# Fast Approximate Energy Minimization with Label Costs

Andrew Delong* Anton Osokin† Hossam N. Isack* Yuri Boykov*

*Department of Computer Science, University of Western Ontario

†Department of Computational Mathematics and Cybernetics, Moscow State University

## Abstract

*The $\alpha$-expansion algorithm [7] has had a significant impact in computer vision due to its generality, effectiveness, and speed. Thus far it can only minimize energies that involve unary, pairwise, and specialized higher-order terms. Our main contribution is to extend $\alpha$-expansion so that it can simultaneously optimize "label costs" as well. An energy with label costs can penalize a solution based on the set of labels that appear in it. The simplest special case is to penalize the number of labels in the solution.*

*Our energy is quite general, and we prove optimality bounds for our algorithm. A natural application of label costs is multi-model fitting, and we demonstrate several such applications in vision: homography detection, motion segmentation, and unsupervised image segmentation. Our C++/MATLAB implementation is publicly available.*

## 1. Some Useful Regularization Energies

In a labeling problem we are given a set of observations $\mathcal{P}$ (pixels, features, data points) and a set of labels $\mathcal{L}$ (categories, geometric models, disparities). The goal is to assign each observation $p \in \mathcal{P}$ a label $f_p \in \mathcal{L}$ such that the joint labeling $f$ minimizes some objective function $E(f)$.

Most labeling problems in computer vision are ill-posed and in need of regularization, but the most useful regularizers often make the problem NP-hard. Our work is about how to effectively optimize two such regularizers: a preference for fewer labels in the solution, and a preference for spatial smoothness. Figure 1 suggests how these criteria cooperate to give clean results. Surprisingly, there is no good algorithm to optimize their combination.[1] Our main contribution is a way to simultaneously optimize both of these criteria inside the powerful $\alpha$-expansion algorithm [7].

**Label costs.** Start from a basic (unregularized) energy $E(f) = \sum_p D_p(f_p)$, where optimal $f_p$ can each be determined independently from the 'data costs'. Suppose, however, that we wish to explain the observations using as few unique labels as necessary. We can introduce *label costs* into $E(f)$ to penalize each unique label that appears in $f$:

$$E(f) = \sum_{p\in\mathcal{P}} D_p(f_p) + \sum_{l\in\mathcal{L}} h_l \cdot \delta_l(f) \qquad (1)$$

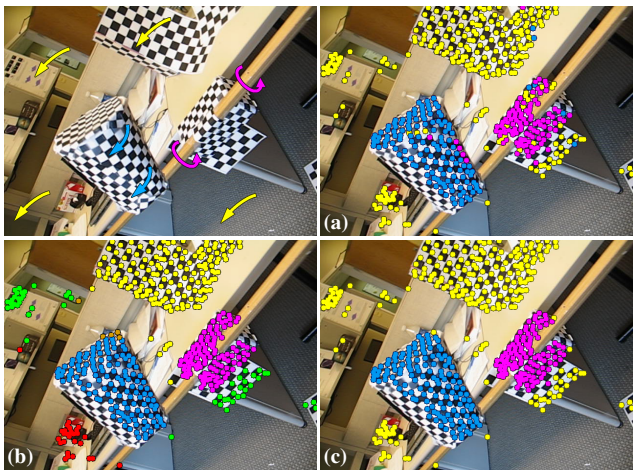



Figure 1. Motion segmentation on the 1RT2RCR sequence [36]. Energy (1) finds 3 dominant motions (a) but labels many points incorrectly. Energy (2) gives coherent segmentations (b) but finds redundant motions. Our energy combines the best of both (c).

where $h_l$ is the non-negative label cost of label $l$, and $\delta_l(\cdot)$ is the corresponding indicator function

$$\delta_l(f) \stackrel{\text{def}}{=} \begin{cases} 1 & \exists p : f_p = l \\ 0 & \text{otherwise.} \end{cases}$$

Energy (1) balances data costs against label costs in a formulation equivalent to the well-studied *uncapacitated facility location* (UFL) problem. Li [26] recently posed multi-model fitting in terms of UFL. For multi-model fitting, where each label corresponds to a candidate model, label costs penalize overly-complex models, preferring to explain the data with fewer, cheaper labels (see Figure 1a).

**Smooth costs.** Spatial smoothness is a standard regularizer in computer vision. The idea here is that groups of observations are often known *a priori* to be positively correlated, and should thus be encouraged to have similar labels. Neighbouring image pixels are a classic example of this. Such pairwise priors can be expressed by the energy

$$E(f) = \sum_{p\in\mathcal{P}} D_p(f_p) + \sum_{pq\in\mathcal{N}} V_{pq}(f_p, f_q) \qquad (2)$$

*† The authors assert equal contribution and thus joint first authorship.

[1] See Addendum on page 13, dated April 25, 2010.

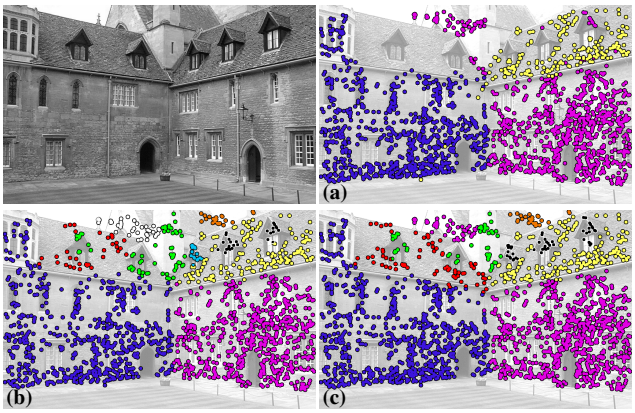



Figure 2. Planar homography detection on VGG (Oxford) Merton College 1 image (right view). Energy (1) finds reasonable parameters for only the strongest 3 models shown in (a), and still assigns a few incorret labels. Energy (2) finds reasonable clusters (b) but fits 9 models, some of which are redundant (nearly co-planar). Our energy ($\star$) finds both good parameters and labels (c) for 7 models.

where each $V_{pq}$ penalizes $f_p \neq f_q$ in some manner. If each $V_{pq}$ defines a metric, then minimizing (2) is known as the *metric labeling* problem [7] and can be optimized effectively with the $\alpha$-expansion algorithm.

This regularizer prefers coherent segmentations, but has no incentive to combine non-adjacent segments and thus a tendency to suggest redundant labels in multi-model fitting (see Figure 1b). Still, spatial smoothness priors are important for a wide array of vision applications.

**Our combined energy.** We propose a discrete energy that essentially combines the UFL and metric labeling problems.

$$E(f) = \overbrace{\sum_{p\in\mathcal{P}} D_p(f_p)}^{\text{data cost}} + \overbrace{\sum_{pq\in\mathcal{N}} V_{pq}(f_p,f_q)}^{\text{smooth cost}} + \overbrace{\sum_{L\subseteq\mathcal{L}} h_L\cdot\delta_L(f)}^{\text{label cost}} \quad (\star)$$

where the indicator function $\delta_L(\cdot)$ is now defined on label subset $L$ as

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

Our energy actually makes a generalization from label costs $h_l$ to label *subset* costs $h_L$, but one can imagine basic per-label costs throughout for simplicity.

Energy ($\star$) balances two demonstrably important regularizers, as suggested by Figure 1c. Figures 2 and 3 show other vision applications where our combined energy makes sense. Section 2 presents our extension to $\alpha$-expansion and corresponding optimality bounds. Section 3 describes a multi-model fitting algorithm based on our energy, and Section 4 discusses connections to standard *expectation maximization* (EM) and $K$-means. Section 5 details our experimental setup. Section 6 discusses applications of high-order label costs, more related works, and possible extensions.

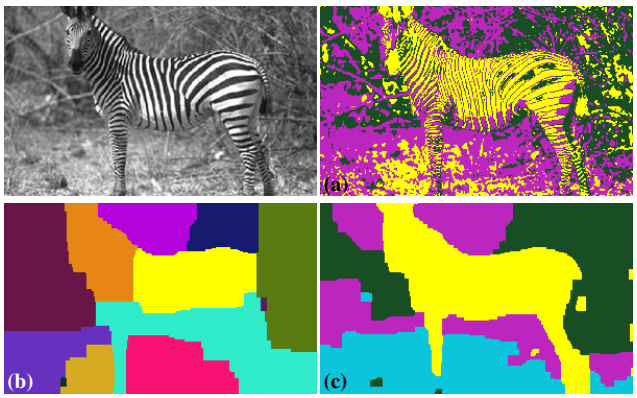



Figure 3. Unsupervised segmentation using histogram models. Energy (1) clusters in colour space, so segments (a) are incoherent. Energy (2) clusters over pixels and must either over-segment or over-smooth (b), just as in [41]. Our energy ($\star$) balances these criteria (c) and corresponds to Zhu & Yuille [42] for segmentation.

## 2. Fast Algorithms to Minimize ($\star$)

Our main technical contribution is to extend the well-known $\alpha$-expansion algorithm [7] to incorporate label costs at each expansion (Section 2.1) and prove new optimality guarantees (Section 2.2). Section 2.3 reviews known results for the 'easy' case (1) with only data and per-label costs.

### 2.1. Expansion moves with label costs

Since minimizing energy ($\star$) is NP-hard for $|\mathcal{L}| \geq 3$, the $\alpha$-expansion algorithm [7] iteratively 'moves' from some current labeling $f'$ to a better one until convergence. Specifically, at each step, some label $\alpha \in \mathcal{L}$ is chosen and variables $f_p$ are simultaneously given a *binary* choice to either stay as $f_p = f'_p$ or switch to $f_p = \alpha$. This binary step is called *expansion* because only the $\alpha$ label can grow and, if each $V_{pq}$ is a metric, the best possible expansion is computed by a single graph cut.

Let $f = \{f_1, \ldots, f_n\}$ and let $f^\alpha$ denote any feasible $\alpha$-expansion w.r.t. current labeling $f'$. The possible labelings $f^\alpha$ can be expressed one-to-one with binary indicator variables $\mathbf{x} = \{x_1, \ldots, x_n\}$ by defining

$$\begin{array}{rcl} x_p = 0 & \iff & f^\alpha_p = f'_p \\ x_p = 1 & \iff & f^\alpha_p = \alpha. \end{array} \quad (3)$$

Let $E^\alpha(\mathbf{x})$ be the energy corresponding to encoding (3) relative to $f'$. The $\alpha$-expansion algorithm computes an optimum $\mathbf{x}^*$, and thereby $f^\alpha$, by a single graph cut.

For example, suppose energy $E(f)$ is such that the optimal expansion w.r.t. labeling $f'$ is $f^\alpha$:

$$f' = \boxed{\beta}\boxed{\alpha}\boxed{\gamma}\boxed{\gamma}\boxed{\beta}\boxed{\beta} \;\rightarrow\; \begin{array}{l} \boxed{\alpha}\boxed{\alpha}\boxed{\alpha}\boxed{\gamma}\boxed{\beta}\boxed{\beta} = f^\alpha \\ \boxed{1}\boxed{\underline{1}}\boxed{1}\boxed{0}\boxed{0}\boxed{0} = \mathbf{x}^* \end{array} \quad (4)$$

where $\underline{1}$ means $x_2$ is fixed to 1. Here only $f_1$ and $f_3$ changed to label $\alpha$ while the rest preferred to keep their labels. The $\alpha$-expansion algorithm iterates the above binary step until finally $E^\alpha(\mathbf{x}') = E^\alpha(\mathbf{x}^*)$ for all $\alpha \in \mathcal{L}$.

**Encoding label costs.** The energy in example (4) was such that $f_5$ and $f_6$ preferred to stay as label $\beta$ rather than switch to $\alpha$. Suppose we want to introduce a cost $h_\beta > 0$ that is added to $E(f)$ if and only if there exists some $f_p = \beta$. This would encourage label $\alpha$ to absorb the entire region that $\beta$ occupies in $f'$. If $h_\beta$ is large enough, the optimal $\alpha$-expansion move would also change $f_5$ and $f_6$:

$$f' = \begin{array}{|c|c|c|c|c|c|c|}\hline \beta & \alpha & \gamma & \gamma & \beta & \beta \\ \hline \end{array} \;\rightarrow\; \begin{array}{|c|c|c|c|c|c|}\hline \alpha & \alpha & \alpha & \gamma & \alpha & \alpha \\ \hline \end{array} = f^\alpha \qquad (5)$$
$$\begin{array}{|c|c|c|c|c|c|}\hline 1 & \underline{1} & 1 & 0 & 1 & 1 \\ \hline \end{array} = \mathbf{x}^*$$

Our main algorithmic contribution is a way to encode such label costs into the expansion step and thereby encourage solutions that use fewer labels.

Energy $E^\alpha(\mathbf{x})$, when expressed as a multilinear polynomial, is a sum of linear and quadratic terms over $\mathbf{x}$. For the specific example (5), we can encode cost $h_\beta$ in $E^\alpha$ by simply adding $h_\beta - h_\beta x_1 x_5 x_6$ to the binary energy. Because this specific term is cubic and $h_\beta \geq 0$, it can be optimized by a single graph cut using the construction in [22].

To encode general label costs for arbitray $L \subseteq \mathcal{L}$ and $f'$, we must optimize the modified expansion energy

$$E^\alpha_h(\mathbf{x}) = E^\alpha(\mathbf{x}) + \sum_{\substack{L \subseteq \mathcal{L} \\ L \cap \mathcal{L}' \neq \emptyset}} \left( h_L - h_L \prod_{p \in \mathcal{P}_L} x_p \right) + C^\alpha(\mathbf{x}) \qquad (6)$$

where set $\mathcal{L}'$ contains the unique labels in the current labeling $f'$, and set $\mathcal{P}_L = \{ p : f'_p \in L \}$. Term $C^\alpha$ simply corrects for the case when $\alpha \notin \mathcal{L}'$ and is discussed later.

Each product term in (6) adds a higher-order clique $\mathcal{P}_L$ beyond the standard $\alpha$-expansion energy $E^\alpha(\mathbf{x})$. Freedman and Drineas [14] generalized the graph construction of [22] to handle terms $c \prod_p x_p$ of arbitrary degree when $c \leq 0$. This means we can transform each product seen in (6) into a sum of quadratic and linear terms that graph cuts can still optimize globally. The transformation for a particular label subset $L \subseteq \mathcal{L}$ with $|\mathcal{P}_L| \geq 3$ is

$$-h_L \prod_{p \in \mathcal{P}_L} x_p = \min_{y_L \in \{0,1\}} h_L \left[ (|\mathcal{P}_L| - 1) y_L - \sum_{p \in \mathcal{P}_L} x_p y_L \right] \qquad (7)$$

where $y_L$ is an auxiliary variable that must be optimized alongside $\mathbf{x}$ whenever $h_L > 0$. Since each $x_p y_L$ term has non-positive coefficient, the overall binary energy can be minimized by a single graph cut [5].

To encode the potential (7) into an $s$-$t$ min-cut graph construction, we reparameterize the right-hand side such that each quadratic monomial has exactly one complemented variable (e.g. $x\bar{y}$) and non-negative coefficient (arc weight). The particular reparameterization we use is

$$-h_L + h_L \bar{y}_L + \sum_{p \in \mathcal{P}_L} h_L \bar{x}_p y_L \qquad (8)$$

where $\bar{x} = 1 - x$. Figure 4 shows the subgraph corresponding to (8) after cancelling the constant $-h_L$ using (7).

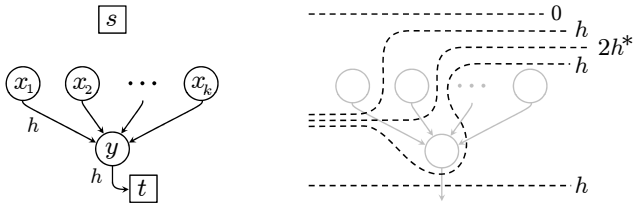



Figure 4. LEFT: Graph construction that encodes $h - hx_1x_2\cdots x_k$ when we define $x_p = 1 \Leftrightarrow p \in T$ where $T$ is the sink side of the cut. RIGHT: In a minimal $s$-$t$ cut, the subgraph contributes cost either 0 (all $x_p = 1$) or $h$ (otherwise). A cost greater than $h$ (e.g. $*$) cannot be minimal because setting $y = 0$ cuts only one arc.

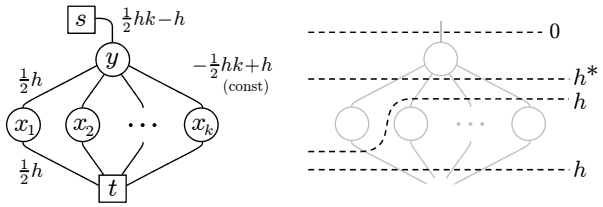



Figure 5. The alternate *undirected* graph construction corresponding to Figure 4 may be easier to understand. The weights are found by reparameterizing (8) such that $\bar{x}y$ and $x\bar{y}$ terms receive identical coefficients. Cut $*$ is not minimal w.r.t. auxiliary variable $y$.

Subgraphs of this type have been used in vision before, most notably the $P^n$ Potts potentials of Kohli et al. [20]. Our indicator potentials $\delta_L(\cdot)$ are different in that, at the binary step (6), each clique $\mathcal{P}_L$ is determined *dynamically* from the current labeling $f'$ and is not expressed as such in the original energy ($\star$). It is easy to represent a $P^n$ Potts potential by combination of label subset cost potentials, but not the other way around. Section 6 elaborates on this point, and mentions a possible extension to our work based on the Robust $P^n$ Potts construction [21].

A final detail is how to handle the case when label $\alpha$ was not used in the current labeling $f'$. The corrective term $C^\alpha$ in (6) incorporates the label costs for $\alpha$ itself:

$$C^\alpha(\mathbf{x}) = \sum_{\substack{L \subseteq \mathcal{L} \setminus \mathcal{L}' \\ \alpha \in L}} \left( h_L - h_L \prod_{p \in \mathcal{P}} \bar{x}_p \right). \qquad (9)$$

If we find that $\mathbf{x}^* = 0$ then label $\alpha$ was not used in $f'$ and it was also not worth expanding it in $f^\alpha$. The term (9) can be encoded by a subgraph analogous to Figure 4, but the following is more efficient: first compute optimal $\mathbf{x}^*$ for (6) without considering $C^\alpha$, then explicitly add it to $E^\alpha_h(\mathbf{x}^*)$ if $\mathbf{x}^* \neq \mathbf{0}$, and reject the expansion if the energy would increase.

In fact, a similar test-and-reject step allows label costs to be trivially incorporated into $\alpha$-$\beta$-swap: before accepting a standard swap move, compare its energy to the energy when all $\beta$ variables become $\alpha$ and vice versa, then apply the move with minimum energy.

### 2.2. Optimality guarantees

In what follows we assume that energy ($\star$) is configured[2] so that $D_p \geq 0$, $V_{pq}$ is a metric [7], and thus $E(f) \geq 0$.

**Theorem 1** *If $f^*$ is a global minimum of energy ($\star$) and $\hat{f}$ is a local minimum w.r.t. $\alpha$-expansion then*

$$E(\hat{f}) \;\leq\; 2cE(f^*) \;+\; \sum_{L \subseteq \mathcal{L}} h_L|L| \qquad (10)$$

*where* $\quad c \;= \max_{pq \in \mathcal{N}} \left( \frac{\max_{\alpha \neq \beta \in \mathcal{L}} V_{pq}(\alpha,\beta)}{\min_{\gamma \neq \zeta \in \mathcal{L}} V_{pq}(\gamma,\zeta)} \right)$

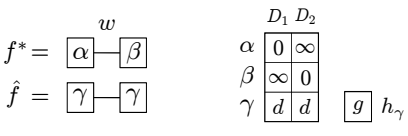
See Appendix A for the proof. The *a priori* bound (10) suggests that for label costs on large subsets the worst-case approximation is poor. The fundamental problem is that $\alpha$-expansion can expand only one label at a time. It may help empirically to order the expansions in a greedy manner, but the Section 2.3 describes a special case for which the greedy algorithm still yields a similar additive bound (see Section 3.5.1 of [10]). We thus do not expect much improvement unless different moves are considered.

For discussion, we note that bound (10) actually follows from a more general bound that does not assume $D_p \geq 0$:

$$E(\hat{f}) \;\leq\; E(f^*) \;+\; (2c-1)E_V(f^*) \;+\; \sum_{L \subseteq \mathcal{L}} h_L|L| \qquad (11)$$

where $E_V$ denotes the smooth cost of energy $E$. This holds for all $\hat{f}$ and $f^*$, so approximation error is actually bounded in terms of smooth cost $\min E_V(f^*)$ rather than by $E(f^*)$ itself. We submit our additive bound (11) as an alternative to the familiar multiplicative bound $E(\hat{f}) \leq 2cE(f^*)$ for $\alpha$-expansion [7]. To see why, consider that the multiplicative bound for $\alpha$-expansion is only tight when $E_D(f^*) = 0$, and does not even hold for $E_D(f^*) < 0$. Yet, replacing data terms with $D'_p(\cdot) := D_p(\cdot) + \epsilon_p$ for arbitrary constant $\epsilon_p$ effects neither the global optima nor the optimal expansions. The $\alpha$-expansion algorithm is indifferent to $\epsilon_p$, and this property distinguishes it from the *isolation heuristic* algorithm for multi-terminal cuts [11]. The isolation heuristic is applicable to metric labeling when $V_{pq}$ are Potts interactions, also has multiplicative bound of 2, but can compute arbitrarily bad solutions to multi-label problems depending on $\epsilon_p$. *The comparative robustness of $\alpha$-expansion is not reflected in the multiplicative bound.*

**Tightness of bounds.** There are non-trivial examples for which the bound (10) is tight. Consider the problem instance shown below, where $w$ is the weight of a pairwise Potts term and $d, g \geq 0$ are constants such that $w < 2d + g$.




(12)

[2] Adding an arbitrary constant to $D_p(\cdot)$ or $V_{pq}(\cdot,\cdot)$ does not affect the optimal labeling, so finite costs can always be made non-negative.

For $d \leq w$ the labeling $\hat{f}$ is a local optimum w.r.t. expansion moves. Plugging $\hat{f}$ and $f^*$ into inequality (10) we get

$$\begin{aligned} 2d+g \;&\leq\; w \;+\; w \;+\; g \\ d \;&\leq\; w \end{aligned} \qquad (13)$$

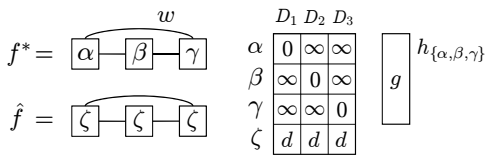
We can therefore bring the *a priori* bound (10) arbitrarily close to equality by setting $w - d \to 0$. In this example the standard multiplicative bound $E(\hat{f}) \leq 2cE(f^*)$ is also tight, but only because $E_D(f^*) = 0$. If we add an arbitrary constant to any $D_p(\cdot)$, our additive bound (11) remains tight whereas the multiplicative bound does not.

If we introduce high-order label costs, such as $h_{\{\alpha,\beta\}}$, the bound (10) is no longer tight. However, bound (11) follows from a tighter *a posteriori* bound w.r.t. specific $\hat{f}$:

$$\begin{aligned} E(\hat{f}) \;\leq\; & E(f^*) \;+\; (2c-1)E_V(f^*) \\ & + E_H(\hat{f}) - E_H(f^*) + \sum_{L \subseteq \mathcal{L} \setminus \hat{\mathcal{L}}} h_L|L \cap \mathcal{L}^*| \end{aligned} \qquad (14)$$

where $E_H(f)$ denotes the label cost of labeling $f$, and sets $\mathcal{L}^*$ and $\hat{\mathcal{L}}$ contain the unique labels in $f^*$ and $\hat{f}$ respectively.

Suppose we again have pairwise Potts terms with weight $w$, and consider the problem instance below for constants $d, g \geq 0$ such that $3w + g < 3d$.




(15)

The labeling $\hat{f}$ is a local optimum for any $d \leq 2w + g$. Plugging $\hat{f}$ and $f^*$ into inequality (14) we get

$$\begin{aligned} 3d \;&\leq\; 3w + g \;+\; 3w \;+\; 0 \;-\; g \;+\; 3g \\ d \;&\leq\; 2w + g \end{aligned} \qquad (16)$$

so setting $d = 2w + g$ makes bound (14) tight. This example demonstrates precisely how high-order label costs can lead to worse approximations.

### 2.3. Easy case: only per-label costs

In the absence of any smooth costs ($V_{pq} \equiv 0$) and higher-order label costs ($h_L = 0$ for $|L| > 1$) our energy reduces to a special case (1) known as the *uncapacitated facility location* (UFL) problem. This well-studied problem was recently applied for motion segmentation, first by Li [26] and then by Lazic et al. [25]. The UFL problem assigns facilities (labels) to each client (variable) such that the cost to clients is balanced against the cost of 'opening' facilities to serve them. Optimizing UFL is NP-hard by simple reduction from SET-COVER, so it is 'easier' than our full energy ($\star$) only in a practical sense.

Li optimizes the integer program corresponding to UFL by *linear programming (LP) relaxation*, then rounds fractional 'facility' variables to 0 or 1 in a straight-forward manner. Because of the heavy LP machinery, this approach is

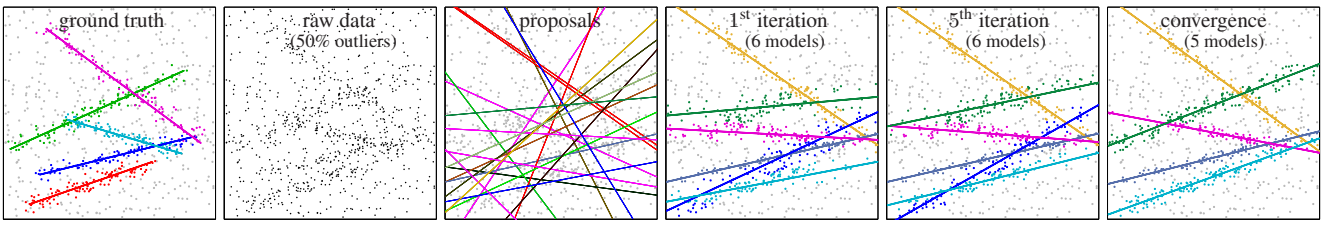



Figure 6. Re-estimation helps to align models over time. Above shows 900 raw data points with 50% generated from 5 line intervals. Random sampling proposes a list of candidate lines (we show 20 out of 100). The 1st segmentation and re-estimation corresponds to Li [26], but only the yellow line and gray line were correctly aligned. The decreasing energies in Figure 7 correspond to better alignments like the subsequent iterations above. If a model loses enough inliers during this process, it is dropped due to label cost (dark blue line).

slow and affords relatively few candidate models in practice. Li implements *four* application-specific heuristics to aggressively prune candidate models "for LP's sake." Lazic et al. optimize the same energy using max-product belief propagation (BP), a message-passing algorithm.

Kuehn & Hamburger [23] proposed a natural greedy algorithm for UFL in 1963. The algorithm starts from an empty set of facilities (labels) and greedily introduces one facility at a time until no facility would decrease the overall cost. The greedy algorithm runs in $O(|\mathcal{L}|^2|\mathcal{P}|)$ time for labels $\mathcal{L}$ and observations $\mathcal{P}$. Hochbaum [17] later showed that greedy yields a $O(\log|\mathcal{P}|)$-approximation in general, and better bounds exist for special cost functions (see [32] for review). Besides the add-facilities-greedily strategy, other greedy moves have been proposed for UFL such as the *greedy-interchange* and *dynamic programming* heuristics (see [9, 10] for review).

Our C++ library implements the greedy heuristic [23] and, when smooth costs are all zero, it is 5–20 times faster than $\alpha$-expansion while yielding similar energies. Indeed, "open facility $\alpha$" is analogous to expansion in this case.

Note that our high-order label costs $h_L$ can also be optimized greedily, but this is not standard and our bound (10) suggests the approximation may become worse than the bound proven by Hochbaum. Babayev [2] and Frieze [15] noted in 1974 that, as a function of open facilities, standard UFL is supermodular (as a minimization problem) and thus yields some form of approximation guarantee [30, 24]. It can be shown however that our generalization of UFL to subset costs $h_L$ is neither supermodular nor submodular.

## 3. Working With a Continuum of Labels

Our experimental Section 5 focuses on *multi-model fitting* problems, which are the most natural applications of energy ($\star$). As was first argued in [19], energies like ($\star$) are powerful criteria for multi-model fitting in general. However, there is a technical hurdle with using combinatorial algorithms for model fitting. In such applications each label represents a specific model, including its parameter values, and the set of all labels $\mathcal{L}$ is a continuum. In line fitting, for example, $\mathcal{L}=\mathbb{R}^2$. Practically speaking, however, the combinatorial algorithms from Section 2 require a *finite* set $\mathcal{L}$ of labels (models). Below we review a technique to effectively explore the continuum of model parameters by working with a finite subset of models at any given iteration $t$.

**PEARL Algorithm** [19]

1 **propose** initial models $\mathcal{L}_0$ by random samples (as in RANSAC)
2 run $\alpha$-**expansion** to compute optimal labeling $f$ w.r.t. $\mathcal{L}_t$
3 **re-estimate** model parameters to get $\mathcal{L}_{t+1}$; $t{:=}t{+}1$; goto **2**

PEARL was the first to use regularization energies and EM-style optimization for geometric multi-model fitting. Other geometric model fitting works have used separate elements such as random sampling [35, 26] (as in RANSAC) or EM-style iteration [3], but none have combined them in a single optimization framework. The experiments in [19] show that their energy-based formulation beats many state-of-the-art algorithms in this area. In other settings (segmentation, stereo) these elements have been combined in various application-specific ways [42, 3, 31, 41].

Our paper introduces a more general energy ($\star$) and a better algorithm for the expansion step of PEARL (step **2**).

**Review of PEARL for ($\star$).** Step **1** of PEARL is to propose an initial set of models $\mathcal{L}_0$. Each proposal is generated by a randomly sampling the smallest subset of data points needed to define a geometric model, exactly as in RANSAC [13]. A larger set of proposals $\mathcal{L}_0$ is more likely to contain models that approximate the true ones. Of course, $\mathcal{L}_0$ will contain many incorrect models as well, but optimizing energy ($\star$) over $\mathcal{L}_0$ (step **2**) will automatically select a small subset of labels from among the best models in $\mathcal{L}_0$.

The initial set of selected models can actually be further improved as follows. From here on, we represent model assignments by two sets of variables: segmentation variables $\{f_p\}$ that for each data point $p$ specifies the index of a model from the finite set $\mathcal{L}_0$, and parameter variables $\{\theta_l\}$ that specify model parameters currently associated with each model index. Then, energy ($\star$) is equivalent to

$$E(f;\theta)=\sum_{p\in\mathcal{P}}D_p(f_p,\theta_{f_p})+\sum_{pq\in\mathcal{N}}V_{pq}(f_p,f_q,\theta_{f_p},\theta_{f_q})+\sum_{L\subseteq\mathcal{L}}h_L(\theta_L)\cdot\delta_L(f). \qquad (\star')$$

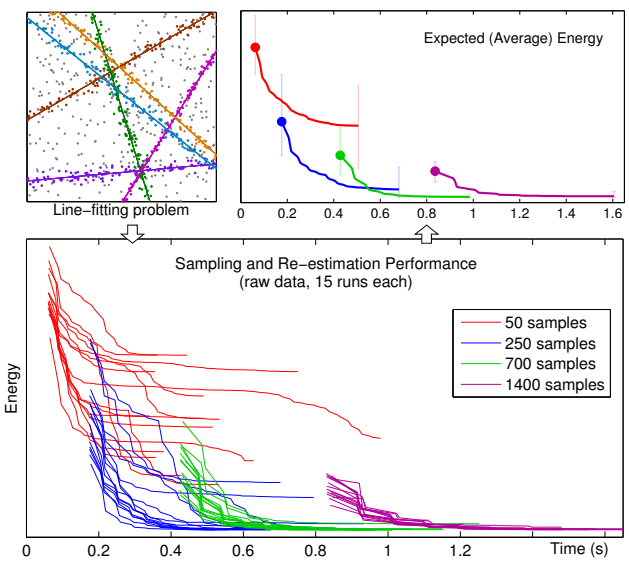



Figure 7. Energy ($\star'$) over time for a line-fitting example (1000 points, 40% outliers, 6 ground truth models). Only label cost regularization was used. Re-estimation reduces energy faster and from fewer samples. The first point (•) in each series is taken after exactly one segmentation/re-estimation, and thus suggests the speed of Li [26] using a fast greedy algorithm instead of LP relaxation.

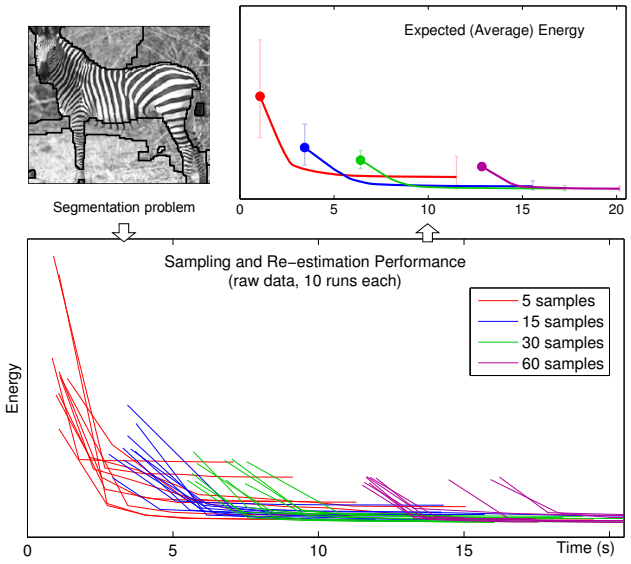



Figure 8. Energy ($\star'$) over time for image segmentation ($222 \times 183$ pixels). Smooth cost and label cost were regularized together. The models are 256-dimensional greylevel histograms. See Section 5.2 for experimental details.

For simplicity, assume that the smoothness terms in ($\star'$) are Potts interaction potentials [7] and the third term represents simple per-label costs as in (1). Then, specific model parameters $\theta_l$ assigned to a cluster of points $\mathcal{P}_l = \{p | f_p = l\}$ only affect the first term in ($\star'$), which is a sum of unary potentials. In most cases, it is easy to compute a parameter value $\hat{\theta}_l$ that locally or even globally minimizes $\sum_{p \in \mathcal{P}_l} D_p(l, \theta_l)$. The re-estimated parameters $\{\hat{\theta}_l\}$ correspond to an improved set of labels $\mathcal{L}_1$ that reduces energy ($\star'$) for fixed segmentation $f$ (step **3**).

Now one can re-compute segmentation $f$ by applying the algorithms in Section 2 to energy ($\star$) over a new set of labels $\mathcal{L}_1$ (step **2** again). PEARL's re-segmentation and re-estimation steps **2** and **3** reduce the energy. Iterating these two steps generates a sequence of re-estimated models $\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2, \ldots$ converging to a better local minima of energy ($\star$). In our experiments, convergence is typically achieved in 5–20 iterations. In most cases, iterating improves the solution significantly beyond the initial result.

Figure 7 shows how re-estimation finds a low energy for line-fitting faster than brute-force random sampling. For this example, the algorithm needs at least 250 random samples to be stable, but more than 700 samples is redundant. Figure 8 shows an analogous plot for unsupervised image segmentation. Recall that Li [26] does not re-estimate beyond the first iteration, and thus corresponds to what we are calling brute-force, i.e. selecting only from among the initial proposals.

**Proposal heuristics.** Re-estimation is a natural way to propose better models from existing ones because it applies to *any* family of models for which a maximum-likelihood estimator can be found. For example, the results in Figures 9 and 10 were both computed with re-estimation alone.

Re-estimation is by no means the only way to propose new models. Another general heuristic is to fit a new model to the inliers of *two* existing models, and then add this new model to the candidate list; this 'merge' heuristic [37] gives energy ($\star'$) an opportunity to jump out of local minima when computing optimal $f$.

The most effective proposal techniques actually tend to be class-specific and make use of the current solution. A simple example for line fitting is to compute a 'merge' proposal only for pairs of lines that are nearly colinear, since we know heuristically that such proposals are more likely to succeed. Li [26] uses a number of "guided sampling" heuristics specific to motion estimation, but they are only used for the *initial* proposals. Such heuristics make our algorithm more robust, but this is not the point of our work and so all our results use basic re-estimation only.

## 4. Relationship to EM and *K*-means

The main goal of this section is to relate our model fitting algorithm to the standard *expectation maximization* (EM) and *K-means* algorithms. Our discussion will focus on *Gaussian mixture models* (GMM). To keep things simple for GMM, we use only data terms and label cost terms, even though our full energy ($\star'$) was designed to handle smoothness priors as well.

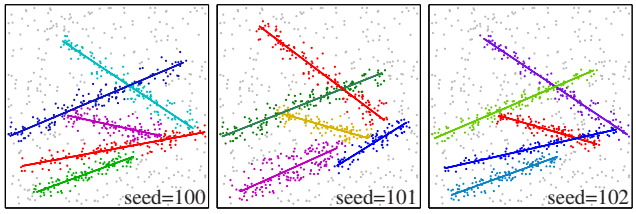



Figure 9. We can also fit line *intervals* to the raw data in Figure 6. The three results above were each computed from a different set $\mathcal{L}$ of random initial proposals. See Section 5.1 for details.

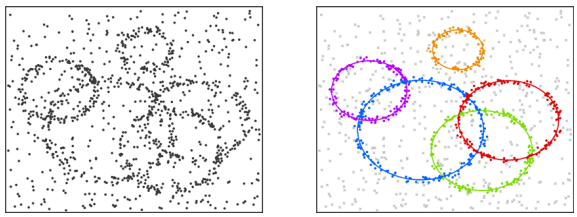


Figure 10. For multi-model fitting, each label can represent a specific model from any family (Gaussians, lines, circles...). Above shows circle-fitting by minimizing geometric error of points.

A number of interesting observations about our model fitting approach can be made:

- $K$-means minimizes a special case of our energy ($\star'$),
- like $K$-means, we make *hard assignments* of models to data points (in contrast to EM), and
- unlike $K$-means, our energy automatically removes unnecessary models from the initial set of proposals.

Sections 4.1–4.3 elaborate on these points, and Section 4.4 shows experimental results to help understand the relationship to EM and $K$-means. Note that our experiments are meant to be illustrative, and do not suggest that we have a state-of-the-art algorithm for GMM.

The main practical conclusion of this section is that **hard assignment works at least as well as soft assignment when models do not overlap too significantly**. We claim that many multi-model fitting applications in computer vision satisfy this property.

### 4.1. Standard Approaches to Finite Mixtures

Let some finite set of observed points $X = \{x_p \,|\, p \in \mathcal{P}\}$ be a mixture of independent samples taken from different probability distributions. These distributions are described by probability density functions $\Pr(x \,|\, \theta_l)$ with distinct parameters from a set $\theta = \{\theta_l \,|\, l \in \mathcal{L}\}$, where $\mathcal{L}$ is a finite set of distribution indices (labels). A set of hidden (unobserved) variables $f = \{f_p \in \mathcal{L} \mid p \in \mathcal{P}\}$ represent indices of specific distributions that generated each data point. The probability of sampling from each distribution is defined by a set of mixing parameters $\omega = \{\omega_l \mid l \in \mathcal{L}\}$ such that

$$\Pr(f_p = l) \;:=\; \omega_l, \qquad \sum_{l\in\mathcal{L}} \omega_l = 1, \quad \omega_l \geq 0.$$

It can be shown that data points in $X$ sampled in this manner correspond to the standard *mixture model* density [4]

$$\Pr(x \,|\, \theta, \omega) = \sum_{l\in\mathcal{L}} \omega_l \cdot \Pr(x \,|\, \theta_l).$$

The problem of estimating a mixture model is to estimate parameters $\theta$ and mixing coefficients $\omega$. We will mainly focus on estimating GMM, i.e. mixtures of normal distributions $\Pr(x \mid \theta_l) = \mathcal{N}(x \mid \mu_l, \Sigma_l)$ where model parameters $\theta_l = \{\mu_l, \Sigma_l\}$ are the mean and covariance matrix.

Standard EM and (elliptical) K-means algorithms can be seen as maximum likelihood (ML) approaches to estimating GMM. The classic EM algorithm [4, 12] finds $\theta, \omega$ that maximize the likelihood function

$$\Pr(X \,|\, \theta, \omega) = \prod_{p\in\mathcal{P}} \left( \sum_{l\in\mathcal{L}} \omega_l \cdot \Pr(x_p \,|\, \theta_l) \right). \qquad (17)$$

As an internal step, EM also computes *responsibilities* $\Pr(f_p = l \mid x_p, \theta, \omega)$ in order to estimate which mixture components could have generated each data point [4].

The elliptical[3] $K$-means algorithm [33] maximizes a different likelihood function on the same probability space

$$\Pr(X \,|\, f, \theta) = \prod_{p\in\mathcal{P}} \Pr\left(x_p \,|\, \theta_{f_p}\right). \qquad (18)$$

In contrast to EM, this approach directly computes labeling $f = \{f_p \mid p \in \mathcal{P}\}$, while mixing coefficients $\omega_l$ are implicitly estimated as percentages of points with $f_p = l$. It is often said that $K$-means performs *hard assignment* of models to data points, whereas EM performs *soft assignment* leaving room for uncertainty in the labeling $f$.

It is possible to derive a version of $K$-means that explicitly estimates mixing weights $\omega$. Assuming that $f_p$ are independent, one gets the following prior on the labeling

$$\Pr(f \,|\, \omega) = \prod_{p\in\mathcal{P}} \Pr(f_p \,|\, \omega) = \prod_{p\in\mathcal{P}} \omega_{f_p}. \qquad (19)$$

By applying this prior to (18), Bayes rule then gives posterior distribution

$$\Pr(f, \omega, \theta \,|\, X) \sim \prod_{p\in\mathcal{P}} \omega_{f_p} \cdot \Pr(x_p \,|\, \theta_{f_p}). \qquad (20)$$

Values of $f, \omega, \theta$ that maximize this distribution are *maximum a posteriori* (MAP) estimates of these parameters. Like the standard $K$-means algorithm, one can maximize (20) by iterating two steps: first optimize over $f$ for fixed $\omega, \theta$ and then (independently) optimize over $\omega$ and $\theta$ for fixed $f$. We refer to this algorithm as *weighted (elliptical) $K$-means*. Note that weighted $K$-means assumes no prior at all on weights $\omega$, whereas standard $K$-means is equivalent to "weighted $K$-means with prior that $\omega_l = \frac{1}{K}$." Figure 11 shows how this difference can affect solutions.

[3] The *elliptical* version of $K$-means explicitly estimates a covariance matrix $\Sigma$ so that each set of parameters is $\theta_l = \{\mu_l, \Sigma_l\}$.

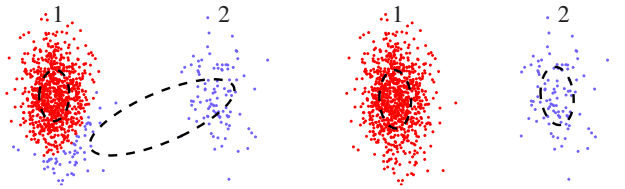



Figure 11. Mixture of two Gaussians where most data points were generated from the first component ($\omega_1 > \omega_2$). Standard $K$-means prefers equal cluster sizes because it assumes $\omega_1 = \omega_2$, whereas weighted $K$-means has no such bias.

### 4.2. Using Energy ($\star'$) for Finite Mixtures

The standard $K$-means algorithm directly minimizes the negative-log of the likelihood function (18), giving energy

$$E(f;\theta) = -\sum_{p\in\mathcal{P}} \ln \Pr(x_p \,|\, \theta_{f_p}). \tag{21}$$

Similarly, the weighted $K$-means algorithm minimizes the negative-log of the posterior distribution (20)

$$E(f;\theta,\omega) = -\sum_{p\in\mathcal{P}} \ln\big(\omega_{f_p}\cdot \Pr(x_p \,|\, \theta_{f_p})\big)\,. \tag{22}$$

Both of these $K$-means energies are expressible as data terms $D_p$ in our energy ($\star'$).

Note that posterior energy (22) is derived from the i.i.d. assumption (19) on assignment variables $f_p$. This assumption holds when the sampling process does not have any coherence or constraints (e.g. occlusions). In some examples, however, variables $f_p$ may be correlated. For example, pairwise interactions could be easily incorporated into a prior for $f$ yielding a posterior energy with the first and second terms in ($\star'$). Such a prior may be also useful for its regularization effect. In the context of GMM estimation, however, it makes more sense to incorporate a different regularization prior $\Pr(\omega)$ on possible combinations of models in the mixture. In particular, if we add a term in the posterior energy to penalize the number of assigned models.

$$E(f;\theta,\omega) = -\sum_{p\in\mathcal{P}} \ln\big(\omega_{f_p}\cdot \Pr(x_p \,|\, \theta_{f_p})\big) \;+\; \sum_{l\in\mathcal{L}} h\cdot\delta_l(f) \tag{23}$$

where $h \geq 0$ is the per-label cost, then this corresponds to some non-standard prior on $\omega$. Energy (23) is a special case of ($\star'$) with the simplest form of label cost regularizer. We use (23) in our GMM experiments in Section 4.4.

Note that basic $K$-means (21) is known to be sensitive to initialization with the correct number of models $K$. If the number of given initial models $K$ is too large, the algorithm will over-fit these $K$ models to data (see Fig.12e). One way to look at energy (23) is that it robustifies $K$-means (22) by penalizing the use of each model. As experiments in Sections 4.4 and 5 show, the exact number of initial models is largely irrelevant for the model fitting algorithms based on energy (23).

There is a standard technique [4] to make EM similarly robust to over-fitting: introduce a Dirichlet prior $\Pr(\omega) = \mathrm{Dir}(\omega \mid \alpha_0)$ that encourages most mixing weights $\omega_l$ to be small. By making our label costs $h_l$ dependent on $\omega_l$, our energy (23) can also incorporate a Dirichlet-like prior on $\omega$. See Appendix B for details. It is interesting to note that, if we apply a Dirichlet prior to weighted $K$-means (20) then we can derive standard $K$-means (18) by taking $\alpha_0 \to \infty$, i.e. encouraging weights $\omega_l$ to be as close to $\frac{1}{K}$ as possible. This "prior on $\omega$" interpretation is another way to understand standard $K$-means' sensitivity to the choice of $K$.

### 4.3. Energy ($\star$) as an information criterion

Regularizers are useful energy terms because they can help to avoid over-fitting. In statistical model selection, various *information criteria* have been proposed to fulfil a similar role. Information criteria penalize overly-complex models, preferring to explain the data with fewer, simpler models (Occam's razor [28]).

For example, consider the well-known *Akaike information criterion* (AIC) [1]:

$$\min_{\Theta} \;\; -2\ln \Pr(X \,|\, \Theta) \;+\; 2|\Theta| \tag{24}$$

where $\Theta$ is a model, $\Pr(X \mid \Theta)$ is a likelihood function and $|\Theta|$ is the number of parameters in $\Theta$ that can vary. This criterion was also discussed by Torr [35] and Li [26] in the context of motion estimation.

Another well-known example is the *Bayesian information criterion* (BIC) [8, 28]:

$$\min_{\Theta} \;\; -2\ln \Pr(X \,|\, \Theta) \;+\; |\Theta|\cdot\ln|\mathcal{P}| \tag{25}$$

where $|\mathcal{P}|$ is the number of observations. The BIC suggests that label costs should be scaled in some proportion (linear or logarithmic) to the number of data points or, in practice, to the estimated number of observations per model. In contrast, AIC over-fits as we add more observations from the true models. See [8] for an intuitive discussion and derivation of BIC in general, particularly Sections 6.3–6.4, and see Torr's work [35] for insights specific to vision.

### 4.4. Experimental Results for GMM Estimation

Figure 12 juxtaposes representative results for GMM estimation by standard EM (17), elliptical $K$-means (21,22), and energy (23). The latter represents a combination of the first and the third terms in ($\star'$). To minimize (23) we iterate PEARL (Sec.3) in combination with the greedy optimization method (Sec.2.3) for each expansion step. Note that the first column paints each point with the color of a model that had the highest *responsibility* according to EM's "soft assignments". The columns for $K$-means and energy (23) show colors corresponding to their "hard assignments".

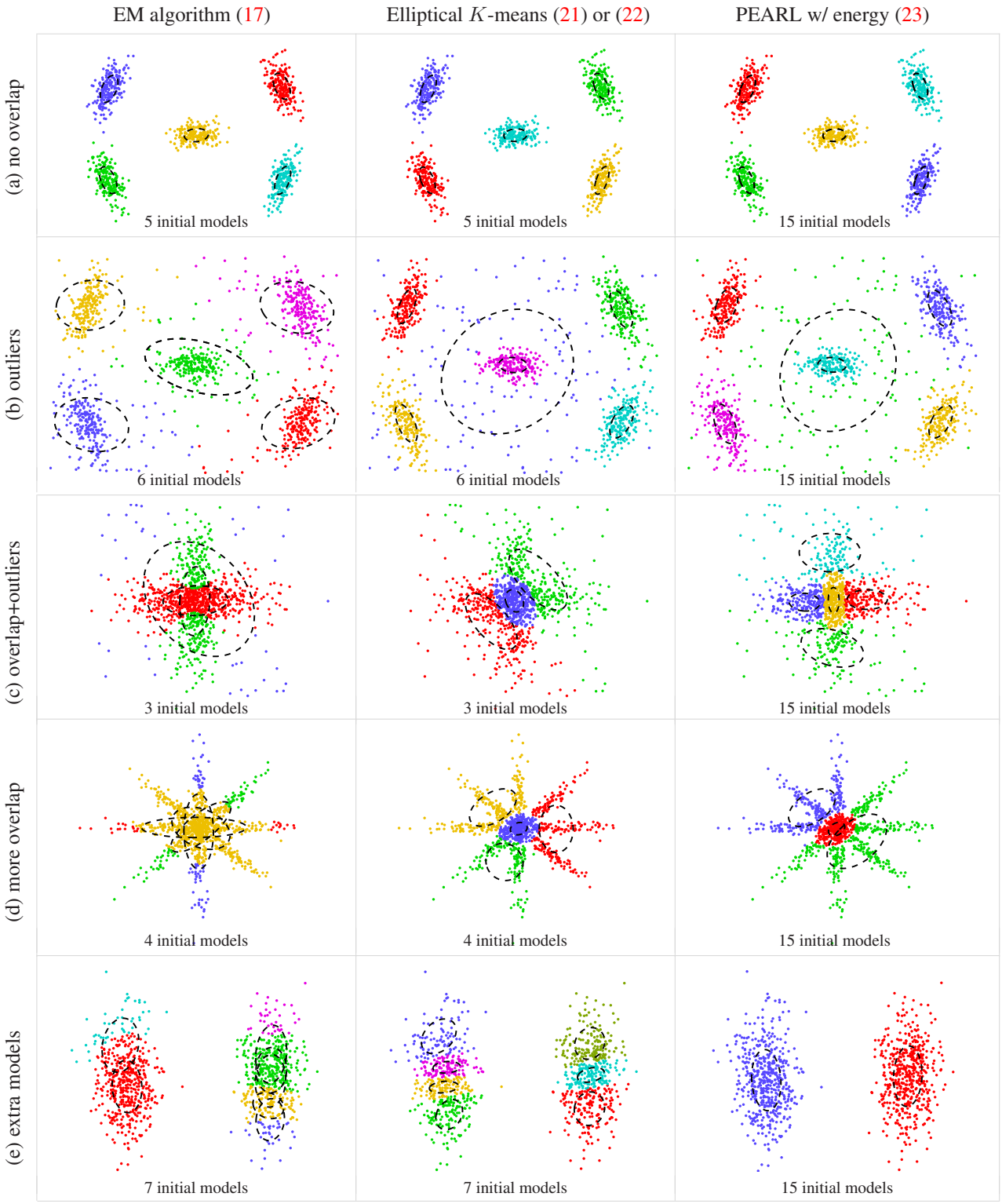



Figure 12. Each row shows how GMM algorithms behave on a particular example. This table is for illustrative purposes, and is *not* meant to be a state-of-the-art comparison. (a) If models do not overlap then all algorithms work. (b) Most algorithms can handle uniform outliers by fitting an extra model. (c) EM finds overlapping models thanks to soft assignment; hard assignment has bias towards isolated models. (d) Even EM (soft assignment) fails with only a little more ambiguity in the data. (e) Standard EM and $K$-means usually fail when given too many initial models, whereas PEARL with energy (23) keeps only enough models to explain the data. See Section 4.4 for discussion.

Figure 13. Unsupervised segmentation by clustering simultaneously over pixels and colour space using Gaussian Mixtures (colour images) and non-parametric histograms (greyscale images). Notice we find coarser clustering on baseball than Zabih & Kolmogorov [41] without over-smoothing. For segmentation, our energy is closer to Zhu & Yuille [42] but our algorithm is more powerful than region-competition.

The middle column in Figure 12 shows the results typical for both standard (21) and weighted $K$-means (22). The two methods worked similarly on all tests in Figure 12 because all models there have approximately the same number of inliers. Such examples can not reveal the bias of standard $K$-means to equalizing mixing weights (see Fig.11).

One important conclusion from Figure 12 is that energy (23) works well on all examples (a,b,e) where the models do not have significant spatial overlap. This case is very common in computer vision problems where models occlude each other rather than intersect.

If $K$-means and EM were initialized with a correct number of models, they also worked very well for spatially non-ovelapping models (a,b), however, EM was more sensitive to outliers in (b). If EM and $K$-means are initialized with a wrong number of models (e) then they overfit these models to data, while regularization energy (23) keeps only the minimal number of necessary models.

In general, EM handled intersecting models in (c) better than the other two methods. Arguably, soft assignments of models to data points help EM to deal with such overlapping models. But, more severe cases of model mixing in (d) are problematic even for EM.

Some additional experiments we made also show that both EM and $K$-means could be sensitive to initial models, particularly if distributions are not exactly Gaussian. Using PEARL to minimize (23) seems less sensitive to initialization in these particular tests. In general, however, our approach benefits from larger number of initial proposals which increases the chances that correct models are found. The last column in Figure 12 shows the minimum number of initial randomly sampled models (proposals) that our algorithm needed to robustly generate good results.

## 5. Applications and Experimental Setup

The experimental setup is essentially the same for each application: generate proposals via random sampling, compute initial data costs $D_p$, and run the iterative algorithm from Section 3. The only components that change are the application-specific $D_p$ and regularization settings. Section 5.1 outlines the setup for basic geometric models: lines, circles, homographies, motion. Section 5.2 describes the unsupervised image segmentation setup.

### 5.1. Geometric multi-model fitting

Here each label $l \in \mathcal{L}$ represents an instance from a specific class of geometric model (lines, homographies), and each $D_p(l)$ is computed by some class-specific measure of geometric error. The strength of per-label costs and smooth costs were tuned for each application.

**Outliers.** All our experiments handle outliers in a standard way: we introduce a special outlier label $\phi$ with $h_\phi = 0$ and $D_p(\phi) = \text{const} > 0$ manually tuned. This corresponds to a uniform distribution of outliers over the domain.

#### 5.1.1 Line fitting

Our line fitting experiments are all synthetic and mainly meant to be illustrative. Our energy ($\star'$) was motivated by applications in vision that involve images (Sections 5.1.2–5.2), but simple models (gaussians, lines) help to understand our energy, our algorithm, and their relation to standard methods.

Data points are sampled i.i.d. from a ground-truth set of line segments (e.g. Figure 6), under reasonably similar noise; outliers are sampled uniformly. Since the data is i.i.d. we set $V_{pq} = 0$ and use the greedy algorithm from Section 2.3. We also use fixed per-label costs as in (23), since keeping them independent of $\theta$ simplifies the

re-estimation of $\theta$ itself. Not surprisingly, the greedy algorithm (Section 2.3) was by far the best algorithm when smooth costs are not involved. Greedy gives similar energies to $\alpha$-expansion (Section 2.1) but is 5–20 times faster.

Figure 6 is a typical example of our line-fitting experiments with outliers. In 2D each line model $l$ has parameters $\theta_l = \{a, b, c, \sigma\}$ where $ax + by + c = 0$ defines the line and $\sigma^2$ is the variance of data; here $a, b, c$ have been scaled such that $a^2 + b^2 = 1$. Each proposal line is generated by selecting two random points from $\mathcal{P}$, fitting $a, b, c$ accordingly, and selecting a random initial $\sigma$ based on a prior. The data cost for a 2D point $x_p = (x_p^{\mathrm{x}}, x_p^{\mathrm{y}})$ is computed w.r.t. orthogonal distance

$$D_p(\,l\,) \;=\; -\ln\Big(\tfrac{1}{\sqrt{2\pi}\sigma}\exp\Big(-\tfrac{(ax_p^{\mathrm{x}}+bx_p^{\mathrm{y}}+c)^2}{2\sigma^2}\Big)\Big)\,. \qquad (26)$$

Figure 7 shows the trend in running time as the number of random initial proposals is increased. For 1000 data points and 700 samples, convergence took .7–1.2 seconds with 50% of execution time going towards computing data costs (26) and performing re-estimation.

Note that (26) does not correspond to a well-defined probabilty density function. The density for unbounded lines cannot be normalized, so lines do not spread their density over a coherent span. Still, in line-fitting experiments it is common to fit full lines to data that was actually generated from line *intervals*, e.g. [19, 43]. The advantage of full lines is that they are a lower-dimensional family of models, but when lines are fit to data generated from intervals this is a model mis-specification, causing discrepancy between the energy being optimized versus the optimal solution from a generative viewpoint. Surprisingly, [19] showed that there are examples where introducing spatial coherence ($V_{pq} > 0$) for i.i.d. line interval data can actually improve the results significantly. We hypothesize that, in this case, spatial coherence can be trained discriminatively to counter the discrepancy caused by fitting unbounded lines to line interval data.

**Line interval fitting.** Figure 9 shows three interval-fitting results, all on the same data. Each solution was computed from a different (random) set of 1500 initial proposals. Line intervals require many more proposals than for lines because intervals are higher-dimensional models. Each result in Figure 9 took 2–4 seconds to converge, with 90% of the execution time going towards computing data costs and performing re-estimation (in MATLAB).

We model an interval from point $a$ to point $b$ as an infinite mixture of isotropic Gaussians $\mathcal{N}\big(\mu, \sigma^2\big)$ for each $\mu$ interpolating $a$ and $b$. The probability of a data point appearing at position $x$ is thus

$$\Pr(x\,|\,a, b, \sigma^2) = \int_0^1 \mathcal{N}\big(x\,|\,(1-t)a + tb, \sigma^2\big)\,\mathrm{d}t. \qquad (27)$$

In two dimensions, the above integral evaluates to

$$\tfrac{1}{4\pi\sigma^2\|a-b\|}\cdot\exp\Big(-\Big(\tfrac{x^{\mathrm{x}}(b^{\mathrm{y}}-a^{\mathrm{x}})-x^{\mathrm{y}}(b^{\mathrm{x}}-a^{\mathrm{y}})+a^{\mathrm{y}}b^{\mathrm{x}}-a^{\mathrm{x}}b^{\mathrm{y}}}{\sqrt{2}\sigma\|a-b\|}\Big)^2\Big)$$
$$\cdot\Big(\mathrm{erf}\Big(\tfrac{(x-b)\cdot(a-b)}{\sqrt{2}\sigma\|a-b\|}\Big)-\mathrm{erf}\Big(\tfrac{(x-a)\cdot(a-b)}{\sqrt{2}\sigma\|a-b\|}\Big)\Big) \qquad (28)$$

where $x = (x^{\mathrm{x}}, x^{\mathrm{y}})$ is and $\mathrm{erf}(\cdot)$ is the *error function*.

Given a set $X_l = \{x_p\colon f_p = l\}$ of inliers for label $l$, we find maximum-likelihood estimators $\theta_l = \{a, b, \sigma\}$ by numerically minimizing the negative-log likelihood

$$E(X_l; a, b, \sigma) = -\sum_p \ln\Pr(x_p\,|\,a, b, \sigma^2). \qquad (29)$$

**Circle fitting.** Figure 10 shows a typical circle-fitting result. Our circle parameters are center-point $a$, radius $r$, and variance $\sigma^2$. We model a circle itself as an infinite mixture of isotropic Gaussians along the circumference. Proposals are generated by randomly sampling three points, fitting a circle, and selecting random $\sigma$ based on some prior. We find maximum-likelihood estimators numerically, much like for line intervals.

### 5.1.2 Homography Estimation

We used our energy to automatically detect multiple homographies in uncalibrated wide-base stereo image pairs. Our setup comes directly from [19] so we give only an outline.

The input comprises two (static) images related by a fundamental matrix. We first detect SIFT features [27] and do exhaustive matching as a preprocessing step; these matches are our observations. The models being estimated are homographies, and each proposal is generated by sampling four potential feature matches. Data costs measure the symmetric transfer error (STE) [16] of a match w.r.t. each candidate homography. Our set of neighbours $pq \in \mathcal{N}$ is determined by a Delaunay triangulation of feature positions in the first image. Re-estimation is done by minimizing the STE of the current inliers via Levenberg-Marquard [16]. Figure 2c shows a representative result.

### 5.1.3 Motion Segmentation

The setup is from [19] and is essentially the same as for homography estimation, except here each model is a fundamental matrix $F = [K'\,t]_\times K' R K^{-1}$ corresponding to a rigid body motion $(R,\, t)$ and intrinsic parameters $K$ [16]. The aim is to select true motions from among the candidates, as in [26].

Again, SIFT matches are our observations. We generate initial proposals by randomly sampling eight matching pairs and computing a fundamental matrix as described in [16], minimizing the non-linear SSD error using Levenberg-Marquard. Data costs measure the squared Sampson's distance [16] of a match with respect to each candidate fundamental matrix. Figure 1 shows a representative result.

### 5.2. Image Segmentation by MDL Criterion

The idea here is to automatically segment an image into coherent parts that have consistent appearence. This is similar to criteria for superpixels except our segments can be any size and need not be contiguous. Figures 3 and 13 show examples of such segmentations.

We formulate the problem as one of finding a *minimum description length* (MDL) representation for the image, meaning a we want to represent the image compactly, in an information-theoretic sense (see [28] for review of MDL). The MDL principle was first proposed for unsupervised segmentation by Zhu & Yuille [42], along with their *region competition* algorithm. When defined over a 2D grid of image pixels, our energy ($\star'$) can implement a discrete version of Zhu & Yuille's energy. Our algorithm is however more powerful because $\alpha$-expansion makes large moves, while region competition relies on local contour evolution and explicit merging of adjacent regions.

In our experiments, the appearance models for greyscale images are 256-dimensional histograms, and for colour images we use Gaussian mixtures in RGB space. Inital proposals were generated by sampling small patches of the image, just like [41, 42]. Figure 8 shows performance at various sampling rates. We used uniform Potts model for pairwise terms, and did not implement segmentation-specific heuristics such as merging or splitting the histograms.

## 6. Discussion

The potential applications of our algorithm are nearly as broad as for $\alpha$-expansion. Our algorithm can be applied whenever observations are known *a priori* to be correlated, whereas standard mixture model algorithms are designed for i.i.d. data.

Our C++ implementation and MATLAB wrapper are available at `http://vision.csd.uwo.ca/code/`. Besides general minimization of ($\star$), the code is further optimized in two important special cases:

1. when the energy reduces to (1) the solution is computed by the greedy UFL algorithm (Section 2.3), and
2. when only a small fraction of labels are feasible for any given data point (e.g. geometric models; labels localized to a patch) we support "sparse data costs" to dramatically speed up computation.[4]

Our new $\alpha$-expansion code optionally uses a simple strategy to invest expansions mainly on 'successful' labels. This is often faster, but can be slower, so we suggest selecting an expansion scheme (adaptive vs. standard cycle) empirically for each application.

Our energy is quite general but this can be a disadvantage in terms of speed. The $\alpha$-expansion step runs in polynomial time for fixed number of positive $h_L$ terms, but higher-order label costs should be used sparingly. Even the set of per-label costs $\{h_l\}$ slows down $\alpha$-expansion by 40–60%, though this is still relatively fast for such difficult energies [34]. This slowdown may be because the Boykov-Kolmogorov maxflow algorithm [6] relies on heuristics that do not work well for large cliques, i.e. subgraphs of the kind in Figure 4. Even if faster algorithms can be developed, our implementation can test the merit of various energies before one invests time in specialized algorithms.

**Category costs.** Our high-order label costs (on *subsets* of labels) seem to be novel, both in vision and in terms of the UFL problem, and can be thought of as a type of co-occurence potential. A natural application is to group labels in a hierarchy of categories and assign a *category cost* to each. This encourages labelings to use fewer categories or, equivalently, to avoid mixing labels from different categories (e.g. kitchen, office, street, forest,...). We anticipate specific applications in object recognition/segmentation and multi-homography/motion estimation.

**Regional label costs.** We can generalize the concept of label costs by making them spatially localized. The label cost term in energy ($\star$) could actually be expressed as

$$\sum_{P\subseteq\mathcal{P}} \sum_{L\subseteq\mathcal{L}} h_L^P \cdot \delta_L(f_P)$$

where our basic energy ($\star$) is a special case that assumes $h_L^P = 0$ for all non-global cliques $P \subsetneq \mathcal{P}$. (Note that the test-and-reject approach to incorporate $C^\alpha$ in Section 2.1 is no longer ideal for this more general case above.)

Such potentials amount to *regional* label cost terms. Regional and high-order label costs are useful together when labels belong to known categories with specific location priors, such as "pay a fixed penalty if any label from {*sky, cloud, sun*} appears in the bottom of an image."

**Relation to $P^n$ Potts [20].** The $P^n$ Potts potential $\psi_P(f_P)$ is defined on clique $P \subseteq \mathcal{P}$ as

$$\psi_P(f_P) \stackrel{\text{def}}{=} \begin{cases} \gamma_\alpha & \text{if } f_p = \alpha \quad \forall p \in P \\ \gamma_{\max} & \text{otherwise} \end{cases}$$

where $\gamma_\alpha \le \gamma_{\max}$ for all $\alpha \in \mathcal{L}$. This potential encodes a label-specific reward $\gamma_{\max} - \gamma_\alpha$ for clique $P$ taking label $\alpha$ in its entirety, and acts either as simple high-order regularization (all $\gamma_\alpha = \text{const}$) or as a form of high-order data cost (label-specific $\gamma_\alpha$).

Let $\bar{\alpha}$ denote the set all labels except $\alpha$, i.e. the set $\mathcal{L} \setminus \{\alpha\}$. Regional label costs over clique $P$ can represent the $P^n$ Potts potential in energy ($\star$) as follows:

1. Set cost $h_{\bar{\alpha}}^P := \gamma_{\max} - \gamma_\alpha$ for each $\alpha \in \mathcal{L}$.
2. Add constant $(1-|\mathcal{L}|)\gamma_{\max} + \sum_\alpha \gamma_\alpha$ to the energy.

Each regional label cost $h_{\bar{\alpha}}^P$ is non-negative by definition of $\psi_P(\cdot)$, thus $P^n$ Potts potentials can be expressed in terms of high-order label costs.

[4] Sparse data costs were not used in our experiments.

The $P^n$ Potts potential and its robust generalization [21] were designed to encourage consistent labelings over specific regions in an image. A special case of our potentials is very closely related to the robust variant: a basic per-label potential $h_l \cdot \delta_l(f)$ can be expressed as a specific (concave) Robust $P^n$ Potts potential. Besides significant conceptual and motivational differences, the main technical difference is that our construction makes no reference to a "dominant label." By constructing a two-label Robust $P^n$ Potts potential at each dynamic clique $\mathcal{P}_L$ in our binary expansion step, we can encode an arbitrary concave penalty on the number of variables taking labels from a specific *subset* of labels. This generalizes our high-order potentials $\delta_L(\cdot)$ if needed.

**Related global interactions.** Label costs can be viewed as a special case of global interactions recently studied in vision, for example, by Werner [38] and Woodford et al. [40]. Werner proposed a cutting plane algorithm to make high-order potentials tractable in an LP relaxation framework. The algorithm is very slow but much more general, and he demonstrates global *class size constraints* (marginal statistics) for image segmentation as a special case. The potential $h_l \cdot \delta_l(f)$ corresponds to a soft constraint that the number of variables taking label $l$ be zero; this cost is concave w.r.t. the number of variables taking $l$. Woodford et al. optimize energies involving marginal statistics and they call these *Marginal Probability Fields* (MPFs). They focus on a number of hard cases with convex costs and propose specialized algorithms based on *dual decomposition*.

**Acknowledgements** We would like to thank **Fredrik Kahl** for referring us to the works of Li [26] and Vidal [36], and for suggesting motion segmentation as an application. We also wish to thank **Lena Gorelick** for corrections and for investing much of her own time to track down bugs in our code. This work was supported by NSERC (Canada) Discovery Grant R3584A02 and Russian President Grant MK-3827.2010.9.

**Addendum, April 25 2010.** We were recently informed that John Winn had developed an "instance cost" potential for $\alpha$-expansion in 2005. It was manifested as a side contribution to the paper "3D LayoutCRF for Multi-View Object Class Recognition and Segmentation" by Hoiem et al. [18] in 2007. Their paper was about supervised part-based object recognition, an extension of the 2D LayoutCRF work by Winn & Shotton [39]. The relevant paragraph in [18] (p.6) mixes binary and multi-label variables in a way such that we are unsure of the exact method of proof/implementation, but the basic idea seems analogous. Our paper studied label costs from a general perspective, including discussion of multiple algorithms, optimality bounds, extensions, and fast special cases. Our work on these algorithms was inspired by an array of generic model-fitting applications in vision that benefit from label costs, e.g. geometric model fitting [35], rigid motion estimation [26, 36], MDL-based segmentation [42], finite mixture models [4].

## A. Optimality proof

**Proof of Theorem 1.** The proof idea follows Theorem 6.1 of [7]. Let us fix some $\alpha \in \mathcal{L}$ and let

$$\mathcal{P}_\alpha \stackrel{\text{def}}{=} \left\{ p \in \mathcal{P} : f^*_p = \alpha \right\}. \tag{30}$$

We can produce a labeling $f^\alpha$ within one $\alpha$-expansion move from $\hat{f}$ as follows:

$$f^\alpha_p = \begin{cases} \alpha & \text{if } p \in \mathcal{P}_\alpha \\ \hat{f}_p & \text{otherwise.} \end{cases} \tag{31}$$

Since $\hat{f}$ is a local optimum w.r.t. expansion moves we have

$$E(\hat{f}) \le E(f^\alpha). \tag{32}$$

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

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

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

$$\begin{aligned} \mathcal{I}^\alpha &= \mathcal{P}_\alpha \cup \{ pq \in \mathcal{N} : p \in \mathcal{P}_\alpha, q \in \mathcal{P}_\alpha \} \\ \mathcal{O}^\alpha &= \mathcal{P} \setminus \mathcal{P}_\alpha \cup \{ pq \in \mathcal{N} : p \notin \mathcal{P}_\alpha, q \notin \mathcal{P}_\alpha \} \\ \mathcal{B}^\alpha &= \{ pq \in \mathcal{N} : p \in \mathcal{P}_\alpha, q \notin \mathcal{P}_\alpha \}. \end{aligned}$$

The following facts now hold:

$$E(f^\alpha)|_{\mathcal{I}^\alpha} = E(f^*)|_{\mathcal{I}^\alpha} \tag{33}$$

$$E(f^\alpha)|_{\mathcal{O}^\alpha} = E(\hat{f})|_{\mathcal{O}^\alpha} \tag{34}$$

$$E(f^\alpha)|_{\mathcal{B}^\alpha} \le cE(f^*)|_{\mathcal{B}^\alpha}. \tag{35}$$

Equation (35) holds because for any $pq \in \mathcal{B}^\alpha$ we have $V(f^\alpha_p, f^\alpha_q) \le cV(f^*_p, f^*_q)$.

Let $E_H$ denote the label cost terms of energy $E$. Using (33), (34) and (35) we can rewrite (32) as

$$E(\hat{f})|_{\mathcal{I}^\alpha} + E(\hat{f})|_{\mathcal{B}^\alpha} + E_H(\hat{f}) \tag{36}$$

$$\le E(f^\alpha)|_{\mathcal{I}^\alpha} + E(f^\alpha)|_{\mathcal{B}^\alpha} + E_H(f^\alpha) \tag{37}$$

$$\le E(f^*)|_{\mathcal{I}^\alpha} + cE(f^*)|_{\mathcal{B}^\alpha} + E_H(f^\alpha) \tag{38}$$

Depending on $\hat{f}$ we can bound $E_H(f^\alpha)$ by

$$E_H(f^\alpha) \le E_H(\hat{f}) + \begin{cases} \sum\limits_{\substack{L \subseteq \mathcal{L} \setminus \hat{\mathcal{L}} \\ \alpha \in L}} h_L & \text{if } \alpha \in \mathcal{L}^* \\ 0 & \text{otherwise.} \end{cases} \tag{39}$$

where sets $\mathcal{L}^*$ and $\hat{\mathcal{L}}$ contain the unique labels in $f^*$ and $\hat{f}$ respectively.

To bound the total energy we sum expressions (36) and (38) over all labels $\alpha \in \mathcal{L}^*$ to arrive at the following:

$$\sum_{\alpha\in\mathcal{L}^*}\Big(E(\hat{f}\,)|_{\mathcal{I}^\alpha}+E(\hat{f}\,)|_{\mathcal{B}^\alpha}\Big) \tag{40}$$
$$\leq \sum_{\alpha\in\mathcal{L}^*}\Big(E(f^*)|_{\mathcal{I}^\alpha}+cE(f^*)|_{\mathcal{B}^\alpha}\Big)+\sum_{L\subseteq\mathcal{L}\setminus\hat{\mathcal{L}}} h_L|L\cap\mathcal{L}^*|.$$

Observe that, for every $pq\in\mathcal{B}=\bigcup_{\alpha\in\mathcal{L}}\mathcal{B}^\alpha$, the term $V_{pq}(\hat{f}_p,\hat{f}_q)$ appears twice on the left side of (40), once for $\alpha=f^*_p$ and once for $\alpha=f^*_q$. Similarly every $V(f^*_p,f^*_q)$ appears $2c$ times on the right side of (40). Therefore equation (40) can be rewritten as

$$\begin{aligned}E(\hat{f}) \;\leq\; & E(f^*)+(2c-1)E_V(f^*)-E(\hat{f})|_{\mathcal{B}} \\ & +E_H(\hat{f})-E_H(f^*)+\sum_{L\subseteq\mathcal{L}\setminus\hat{\mathcal{L}}} h_L|L\cap\mathcal{L}^*|.\end{aligned} \tag{41}$$

The above inequality is a tight *a posteriori* bound on $E(\hat{f})$ w.r.t. a specific local optimum $\hat{f}$ and global optimum $f^*$. Observe that

$$\begin{aligned}E_H(\hat{f})-E_H(f^*) \;+\; & \sum_{L\subseteq\mathcal{L}\setminus\hat{\mathcal{L}}} h_L|L\cap\mathcal{L}^*| \\ =\; & \sum_{\substack{L\subseteq\mathcal{L}\setminus\mathcal{L}^*\\ L\cap\hat{\mathcal{L}}\neq\emptyset}} h_L+\sum_{\substack{L\subseteq\mathcal{L}\setminus\hat{\mathcal{L}}\\ L\cap\mathcal{L}^*\neq\emptyset}} h_L\big(|L\cap\mathcal{L}^*|-1\big) \\ \leq\; & \sum_{L\subseteq\mathcal{L}} h_L|L|.\end{aligned} \tag{42}$$

Using (42) and the assumption $D_p\geq 0$ we simplify (41) to give *a priori* bound (10). ∎

## B. Label costs as Dirichlet prior

The label costs in energy (23) help to avoid the inevitable over-fitting by $K$-means when $K$ is set too large. There is a standard technique [4] to make EM similarly robust to over-fitting: introduce a prior $\Pr(\omega)$ that encourages each mixing weight $\omega_l$ to be small. This can be done with a Dirichlet prior $\Pr(\omega)=\mathrm{Dir}(\omega\,|\,\alpha_0)$ with $0<\alpha_0<1$. Since we must have $\sum\omega_l=1$, the prior thus chooses a few large mixing weights to explain the data.

We now show how, by making $h_l$ dependent on $\omega_l$, we can can incorporate a Dirichlet-like prior on $\omega$ using our energy (23). First consider this prior $\Pr(\omega)$ when applied to the weighted $K$-means posterior (20):

$$\Pr(f,\omega,\theta\,|\,X)\sim\prod_{p\in\mathcal{P}}\omega_{f_p}\cdot\Pr(x_p\,|\,\theta_{f_p})\prod_{l\in\mathcal{L}}\omega_l^{\alpha_0-1}. \tag{43}$$

The negative-log is the sum of (22) and an extra term dependent on $\omega$:

$$(1-\alpha_0)\sum_{l\in\mathcal{L}}\ln\omega_l. \tag{44}$$

So, with this prior, $K$-means can no longer estimate $\omega$ by simply setting each $\omega_l=\frac{N_l}{|\mathcal{P}|}$ where $N_l$ is the number of $f_p=l$ in the labeling. Instead, we must find $\omega$ minimizing

$$\begin{aligned}&\sum_{p\in\mathcal{P}}-\ln\omega_{f_p}\;+\;(1-\alpha_0)\sum_{l\in\mathcal{L}}\ln\omega_l \\ &=\sum_{l\in\mathcal{L}}(1-\alpha_0-N_l)\cdot\ln\omega_l\end{aligned} \tag{45}$$

subject to $\sum\omega_l=1$. Notice that for $\alpha_0<1$ and $N_l\leq 1$, this objective function is unbounded below so, in practice, we must add constrains $\omega_l\geq\epsilon$ to (45) where $\epsilon>0$ is some constant lower bound on the mixing coefficients[5]. Using (45) to re-estimate $\omega$ for fixed $f$, the weighted $K$-means algorithm can then optimize over this prior.

There is good reason, with this prior, to improve the segmentation step in $K$-means. The naïve way to re-estimate $f$ for fixed $\omega$ does not take the additive term (44) into account. If we could optimize over $f$ and $\omega$ simultaneously, even in some restricted sense, then our algorithm would be strictly more powerful. Label costs allow (44) to be partially optimized simultanously alongside $f$. Specifically, if label $l$ loses *all* support in $f$, then the corresponding change in (44) is considered by our algorithm, unlike $K$-means.

To see how this works, suppose we are estimating $f$ with respect to current mixing coefficients $\omega'$. Consider some label $l\in\mathcal{L}$ that already has support ($N_l>0$) in the current labeling. If we hypothetically knew that $N_l=0$ (i.e. label $l$ lost all support in $f$) then the optimal $\omega_l$ would be $\epsilon$. So, during the estimation step for $f$, we use a label cost $h_l$ that encodes the drop in (44) if $N_l$ becomes zero in $f$:

$$\begin{aligned}h_l(\omega_l) \;&=\; (1-\alpha_0)\ln\omega'_l\;-\;(1-\alpha_0)\ln\epsilon \\ &=\; (1-\alpha_0)\ln\left(\frac{\omega'_l}{\epsilon}\right).\end{aligned} \tag{46}$$

Since $h_l(\omega_l)\geq 0$ the algorithms from Section 2 apply. We can thereby optimize simultaneously over $f$ and $\omega$ such that each $\omega_l$ makes a discrete jump from $\omega'_l$ to $\epsilon$ if it loses support in the labeling.

The fact that $\omega$ is not normalized during this step is not a problem when $\alpha_0<1$. This is because an expansion move can only over-estimate the true cost of any move, and therefore can never increase the energy. The algorithms in Section 2 along with (46) are a strict improvement in the sense that they can jump out of local minima that a naïve algorithm could not.

## References

[1] H. Akaike. A new look at statistical model identification. *IEEE Trans. on Automatic Control*, 19:716–723, 1974. 8

[2] D. A. Babayev. Comments on the note of Frieze. *Mathematical Programming*, 7(1):249–252, December 1974. 5

[3] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *ICCV*, 1999. 5

[5] Another way to avoid singularities when $\alpha_0<1$ might be to use a Smoothed Dirichlet prior [29] instead. Then, an approximate maximum likelihood estimator can be found by closed-form expression.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, August 2006. 7, 8, 13, 14

[5] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Appl. Math.*, 123(1-3):155–225, 2002. 3

[6] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE TPAMI*, 29(9):1124–1137, 2004. 12

[7] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE TPAMI*, 23(11):1222–1239, 2001. 1, 2, 4, 6, 13

[8] K. P. Burnham and D. R. Anderson. *Model Selection and Multimodel Inference*. Springer, 2002. 8

[9] G. Cornuejols, M. L. Fisher, and G. L. Nemhauser. Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms. *Management Science*, 23(8):789–810, 1977. 5

[10] G. Cornuejols, G. L. Nemhauser, and L. A. Wolsey. The Uncapacitated Facility Location Problem. Technical Report 605, Op. Research, Cornell University, August 1983. 4, 5

[11] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The Complexity of Multiterminal Cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994. 4

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977. 7

[13] M. A. Fischler and R. 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. 5

[14] D. Freedman and P. Drineas. Energy minimization via graph cuts: settling what is possible. In *CVPR*, June 2005. 3

[15] A. M. Frieze. A cost function property for plant location problems. *Mathematical Programming*, 7(1):245–248, December 1974. 5

[16] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 11

[17] D. S. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming*, 22(1):148–162, 1982. 5

[18] D. Hoiem, C. Rother, and J. Winn. 3D LayoutCRF for Multi-View Obect Class Recognition and Segmentation. In *CVPR*, 2007. 13

[19] H. N. Isack and Y. Boykov. Energy-based Geometric Multi-Model Fitting. Technical Report 735, University of Western Ontario, March 2010. (Submitted to IJCV). 5, 11

[20] P. Kohli, M. P. Kumar, and P. H. S. Torr. $\mathcal{P}^3$ & Beyond: Solving Energies with Higher Order Cliques. In *CVPR*, 2007. 3, 12

[21] P. Kohli, L. Ladický, and P. H. S. Torr. Robust Higher Order Potentials for Enforcing Label Consistency. *IJCV*, 82(3):302–324, 2009. 3, 13

[22] V. Kolmogorov and R. Zabih. What Energy Functions Can Be Optimized via Graph Cuts. *IEEE TPAMI*, 26(2):147–159, 2004. 3

[23] A. A. Kuehn and M. J. Hamburger. A Heuristic Program for Locating Warehouses. *Manag. Sci.*, 9(4):643–666, 1963. 5

[24] A. Kulik, H. Schachnai, and T. Tamir. Maximizing submodular set functions subject to multiple linear constraints. In *SODA*, 2009. 5

[25] N. Lazic, I. Givoni, B. Frey, and P. Aarabi. FLoSS: Facility Location for Subspace Segmentation. In *ICCV*, 2009. 4

[26] H. Li. Two-view Motion Segmentation from Linear Programming Relaxation. In *CVPR*, 2007. 1, 4, 5, 6, 8, 11, 13

[27] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60:91–110, 2004. 11

[28] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. 8, 12

[29] R. Nallapati. *Smoothed Dirichlet distribution: Understanding Cross-entropy ranking in information retrieval*. PhD thesis, University of Massachusetts Amherst, 2006. 14

[30] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions – I. *Mathem. Programming*, 14(1):265–294, 1978. 5

[31] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive Foreground Extraction using Iterated Graph Cuts. In *SIGGRAPH*, 2004. 5

[32] D. B. Shmoys, E. Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract). In *ACM STOC*, pages 265–274, 1998. 5

[33] K. K. Sung and T. Poggio. Example based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:39–51, 1995. 7

[34] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE TPAMI*, 30(6):1068–1080, June 2008. 12

[35] P. H. S. Torr. Geometric Motion Segmentation and Model Selection. *Philosophical Trans. of the Royal Society A*, pages 1321–1340, 1998. 5, 8, 13

[36] R. Tron and R. Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *CVPR*, 2007. 1, 13

[37] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. SMEM Algorithm for Mixture Models. *Neural Computation*, 12(9):2109–2128, 2000. 6

[38] T. Werner. High-arity Interactions, Polyhedral Relaxations, and Cutting Plane Algorithm for Soft Constraint Optimisation (MAP-MRF). In *CVPR*, June 2008. 13

[39] J. Winn and J. Shotton. The Layout Consistent Random Field for Recognizing and Segmenting Partially Occluded Objects. In *CVPR*, June 2006. 13

[40] O. J. Woodford, C. Rother, and V. Kolmogorov. A Global Perspective on MAP Inference for Low-Level Vision. In *ICCV*, October 2009. 13

[41] R. Zabih and V. Kolmogorov. Spatially Coherent Clustering with Graph Cuts. In *CVPR*, June 2004. 2, 5, 10, 12

[42] S. C. Zhu and A. L. Yuille. Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *TPAMI*, 18(9):884–900, 1996. 2, 5, 10, 12, 13

[43] M. Zuliani, C. S. Kenney, and B. S. Manjunath. The multiRANSAC algorithm and its application to detect planar homographies. In *ICIP*, 2005. 11