006).

Understanding the approximation bound of α-expansion will be helpful for understanding our generalized bound in Section 6. The following holds for any energy $E = D + V$ with[1] $D_p(\cdot) \geq 0$ and metric $V(\cdot,\cdot)$.

Theorem 1 (Veksler (1999)) *If f^* is a global minimum of $E = D + V$, and $\hat{f}$ is a local minimum w.r.t. expansion moves, then*

$$E(\hat{f}) \;\leq\; 2cE(f^*) \quad \textit{where } c = \frac{\max_{\alpha\neq\beta\in\mathcal{L}} V(\alpha,\beta)}{\min_{\gamma\neq\zeta\in\mathcal{L}} V(\gamma,\zeta)} \tag{7}$$

[1] Note that α-expansion itself does not require $D_p(\cdot) \geq 0$; this assumption is only needed for analysis of worst-case bounds.

In other words, α-expansion is a $2c$-approximation algorithm where $c \geq 1$ depends on the ratio of largest to smallest costs in V. Below are some $V(\cdot,\cdot)$ terms commonly used in vision, shown in matrix form for $|\mathcal{L}| = 5$.

Potts (1952)

0	1	1	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

$c = 1$

linear

0	1	2	3	4
1	0	1	2	3
2	1	0	1	2
3	2	1	0	1
4	3	2	1	0

$c = 4$

truncated linear

0	1	2	2	2
1	0	1	2	2
2	1	0	1	2
2	2	1	0	1
2	2	2	1	0

$c = 2$

Underneath we see the coefficient c corresponding to each case. The simplest potential (Potts) penalizes $f_p \neq f_q$ equally, and gives the best approximation bound. When the range of values is large, *e.g.* for the 'linear' penalty of $|f_p - f_q|$, the bound (7) gets worse. Our h-fusion algorithm beats the α-expansion bound for a wide class of 'hierarchical' costs.

Delong et al. (2012) showed that incorporating label costs $H(\cdot)$ into α-expansion can worsen the above bound. If arbitrary label costs $H(L) \geq 0$ are assumed on arbitrary subsets $L \subset \mathcal{L}$ then the bound is as follows.

Theorem 2 (Delong et al. (2012)) *If f^* is a global minimum of $E = D + V + H$, and $\hat{f}$ is a local minimum w.r.t. α-expansion, then the following bound is tight:*

$$E(\hat{f}) \;\leq\; (2c + c_2)E(f^*) \;+ \sum_{L\subset\mathcal{L}} H(L) \tag{8}$$

where $c_2 = \max_{L:H(L)>0}(|L|-1)$.

This tells us that, if arbitrary label costs are assumed, standard α-expansion is no longer a constant-ratio approximation algorithm (recall Observation 1) and furthermore the bound gets worse (c_2 gets larger) if costs are defined on large subsets L.

4 Energies with Hierarchical Costs

We wish to minimize an energy of the general form (1), but we assume the labels are grouped in some kind of hierarchy. Depending on the application, the grouping will likely be either semantic (hierarchy of object labels) or geometric (families of geometric models). One option for minimizing such energies is to ignore the grouping and simply apply α-expansion. However, Theorem 2 suggests caution, because α-expansion finds poor local minima when, for example, strong high-order label costs are involved.

We express the label grouping structure as follows. The leaves of the tree are the actual labels $\mathcal{L}$. The root of the tree we denote by r. For trees of non-trivial structure, we call the extra nodes *pseudo-labels* and denote them by the set $\mathcal{S}$, so the set of all intermediate nodes is $\mathcal{S} \cup \{r\}$. The structure of the tree itself can be defined by a child-to-parent map $\pi(\cdot)$

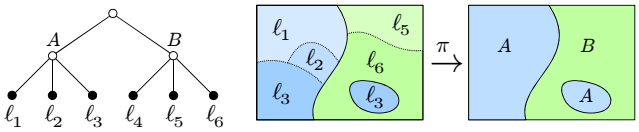


Fig. 2 A hierarchical grouping of label set $\mathcal{L} = \{\ell_1, \ldots, \ell_6\}$ into groups $\mathcal{S} = \{A, B\}$ where, for example, the parent $\pi(\ell_4) = B$. At right is a possible labeling f and the pseudolabeling $(\pi \circ f)$ it induces.

where for each node $i \in \mathcal{L} \cup \mathcal{S}$ a parent $\pi(i) \in \mathcal{S} \cup \{r\}$ is defined. We use $\mathcal{T} = \mathcal{L} \cup \mathcal{S} \cup \{r\}$ to denote all tree nodes. Figure 2 shows a simple label hierarchy.

Merely declaring the labels to be 'grouped' does not in itself change energy $E(f)$ (we still have $f_p \in \mathcal{L}$) nor is the standard α-expansion algorithm 'aware' of a label hierarchy. However, in Sections 4.1–4.2 we describe energies for which a 'good' tree can be defined so that our h-fusion algorithm (Section 5) is provably better than α-expansion. In fact if one defines a flat tree ($\mathcal{S} = \{\}$) then our algorithm and approximation bounds all reduce to those of Boykov et al. (2001) and Delong et al. (2012) as a special case.

Sections 4.1 and 4.2 develop key definitions: the class of *h-metrics* for smoothness costs V, and the class of *h-subsets* for label costs H. These definitions are directly motivated by our h-fusion algorithm and how its computation is organized. If an energy satisfies our h-metric and h-subset assumptions, then it can be minimized by h-fusion (Section 5).

4.1 Hierarchical Smoothness Costs (h-Potts, h-metrics)

The following notation will be helpful for discussing a tree defined by parent function π. We use $\pi^n(i)$ to denote n applications of π, as in $\pi(\cdots\pi(i))$. The set of children of a node j is denoted by

$$\mathcal{I}(j) = \{\, i \in \mathcal{T} \,:\, \pi(i) = j \,\}\,.$$

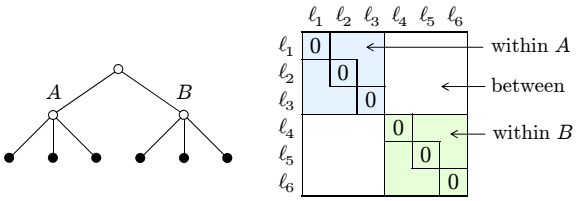
The set of all nodes in the subtree rooted at j is denoted by

$$\text{subtree}(j) = \{\, i \in \mathcal{T} \,:\, \pi^n(i) = j \;\text{ for some }\; n \geq 0 \,\}\,.$$

The set of labels belonging to the subtree of node i is

$$\mathcal{L}_i = \{\, \ell \in \mathcal{L} \,:\, \ell \in \text{subtree}(i) \,\}$$

A hierarchical grouping of labels $\mathcal{L}$ induces a grouping of the smoothness cost values inside each $V(\cdot,\cdot)$ potential. Looking at the tree structure in Figure 2 we can say that labels ℓ_1 and ℓ_2 are both in group A whereas ℓ_1 and ℓ_4 are from different groups; thus $V(\ell_1, \ell_2)$ can be interpreted as a "within-group cost" and $V(\ell_1, \ell_4)$ as a "between-group cost." An example is illustrated below, where regions of the $|\mathcal{L}| \times |\mathcal{L}|$ matrix are delineated.



Individual costs inside each block can vary, though it is helpful to consider the simple case when the cost within each block is constant. So, we now define *hierarchical Potts*, a natural class of hierarchical smoothness costs V parameterized by node-based 'transition' costs $\{w_i\}_{i \in \mathcal{S} \cup \{r\}}$. In what follows, $\text{lca}(\alpha, \beta)$ denotes the *lowest common ancestor* of nodes α and β with respect to the tree structure π.

Definition 2 (Delong et al. (2011)) Given tree structure π, for each node $i \in \mathcal{T}$ let $w_i \geq 0$ be its *transition cost* so that $V(\alpha, \beta) = w_{\text{lca}(\alpha,\beta)}$ for all $\alpha, \beta \in \mathcal{L}$ and $w_i = 0$ for each leaf $i \in \mathcal{L}$. We then say that (V, π) forms a *hierarchical Potts* (h-Potts) potential.

For example, Delong et al. (2011) use a two-level tree where w_r is the transition cost between 'super-labels' and each w_i for $i \in \mathcal{S}$ is the transition cost between 'sub-labels' in group i. They show that if $w_i \leq 2w_r$ then V is metric and standard α-expansion can be applied. For our h-fusion algorithm to apply, a simple sufficient condition is that $w_i \leq w_{\pi(i)}$ for all $i \in \mathcal{L} \cup \mathcal{S}$.

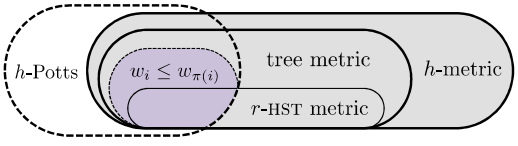
Now we define *h-metrics*, a class of hierarchical smoothness costs where V is not necessarily parameterized by w_i. As we shall see in Section 5, the h-metric assumption is necessary for our specialized algorithm.

Definition 3 We say that pair (V, π) forms a *hierarchical metric* (h-metric) if π is irreducible[2] and for every $i \in \mathcal{L} \cup \mathcal{S}$

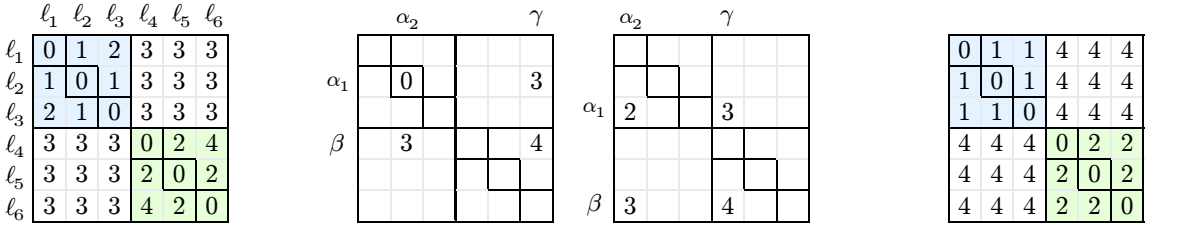
$$V(\alpha_1, \alpha_2) + V(\beta, \gamma) \;\leq\; V(\alpha_1, \gamma) + V(\beta, \alpha_2) \qquad (9)$$
$$\forall\, \alpha_1, \alpha_2 \in \mathcal{L}_i,\; \beta, \gamma \in \mathcal{L}_{\pi(i)} \setminus \mathcal{L}_i$$

Note that for a flat tree ($\mathcal{S} = \{\}$) each set $\mathcal{L}_i = \{i\}$ so we always have $\alpha_1 = \alpha_2$. It is easy to show, then, that the h-metric constraint (9) on a flat tree reduces to the standard α-expansion constraint (6). Figure 3 gives a concrete example of an h-metric and shows how (9) constrains the costs that V can encode.

For a fixed tree structure, the relationship between h-Potts, h-metrics, tree metrics and r-HST metrics is shown by the set inclusion diagram below (see Appendix A for proof). Our h-fusion algorithm will be applicable to the shaded cases.



[2] A tree is irreducible if all its internal nodes have at least two children, *i.e.* there are no 'redundant' parent nodes and so for each i there exists some γ, ζ such that $\text{lca}(\gamma, \zeta) = i$.



(a) example h-metric two configurations where (a) satisfies (9) (b) an h-Potts with $w_i < w_{\pi(i)}$

Fig. 3 The costs in (a) define an h-metric for the tree structure in Figure 2. Inequality (9) is satisfied for every tuple $(\alpha_1, \alpha_2, \beta, \gamma)$ required; the two tuples emphasized at center are $(\ell_2, \ell_2, \ell_4, \ell_6)$ where (9) becomes $0 + 4 \leq 3 + 3$, and $(\ell_3, \ell_1, \ell_6, \ell_4)$ where it becomes $2 + 4 \leq 3 + 3$. On the right, (b) shows a totally different h-metric that is also h-Potts with $w_i < w_{\pi(i)}$.

4.2 Hierarchical Label Costs (h-subsets)

We have already defined a notion of 'hierarchical' smoothness costs (h-metrics), and we now do the same for label costs. As we shall see, if an energy $E(f)$ has hierarchical costs with respect to some tree, then $E(f)$ can be minimized by our h-fusion algorithm on that tree (Section 5).

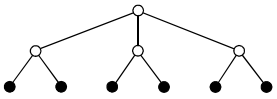
Definition 4 Given an irreducible tree structure π, we define its *hierarchical label subsets* (h-subsets) as

$$\mathcal{H} = \left\{ L \subseteq \mathcal{L} : L \cap \mathcal{L}_i = \emptyset \ \vee \ L \subset \mathcal{L}_i \ \vee \ L \supseteq \mathcal{L}_i \ \forall i \in \mathcal{T} \right\}$$

or, equivalently

$$\mathcal{H} = \left\{ L \subseteq \mathcal{L} : L = \cup_{i \in I} \mathcal{L}_i \text{ for some siblings } I \subset \mathcal{I}(j) \right\}$$

For example, assume we have $\mathcal{L} = \{\ell_1, \ldots, \ell_6\}$ grouped in the structure shown below:



The hierarchical label subsets are

$$\begin{aligned} \mathcal{H} = \big\{ &\{\}, \{\ell_1\}, \{\ell_2\}, \{\ell_3\}, \{\ell_4\}, \{\ell_5\}, \{\ell_6\}, \\ &\{\ell_1, \ell_2\}, \{\ell_3, \ell_4\}, \{\ell_5, \ell_6\}, \\ &\{\ell_1, \ell_2, \ell_3, \ell_4\}, \{\ell_1, \ell_2, \ell_5, \ell_6\}, \{\ell_3, \ell_4, \ell_5, \ell_6\} \\ &\{\ell_1, \ell_2, \ell_3, \ell_4, \ell_5, \ell_6\} \big\}. \end{aligned}$$

Note that sets like $\{\ell_2, \ell_3\}$ and $\{\ell_1, \ell_2, \ell_3\}$ are not in $\mathcal{H}$ because they cannot be generated from a union of siblings in the particular hierarchy chosen.

Definition 5 Given a tree π we say that (H, π) form *hierarchical label costs* if $H(L) > 0 \ \Rightarrow \ L \in \mathcal{H}$, *i.e.* if label costs appear only on the h-subsets.

Note that for a flat tree, the set $\mathcal{H} = 2^{\mathcal{L}}$ and so all subsets are considered 'hierarchical' in this degenerate case.

5 Hierarchical Fusion Algorithm (h-fusion)

Recall that the α-expansion algorithm (Section 3, Boykov et al., 2001) minimizes a multi-label energy by constructing a sequence of binary energies. Our h-fusion algorithm constructs a hierarchical sequence of *multi-label* energies, each of which is solved by running α-expansion as a subroutine. These intermediate multi-label energies are designed to 'stitch' or to 'fuse' labelings that were computed earlier in the sequence. As we shall see, this procedure provides better optimality guarantees than α-expansion for a wide class of energies, particularly those with strong label costs.

We name our algorithm h-fusion after the binary *fusion moves* of Lempitsky et al. (2010). They propose an iterative algorithm to minimize energies of the form $E = D + V$. Given a two candidate labelings $\hat{f}^A$ and $\hat{f}^B$, they try to find a lower-energy labeling by 'fusing' the best parts of each. This is highly analogous to *optimized crossover* operations (Aggarwal et al., 1997), a successful technique in genetic algorithms (discussed in Section 8). Figure 4 illustrates how new labelings are generated this way in our hierarchy. The key insight of Lempitsky et al. is that all fusion moves can be generated by a binary vector $\mathbf{x} = (x_p)_{p \in \mathcal{P}}$ where the move itself is $f = \mathbf{x}\hat{f}^A + \bar{\mathbf{x}}\hat{f}^B$. Note the similarity with the inner loop of α-expansion: an expansion move from current labeling $\hat{f}$ can now be thought of as the binary fusion $f = \mathbf{x}\boldsymbol{\alpha} + \bar{\mathbf{x}}\hat{f}$ with constant labeling $\boldsymbol{\alpha} = (\alpha, \ldots, \alpha)$. They propose a tennis-tournament strategy for selecting pairs of labelings to fuse, and they stop after the energy fails to decrease for some number of attempts. However, they cannot always find an optimal fusion because they fuse arbitrary labelings; the resulting binary energy is *non-submodular* and therefore NP-hard in general (see Section 3.2). They must rely on minimization methods like QPBO (Kolmogorov and Rother, 2007; Rother et al., 2007) with no approximation guarantees.

In contrast, we construct *multi-label* subproblems and approximately minimize each one with α-expansion. Ours is therefore a *graph cut* approach that does not rely on QPBO, message passing (Kolmogorov, 2006), or other LP relaxations (Werner, 2008). Furthermore we,

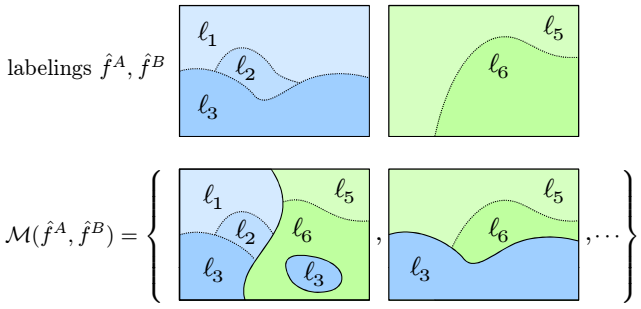


Fig. 4 Given labelings $\hat{f}^A$ and $\hat{f}^B$, the set of all possible fusions $\mathcal{M}(\hat{f}^A, \hat{f}^B)$ includes the two 'stitched' labelings shown at bottom. Our move space $\mathcal{M}(\hat{f}^1, \ldots, \hat{f}^k)$ fuses any number of labelings.

- handle energies with label costs ($E = D + V + H$),
- characterize the subclass of energies (h-metrics, h-subsets) for which h-fusion is applicable, and
- prove approximation bounds that generalize and improve upon those of α-expansion.

To the best of our knowledge, we are the first to incorporate high-arity costs (label costs) into a fusion-based algorithm.

At each step of our h-fusion algorithm, the main subproblem is to find $\operatorname{argmin}_{f \in \mathcal{M}(\hat{f}^1, \ldots, \hat{f}^k)} E(f)$ where each $\hat{f}^i$ is a fixed labeling with $\hat{f}^i_p \in \mathcal{L}_i$ and the set of all possible fusions is

$$\mathcal{M}(\hat{f}^1, \ldots, \hat{f}^k) = \{\, f : f_p \in \{\hat{f}^i_p\}_{i \in \{1..k\}} \,\} \tag{10}$$

Crucially, in our framework no two labelings $\hat{f}^i$ and $\hat{f}^{i'}$ can contain the same labels, unlike the 'tennis-tournament' scheme. Given a tree structure π on label set $\mathcal{L}$, our algorithm generates labelings in a bottom-up fashion with respect to π. Figure 5 shows the order of sub-problems on a simple tree structure. For each internal node j a multi-label fusion energy is constructed and approximately minimized by α-expansion. The local minima computed at one level of the tree are subsequently fused at the next-higher level. Recursive code is given below, where calling h-FUSION(r) on root node r builds the final labeling in bottom-up manner.

h-FUSION(j) — for node j, outputs labeling $\hat{f}^j$ with $\hat{f}^j_p \in \mathcal{L}_j$

```
1 if j ∈ 𝓛
2     return (j, j, ...)                  // at leaf; return constant labeling
3 for i ∈ 𝓘(j)                            // compute child labelings
4     f̂^i := h-FUSION(i)
5 E' := CONSTRUCTFUSIONENERGY(j)
6 g  := α-EXPANSION(E')                   // each g_p is a child index ∈ 𝓘(j)
7 f_p := f̂^{g_p}_p   ∀p ∈ 𝒫              // convert child index to label in 𝓛
8 return f                                // local minimum at node j
```

The key question for h-FUSION is how to set up E' (line **5**) so that it encodes our original energy E over all possible fusions, *i.e.* over all labelings in $\mathcal{M}(\{\hat{f}^i\}_{i \in \mathcal{I}(j)})$.

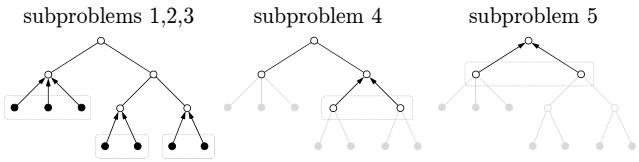


Fig. 5 The h-fusion steps on a tree. Subproblems 1,2,3 independently generate labelings using mutually exclusive labels. Subsequent steps fuse these labelings a bottom-up manner.

Given a set of labelings $\{\hat{f}^i\}_{i \in \mathcal{I}(j)}$ there is a one-to-one correspondence between mappings $g : \mathcal{P} \to \mathcal{I}(j)$ and labelings $f \in \mathcal{M}(\{\hat{f}^i\}_{i \in \mathcal{I}(j)})$. We let $f(g)$ be the labeling $f_p = \hat{f}^{g_p}_p$ corresponding to g. We can then design an unconstrained energy E' such that $E'(g) = E(f(g))$ for all g.

Again, we define E' to be an energy over child *indices*. It will involve the familiar terms

$$E'(g) = D'(g) + V'(g) + H'(g) \tag{11}$$

where D' takes the usual form and V' takes the form

$$V'(g) = \sum_{pq \in \mathcal{N}} V'_{pq}(g_p, g_q).$$

As we shall see, the label costs H' of our fusion energy must take the form of *local label costs* (Delong et al., 2012)

$$H'(g) = \sum_{I \subseteq \mathcal{I}(j)} H'_{P_I}(I)\, \delta_I(g_{P_I}) \qquad \text{(\textit{local} label costs)}$$

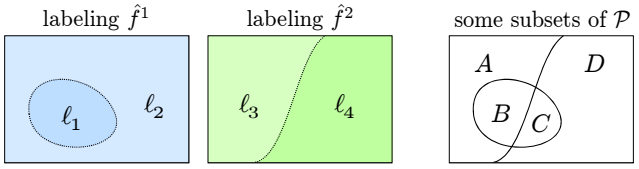
where cost $H'_{P_I}(I)$ is only applied for particular subset of variables $P_I \subseteq \mathcal{P}$ associated with child indices $I \subseteq \mathcal{I}(j)$. In other words, we need E' to encode costs of the form "pay if g uses this child *here*." The precise costs encoded by D', V' and H' are shown in CONSTRUCTFUSIONENERGY below.

CONSTRUCTFUSIONENERGY(j)

```
1 D'_p(i) := D_p(f̂^i_p)                          ∀ p ∈ 𝒫, i ∈ 𝓘(j)
2 V'_pq(i,i') := w_pq · V(f̂^i_p, f̂^{i'}_q)       ∀ pq ∈ 𝒩, i, i' ∈ 𝓘(j)
3 for each L ∈ ℋ such that H(L) > 0
4     I := { i ∈ 𝓘(j) : L ∩ 𝓛_i ≠ ∅ }            // relevant child indices
5     P_I := { p ∈ 𝒫 : ∃ f̂^i_p ∈ L, i ∈ I }       // relevant pixel indices
6     H'_{P_I}(I) := H(L)                        // set up local label cost
7 return E' = (D', V', H')
```

The correctness of D' and V' are self evident; the algorithm of Kumar and Koller (2009) includes lines **1–2** but on a more restrictive class of metrics. However, our work was mainly motivated by label costs. It is not obvious how lines **4–6** encode original label cost $H(L)$ as a local label cost $H'_{P_I}(I)$. We now verify its correctness, first by simple example and then by proving it in general.

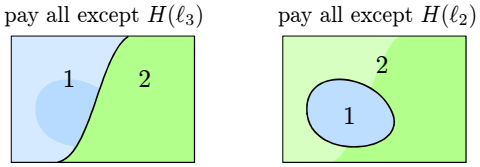
Example 1 Suppose E' must fuse the two child labelings $\hat{f}^1$ and $\hat{f}^2$ shown below.



Assume $\mathcal{H} = \{\{\ell_1\}, \{\ell_2\}, \{\ell_3\}, \{\ell_4\}, \{\ell_1, \ell_2\}, \{\ell_3, \ell_4\}\}$ and $H(L) > 0$ for each $L \in \mathcal{H}$. Each $g_p \in \{1, 2\}$ and so $E'(g)$ can account for $H(L)$ by setting

$$H'_{\mathcal{P}}(1) := H(\{\ell_1, \ell_2\}) \qquad H'_{B\cup C}(1) := H(\ell_1)$$
$$H'_{\mathcal{P}}(2) := H(\{\ell_3, \ell_4\}) \qquad H'_{A\cup D}(1) := H(\ell_2)$$
$$H'_{A\cup B}(2) := H(\ell_3)$$
$$H'_{C\cup D}(2) := H(\ell_4).$$

Costs $H'_{\mathcal{P}}(i)$ (left-hand column) are global label costs in the fusion energy. The other costs $H'_P(i)$ (right-hand column), are localized label costs that encode the original (global) per-label costs $H(\ell)$. In this example, the fusion below at left should pay all label costs except $H(\ell_3)$, whereas the fusion at right should pay all costs except $H(\ell_2)$.



Theorem 3 *If energy E has hierarchical label costs (H, π) then $E'(g) = E(f(g))$ for all $g : \mathcal{P} \to \mathcal{I}(j)$.*

Proof The correctness of D' and V' are self-evident so we focus on proving that $H'_{P_I}(I)$ correctly encodes $H(L)$ for some subset of labels $L \in \mathcal{H}$. This reduces to showing that $\delta_L(f(g)) = \delta_I(g_P)$ where indices I and pixels $P = P_I$ are as defined on lines **4** and **5** of CONSTRUCTFUSIONENERGY. In other words, we must prove that

$$\exists \hat{f}_p^{g_p} \in L,\ p \in \mathcal{P} \quad \Longleftrightarrow \quad \exists g_p \in I,\ p \in P \tag{12}$$

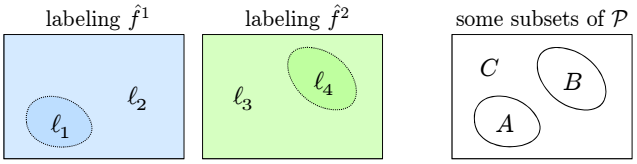
where we can assume $\hat{f}_p^{g_p} \in \mathcal{L}_{g_p}$ due to the way h-fusion works.

Because we assume hierarchical label costs, each $L \in \mathcal{H}$ belongs to one of four cases derived from Definition 4.

1. If $L \cap \mathcal{L}_i = \emptyset$ for all $i \in \mathcal{I}(j)$, then we know that any $\hat{f}_p^{g_p} \notin L$, then the cost $H(L)$ is not applied in subtree j. By definition $I = \emptyset$ ensuring $\delta_L(f(g)) = \delta_\emptyset(g_P) = 0$, which is correct.
2. If $L \subset \mathcal{L}_i$ for some $i \in \mathcal{I}(j)$, then by definition $I = \{i\}$ and $P = \{p : \hat{f}_p^i \in L\}$. Since $g_p \neq i \Rightarrow \hat{f}_p^{g_p} \notin L$ and so $f(g)$ contains a label in L if and only if $g_p = i$ for some $p \in P$. Therefore $\delta_L(f(g)) = \delta_i(g_P)$ holds in this case.
3. If $\mathcal{L}_i \subseteq L \subset \mathcal{L}_j$ for some $i \in \mathcal{I}(j)$, then clearly $P = \mathcal{P}$. Since $L \in \mathcal{H}$ we must also have $L = \cup_{i\in I}\mathcal{L}_i$, and so $f(g)$ uses a label in L if an only if g uses a label in I. Therefore $\delta_L(f(g)) = \delta_I(g)$ holds in this case.
4. If $L \supseteq \mathcal{L}_j$ then $I = \mathcal{I}(j)$ and $P = \mathcal{P}$, so $H(L)$ can be added to E' as a constant or simply ignored. ■

Looking at the proof of Theorem 3 we can see that the structure of h-subsets is especially needed for the third case to hold. If we allow a subset $L \notin \mathcal{H}$, then α-expansion could not be applied to the resulting E' because the internal binary steps would be non-submodular and potentially NP-hard. The purpose of Example 2 is to demonstrate why $H(L) > 0$ for arbitrary L can be problematic.

Example 2 Suppose E' must fuse the two child labelings $\hat{f}^1$ and $\hat{f}^2$ shown below.



Further suppose that $H(\{\ell_1, \ell_4\}) > 0$ in the original energy. This cost should be paid in $E'(g)$ if and only if any g_p variable in region A is assigned child index 1 *or* any g_p in region B is assigned child index 2. However, this potential cannot be encoded as a label cost any sort. Furthermore, encoding $H(\{\ell_1, \ell_4\})$ would result in a non-submodular fusion energy. Let $E'(i, j)$ denote the label cost of assigning index i to pixels A and index j to pixels B, then

$$\begin{aligned} E'(1,1) + E'(2,2) &= 2H(\{\ell_1, \ell_4\}) \\ &> H(\{\ell_1, \ell_4\}) = E'(1,2) + E'(2,1). \end{aligned}$$

That $E'(i, j)$ is not submodular follows from inequality (4) if one assumes $|A| = |B| = 1$, and so α-expansion is not applicable inside h-fusion. In fact, because this example has only two labels, we can conclude that the $\alpha\beta$-swap algorithm (Boykov et al., 2001) is also inapplicable to E'.

Finally, we establish that h-metrics give a precise characterization of smoothness costs V that h-fusion can handle.

Theorem 4 *The h-fusion algorithm is applicable to V using tree π if and only if (V, π) forms an h-metric.*

Proof Recall from (6) that α-expansion is applicable if and only if V satisfies, for all $\alpha, \beta, \gamma \in \mathcal{L}$,

$$V(\alpha, \alpha) + V(\beta, \gamma) \leq V(\alpha, \gamma) + V(\beta, \alpha). \tag{13}$$

Actually, if (13) holds for $\beta, \gamma \in \mathcal{L}\backslash\{\alpha\}$ the it trivially holds for all $\beta, \gamma \in \mathcal{L}$. This observation matters for h-fusion.

In the h-fusion case, each local fusion metric V'_{pq} on line **2** of CONSTRUCTFUSIONENERGY must satisfy this constraint and so α-expansion can fuse a collection of labelings $\{\hat{f}^i\}_{i\in\mathcal{I}(j)}$ if and only if, for all $i \in \mathcal{I}(j),\ i', i'' \in \mathcal{I}(j) \setminus \{i\}$,

$$V(\hat{f}_p^i, \hat{f}_q^i) + V(\hat{f}_p^{i'}, \hat{f}_q^{i''}) \leq V(\hat{f}_p^i, \hat{f}_q^{i''}) + V(\hat{f}_p^{i'}, \hat{f}_q^i) \tag{14}$$

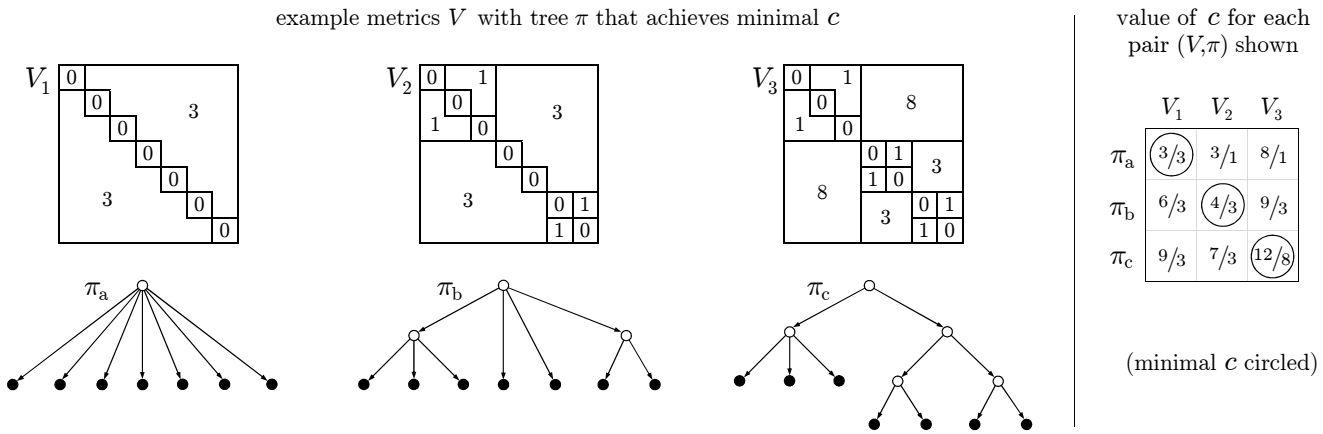


Fig. 6 The top row shows three example smoothness cost matrices. The first is a standard (flat) Potts potential with penalty $V_1(\ell,\ell') = 3$ for any $\ell \neq \ell'$ and so $c = 1$ for a flat tree π_a. Metrics V_2 and V_3 have varying penalties and so a flat tree yields $c = 3$ and $c = 8$ respectively. However, by applying h-fusion on tree structures π_b and π_c respectively (bottom row), we can achieve better c for these particular metrics. The table at right shows other values of c and demonstrates that the choice of tree is important for achieving a good bound.

Note that $\hat{f}_p^i$ and $\hat{f}_q^i$ could each be any label in $\mathcal{L}_i$ and are *not* necessarily identical, unlike the α-expansion case. The constraints on the original metric V for h-fusion will therefore be more restrictive than for α-expansion. Since inequality (14) must hold for all possible labelings $\hat{f}^i$, $\hat{f}^{i'}$, and $\hat{f}^{i''}$ then it is equivalent to

$$V(\alpha_1,\alpha_2) + V(\beta,\gamma) \leq V(\alpha_1,\gamma) + V(\beta,\alpha_2) \qquad (15)$$
$$\alpha_1,\alpha_2 \in \mathcal{L}_i,\ \beta,\gamma \in \mathcal{L}_j \setminus \mathcal{L}_i$$

Since $j = \pi(i)$ then inequalities (15) are identical to (9). ■

6 Approximation Bounds of h-Fusion

Since α-expansion has an approximation bound and we use it as our main subroutine, it is natural to ask if h-fusion has some bound of its own. If we use a flat tree π and assume $E = D + V$, then h-fusion reduces to classical α-expansion and we directly inherit the $2c$-approximation where

$$c = \frac{\max_{\alpha,\beta\in\mathcal{L}} V(\alpha,\beta)}{\min_{\gamma\neq\zeta\in\mathcal{L}} V(\gamma,\zeta)}. \qquad (16)$$

Our goal is to derive a generalized bound for h-fusion with arbitrary tree π and arbitary label costs (*i.e.* $\mathcal{H} = 2^{\mathcal{L}}$ in (1)). Like the α-expansion bound in Theorem 2, the quality of our new bound will involve some c and c_2 that depend on the particular energy. As we shall see, these two coefficients can be much smaller for our algorithm. We begin by defining some useful quantities for expressing the h-fusion bound.

Definition 6 Given smoothness costs V and a particular tree π, we define two quantities for each node i:

$$V_i^{\max} = \max_{\alpha,\beta\in\mathcal{L}_i} V(\alpha,\beta) \qquad V_i^{\min} = \min_{\substack{\gamma,\zeta\in\mathcal{L}_i \\ \mathrm{lca}(\gamma,\zeta)=i}} V(\gamma,\zeta).$$

In other words, $V_i^{\max}$ is the maximum cost for any pair of labels in the subtree of node i, and $V_i^{\min}$ is the minimum cost for two labels from *different* subtrees descended from i. For example, in Figure 3a we have $V_A^{\max} = 2, V_A^{\min} = 1$ and $V_B^{\max} = 4, V_B^{\min} = 2$. For the root node r, this example has $V_r^{\max} = 4$ and $V_r^{\min} = 3$.

Definition 7 Given an h-metric (V,π) where V is also semi-metric, we define the additional quantities

$$b_j = V_j^{\max} + \max_{i\in\mathcal{I}(j)} b_i \qquad c = \max_{j\in\mathcal{T}\setminus\mathcal{L}} \left(\frac{b_j}{V_j^{\min}}\right)$$

Observation 3 *If π defines a flat tree, then c in Definition 7 reduces to quantity* (16) *from the α-expansion bound.*

The ratio c is most important because it bounds the worst-case approximation error. As we will show, when c is large for standard α-expansion, choosing a non-flat tree can result in a much smaller constant for h-fusion and thereby a better bound. The easiest way to understand how V and π affect c is by looking at a few numeric examples. Figure 6 examines specific values of c for various pairs of smoothness cost matrices V and trees π. These examples suggests that for each choice of V there exists an optimal choice of π to give the best approximation bound. Since for every metric V we can use a flat tree, we can always find a tree for which h-fusion's bound is at least as good as α-expansion's.

We now consider label costs in h-fusion and generalize the related coefficient c_2. The cardinality of set $I \subset \mathcal{I}(j)$ on line **4** of CONSTRUCTFUSIONENERGY is an important quantity affecting c_2 for h-fusion. In general, the smaller $|I|$ the better the bound. (Note that we use $\subset$ to mean $\subsetneq$ throughout this paper.)

Definition 8 We define c_2 to be the maximum cardinality of any index set I (CONSTRUCTFUSIONENERGY, line **4**) that

is a strict subset of $\mathcal{I}(j)$, minus 1. That is, the constant

$$c_2 = \max_{H(L)>0} |I(L)| - 1$$

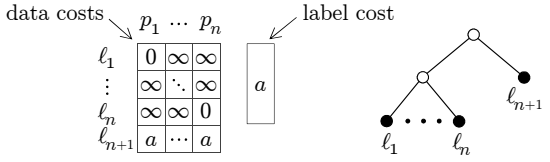
where $I(L) \subset \mathcal{I}(j)$ for some j and $\bigcup_{i \in I(L)} \mathcal{L}_i = L$.

The fact that $I(L)$ will always be the union of some siblings in the tree follows from the assumption that L is an h-subset.

Observation 4 *If π defines a flat tree, then c_2 in Definition 8 reduces to the same quantity for α-expansion in Theorem 2.*

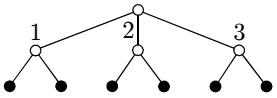
To see how h-fusion can beat α-expansion at minimizing energies with label costs, consider the following worst-case example for α-expansion.

Example 3 Suppose variables $\mathcal{P} = \{p_1, \ldots, p_n\}$ and labels $\mathcal{L} = \{\ell_1, \ldots, \ell_{n+1}\}$. The data costs $D_p(\cdot)$ are shown in the table below, where $a > 0$, and smoothness costs are zero. We also assume a label cost $H(\{\ell_1, \ldots \ell_n\}) = a$, as illustrated.



The labeling $\hat{f} = (\ell_{n+1}, \ldots, \ell_{n+1})$ is a local minimum for α-expansion because no individual variable wants to pay the (shared) label cost in order to switch its label. However the globally optimal labeling is $f^* = (\ell_1, \ldots, \ell_n)$ and, since $E(\hat{f}) = nE(f^*)$, the α-expansion solution can be made arbitrarily bad. The h-fusion algorithm will find f^* if we use the tree shown above, at right. Notice that $c_2 = 0$ for this tree, whereas $c_2 = n - 1$ for a flat tree (α-expansion).

For an example of non-trivial c_2, consider again the six-label tree structure:



If the only label cost $H(L) > 0$ is on subset $L = \{\ell_5, \ell_6\}$ then $I(L) = \{3\}$ and so $c_2 = |I(L)| - 1 = 0$. If instead we have a label cost on $L = \{\ell_1, \ell_2, \ell_5, \ell_6\}$ then $I(L) = \{1, 3\}$ yielding coefficient $c_2 = 1$. Again, the bound of h-fusion is stronger for energies where c_2 is small.

Using our definitions of c and c_2 we state the main theorem of this work: an improvement upon the bound of Delong et al. (2012). For the purposes of the bound we assume $D_p(\cdot) \geq 0$ and that V is semi-metric.

Theorem 5 *If f^* is a global minimum of $E = D + V + H$ with h-metric (V, π) and hierarchical label costs (H, π), then the solution $\hat{f}$ computed by h-fusion is bounded by*

$$E(\hat{f}) \leq (2c + c_2)E(f^*) + \sum_{L \in \mathcal{H}} H(L) \tag{17}$$

where c and c_2 are constants given in Definitions 7 and 8 respectively (possibly much smaller than in Theorem 2).

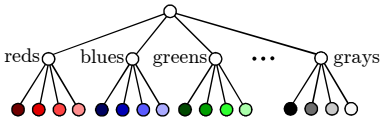
Proof See Appendix B. ■

In the presence of arbitrary label costs, this is still not a constant-ratio approximation bound, but we can construct a worst-case example to show that our bound is indeed tight (see Delong, 2011).

7 Application: Hierarchical Color Segmentation

We use hierarchical color segmentation as a simple, illustrative example because it allows us to visualize the effects of hierarchical smooth and label costs. Given an image we wish to group pixels with similar color. We treat segmentation as labeling where each label represents a color; the labels essentially re-colorize the image. However, we explicitly divide the possible colors into groups, and seek a pixel labeling that relies only on a few groups of colors. For example, a natural way to group colors is by hue, and the goal is then to re-color the image using as few hues as possible while staying reasonably faithful to the original image.

The tree below illustrates one possible grouping (hierarchy) of color labels. Each leaf corresponds to a specific color label (*e.g.* dark red, red, bright red,...) while each subtree corresponds to a group of labels (*e.g.* reds, blues, greens, grays,...)



In order to limit the number of hues used in re-colorization we introduce group costs in addition to regular label costs. A cost is associated with each group of colors L and is represented by a label subset cost $H(L) > 0$. It is paid whenever any of the colors in the group is used in the labeling. For the smoothness costs V we use hierarchical Potts smoothness terms between and within color groups to encourage smooth re-colorization. If I_p is an image pixel, the data cost $D_p(\ell)$ is proportional to squared distance between I_p and the color represented by label ℓ.

We thereby formulate the re-colorization problem as hierarchical energy. We compare α-expansion and h-fusion when applied to this energy. In all the re-colorization experiments our color hierarchy consists of 121 groups of colors, each containing 20 different shades varying from dark to bright. This results in 2420 labels in total. We then demonstrate qualitative (the resulting re-colorizations) and quantitative (running time and energy value) comparisons. In all experiments we set $w_i = 1$ and $w_r = 2$. Each invocation of α-expansion performed two cycles only (a cycle expands on each label exactly once). This limitation was applied to all instances of α-expansion within h-fusion as well. (Allowing α-expansion to converge takes much longer but only decreases the energy by $< 0.01\%$ for both algorithms.)

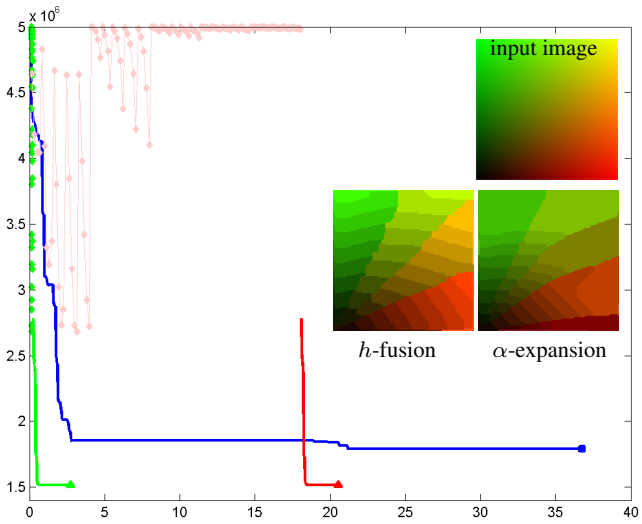


Fig. 7 Synthetic example. Qualitative and quantitative comparison between α-expansion and h-fusion. The blue line corresponds to energy attained by α-expansion as a function of time. The pink line with diamonds correspond to energies of child-labelings optimized sequentially in the first step of h-fusion. Child-labelings are then stitched together in a fusion step. The energy of the final fusion is shown by the red line. The green line represents the energy of h-fusion if all child-labelings are computed in parallel. See text for details.

Fig. 8 Input images used for the experiments shown in Fig. 9.

Consider the input image shown in Fig. 7 top. It has a smooth gradient of color varying from black to yellow through the shades of green and red. Both α-expansion and h-fusion algorithms result in re-colorizations with 4 groups of colors each, namely shades of green, yellow, orange and red (see bottom left and right). However, it can be seen from the result of α-expansion (bottom right) that the optimization got stuck in a local minimum. By expanding on a wrong group of colors first (wrong hue), α-expansion was unable to match the bright portion of the image well. At this point expanding on any one label of a better matched hue did not justify adding the extra group costs associated with this new hue. In the case of h-fusion the algorithm is able to replace one group of hues with another at once and therefore attain a lower energy (see Fig. 7 bottom-left).

The plot in Fig. 7 provides quantitative comparison between α-expansion and h-fusion in terms of running time and energy values. The blue line corresponds to energy value attained by α-expansion as a function of time. The h-fusion

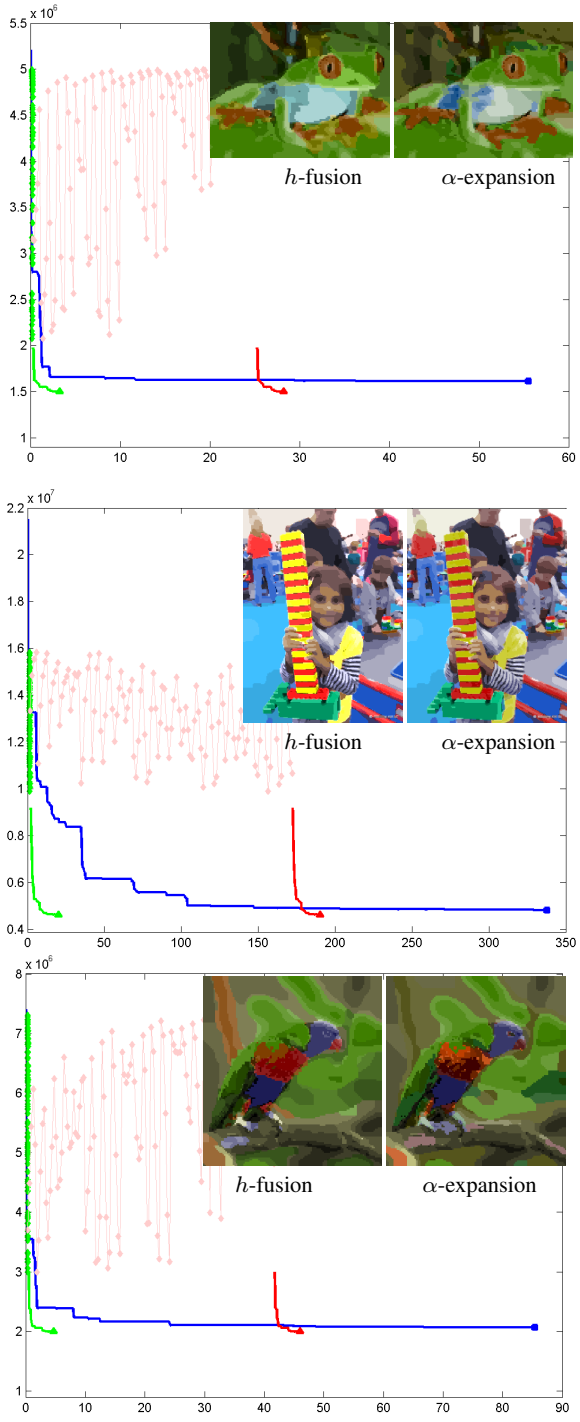


Fig. 9 Qualitative and quantitative comparison between α-expansion and h-fusion on input images shown in Fig. 8. Again, blue, red-pink and green lines correspond to α-expansion, sequential h-fusion and parallel h-fusion respectively. See text for details. Note that the re-colorizations obtained with h-fusion are more faithful to the original images than those obtained with α-expansion. For example the lego piece in the image of a girl is much brighter and, in the parrot image, the parrot's chest and the tree branch are colored more faithfully.

algorithm begins with optimizing a set of sub-energies corresponding to child-labelings. Each child labeling is restricted to one sub-tree of labels and essentially re-colors the image with the colors from that group only. (For example one child-labeling re-colors the image with the shades of red, another with the shades of green...) The sub-energies are independent and can be optimized either sequentially or in parallel. When sub-energies are optimized sequentially we represent each sub-energy with a pink diamond and plot them as a function of cumulative time. After all sub-energies are optimized, h-fusion algorithm fuses the resulting child-labelings by running α-expansion (starting from the child-labeling with the minimal energy. Again we limit the h-fusion to two cycles only). The energy of h-fusion is represented by the red line and attains a lower energy than regular α-expansion.

Unlike α-expansion, the running time of h-fusion can be dramatically improved by minimizing sub-energies in parallel. This is illustrated by the green line in the plot of Fig. 7. In our specific application the parallel version of h-fusion is faster by a factor of 10–15 compared to sequential h-fusion. At any time in our experiments, the energy curve of parallel h-fusion is dramatically below that of α-expansion, and terminates 20–30 times faster. In theory this speed-up factor should grow linearly with the number of siblings at each level of the label hierarchy. The speed-up is due to the fact that running one expansion cycle with h-fusion is more efficient than with regular α-expansion. This is because the number of unique possible labels in h-fusion corresponds to the number of groups in the hierarchy (121 in our case) while the number of unique labels for α-expansion corresponds to the number of leaves in the hierarchy (2420 colors in our case).

Figure 9 shows similar results for additional input images shown in Fig. 8. For all the experiments sequential h-fusion (pink-red line) attained lower energy and in shorter time than α-expansion (blue line), with even more significant speedup in the case of parallel optimization of the sub-energies in h-fusion (green line).

8 Discussion

The main results of this paper are a characterization of hierarchical costs (h-metrics and h-subsets), the h-fusion algorithm itself, and a significant improvement on the approximation bound of α-expansion. These results are theoretical, but we foresee a number of applications for such energies.

Applications of hierarchical costs We presented hierarchical color segmentation as the simplest possible example that illustrates (a) the nature of energies with hierarchical costs, and (b) the qualitative and quantitative benefits of h-fusion for such energies. However, computer vision is full of problems for which hierarchical costs are natural.

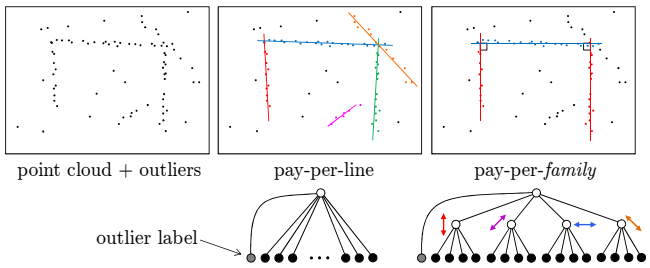
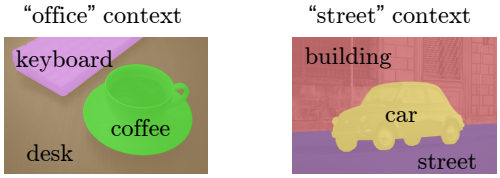


Fig. 10 Depiction of how hierarchical line fitting might work with an energy of the form $E = D + H$. Each label corresponds to a possible line (*e.g.* from random sampling), and each point wants to be labeled by a nearby line. Label cost $H(\ell)$ discourages line ℓ from being used unless there are enough supporting points—otherwise the points take the outlier label (constant penalty per point). However, if we group lines by orientation, we could add costs $H(L)$ where L is a family of lines, encouraging solutions that use a few families of parallel lines.

The most obvious is using hierarchical context (*e.g.* Choi et al., 2010) for image segmentation, where in theory we could group the labels into some appropriate context as depicted below.



This is a very rudimentary form of context but can be integrated with segmentation via an energy with hierarchical V and H terms.

In vision it is also common to assign labels that have geometric meaning, such as depths (*e.g.* Boykov et al., 2001; Ladický et al., 2010b), homographies or motions (*e.g.* Birchfield and Tomasi, 1999; Isack and Boykov, 2011). For example, Isack and Boykov (2011) start with a set of observations (points, matches, *etc.*) and use random sampling to generate hundreds of candidate geometric models, much the way RANSAC does (Fischler and Bolles, 1981). They formulate the model fitting problem as a *labeling* problem where each label represents a candidate model. They find a labeling that corresponds to a good configurations of models, and do this by minimizing an energy of the form $E = D + V + H$. However, there are many situations where geometric models fall into a natural hierarchy. Figure 10 is a hypothetical example to illustrate this point. Analogous hierarchical relationships exist between, for example, a fundamental matrix (a rigid motion) and the family of homographies (families of correspondences) compatible with that fundamental matrix (Hartley and Zisserman, 2003).

Furthermore, hierarchical costs can be useful for detecting patterns, for compression, and for learning a database of inter-dependent patches from images (Gorelick et al., 2011). Outside vision, Sefer and Kingsford (2011) showed that the r-HST metrics of Kumar and Koller are effective at identifying protein function; our work could extend their results.

Relation to r-HST metrics Recall that, at a high level, the h-fusion process shown in Figure 5 is the same as that used by Kumar and Koller (2009). Given a metric V, they find the set of r-HST *metrics* that best approximates V and try to minimize an energy of the form $E = D + V$ using a bottom-up fusion process. The main idea of an r-HST metric is as follows. Assume we are given a tree with distances $d(i, j)$ defined on each edge from child i to parent j. Assume that the distance from j to all its children is uniform, *i.e.* $d(i, j) = d(i', j)$ for all $i, i' \in \mathcal{I}(j)$. Further assume that we know the parent-to-child distance gets cheaper by a factor of r as we descend the tree, *i.e.* $\frac{d(i,j)}{d(k,i)} \geq r$ for some constant $r > 1$. The total distance between two leaf nodes α and β is the cumulative sum of edge distances along the path from α to β in the tree. If the 'costs' of a pairwise potential $V(\alpha, \beta)$ correspond to such a distance function for all α, β, then V is said to be an r-HST metric.

Our concept of an h-metric is expressed directly in terms of constraints on $V(\cdot, \cdot)$, not on edges or distances traversed in the tree. Furthermore, r-HST metrics are a strict subset of h-metrics (see Appendix A).

Generalizing facility location In the optimization and operations research communities, *uncapacitated facility location* (UFL) is a well-studied problem (*e.g.* Shmoys et al., 1998). UFL assigns a 'facility' to serve each client such that the cost to clients and the cost of opening facilities is jointly minimized. UFL is connected to our energy because if we let $\mathcal{L}$ denote the facilities and $\mathcal{P}$ denote the clients then every problem instance can be expressed as minimizing an energy of the form

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{\ell \in \mathcal{L}} H(\ell)\delta_\ell(f). \tag{18}$$

In vision, the UFL objective has recently been applied to motion segmentation by Li (2007) and by Lazic et al. (2009), but goes all the way back to Torr and Murray (1994).

There exist variants of UFL that allow for a hierarchy of facilities, *e.g.* Svitkina and Tardos (2006) and Sahin and Süral (2007). This generalization allows for more realistic modeling of complex interdependencies between facilities themselves. Some of these works derive constant-factor approximation bounds for hierarchical facility location, *e.g.* Kantor and Peleg (2009), but all such works assume *metric* client costs where the costs $D_p(\cdot)$ are computed as distances from a particular center. Without this assumption, Feige's hardness result still holds. Strategies for optimizing hierarchical UFL include linear programming relaxation, primal-dual algorithms and, very recently, message passing algorithms (Givoni et al., 2011).

We can encode a kind of hierarchical facility cost with our framework as follows. Suppose facilities ℓ_1 and ℓ_2 require the services of facility ℓ_3, which costs 50 to open. A label cost $H(\{\ell_1, \ell_2, \ell_3\}) := 50$ correctly accounts for the shared dependency of ℓ_1 and ℓ_2 on ℓ_3. If we furthermore have a facility ℓ_4 that depends on both ℓ_3 and some facility ℓ_5 (cost 80), then our label costs should instead be $H(\{\ell_1, \ell_2, \ell_3, \ell_4\}) := 50$ and $H(\{\ell_4, \ell_5\}) := 80$.

Furthermore, our h-fusion algorithm can handle smoothness costs V, which to the best of our knowledge are novel for UFL. In the UFL setting, $V(f_p, f_q)$ can encode an explicit preference that clients p and q be serviced by the same facility. When clients are social, there are many scenarios where such a preference makes sense. When client costs D are metric (*e.g.* Euclidean distance) then this preference is implicitly encoded in D. However, when the client costs are not metric, such as clients connected by an irregular network despite being physically close, then our smoothness costs V may be useful for modeling such problems.

Improving the bound Recall from Observation 1 that for the SET-COVER problem the best we can hope for is a $\ln |\mathcal{P}|$-approximation. Yet one can formulate SET-COVER using an energy of the form (18), so minimizing energy $E = D + V + H$ is at least as hard. However, Hochbaum (1982) gave a simple greedy algorithm for SET-COVER and proved that it yields precisely a $\ln |\mathcal{P}|$-approximation, the best possible according to Feige (1998). If label costs are arbitrary in (18), then α-expansion's bound is also arbitrarily bad. So, there is a huge gap between what α-expansion can achieve on (18) versus what Hochbaum's greedy algorithm can guarantee. For energies of the form $E = D + H$, it may be possible to extend Hochbaum's algorithm and use it as a subroutine within h-fusion (rather than using α-expansion). One may ask if h-fusion could inherit a better approximation bound in that case. We also do not know if her approach can be applied in the presence of smoothness costs V.

Relation to genetic algorithms Within our framework, the inner α-expansion subroutine is performing a sequence of *fusion moves* like proposed by Lempitsky et al. (2010). We point out that a binary fusion move is essentially an *optimized crossover* operation, already used to some success in genetic algorithms (Aggarwal et al., 1997; Meyers and Orlin, 2007). A standard concern for genetic algorithms is how to maintain *population diversity* so that, when two chromosomes (labelings) are crossed, there is a chance that the descendant will be better. Our h-fusion process forces a kind of population diversity based on a tree: the labelings in our multi-label fusion each contain labels from *different* subtrees. It is interesting that this structured-diversity gives a provably better approximation bound in our case.

Acknowledgements We wish to thank the anonymous reviewers for careful reading and helpful comments. This work was supported by NSERC Discovery Grant R3584A02, the Canadian Foundation for Innovation (CFI), and the Early Researcher Award (ERA) program.

References

Charu C. Aggarwal, James B. Orlin, and Ray P. Tai. Optimized Crossover for the Independent Set Problem. *Operations Research*, 45(2):226–234, 1997.

Ravindra K. Ahuja, Özlem Ergun, James B. Orlin, and Abraham P. Punnen. A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*, 123(1–3):75–202, 2002.

Olga Barinova, Victor Lempitsky, and Pushmeet Kohli. On the Detection of Multiple Object Instances using Hough Transforms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010.

Yair Bartal. On approximating arbitrary metrics by tree metrics. In *ACM Symposium on Theorey of Computing (STOC)*, 1998.

Stan Birchfield and Carlo Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *International Conference on Computer Vision (ICCV)*, 1999.

Endre Boros and Peter L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1–3):155–225, 2002.

Yuri Boykov and Marie-Pierre Jolly. Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. In *International Conference on Computer Vision (ICCV)*, 2001.

Yuri Boykov and Vladimir Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)*, 29(9):1124–1137, 2004.

Yuri Boykov and Olga Veksler. Graph Cuts in Vision and Graphics: Theories and Applications. In Nikos Paragios, Yunmei Chen, and Olivier Faugeras, editors, *Handbook of Mathematical Models in Computer Vision*, pages 79–96. Springer US, 2006.

Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)*, 23(11):1222–1239, 2001.

Myung Jin Choi, Joseph J. Lim, Antonio Torralba, and Alan S. Willsky. Exploiting Hierarchical Context on a Large Database of Object Categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010.

William Cunningham and Lawrence Tang. Optimal 3-Terminal Cuts and Linear Programming. In *Integer Programming and Combinatorial Optimization*, volume 1610 of *LNCS*, pages 114–125. 1999.

Andrew Delong. *Advances in Graph-Cut Optimization*. PhD thesis, University of Western Ontario, October 2011.

Andrew Delong, Lena Gorelick, Frank R. Schmidt, Olga Veksler, and Yuri Boykov. Interactive Segmentation with Super-Labels. In *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, July 2011.

Andrew Delong, Anton Osokin, Hossam N. Isack, and Yuri Boykov. Fast Approximate Energy Minimization with Label Costs. *International Journal of Computer Vision (IJCV)*, 96(1):1–27, 2012. (Earlier version in CVPR 2010).

Uriel Feige. A Threshold of $\ln n$ for Approximating Set Cover. *Journal of the ACM*, 45(4):634–652, 1998.

P.F. Felzenszwalb, G. Pap, É. Tardos, and R. Zabih. Globally optimal pixel labeling algorithms for tree metrics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

Inmar E. Givoni, Clement Chung, and Brendan J. Frey. Hierarchical Affinity Propagation. In *Uncertainty in Artificial Intelligence (UAI)*, July 2011.

Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum-flow problem. *Journal of the Association for Computing Machinery (JACM)*, 35(4):921–940, 1988.

Lena Gorelick, Andrew Delong, Olga Veksler, and Yuri Boykov. Recursive MDL via Graph Cuts: Application to Segmentation. In *International Conference on Computer Vision (ICCV)*, November 2011.

D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society B*, 51(2):271–279, 1989.

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

Dorit S. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming*, 22(1):148–162, 1982.

Hossam N. Isack and Yuri Boykov. Energy-based Geometric Multi-Model Fitting. *International Journal on Computer Vision (IJCV)*, June 2011. doi:10.1007/s11263-011-0474-7.

Evangelos Kalogerakis, Aaron Hertzmann, and Karan Singh. Learning 3D mesh segmentation and labeling. In *ACM SIGGRAPH*, 2010.

Erez Kantor and David Peleg. Approximate hierarchical facility location and applications to the bounded depth Steiner tree and range assignment problems. *Journal of Discrete Algorithms*, 7(3):341–362, 2009.

Jon Kleinberg and Éva Tardos. Approximation Algorithms for Classification Problems with Pairwise Relationships: Metric Labeling and Markov Random Fields. *Journal of the ACM*, 49(5), 2002.

Vladimir Kolmogorov. Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(10):1568–1583, October 2006.

Vladimir Kolmogorov and Carsten Rother. Minimizing non-submodular functions with graph cuts—a review. *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)*, 29 (7), 2007.

Vladimir Kolmogorov and Ramin Zabih. What Energy Functions Can Be Optimized via Graph Cuts. *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)*, 26(2):147–159, 2004.

M. Pawan Kumar and Daphne Koller. MAP estimation of semi-metric MRFs via hierarchical graph cuts. In *Conference on Uncertainty in Artificial Intelligence*, pages 313–320, June 2009.

L'ubor Ladický, Chris Russell, Pushmeet Kohli, and Philip H. S. Torr. Graph Cut based Inference with Co-occurrence Statistics. In *European Conference on Computer Vision (ECCV)*, September 2010a.

L'ubor Ladický, Paul Sturgess, Chris Russell, Sunando Sengupta, Yalin Bastanlar, William Clocksin, and Philip H. S. Torr. Joint Optimisation for Object Class Segmentation and Dense Stereo Reconstruction. In *British Machine Vision Conference (BMVC)*, 2010b.

Nevena Lazic, Inmar Givoni, Brendan J. Frey, and Parham Aarabi. FLoSS: Facility Location for Subspace Segmentation. In *International Conference on Computer Vision (ICCV)*, 2009.

Victor Lempitsky, Carsten Rother, Stephan Roth, and Andrew Blake. Fusion moves for markov random field optimization. *IEEE Transactions on Pattern Analysis and Machine Inference (TPAMI)*, 32: 1392–1405, August 2010.

Hongdong Li. Two-view Motion Segmentation from Linear Programming Relaxation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

Stan Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer, 1994.

C. Meyers and J.B. Orlin. Very large-scale neighborhood search techniques in timetabling problems. *Practice and Theory of Automated Timetabling VI*, page 24, 2007.

Carl Olsson, Martin Byröd, Niels Christian Overgaard, and Fredrik Kahl. Extending Continuous Cuts: Anisotropic Metrics and Expansion Moves. In *International Conference on Computer Vision*, October 2009.

Thomas Pock, Thomas Schoenemann, Gottfried Graber, Horst Bischof, and Daniel Cremers. A Convex Formulation of Contin-

uous Multi-Label Problems. In *European Conference on Computer Vision (ECCV)*, October 2008.

Thomas Pock, Antonin Chambolle, Horst Bischof, and Daniel Cremers. A Convex Relaxation Approach for Computing Minimal Partitions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009.

Renfrey B. Potts. Some generalized order-disorder transformations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 48:106–109, 1952.

C. Rother, S. Kumar, V. Kolmogorov, and A. Blake. Digital tapestry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*, 2005.

Carsten Rother, Vladimir Kolmogorov, Victor Lempitsky, and Martin Szummer. Optimizing Binary MRFs via Extended Roof Duality. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007.

Güvenç Sahin and Haldun Süral. A review of hierarchical facility location models. *Computers and Operations Research*, 34(8):2310–2331, 2007.

Emre Sefer and Carl Kingsford. Metric Labeling and Semi-metric Embedding for Protein Annotation Prediction. In *Research in Computational Molecular Biology*, 2011.

David B. Shmoys, Éva Tardos, and Karen Aardal. Approximation algorithms for facility location problems. In *ACM Symposium on Theory of Computing (STOC)*, pages 265–274, 1998.

Petter Strandmark and Fredrik Kahl. Parallel and distributed graph cuts by dual decomposition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010.

Zoya Svitkina and Éva Tardos. Facility location with hierarchical facility costs. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2006.

Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A Comparative Study of Energy Minimization Methods for Markov Random Fields. In *European Conference on Computer Vision (ECCV)*, pages 16–29, 2006.

Philip H. S. Torr. Geometric Motion Segmentation and Model Selection. *Philosophical Transactions of the Royal Society A*, pages 1321–1340, 1998.

Phillip H. S. Torr and D. Murray. Stochastic motion clustering. *European Conference on Computer Vision (ECCV)*, 1994.

Olga Veksler. *Efficient Graph-Based Energy Minimization Methods in Computer Vision*. PhD thesis, Cornell University, 1999.

Tomáš Werner. High-arity Interactions, Polyhedral Relaxations, and Cutting Plane Algorithm for Soft Constraint Optimisation (MAP-MRF). In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008.

Oliver J. Woodford, Carsten Rother, and Vladimir Kolmogorov. A Global Perspective on MAP Inference for Low-Level Vision. In *International Conference on Computer Vision (ICCV)*, October 2009.

Jing Yuan and Yuri Boykov. TV-Based Multi-Label Image Segmentation with Label Cost Prior. In *British Machine Vision Conference (BMVC)*, Sept 2010.

Quan Zhou, Tianfu Wu, Wenyu Liu, and Song-Chun Zhu. Scene Parsing by Data-Driven Cluster Sampling. *International Journal of Computer Vision (IJCV)*, 2011. under review.

Song-Chun Zhu and Alan L. Yuille. Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 18(9):884–900, 1996.

A (Proof of Metric Relationships)

Pair (V, π) forms a *tree metric* if V represents an edge-weighted distance in tree π. This means that $V(\alpha, \beta) = d(\alpha, \beta)$ where $d(\alpha, \beta)$ is the sum of edge weights $d_{ij} \geq 0$ along a path from leaf α to leaf β. A tree metric (V, π) is therefore entirely parameterized by its edge weights d_{ij} where $j = \pi(i)$. An r-HST metric is just a tree metric where edge costs get cheaper by a factor of $\frac{1}{r} < 1$ as we descend the tree, *i.e.* $d_{ij} \leq \frac{1}{r} d_{jk}$ for $j = \pi(i), k = \pi(j)$. So, r-HST metrics are a subclass of tree metrics by definition.

[tree metrics $\subset$ h-metrics]: For a tree metric to be an h-metric, d must satisfy (according to Definition 3, page 6)

$$d(\alpha_1, \alpha_2) + d(\beta, \gamma) \;\leq\; d(\alpha_1, \gamma) + d(\beta, \alpha_2) \qquad \forall\, \alpha_1, \alpha_2 \in \mathcal{L}_i,\ \beta, \gamma \in \mathcal{L}_{\pi(i)} \setminus \mathcal{L}_i \tag{19}$$

For each $i \in \mathcal{L} \cup \mathcal{S}$ use shorthand $j = \pi(i)$ and consider that

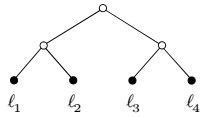
$$d(\alpha_1, \alpha_2) \leq d(\alpha_1, i) + d(i, \alpha_2), \tag{20}$$

$$d(\beta, \gamma) \leq d(\beta, j) + d(j, \beta), \tag{21}$$

$$d(\alpha_1, \gamma) = d(\alpha_1, j) + d(j, \gamma), \tag{22}$$

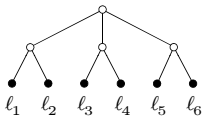
$$d(\beta, \alpha_2) = d(\beta, j) + d(j, \alpha_2). \tag{23}$$

Use inequalities (20) and (21) to replace the left-hand side of (19) and cancel terms with (22) and (23) to get $d(\alpha_1, i) + d(i, \alpha_2) \leq d(\alpha_1, j) + d(j, \alpha_2)$, which is clearly satisfied since $d_{ij} \geq 0$. To see that some (non-h-Potts) h-metrics are not tree metrics, consider the tree and symmetric smoothness cost $V(\cdot, \cdot)$ below.



	ℓ_1	ℓ_2	ℓ_3	ℓ_4
ℓ_1	0	2	2	4
ℓ_2		0	2	2
ℓ_3			0	2
ℓ_4				0

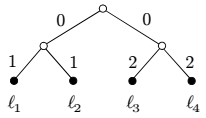
[(h-Potts $\cap$ h-metrics) $\not\subseteq$ tree metrics]: The example below is a simple h-Potts potential which is also an h-metric but is not a tree metric.



	ℓ_1	ℓ_2	ℓ_3	ℓ_4	ℓ_5	ℓ_6
ℓ_1	0	2	4	4	4	4
ℓ_2		0	4	4	4	4
ℓ_3			0	2	4	4
ℓ_4				0	4	4
ℓ_5					0	6
ℓ_6						0

The fact that it is not a tree metric can be verified by setting up a linear program relating edge costs d_{ij} to node costs w_i, and noting that the system is infeasible if $d_{ij} \geq 0$.

[(h-Potts with $w_i \leq w_{\pi(i)}$) $\subset$ (h-Potts $\cap$ tree metrics)]: If node costs $\{w_i\}_{i \in \mathcal{S} \cup \{r\}}$ are non-negative and do not increase as we descend the tree (*i.e.* $w_i \leq w_{\pi(i)}$) then we can construct a tree metric by induction. Given some node $j \in \mathcal{S} \cup \{r\}$, assume we have non-negative edge costs so that, for each child $i \in \mathcal{I}(j)$, $d(\alpha, i) = \frac{1}{2} w_i$ for all $\alpha \in \mathcal{L}_i$. Then we can assign cost $d_{ij} = \frac{1}{2}(w_j - w_i)$ to each child edge of j to get $d(\alpha, j) = \frac{1}{2} w_j$ for all $\alpha \in \mathcal{L}_j$. Since $w_i \leq w_j$ we also have a tree metric for subtree j. It is not necessary to assume $w_i \leq w_{\pi(i)}$ for an h-Potts potential to be a tree metric, as the example below demonstrates (edge costs are shown on the tree).



	ℓ_1	ℓ_2	ℓ_3	ℓ_4
ℓ_1	0	2	3	3
ℓ_2		0	3	3
ℓ_3			0	4
ℓ_4				0

[(h-Potts $\cap$ r-HST metrics) $\subset$ (h-Potts with $w_i \leq w_{\pi(i)}$)]: As described by Kumar and Koller (2009), an r-HST metric has a constant edge cost d_{ij} between node j and all of its children $i \in \mathcal{I}(j)$. In other words, an r-HST metric is actually parameterized by one common 'edge' cost per parent node $\{d_j\}_{j \in \mathcal{S} \cup \{r\}}$, where $0 \leq d_i \leq \frac{1}{r} d_{\pi(i)}$ for all $i \in \mathcal{S}$. It is easy to see that, for an h-Potts potential to be an r-HST metric, it must have $w_i = w_j - 2d_j$ where $j = \pi(i)$. Thus $d_j \geq 0$ implies $w_i \leq w_j$. Also note that $r > 1$ means quantity $w_j - w_i$ must decrease at a *rate* of $\frac{1}{r}$ as we descend the tree. ■

B (Proof of Theorem 5)

Proof Without loss of generality we assume that all weights $w_{pq} = 1$. Consider any local minimum $\hat{f}^j$ computed by h-fusion at internal node j, and let us choose some child node $i \in \mathcal{I}(j)$. We first define a useful set of pixels for i with respect to a global optimum f^*

$$\mathcal{P}_i \;=\; \{\, p : f^*_p \in \mathcal{L}_i \,\}.$$

This set contains all pixels assigned a label within subtree i, and so for any other child $i' \neq i$ we know that $\mathcal{P}_i \cap \mathcal{P}_{i'} = \emptyset$.

We can produce a labeling $\hat{f}^{j\otimes i}$ within one h-fusion move from local minimum $\hat{f}^j$ as follows:

$$\hat{f}^{j\otimes i}_p \;=\; \begin{cases} \hat{f}^i_p & \text{if } p \in \mathcal{P}_i \\ \hat{f}^j_p & \text{otherwise.} \end{cases}$$

Since each $\hat{f}^j$ is known to be a local optimum w.r.t. expansion moves for each $i \in \mathcal{I}(j)$ we know that

$$E(\hat{f}^j) \;\leq\; E(\hat{f}^{j\otimes i}). \tag{24}$$

The general strategy to use (24) for different i to build an inequality that is ultimately of the form $E(\hat{f}^j) \leq E(f^*) + \text{error}$. This will be achieved by breaking the energy terms in E into parts in such a way that a recursive inequality can be established. The recursive inequality will then be expanded until all terms can be bounded relative to $E(f^*)$.

Let $E(\cdot)|_{\mathcal{A}}$ denote a restriction of the summands of energy (1) to only the following terms:

$$E(f)|_{\mathcal{A}} \;=\; \sum_{p\in\mathcal{A}} D_p(f_p) \;+\; \sum_{pq\in\mathcal{A}} V(f_p, f_q).$$

We separate the unary and pairwise terms of $E(f)$ via interior, exterior, and boundary sets with respect to pixels $\mathcal{P}_i$:

$$\begin{aligned} \mathcal{A}_i &= \mathcal{P}_i \qquad \cup \{\, pq \in \mathcal{N} \colon p, q \in \mathcal{P}_i \,\} \\ \overline{\mathcal{A}}_i &= \mathcal{P} \setminus \mathcal{P}_i \cup \{\, pq \in \mathcal{N} \colon p, q \notin \mathcal{P}_i \,\} \\ \partial\mathcal{A}_i &= \qquad\qquad \{\, pq \in \mathcal{N} \colon p \in \mathcal{P}_i, q \notin \mathcal{P}_i \,\}. \end{aligned}$$

Let $E_H(f)$ denote the total label cost incurred by a labeling f, *i.e.* the sum of label cost terms. The following facts now hold:

$$E(\hat{f}^{j\otimes i})|_{\mathcal{A}_i} = E(\hat{f}^i)|_{\mathcal{A}_i} \tag{25}$$

$$E(\hat{f}^{j\otimes i})|_{\overline{\mathcal{A}}_i} = E(\hat{f}^j)|_{\overline{\mathcal{A}}_i}. \tag{26}$$

We have not accounted for the label costs yet, but for simplicity we break this proof into two parts: part 1 derives the coefficient c related to smoothness costs V, and part 2 derives the coefficient c_2 related to label costs H. For part 1 we can assume there are no label costs at all.

Part 1. Derive coefficient c for smoothness cost bound

Using (25) and (26) we can cancel out all the $\overline{\mathcal{A}}_i$ terms and rewrite (24) as

$$E(\hat{f}^j)|_{\mathcal{A}_i\cup\partial\mathcal{A}_i} \;\leq\; E(\hat{f}^i)|_{\mathcal{A}_i} + E(\hat{f}^{j\otimes i})|_{\partial\mathcal{A}_i} \tag{27}$$

For each $i \in \mathcal{I}(j)$ inequality (27) contains a subset of all the energy terms in $E(\hat{f}^j)|_{\mathcal{A}_j}$ pertaining to pixels $\mathcal{P}_i$. Let $\mathcal{I}^* = \{i \in \mathcal{I}(j) : \mathcal{P}_i \neq \emptyset\}$ be the set of children whose sub-trees contain a label used by f^*. If we sum inequality (27) over all $i \in \mathcal{I}^*$, the left-hand side will contain *all* the terms in $E(\hat{f}^j)|_{\mathcal{A}_j}$ (and more). Adding up all the left-hand sides we have

$$\begin{aligned} &\sum_{i\in\mathcal{I}^*} E(\hat{f}^j)|_{\mathcal{A}_i\cup\partial\mathcal{A}_i} \\ &\quad= E(\hat{f}^j)|_{\mathcal{A}_j\cup\partial\mathcal{A}_j} + \sum_{i\in\mathcal{I}^*} E(\hat{f}^j)|_{\partial\mathcal{A}_i\setminus\partial\mathcal{A}_j} \qquad (28) \\ &\quad\geq E(\hat{f}^j)|_{\mathcal{A}_j}. \end{aligned}$$

Using (28) and likewise adding up the right-hand sides of (27) we have

$$E(\hat{f}^j)|_{\mathcal{A}_j} \leq \sum_{i\in\mathcal{I}^*} E(\hat{f}^i)|_{\mathcal{A}_i} + E(\hat{f}^{j\otimes i})|_{\partial\mathcal{A}_i} \tag{29}$$

$$= \sum_{i\in\mathcal{I}^*} E(\hat{f}^i)|_{\mathcal{A}_i} + E(\hat{f}^{j\otimes i})|_{\partial\mathcal{A}_i\cap\partial\mathcal{A}_j} + E(\hat{f}^{j\otimes i})|_{\partial\mathcal{A}_i\setminus\partial\mathcal{A}_j} \tag{30}$$

$$= \sum_{i\in\mathcal{I}^*} E(\hat{f}^i)|_{\mathcal{A}_i} + \sum_{pq\in\partial\mathcal{A}_i\cap\partial\mathcal{A}_j} V(\hat{f}^i_p, \hat{f}^j_q) \;+\; \sum_{pq\in\partial\mathcal{A}_i\setminus\partial\mathcal{A}_j} V(\hat{f}^i_p, \hat{f}^j_q) \tag{31}$$

The first important observation about (31) is that each $E(\hat{f}^i)|_{\mathcal{A}_j}$ term on the right-hand side can be substituted by recursively applying the inequality itself. We can recursively substitute, branching further and further down the tree, until the path finally stops at a leaf $\ell \in \mathcal{L}$ giving us base case $E(\hat{f}^\ell)|_{\mathcal{A}_\ell} = \sum_{p\in\mathcal{P}_\ell} D_p(f^*_p)$. The sets $\{\mathcal{P}_\ell\}_{\ell\in\mathcal{L}}$ must be disjoint and their union is $\mathcal{P}_j$ so expression (31), when fully expanded, becomes roughly

$$= \sum_{p\in\mathcal{A}_j} D_p(f^*_p) \;+ \text{pairwise terms of the form } V(\hat{f}^i_p, \hat{f}^{\pi(i)}_q). \tag{32}$$

The second observation about (31) is that each edge pq on an outer boundary $\partial\mathcal{A}_i \cap \partial\mathcal{A}_j$ appears once in the sum over $\mathcal{I}^*$ whereas each edge on an interior boundary $\partial\mathcal{A}_i \setminus \partial\mathcal{A}_j$ appears twice: once for $p \in \mathcal{A}_i$ and once for some $q \in \mathcal{A}_{i'}$. By careful accounting we collect all the $V(\hat{f}^i_p, \hat{f}^{\pi(i)}_q)$ terms generated by the recursive substitution and express (31) as[3]

$$\begin{aligned} &= \sum_{p\in\mathcal{A}_j} D_p(f^*_p) \\ &\quad + \sum_{pq\in\mathcal{A}_j} \left(\sum_{i\in\mathcal{J}(f^*_p; f^*_q)} V(\hat{f}^i_p, \hat{f}^{\pi(i)}_q) \;+\; \sum_{i\in\mathcal{J}(f^*_q; f^*_p)} V(\hat{f}^{\pi(i)}_p, \hat{f}^i_q) \right) \end{aligned} \tag{33}$$

where we define $\mathcal{J}(\ell; \ell')$ to be the set of nodes along the path from a label $\ell \in \mathcal{L}$ up to, but not including, the lowest common ancestor of ℓ and ℓ', namely

$$\mathcal{J}(\ell; \ell') = \left\{\ell, \pi(\ell), \ldots, \pi^{n-1}(\ell)\right\} \quad \text{where} \quad \pi^n(\ell) = \text{lca}(\ell, \ell').$$

All that remains is to bound each $V(\hat{f}^i_p, \hat{f}^{\pi(i)}_q)$ in terms of $V(f^*_p, f^*_q)$ using b_i described in Definition 7. From now on we use $a_i = V^{\max}_i$ and $d_i = V^{\min}_i$ as shorthand. For a particular edge pq shown in (33) we must have each $V(\hat{f}^i_p, \hat{f}^{\pi(i)}_q) \leq a_{\pi(i)}$ and so their sum is

$$\sum_{i\in\mathcal{J}(f^*_p; f^*_q)} V(\hat{f}^i_p, \hat{f}^{\pi(i)}_q) \;\leq\; a_{\pi(f^*_p)} + \ldots + a_{\text{lca}(f^*_p, f^*_q)} \;\leq\; b_{\text{lca}(f^*_p, f^*_q)}. \tag{34}$$

We also know that $V(f^*_p, f^*_q) \geq d_{\text{lca}(f^*_p, f^*_q)}$ so we can use ratio $\frac{b_{\text{lca}(f^*_p, f^*_q)}}{d_{\text{lca}(f^*_p, f^*_q)}}$ to bound the approximation error at each edge pq appearing in (33), giving upper-bound

$$\leq \sum_{p\in\mathcal{A}_j} D_p(f^*_p) + \sum_{pq\in\mathcal{A}_j} \left(2\, \frac{b_{\text{lca}(f^*_p, f^*_q)}}{d_{\text{lca}(f^*_p, f^*_q)}} V(f^*_p, f^*_q) \right). \tag{35}$$

[3] Due to our assumption that V is semi-metric and so $V(\ell, \ell) = 0$, we can simply sum over all $pq \in \mathcal{A}_j$ instead of only where $f^*_p \neq f^*_q$.

If j is the root of the tree, then $\{p \in \mathcal{A}_j\} = \mathcal{P}$ and $\{pq \in \mathcal{A}_j\} = \mathcal{N}$. Using the fact that any ratio $\frac{b_i}{d_i}$ is bounded from above by quantity c (Definition 7) we arrive at

$$\leq \sum_{p \in \mathcal{P}} D_p(f^*_p) \; + \; 2c \sum_{pq \in \mathcal{N}} V(f^*_p, f^*_q) \tag{36}$$

$$= E(f^*) \; + \; (2c-1) \sum_{pq \in \mathcal{N}} V(f^*_p, f^*_q) \tag{37}$$

$$\leq 2cE(f^*). \tag{38}$$

This completes the proof of Part 1. When there are only smoothness costs, $E(\hat{f}) \leq 2cE(f^*)$ where $\hat{f}$ is the labeling generated at the root of the tree.

Part 2. Derive coefficient c_2 for label cost bound

We now revisit from (27) onward but with the assumption that there are hierarchical label costs.

Let $E_H(f)$ denote the total label cost incurred by a labeling f, *i.e.* the sum of label cost terms. We can bound the label cost $E_H(\hat{f}^{j \otimes i})$ of our fused labeling by

$$E_H(\hat{f}^{j \otimes i}) \; \leq \; E_H(\hat{f}^j) \; + \sum_{\substack{L \subseteq \mathcal{L} \setminus \hat{\mathcal{L}}_j \\ L \cap \hat{\mathcal{L}}_i \neq \emptyset}} H(L) \tag{39}$$

where $\hat{\mathcal{L}}_j$ and $\hat{\mathcal{L}}_i$ are the sets of unique labels appearing in $\hat{f}^j$ and $\hat{f}^i$ respectively.

Recall from Part 1 that, looking at the key inequality (24), we can break the energy terms on each side into parts based on sets $\mathcal{A}_i, \overline{\mathcal{A}}_i$, and $\partial\mathcal{A}_i$. Because $E(\hat{f}^{j \otimes i})|_{\overline{\mathcal{A}}_i} = E(\hat{f}^j)|_{\overline{\mathcal{A}}_i}$ these terms cancel out, and we can substitute $E(\hat{f}^{j \otimes i})|_{\mathcal{A}_i} = E(\hat{f}^i)|_{\mathcal{A}_i}$. Along with bound (39) and canceling the $E_H(\hat{f}^j)$ terms we can now rewrite (24) as

$$E(\hat{f}^j)|_{\mathcal{A}_i \cup \partial\mathcal{A}_i} \; \leq \; E(\hat{f}^i)|_{\mathcal{A}_i} + E(\hat{f}^{j \otimes i})|_{\partial\mathcal{A}_i} + \sum_{\substack{L \subseteq \mathcal{L} \setminus \hat{\mathcal{L}}_j \\ L \cap \hat{\mathcal{L}}_i \neq \emptyset}} H(L) \tag{40}$$

Again, let $\mathcal{I}^* = \{i \in \mathcal{I}(j) : \mathcal{P}_i \neq \emptyset\}$ be the set of child nodes that contain a label used by f^* in their subtree. We sum inequality (40) over all $i \in \mathcal{I}^*$ to arrive at a recursive expression, this time incorporating errors incurred by label costs. The key observation is that a particular label cost $H(L)$ appears once on the right-hand side for each element in the set $\mathcal{I}^*_L = \{i \in \mathcal{I}^* : L \cap \hat{\mathcal{L}}_i \neq \emptyset\}$. The sum of inequalities (40) thus implies

$$E(\hat{f}^j)|_{\mathcal{A}_j} \leq \left(\sum_{i \in \mathcal{I}^*} E(\hat{f}^i)|_{\mathcal{A}_i} + E(\hat{f}^{j \otimes i})|_{\partial\mathcal{A}_i} \right) + \sum_{L \subseteq \mathcal{L} \setminus \hat{\mathcal{L}}_j} H(L) \cdot |\mathcal{I}^*_L| \tag{41}$$

where the quantity in parentheses is identical to that of Part 1.

The above inequality can be recursively expanded for each $E(\hat{f}^i)|_{\mathcal{A}_i}$ until the recursion stops at a label used by f^*. We already know that, after recursive substitution, the quantity in parentheses is bounded by (35). We now must bound the total label cost accumulated by recursive application of (41). The central question is whether a particular subset L that appears in (41) with $|\mathcal{I}^*_L| > 0$ for node j can appear *again* when we recursively substitute the children $i \in \mathcal{I}^*$. If the answer were 'yes' then each label cost $H(L)$ could appear more than $|\mathcal{I}^*_L|$ total times by the end of recursive expansion, leading to a worse bound. Fortunately, Lemma 1 (after this proof) says that this is not the case; each L appearing in the sum for j and child i (40) can never reappear in the sums for i or its children.

From now on we assume j is the root of the tree structure, and so $\hat{f}^j = \hat{f}$, *i.e.* the final labeling output by h-fusion. If we let $\mathcal{H}^*$ denote the set of all subsets L generated by recursive substitution of (41), we can thereby write

$$E(\hat{f}) \; \leq \; (36) \; + \sum_{L \in \mathcal{H}^*} H(L) \cdot |\mathcal{I}^*_L| \tag{42}$$

Note that the left-hand side of (42) is still $E(\hat{f}^j)|_{\mathcal{A}_j}$ which does *not* include the label costs incurred by $\hat{f}^j$. By adding $E_H(\hat{f}^j)$ to both sides we have $E(\hat{f}^j)|_{\mathcal{A}_j} + E_H(\hat{f}^j) = E(\hat{f})$ on the left side, giving a new inequality below.

$$E(\hat{f}) \; \leq \; (36) \; + E_H(\hat{f}) + \sum_{L \in \mathcal{H}^*} H(L) \cdot |\mathcal{I}^*_L| \tag{43}$$

$$= \; (37) \; + E_H(\hat{f}) - E_H(f^*) + \sum_{L \in \mathcal{H}^*} H(L) \cdot |\mathcal{I}^*_L| \tag{44}$$

All that is left is to re-group the summands in the last three terms (the label cost terms) in a way that proves our theorem. First we rewrite the three sums more explicitly, using $\hat{\mathcal{L}}$ and $\mathcal{L}^*$ to denote the unique labels used by $\hat{f} = \hat{f}^j$ and f^* respectively.

$$\sum_{\substack{L \in \mathcal{H} \\ L \cap \hat{\mathcal{L}} \neq \emptyset}} H(L) \; - \sum_{\substack{L \in \mathcal{H} \\ L \cap \mathcal{L}^* \neq \emptyset}} H(L) \; + \sum_{L \in \mathcal{H}^*} H(L) \cdot |\mathcal{I}^*_L|$$

$$= \sum_{\substack{L \in \mathcal{H} \\ L \cap \mathcal{L}^* = \emptyset \\ L \cap \hat{\mathcal{L}} \neq \emptyset}} H(L) \; - \sum_{\substack{L \in \mathcal{H} \\ L \cap \mathcal{L}^* \neq \emptyset \\ L \cap \hat{\mathcal{L}} = \emptyset}} H(L) \; + \sum_{L \in \mathcal{H}^*} H(L) \cdot |\mathcal{I}^*_L| \tag{45}$$

First note that if $|\mathcal{I}^*_L| > 1$ then this means $L \supset \mathcal{L}_i$ for some $\mathcal{L}_i \cap \mathcal{L}^* \neq \emptyset$ and so $L \cap \mathcal{L}^* \neq \emptyset$ also. We can break the last sum in (45) into two parts based on whether $L \cap \mathcal{L}^* \neq \emptyset$.

$$= \sum_{\substack{L \in \mathcal{H} \\ L \cap \mathcal{L}^* = \emptyset \\ L \cap \hat{\mathcal{L}} \neq \emptyset}} H(L) + \sum_{\substack{L \in \mathcal{H}^* \\ L \cap \mathcal{L}^* = \emptyset}} H(L) - \sum_{\substack{L \in \mathcal{H} \\ L \cap \mathcal{L}^* \neq \emptyset \\ L \cap \hat{\mathcal{L}} = \emptyset}} H(L) + \sum_{\substack{L \in \mathcal{H}^* \\ L \cap \mathcal{L}^* \neq \emptyset}} H(L) \cdot |\mathcal{I}^*_L| \tag{46}$$

We can also show that $L \in \mathcal{H}^* \; \Rightarrow \; L \cap \hat{\mathcal{L}} = \emptyset$ as follows. If $L \in \mathcal{H}^*$ then there must be some node i such that $L \cap \hat{\mathcal{L}}_i = \emptyset$ and $L \subset \mathcal{L}_i$. We know from (59) in Lemma 1 that $\hat{\mathcal{L}} \cap \mathcal{L}_i \subseteq \hat{\mathcal{L}}_i$, so this implies $\emptyset = L \cap \hat{\mathcal{L}}_i \supseteq L \cap (\hat{\mathcal{L}} \cap \mathcal{L}_i) = L \cap \hat{\mathcal{L}}$. This means the two leftmost sums of (46) have disjoint L and can be bounded by simply $\sum_{L \in \mathcal{H}} H(L)$. It furthermore implies that, for every L appearing in the rightmost sum of (46), the same L must appear in the negative sum. Putting these together we have upper bound on label costs

$$\leq \sum_{L \in \mathcal{H}} H(L) \; + \sum_{\substack{L \in \mathcal{H}^* \\ L \cap \mathcal{L}^* \neq \emptyset}} H(L) \cdot (|\mathcal{I}^*_L| - 1) \tag{47}$$

$$\leq \sum_{L \in \mathcal{H}} H(L) \; + c_2 \cdot \sum_{\substack{L \in \mathcal{H}^* \\ L \cap \mathcal{L}^* \neq \emptyset}} H(L) \tag{48}$$

$$\leq \sum_{L \in \mathcal{H}} H(L) \; + c_2 E_H(f^*) \tag{49}$$

We can therefore revise bound (44) to

$$E(\hat{f}) \; \leq \; (37) \; + \; c_2 E_H(f^*) \; + \sum_{L \in \mathcal{H}} H(L) \tag{50}$$

$$\leq \; E(f^*) + (2c-1)E(f^*) + c_2 E(f^*) + \sum_{L \in \mathcal{H}} H(L) \tag{51}$$

$$\leq \; (2c + c_2)E(f^*) + \sum_{L \in \mathcal{H}} H(L) \tag{52}$$

Inequality (52) is main result of our Theorem. ■

Lemma 1 *If label subset L appears in the summand of* (40) *for node j and child i, then L does not appear in the summands of* (40) *for any $k \in \mathrm{subtree}(i)$.*

Proof To be clear, let us restate the claim more formally. Let $\mathcal{H}^{j \otimes i}$ denote all subsets L appearing in the label cost summands of (40) when applied to node j and child i, *i.e.*

$$\mathcal{H}^{j \otimes i} \stackrel{\mathrm{def}}{=} \{\, L \,:\, L \cap \hat{\mathcal{L}}_j = \emptyset,\, L \cap \hat{\mathcal{L}}_i \neq \emptyset \,\} \tag{53}$$

We must prove that $L \in \mathcal{H}^{j \otimes i} \;\Rightarrow\; L \notin \mathcal{H}^{k \otimes l}$ for any $k \in \mathrm{subtree}(i)$ and $l \in \mathcal{I}(k)$.

First note that for each $L \in \mathcal{H}^{j \otimes i}$ we have

$$L \cap \hat{\mathcal{L}}_j = \emptyset \;\Rightarrow\; L \not\supseteq \mathcal{L}_j \tag{54}$$

$$L \cap \hat{\mathcal{L}}_i \neq \emptyset \;\Rightarrow\; L \cap \mathcal{L}_i \neq \emptyset \tag{55}$$

By the hierarchical label cost assumption (Definition 4) we can use (54) and (55) to conclude that $L \in \mathcal{H}^{j \otimes i} \;\Rightarrow\; L \subset \mathcal{L}_j$.

Now consider the set $\mathcal{H}^{j \otimes i} \cap \mathcal{H}^{k \otimes l}$. By the definition (53) an element L of this joint set must satisfy at least the following conditions:

$$L \cap \hat{\mathcal{L}}_i \neq \emptyset \tag{56}$$

$$L \cap \hat{\mathcal{L}}_k = \emptyset \tag{57}$$

$$L \subset \mathcal{L}_k. \tag{58}$$

However, no subset L can satisfy all three conditions, as we now show. In the h-fusion algorithm, if $\hat{f}^i$ contains a label $\ell \in \mathcal{L}_k$, then $\hat{f}^k$ *must* contain ℓ as well—after all, there is no other way that a label in $\mathcal{L}_k$ could have propagated up to $\hat{f}^i$. This relation can be restated as

$$\hat{\mathcal{L}}_i \cap \mathcal{L}_k \subseteq \hat{\mathcal{L}}_k \quad \forall\, k \in \mathrm{subtree}(i) \tag{59}$$

Starting from (56) we can say

$$\begin{aligned} & L \cap \hat{\mathcal{L}}_i \neq \emptyset \\ \Rightarrow\; & L \cap (\hat{\mathcal{L}}_i \cap \mathcal{L}_k) \neq \emptyset && \text{by (58)} \\ \Rightarrow\; & L \cap (\hat{\mathcal{L}}_k) \neq \emptyset && \text{by (59)} \end{aligned}$$

which contradicts requirement (57). Thus $\mathcal{H}^{j \otimes i} \cap \mathcal{H}^{k \otimes l} = \emptyset$ for all $k \in \mathrm{subtree}(i)$ and so L cannot reappear. ■