International Journal of Computer Vision, vol. 96, no. 1, pp. 1–27, January 2012

Fast Approximate Energy Minimization with Label Costs

Andrew Delong · Anton Osokin · Hossam N. Isack · Yuri Boykov

Received: October 23, 2010 / Accepted: March 17, 2011 / Published online: July 15, 2011

Abstract The α-expansion algorithm has had a significant impact in computer vision due to its generality, effectiveness, and speed. It is commonly used to minimize energies that involve unary, pairwise, and specialized higher-order terms. Our main algorithmic contribution is an extension of α-expansion that also optimizes "label costs" with well-characterized optimality bounds. Label costs penalize a solution based on the set of labels that appear in it, for example by simply penalizing the number of labels in the solution.

Our energy has a natural interpretation as minimizing description length (MDL) and sheds light on classical algorithms like K-means and expectation-maximization (EM). Label costs are useful for multi-model fitting and we demonstrate several such applications: homography detection, motion segmentation, image segmentation, and compression. Our C++ and MATLAB code is publicly available.*

Keywords Energy minimization · Multi-model fitting · Metric labeling · Graph cuts · Minimum description length

1 Some Useful Regularization Energies

In a labeling problem we are given a set of observations $\mathcal{P}$ (pixels, features, data points) and a finite 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 $E(f)$.

Andrew Delong · Hossam N. Isack · Yuri Boykov
Department of Computer Science, University of Western Ontario.
E-mail: andrew.delong@gmail.com
E-mail: isack.hossam@gmail.com
E-mail: yuri@csd.uwo.ca

Anton Osokin
Department of Computational Mathematics and Cybernetics, Moscow State University.
E-mail: anton.osokin@gmail.com

The authors assert equal contribution and joint first authorship.

*http://vision.csd.uwo.ca/code/

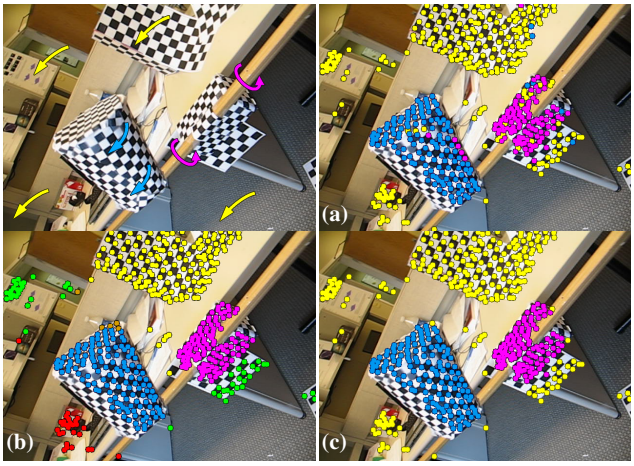


Fig. 1 Motion segmentation on the 1RT2RCR sequence [56]. Energy (1) finds 3 dominant motions (a) but labels many points incorrectly. Energy (2) gives coherent segmentations (b) but finds redundant motions. Our energy combines the best of both (c).

Most labeling problems in computer vision and machine learning 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 energies with two such regularizers: a preference for fewer unique labels in the solution (*label costs*), and a preference for spatial smoothness (*smooth costs*). Figures 1, 2, and 3 suggest how these criteria cooperate to give clean results.

Regularization combining smoothness and label costs has a long history in vision going back to well known papers by Leclerc [41], Zhu & Yuille [63], and many others. Until recently, however, label cost optimization problems were not addressed by powerful combinatorial algorithms that can guarantee certain optimality bounds and which are widely used for other problems in vision. The main contributions of our work (originally reported in [18]) are as follows. We are first to describe a general label cost functional ($\star$) that depends on a specific subset of used labels,

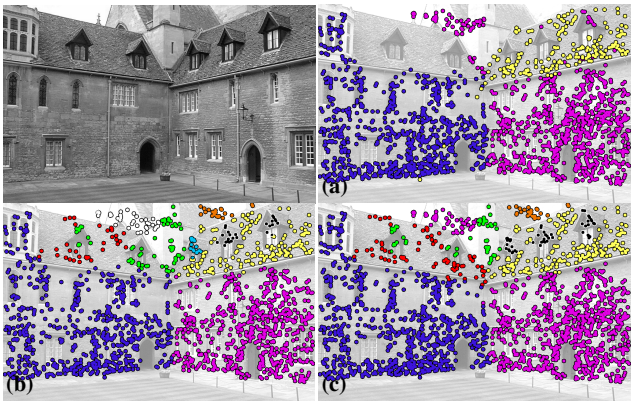


Fig. 2 Planar homography detection on VGG (Oxford) Merton College 1 image (right view). Energy (1) finds reasonable parameters for only the strongest 3 models shown in (a), and still assigns a few incorrect labels. Energy (2) finds reasonable clusters (b) but fits 9 models, some of which are redundant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

rather than on a number of labels. Moreover, we propose several combinatorial optimization algorithms with guaranteed optimality bounds for minimizing energies combining data costs, smooth costs, and label costs.

Label costs. Start by considering a basic (unregularized) energy $E(f) = \sum_p D_p(f_p)$, where optimal f_p can be determined trivially by minimizing over independent '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:

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

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 [42] recently posed multi-body motion estimation in terms of UFL. For multi-model fitting, each label corresponds to a candidate model and 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

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

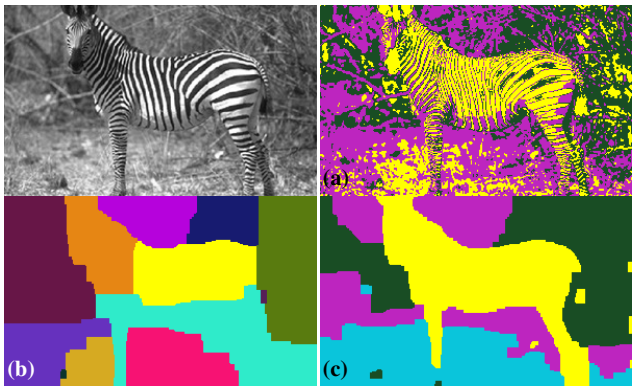


Fig. 3 Unsupervised segmentation using histogram models. Energy (1) clusters in colour space, so segments (a) are incoherent. Energy (2) clusters over pixels and must either over-segment or over-smooth (b), just as in [62]. Our energy ($\star$) balances these criteria (c) and corresponds to Zhu & Yuille [63] for segmentation.

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 [11,32] and can be optimized effectively with the α-expansion algorithm.

This regularizer prefers spatially 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.

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

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 generalizes 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 label cost energy makes sense.

Related work. A number of recent publications have relied on label costs in some form. For example, in [18] we proposed our subset costs in ($\star$) as a form of *co-occurrence cost* in object recognition. This application was thoroughly and independently developed by Ladický *et al.* [39], also within an α-expansion framework but with a heuristic extension; see Section 7 for discussion. Others have independently proposed label cost energies for specific applications. For example, we learned from personal correspondence that John Winn developed an extension of α-expansion to *instance cost* potentials in 2004 that only appeared as part of

a supervised part-based object recognition framework [30], though his approach to deriving an algorithm is quite different from ours[1]. Special case energy (1) corresponds to objective functions studied in vision by Torr [55] and in a number of independent later works for specific applications [42,40, 4]. Our combined energy ($\star$) has recently been extended to convex continuous *total variation* (TV) formulations [61].

Label costs can be viewed as a special case of other global interactions recently studied in vision, for example by Werner [59] and Woodford *et al.* [60]. Werner proposed a cutting plane algorithm to make certain 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* for enforcing simple marginal statistics in image segmentation. Our 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 (but slow) algorithms based on *dual decomposition*.

Our paper studies 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: geometric model fitting [55], rigid motion estimation [42,56], MDL-based segmentation [63], finite mixture models [6]. This paper presents a number of synthetic and real examples illustrating generic applications for the label costs and evaluating the proposed optimization techniques.

Our paper has the following structure. Section 2 presents our extension to α-expansion and corresponding optimality bounds. We also analyze fast UFL heuristics for a special case of ($\star$) without smooth costs. 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 experiments illustrating generic applications in vision. Section 6 empirically compares a number of alternative combinatorial optimization algorithms applicable to label cost energies. Besides the extended version of α-expansion designed specifically for energy ($\star$), we tested a number of alternative methods based on standard α-expansion [11] for (2) with additional heuristics addressing the label costs term. Section 7 discusses applications of high-order label costs, more related works, and possible extensions.

[1] In [30] the algorithm is briefly described on page 6 and mixes binary and multi-label variables in a way such that we are unsure of the exact method of implementation/proof, but the goal is clearly analogous to a special case of our extended α-expansion for energy ($\star$).

2 Fast Algorithms to Minimize ($\star$)

Our main technical contribution is to extend the well-known α-expansion algorithm [11] to incorporate label costs at each expansion (Section 2.1) and prove new optimality guarantees (Section 2.3). Section 2.4 reviews known results for the 'easy' case (1) with only data and per-label costs.

2.1 Expansion Moves with Label Costs

Minimizing the multi-label energy ($\star$) is NP-hard in general for $|\mathcal{L}| \geq 3$. The α-expansion algorithm [11] maintains a current labeling f' and iteratively 'moves' to a better one until no improvements can be made. At each iteration, 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 key step (line **4** below) is called *expansion* because label α is given a chance to grow arbitrarily. If each V_{pq} is a metric [11], the best possible expansion move can be computed efficiently by a single graph cut.

ALPHAEXPANSION [11]

```
1 f' := arbitrary labeling
2 repeat
3    for each α ∈ L
4       f^α := arg min_f E(f)  where f is an α-expansion of f'
5       if E(f^α) < E(f')
6          f' := f^α
7 until converged
```

We now describe the binary expansion step in more detail. Let labeling $f = \{f_1, \ldots, f_n\}$ and let f^α denote a feasible α-expansion w.r.t. current labeling f'. The possible labelings f^α can be expressed one-to-one with binary indicator variables $\mathbf{x} = \{x_1, \ldots, x_n\}$ by defining

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

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

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

$$\begin{array}{rcl} f' = \boxed{\beta}\boxed{\alpha}\boxed{\gamma}\boxed{\gamma}\boxed{\beta}\boxed{\beta} & \rightarrow & \boxed{\alpha}\boxed{\alpha}\boxed{\alpha}\boxed{\gamma}\boxed{\beta}\boxed{\beta} = f^\alpha \\ & & \boxed{1}\boxed{\underline{1}}\boxed{1}\boxed{0}\boxed{0}\boxed{0} = \mathbf{x}^* \end{array} \tag{4}$$

where $\underline{1}$ means x_2 is fixed to 1. Here only f_1 and f_3 changed to label α while the rest preferred to keep their labels. The α-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 β rather than switch to α. Suppose we introduce a cost $h_\beta > 0$ that is added to $E(f)$ if and only if there exists some $f_p = \beta$. The binary energy for an expansion move must encode a potential reward

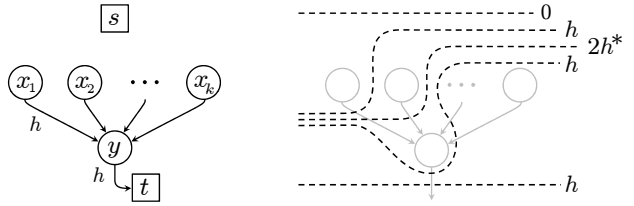


Fig. 4 LEFT: Graph construction that encodes $h - hx_1x_2\cdots x_k$ when we define $x_p = 1 \Leftrightarrow p \in T$ where T is the sink side of the cut. RIGHT: In a minimal s-t cut, the subgraph contributes cost either 0 (all $x_p = 1$) or h (otherwise). A cost greater than h (*e.g.* $*$) cannot be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

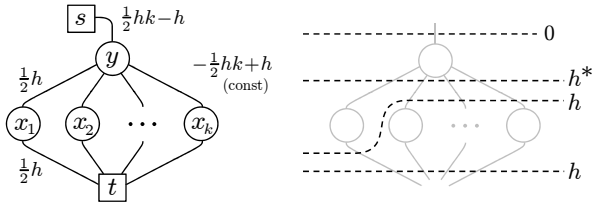


Fig. 5 The alternate *undirected* graph construction corresponding to Figure 4 may be easier to understand. The weights are found by reparameterizing (8) such that $\bar{x}y$ and $x\bar{y}$ terms receive identical coefficients. Cut $*$ is not minimal w.r.t. auxiliary variable y.

of h_β for replacing all $f'_p = \beta$ with label α. If h_β is large enough, the optimal expansion move for our small example would affect f_5 and f_6:

$$f' = \underset{1}{\boxed{\beta}}\boxed{\alpha}\boxed{\gamma}\boxed{\gamma}\underset{5}{\boxed{\beta}}\underset{6}{\boxed{\beta}} \;\rightarrow\; \begin{array}{l}\boxed{\alpha}\boxed{\alpha}\boxed{\alpha}\boxed{\gamma}\boxed{\alpha}\boxed{\alpha} = f^\alpha \\ \boxed{1}\boxed{1}\boxed{1}\boxed{0}\boxed{1}\boxed{1} = \mathbf{x}^*\end{array} \tag{5}$$

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_β in E^α 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 [37].

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

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

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^α 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 α-expansion energy $E^\alpha(\mathbf{x})$. Freedman and Drineas [24] generalized the graph construction of [37] 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

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

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 [8].

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

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

where $\bar{x} = 1 - x$. Figures 4 and 5 show subgraphs corresponding to (8) after cancelling the constant $-h_L$ using (7).

Subgraphs of this type have been used in vision before, most notably the P^n Potts potentials of Kohli *et al.* [33]. 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 ($\star$). A P^n Potts potential can be represented by a combination label subset costs but not the other way around. The idea is to apply 'regional' subset costs derived from the coefficients of the P^n Potts potential. Section 7 describes this transformation in detail.

A final detail for α-expansion is the case when label α was not present in the current labeling f'. The corrective term C^α in (6) incorporates the label costs for α itself:

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

If we find that $\mathbf{x}^* = 0$ then label α was not used in f' and it was also not worth expanding it in f^α. 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^α, 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.

2.2 Swap Moves with Label Costs

Label costs can be trivially incorporated into $\alpha\beta$-swap by a test-and-reject approach similar to above: before accepting a standard swap move, compare its energy to the energy when all β variables become α and vice versa, then apply the move with minimum energy.

2.3 Optimality Guarantees

In what follows we assume that energy ($\star$) is configured[2] so that $D_p \geq 0$, V_{pq} is a metric [11], 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. α-expansion then*

$$E(\hat{f}) \;\leq\; (2c+d)E(f^*) \;+ \sum_{L \subset \mathcal{L}} h_L \qquad (10)$$

where

$$c = \max_{pq \in \mathcal{N}} \left(\frac{\max_{\alpha \neq \beta \in \mathcal{L}} V_{pq}(\alpha,\beta)}{\min_{\gamma \neq \zeta \in \mathcal{L}} V_{pq}(\gamma,\zeta)} \right), \quad d = \max_{\substack{L \subset \mathcal{L} \\ h_L > 0}} |L| - 1.$$

See Appendix A for the proof of (10) and also of (11) below. These bounds suggest the following properties in practice:

- if label costs are modest we inherit an approximation guarantee comparable to α-expansion,
- if label costs are arbitrarily large the bound is poor, and
- if the optimal solution includes label costs defined over large subsets then the bound worsens.

Poor local minima are caused by the fact that α-expansion allows only one label to expand at a time. Performing expansions in greedy order (rather than arbitrary order) may help empirically, but a hardness result of Feige [21] still applies to our problem (discussed in Section 2.4).

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

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

where $E_V(f)$ denotes the total smooth cost of labeling f, $E_H(f)$ total label cost, and $\mathcal{L}^*$ the set of unique labels in f^*. This holds for all $\hat{f}$ and f^*, so the approximation error is determined by the minimum of the three additive terms above over all global optima f^*. The additive bound (11) is informative in a way that the familiar multiplicative bound $E(\hat{f}) \leq 2cE(f^*)$ for α-expansion is not. To see why, consider that the multiplicative bound for α-expansion is only tight when the total data cost $E_D(f^*) = 0$, and does not even hold for $E_D(f^*) < 0$. Yet, biasing the data costs with some $D'_p(\cdot) := D_p(\cdot) + \epsilon_p$ for arbitrary constant ϵ_p affects neither the global optima nor the optimal expansion moves. The α-expansion algorithm is indifferent to ϵ_p, and this property distinguishes it from the *isolation heuristic* algorithm for multi-terminal cuts [17]. 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 ϵ_p. The comparative robustness of α-expansion is not reflected in the multiplicative bound.

[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.

Worst-case examples. The simplified bound (10) describes the worst-case performance in special cases, but bound (11) is tight more generally. The table below describes a worst-case problem instance with $\mathcal{P} = \{p, q\}$ and $\mathcal{L} = \{\alpha, \beta, \gamma\}$. We also assume a label cost $h_\gamma \geq 0$ and a Potts potential that penalizes $f_p \neq f_q$ with weight $w > 0$.

data costs	p	q	
α	0	∞	$f^* = (\alpha, \beta)$
β	∞	0	
γ	w	w	$\hat{f} = (\gamma, \gamma)$
label cost	h_γ		

(12)

This example has global optimum $E(f^*) = w$ and so the local minimum $E(\hat{f}) = 2w + h_\gamma$ is tight with respect to (10). Note that by adding positive ϵ_p to each $D_p(\cdot)$ our additive bound (11) remains tight, unlike the multiplicative bound.

More generally we can design bad local minima from the following n-variable problem structure. Let $a, b, h \geq 0$ be constants such that $a = h + w$ where w is still the weight of all Potts potentials. Let $\mathcal{N} = \{\{1,2\}, \{3,4\}, \ldots\}$ be the neighbour set for Potts potentials. The data costs and label costs in the table below have optimal labeling $f^* = \{1, \ldots, n\}$, yet labeling $\hat{f} = \{n+1, n+1, n+2, n+2, \ldots\}$ is a local minimum w.r.t. expansion moves. (A blank entry signifies $D_p = \infty$)

data costs	n variables						label subset costs	
n labels	0						h	$E(f^*) = h + \frac{1}{2}nw$
		0						
			0					
				0				
					0			
						0		
	a	a					b	$E(\hat{f}) = na + \sum b$
			a	a			b	
					a	a	b	

(13)

We verify that $\hat{f}$ is generally tight for bound (11) as follows

$$\begin{aligned} E(\hat{f}) = na + \textstyle\sum b &= nh + nw + \textstyle\sum b \\ &= E(f^*) + \tfrac{1}{2}nw + (n-1)h + \textstyle\sum b \quad (14) \\ &= E(f^*) + E_V(f^*) + dE_H(f^*) + \textstyle\sum b. \end{aligned}$$

The above is tight for (10) when $h = 0$ and nearly tight when $w = 0$ aside for one double-counted label cost h. This example demonstrates how high-order label costs in the optimal labeling can worsen the approximation.

2.4 Energies with Only Per-label Costs

In the absence of smooth costs ($V_{pq} = 0$) and higher-order label costs ($h_L = 0$ for $|L| > 1$) our energy reduces to special case (1) known as the *uncapacitated facility location* (UFL) problem. 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.

In vision, the UFL problem has recently been applied to motion segmentation by Li [42] and by Lazic *et al.* [40]. Each facility represents a potential rigid motion, and each

client is a correspondence that must be assigned to one motion. The goal is then to choose a good subset of motions, much like Figure 1a. Li optimizes the integer program corresponding to UFL by *linear programming (LP) relaxation*, then rounds fractional facility variables to $\{0,1\}$ in a straightforward manner. Because general LP solvers are slow, this approach affords relatively few candidate models in practice. Li implements four application-specific heuristics to aggressively prune out candidate models before building an LP problem instance. Lazic *et al.* optimize the same energy using max-product belief propagation (BP), a message-passing algorithm. More recently, Barinova *et al.* [4] used UFL to model a class of object-detection problems and used the same greedy algorithm as our concurrent work [18].

The general[3] UFL problem is NP-hard by simple reduction from SET-COVER. A hardness result for SET-COVER by Feige [21] implies that UFL cannot be approximated better than $(1-\epsilon)\ln|\mathcal{P}|$ for $\epsilon > 0$ in polynomial time unless the complexity class $\mathrm{NP} \subseteq \mathrm{DTIME}[n^{O(\log\log n)}]$. Kuehn & Hamburger [38] proposed a natural *greedy* algorithm where facilities are opened one at a time. Cornuejols *et al.* [15] showed that the greedy algorithm provides a constant-factor approximation bound, but only with respect to the gap between best and worst solutions; this bound is not informative when the range of costs involved are prohibitively large. Hochbaum [29] later proposed a *set-greedy* algorithm that achieves a $\ln|\mathcal{P}|$-approximation regardless of the costs involved, which is optimal in the sense outlined by Feige. Hochbaum also showed that neither greedy nor set-greedy is strictly better than the other, and that the best choice depends on the problem instances at hand. We present the original greedy algorithm rather than the set-greedy algorithm.

Greedy UFL. In terms of our multi-label energy (1), the greedy UFL algorithm starts from an empty set of labels and greedily introduces one label at a time until no subsequent label would allow the overall cost to decrease. Once a label l is introduced, its cost h_l is assumed to be paid for regardless of subsequent steps. To express the greedy algorithm we introduce a function of label subsets $Z(S)$ where $S \subseteq \mathcal{L}$. The problem of minimizing $E(f)$ in (1) can then be rewritten as

$$\min_f E(f) = \min_{S\subseteq\mathcal{L}} Z(S) \tag{15}$$

$$\text{where } Z(S) = \sum_{p\in\mathcal{P}} \min_{l\in S} D_p(l) + \sum_{l\in S} h_l \tag{16}$$

and $Z(\{\})$ is defined to be $+\infty$. The overall algorithm is described in pseudo-code below.

GREEDYUFL [38, 16]

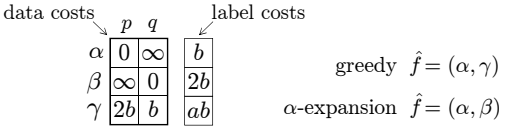
```
1 S := {}
2 while exists l ∉ S such that Z(S ∪ {l}) < Z(S)
3     j := arg min_{l∉S} Z(S ∪ {l}) − Z(S)
4     S := S ∪ {j}
```

The greedy algorithm runs in $O(|\mathcal{L}|^2|\mathcal{P}|)$ time for label set $\mathcal{L}$ and variable set $\mathcal{P}$. Our C++ library implements GREEDYUFL and it is 5–20 times faster than α-expansion for energies of the form (1) while yielding similar results. Besides this classic heuristic, other greedy moves have been proposed for UFL such as the *greedy-interchange* and *dynamic programming* heuristics (see [15, 16] for a review).

Babayev [3] and Frieze [25] noted in 1974 that the set function $Z(S)$ is supermodular (as a minimization problem), *i.e.* it can be shown that

$$Z(S\cup\{j,k\}) - Z(S\cup\{k\}) \geq Z(S\cup\{j\}) - Z(S). \tag{17}$$

The greedy bound for UFL by Cornuejols *et al.* [15] then follows from a general bound on minimizing supermodular functions by Nemhauser *et al.* [46]. Note that introducing a new label $j \notin S$ in step **3** does not consider the potential reward for eliminating labels from S once j is made available. This is in contrast to a j-expansion move with label costs, where introducing j may be beneficial because existing labels could be eliminated despite $Z(S\cup\{j\})$ not reflecting this in the classical algorithm. The 2-variable problem instance below illustrates this difference for some constants $a > 1$, $b > 0$. GREEDYUFL finds an arbitrarily poor energy of $(2+a)b$ whereas α-expansion with label costs finds an energy of $3b$ regardless of initial labeling.



(18)

Our subset costs h_L suggest a generalization of the classic UFL problem to add *facility subset costs*. Each subset cost represents a *shared* setup cost for opening particular set of facilities, after which the individual facilities can be opened with their own costs h_l for $l \in L$. The greedy algorithm can be adapted to this generalized UFL problem, but it can be shown that the new $Z(S)$ corresponding to (16) is no longer supermodular and so the approximation results of Cornuejols *et al.* no longer apply.

Finally, the greedy algorithm may be enhanced by applying the *tabu search* meta-heuristic to the UFL problem [50]. Empirical results in [50] show that tabu search finds global optima for many examples in the UFL literature at reasonable increase in running time.

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 ($\star$). The goal is to estimate parameters for an unknown

[3] *Metric*-UFL is a special case that can be approximated to within a constant factor [49]. In our work we assume arbitrary costs $D_p(\cdot)$. Unfortunately, some papers refer to metric-UFL simply as UFL.

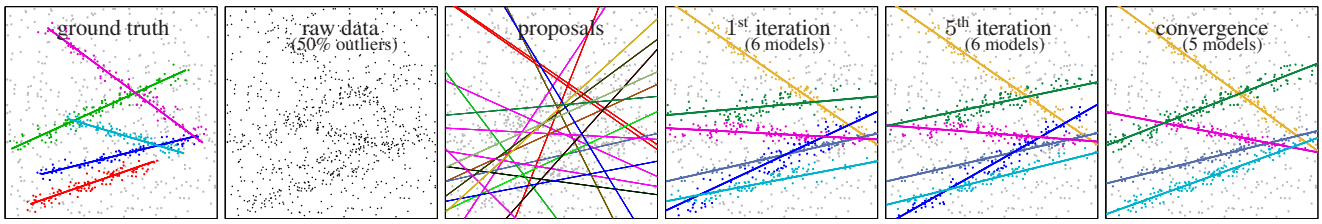


Fig. 6 Re-estimation helps to align models over time. Above shows 900 raw data points with 50% generated from 5 line intervals. Random sampling proposes a list of candidate lines (we show 20 out of 100). The 1st segmentation and re-estimation corresponds to Li [42], but only the yellow line and gray 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).

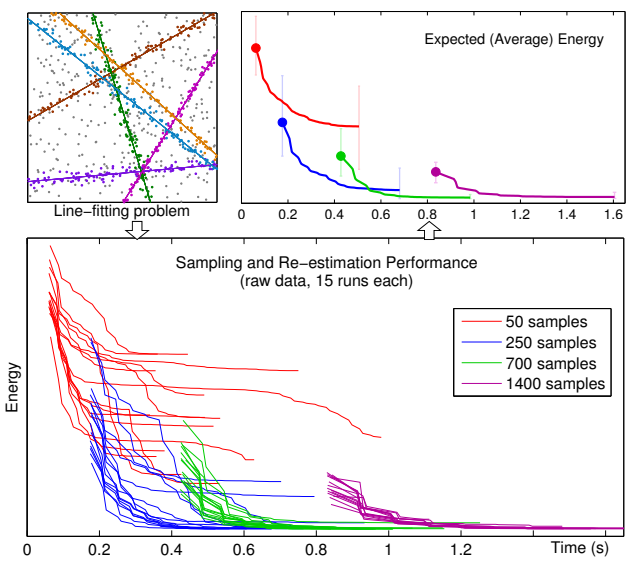


Fig. 7 Energy ($\star'$) over time for a line-fitting example (1000 points, 40% outliers, 6 ground truth models). Only label cost regularization was used. Re-estimation reduces energy faster and from fewer samples. The first point (•) in each series is taken after exactly one segmentation/re-estimation, and thus suggests the speed of Li [42] using a fast greedy algorithm instead of LP relaxation.

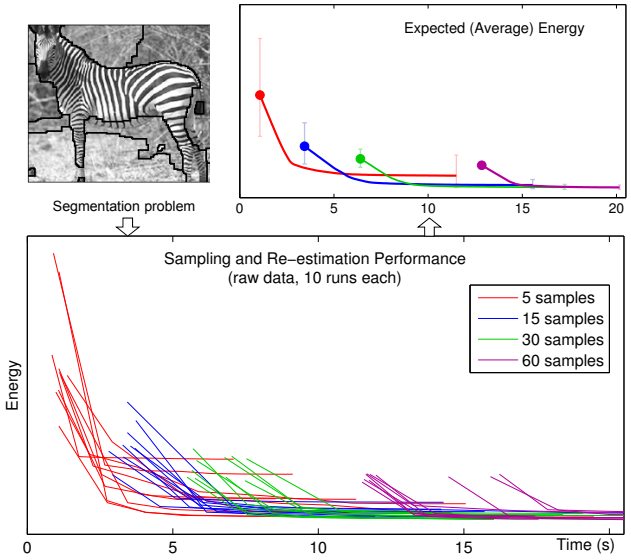


Fig. 8 Energy ($\star'$) over time for image segmentation (222×183 pixels). Smooth cost and label cost were regularized together. The models are 256-dimensional greylevel histograms. See Section 5.2 for experimental details.

number of models supported by noisy data with outliers. As was first argued in [31], 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 [31]

1 **propose** initial models $\mathcal{L}_0$ (*e.g.* randomly sample data points)
2 run α-**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 iterative optimization for geometric multi-model fitting. Other geometric model fitting works have used separate elements such as RANSAC-style random sampling [55, 42] or EM-style iteration [5], but none have combined them in a single optimization framework. The experiments in [31] 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 [63,5,48,62].

Our paper suggests better algorithms for the expansion step of PEARL (step **2**), proposes a more general form of label costs in energy ($\star$), describes fast methods for the special case without the spatial smoothness term, and discusses a broader class of multi-model fitting problems in vision.

Review of PEARL for ($\star$). For simplicity, we will discuss PEARL in the context of geometric model fitting, as in [31]. Figure 6 illustrates the algorithm's progression. Step **1** of PEARL is to propose an initial set of models $\mathcal{L}_0$. Each proposal can be generated by randomly sampling the smallest subset of data points needed to define a geometric model, exactly as in RANSAC [23]. 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 $\mathcal{L}_0$, see iteration 1 in Fig.6. In this example we used only the label cost regularizer in ($\star$) ignoring the spatial smoothness term, and data fidelity $D_p(l)$ represented an orthogonal distance from point p to line l, see Sec.5.1.1. We also fit one additional outlier model ϕ with $D_p(\phi) = \text{const}$.

The initial set of selected models can 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

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

For simplicity, assume that the smoothness terms in ($\star'$) are Potts interaction potentials [11] and the third term represents simple per-label costs as in (1). Then, specific model parameters θ_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 Sec.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**-**3** reduce the energy. Iterating these steps generates a sequence of re-estimated models $\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2, ...$ 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, see Fig.6.

Figure 7 shows effectiveness of re-estimation. Starting with only 250 samples (blue plot), re-estimation converges to better solutions than those computed from 1400 samples without re-estimation (a first thick dot on the violet plot). 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 color-model fitting in unsupervised image segmentation, see Sec.5.2. Recall that Li [42] does not re-estimate beyond the first iteration. His solutions correspond to thick dots at the begging of each plot in Fig.7. This approach would heavily rely on brute-force random sampling to find solutions of the same quality that we can find with only 250 samples.

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 13 and 14 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 [58] gives energy ($\star'$) an opportunity to jump out of local minima when computing optimal f. The algorithm in [31] finds lower energy solutions when new 'merge' proposals are added (compare α-SM and α-BM curves in our Section 6).

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 collinear. Li [42] uses a number of "guided sampling" heuristics specific to motion estimation, but they are only used for the *initial* proposals. In general, proposal heuristics can make our algorithms in Section 2 more robust but this is not the point of our work, 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), but we will also consider a geometric example of fitting multiple lines to noisy data points with outliers. 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.

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. Sections 4.4 and 4.5 show experimental results to help understand the relationship to EM and K-means. Note that our experiments are meant to be illustrative. In particular, we 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 have (nearly) non-overlapping spatial support**. We claim that many multi-model fitting applications in computer vision satisfy this property, see Figs.1,2,3. Note that in contrast to K-means or EM algorithm our method can also use spatial smoothness prior that is often needed in vision. In this section, however, we focus on a special case of ($\star'$) ignoring the smoothness term mainly to discuss the relationships with the classical multi-model fitting methods.

4.1 Standard Approaches to Finite Mixtures

Let some finite set of observed points $X = \{x_p \mid p \in \mathcal{P}\}$ be a mixture of independent samples taken from different probability distributions. These distributions are described by probability density functions $\Pr(x \mid \theta_l)$ with distinct parameters from a set $\theta = \{\theta_l \mid l \in \mathcal{L}\}$, 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 [6]

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

The problem of estimating a mixture model is to estimate parameters θ and mixing coefficients ω. 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.

Objective functions for EM. The classic EM algorithm [6, 19] finds maximum likelihood (ML) estimators for GMM. The ML objective is to find parameters θ and weights ω that maximize the likelihood function

$$\Pr(X \mid \theta, \omega) = \prod_{p \in \mathcal{P}} \left(\sum_{l \in \mathcal{L}} \omega_l \cdot \Pr(x_p \mid \theta_l) \right). \tag{19}$$

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

The EM algorithm can be generalized [6] to compute *maximum a posteriori* (MAP) estimates of θ and ω maximizing the posterior $\Pr(\theta, \omega \mid X) \propto \Pr(X \mid \theta, \omega) \Pr(\theta) \Pr(\omega)$. For example, a common MAP objective is

$$\Pr(\theta, \omega \mid X) \propto \prod_{p \in \mathcal{P}} \left(\sum_{l \in \mathcal{L}} \omega_l \cdot \Pr(x_p \mid \theta_l) \right) \cdot \prod_{l \in \mathcal{L}} \omega_l^{\alpha - 1} \tag{20}$$

which combines the ML objective (19) with a uniform prior on θ and Dirichlet prior on weights ω

$$\Pr(\omega) = \mathrm{Dir}(\omega \mid \alpha) \propto \prod_{l \in \mathcal{L}} \omega_l^{\alpha - 1}, \qquad \alpha > 0. \tag{21}$$

The Dirichlet prior is a uniform distribution for $\alpha = 1$ but for $\alpha < 1$ it prefers to estimate ω such that most ω_l are close to zero. A smaller choice of α creates a stronger sparsity effect on ω, and so α is called a *sparsity parameter*. In theory, this prior should encourage mixture models where most components are close to zero. According to [22] and in our own experience (see Fig.12), negative values of α are often necessary in practice to effectively remove redundant models. However, the Dirichlet prior is not a proper (integrable) distribution for $\alpha \leq 0$.

Objective functions for K-means. Standard K-means can also be seen as an ML approach to estimating mixture models. The elliptical[4] K-means algorithm [51] maximizes the following likelihood on the same probability space

$$\Pr(X \mid f, \theta) = \prod_{p \in \mathcal{P}} \Pr\left(x_p \mid \theta_{f_p}\right). \tag{22}$$

In contrast to EM, this approach directly computes labeling $f = \{f_p \mid p \in \mathcal{P}\}$ rather than responsibilities, while mixing coefficients ω_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 ω. Assuming that f_p are independent, one gets the following prior on the labeling

$$\Pr(f \mid \omega) = \prod_{p \in \mathcal{P}} \Pr(f_p \mid \omega) = \prod_{p \in \mathcal{P}} \omega_{f_p}. \tag{23}$$

Combining this prior with likelihood (22) and assuming non-informative (uniform) priors for ω and θ, Bayes rule then gives posterior distribution

$$\Pr(f, \theta, \omega \mid X) \propto \prod_{p \in \mathcal{P}} \omega_{f_p} \cdot \Pr(x_p \mid \theta_{f_p}). \tag{24}$$

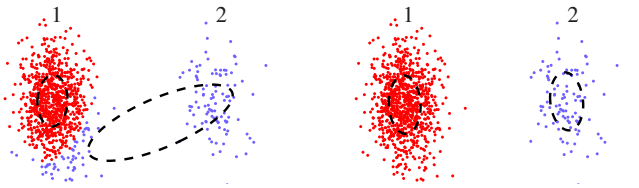
Values of f, θ, ω maximizing this distribution are MAP estimates of these parameters. Like the standard K-means algorithm, one can maximize (24) by iterating two steps: first optimize over f for fixed θ, ω and then (independently) optimize over ω and θ for fixed f. We refer to this algorithm as *weighted (elliptical) K-means*.

Discussion of priors. Instead of a uniform prior on ω used in (24) one can add any informative prior for mixture weights. For example, the Dirichlet prior (21) gives posterior

$$\Pr(f, \theta, \omega \mid X) \propto \prod_{p \in \mathcal{P}} \omega_{f_p} \cdot \Pr(x_p \mid \theta_{f_p}) \cdot \prod_{l \in \mathcal{L}} \omega_l^{\alpha - 1}. \tag{25}$$

For $\alpha < 1$ this posterior encourages sparsity of weights ω. Objectives (22) and (24) can be derived from (25) for other values of α. Setting $\alpha = 1$ gives the uniform prior on ω and (25) reduces to the weighted K-means posterior (24). Setting α very large ($\alpha \to \infty$) encourages equal weights $\omega_l = \frac{1}{K}$ and so (25) reduces to the standard K-means likelihood (22).

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



elliptical K-means (22,28) weighted ellpt. K-means (24,29)

Fig. 9 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.

Figure 9 shows how this difference can affect solutions. Standard K-means' bias to equal-size components is another way to understand its sensitivity to the choice of K.

As an alternative to Dirichlet prior, one can impose a sparsity prior similar to the *spike-and-slab* distribution [45]. We consider a modification that we call *step-and-slab* prior

$$\Pr(\omega) = \text{Sts}(\omega\,|\,\gamma) \propto \prod_{l\in\mathcal{L}} \psi_\varepsilon(\omega_l\,|\,\gamma) \tag{26}$$

where for some $\gamma \in (0,1)$ and infinitesimally small $\varepsilon > 0$

$$\psi_\varepsilon(\omega_l\,|\,\gamma) := \begin{cases} 1, & \text{if } \omega_l \le \varepsilon \\ \gamma, & \text{if } \omega_l > \varepsilon. \end{cases}$$

Note that γ is a *sparsity parameter* analogous to α in (21)[5]. Step-and-slab sparsity prior (26) yields posterior

$$\Pr(f,\theta,\omega\,|\,X) \propto \prod_{p\in\mathcal{P}} \omega_{f_p}\cdot\Pr(x_p\,|\,\theta_{f_p}) \cdot \prod_{l\in\mathcal{L}} \psi(\omega_l). \tag{27}$$

As discussed in Section 4.2, this posterior distribution obtained from step-and-slab sparsity prior (26) corresponds to posterior energy like ($\star'$) with label costs $h = \log\frac{1}{\gamma}$ but with no smooth costs. In contrast to the properties of the Dirichlet prior discussed earlier, (26) can achieve arbitrarily strong sparsity for small $\gamma > 0$ remaining a proper distribution.

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

The standard K-means directly minimizes the negative-log of the likelihood function (22), giving energy

$$E(f;\theta) = -\sum_{p\in\mathcal{P}} \log\Pr(x_p\,|\,\theta_{f_p}). \tag{28}$$

Similarly, the weighted K-means algorithm minimizes the negative-log of the posterior distribution (24)

$$E(f;\theta,\omega) = -\sum_{p\in\mathcal{P}} \log\big(\omega_{f_p}\cdot\Pr(x_p\,|\,\theta_{f_p})\big)\,. \tag{29}$$

[5] Each mixture component weight ω_l may have a separate sparsity parameter γ_l in step-and-slab prior (26). This is similar to Dirichlet prior generally defined by a sequence of parameters α_l. We use uniform sparsity parameters γ and α only for simplicity.

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

Note that posterior energy (29) is derived from the i.i.d. assumption (23) 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 dependent. 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 regularize using some sparsity prior, for example (26). The negative logarithm of the corresponding posterior distribution (27) gives posterior energy

$$E(f;\theta,\omega) = -\sum_{p\in\mathcal{P}} \log\big(\omega_{f_p}\cdot\Pr(x_p\,|\,\theta_{f_p})\big) + \sum_{l\in\mathcal{L}} \log\frac{1}{\gamma}\cdot[\omega_l > \varepsilon]$$

where $[\cdot]$ are *Iverson brackets*. The next theorem shows that the last term in this energy is essentially the label cost.

Theorem 2 *For sufficiently small $\varepsilon > 0$, the energy above has the same global minimum as the label cost functional*

$$E(f;\theta,\omega) = -\sum_{p\in\mathcal{P}} \log\big(\omega_{f_p}\cdot\Pr(x_p\,|\,\theta_{f_p})\big) + \sum_{l\in\mathcal{L}} h\delta_l(f) \tag{30}$$

for $h = \log\frac{1}{\gamma}$. That is, minimization of label cost energy (30) *is equivalent to MAP estimation for posterior* (27)*.*

The proof of this theorem is in appendix B. Energy (30) is a special case of ($\star'$) with the simplest form of label cost regularizer. We use (30) in our GMM experiments in Section 4.4 and line-fitting experiments in Section 4.5.

Note that the K-means algorithm for (28) is very sensitive to initialization even if the right number of models K is given, see Fig.11. If the number of given initial models K is too large, the algorithm will over-fit these K models to data, see Fig.10e. The extra label cost term in energy (30) removes many problems associated with fixed K. We initialize our method with a relatively large number of randomly sampled models and minimization of (30) leads to a solution with a small number of good models, see Fig.6. Our approach based on energy (30) is fairly robust to local minima and it is stable with respect to the set of randomly sampled initial models as long as is it large enough.

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 [44]).

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

$$\min_{\Theta} \; -2\log \Pr(X\,|\,\Theta) \;+\; 2|\Theta| \tag{31}$$

where Θ is a model, $\Pr(X \mid \Theta)$ is a likelihood function and $|\Theta|$ is the number of parameters in Θ that can vary. This criterion was also discussed by Torr [55] and Li [42] in the context of motion estimation.

Another well-known example is the *Bayesian information criterion* (BIC) [13,44]:

$$\min_{\Theta} \; -2\log \Pr(X\,|\,\Theta) \;+\; |\Theta|\cdot\log|\mathcal{P}| \tag{32}$$

where $|\mathcal{P}|$ is the number of observations. The BIC suggests that label costs should be scaled in logarithmic proportion 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 [13] for an intuitive discussion and derivation of BIC in general, particularly Sections 6.3–6.4, and see Torr's work [55] for insights specific to vision.

4.4 Experimental Results for GMM Estimation

Figure 10 juxtaposes representative GMM estimation results by basic EM (19), EM with Dirichlet prior (20), elliptical K-means (28,29), and our approach to label cost energy (30). For simplicity, Fig.10 represents EM's "soft assignment" at each point p using only one color corresponding to the model with the highest *responsibility*, see appendix C. The results for K-means and energy (30) show colors corresponding to their "hard assignments".

Implementation of (weighted) elliptical K-means maximizing (28,29) is fairly straightforward. Some details for optimizing (19) and (20) via EM algorithm are provided in appendix C. Since (20) automatically controls sparsity of the solution, we can initialize this version of EM with a large number of randomly sampled models. As discussed in [22], this makes EM robust to initialization and helps to avoid local minima.

Energy (30) represents a combination of the first and the third terms in ($\star'$). To minimize (30) we iterate PEARL (Sec.3) in combination with the greedy optimization method (Sec.2.4) for each expansion step. Similarly to [22] and to our EM approach for (20), optimization of (30) via PEARL avoids local minima when initialized with a large set of randomly sampled models.

The second column in Figure 10 shows the results typical for both standard (28) and weighted K-means (29). The two methods worked similarly on all tests in Figure 10 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.9).

One important conclusion from Figure 10 is that energy (30) 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 basic EM (19) were initialized with a correct number of models, they also worked very well for spatially non-overlapping models (a,b), however, EM was more sensitive to outliers in (b). If basic EM and K-means are initialized with a wrong number of models (e) they overfit these models to data, while Dirichlet-based posterior (20) and label cost energy (30) keep the minimal number of necessary models.

In general, EM handled intersecting models in (c) better than K-means and our method with (30). Arguably, soft assignments of models to data points help EM to deal with such overlapping models. More severe cases of model mixing in (d) were problematic for basic EM with a fixed number of models (19) due to local minima. However, EM for Dirichlet-based posterior (20) could avoid such local minima by selecting good models from a large initial sample.

In general, our approach with (30) and EM with (20) benefit from larger number of initial proposals which increases the chances that correct models are found. The 2 right columns in Figure 10 show the minimum number of initial randomly sampled models (proposals) that these algorithms needed to robustly generate good results.

4.5 Experimental Results for Geometric Model Fitting

Figures 11 and 12 show representative multi-line fitting results by basic EM (19), EM with Dirichlet prior (20), elliptical K-means (28,29), and our approach to label cost energy (30). As before, we represent EM's "soft assignment" at each point using only the color of the model with the highest *responsibility*, see appendix C. The results for K-means and energy (30) show colors of their "hard assignments".

The data set for experiments in Figs.11-12 consists of 300 inliers for 5 lines and 180 outliers. Each line model $\theta = \{a, b, c, \sigma\}$ includes noise variance σ. Log-likelihood $D_p(\,l\,) \;=\; -\log Pr(x_p|\theta_l)$ for a given data point x_p and line θ_l assumes Gaussian orthogonal error and is given in (33). We also fit one uniform outlier model ϕ with likelihood $Pr(x_p|\phi) = \text{const} > 0$ where const was manually tuned. Some additional general details about the experimental set-up for line fitting can be found in Sec.5.1.1. Optimization of functionals (19), (20), (28), (29), and (30) via EM, K-means, and PEARL is implemented as in the previous section. Some details for EM are in appendix C.

Figure 11 demonstrates that the standard K-means for (28), (29), and basic EM algorithm for (19) are very sensitive to local optima. Figure 12a shows that such local minima are avoided by optimization algorithms that select a few

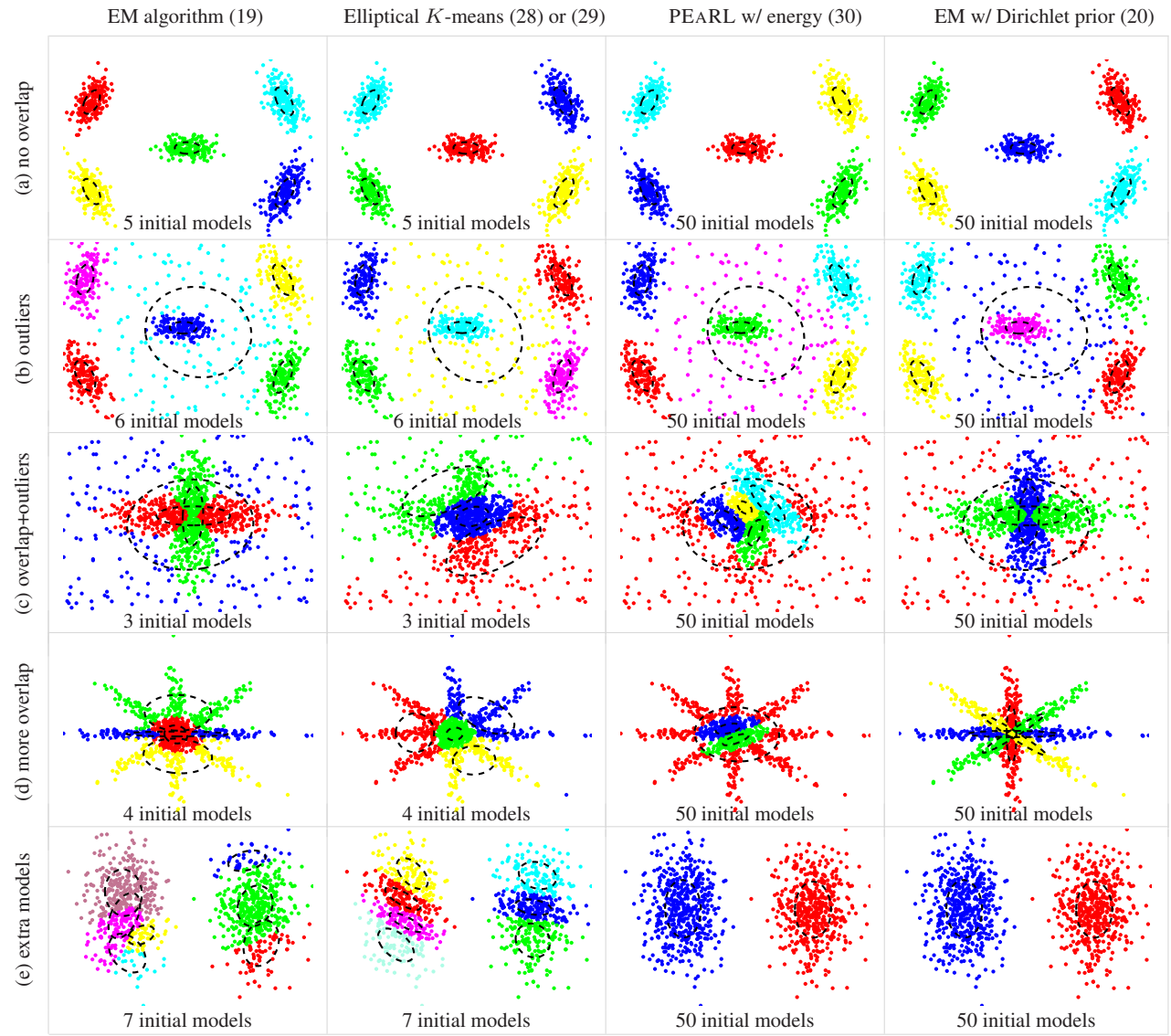


Fig. 10 Each row shows how GMM algorithms behave on a particular example. This table is for illustrative purposes, and is *not* meant to be a state-of-the-art comparison. (a) If models do not overlap then all algorithms work. (b) Most algorithms can handle uniform outliers by fitting an extra model. (c) EM finds overlapping models thanks to soft assignment; hard assignment has bias towards isolated models. (d) Basic EM (19) may easily get stuck in local minima with only a little more ambiguity in the data. But, EM with sparsity prior (20) can avoid such minima by choosing solution from a large set of model samples. Bad solution by PEARL in this case of heavy spatial overlap between the models is due to "hard assignments". (e) Basic EM and K-means usually fail when given too many initial models, whereas PEARL with label cost energy (30) and EM with Dirichlet-based posterior (20) keep the minimum number of models explaining the data. See Section 4.4 for discussion.

good lines from a large pool of initial models using sparsity control: label costs in (30) or Dirichlet prior in (20). The number of models generated by (30) and (20) is controlled by parameters h and α, see Fig.12(b,c).

Our main conclusion from Section 4 is that "hard assignments" have no particular disadvantages in cases where spatial overlap between the observed models constitutes only a small portion of their support. In image analysis problems (*e.g.* Figs.1,2,3) models often correspond to separate objects with distinct spatial support. Objects normally "occlude" each other rather than "intersect". Thus, "hard assignments" should be appropriate for many multi-model fitting problems in computer vision. In contrast to standard "soft assignment" methods like EM, besides sparsity prior (label costs) our general approach to model fitting can also integrate a spatial smoothness prior - the second term in ($\star'$) that was ignored in this section. Figs.1,2,3 show that this combination of regularizers is useful in vision.

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

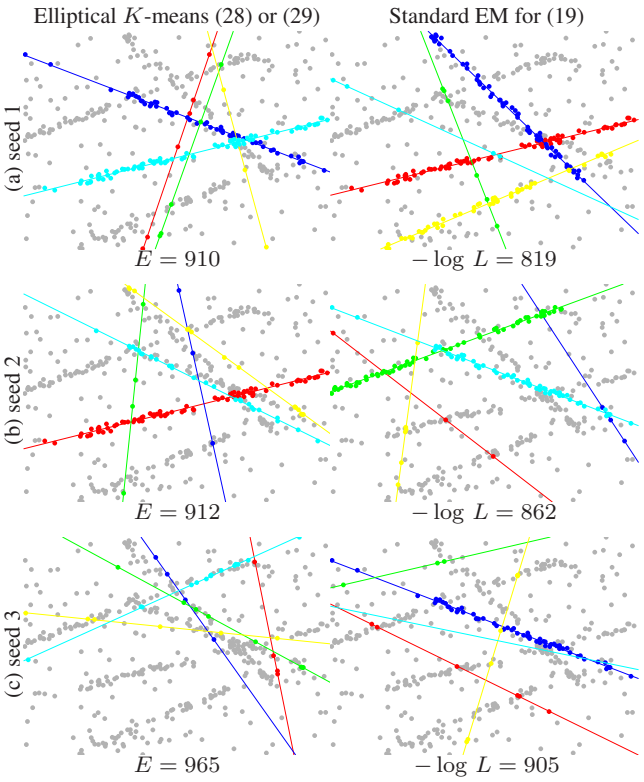


Fig. 11 Standard K-means and EM with a fixed number of models get stuck in local minima. The data points include (in total) 300 inliers for 5 lines and 180 outliers. Here we assumed that the correct number of models is known and estimated $K = 5$ lines and one outlier model. Solutions in (a)-(c) correspond to different initializations with 5 randomly sampled lines. The ground truth configuration has energy $E = 797$ in (29) and log-likelihood $-\log L = 721$ in (19).

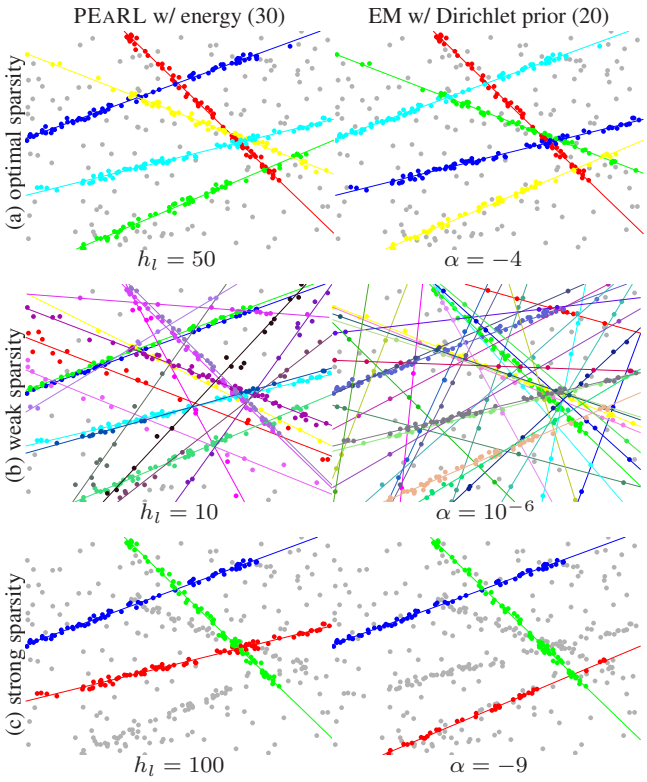


Fig. 12 Label costs in (30) or sparsity prior in (20) significantly improve the results on the data from Fig.11. Now a small number of models near ground truth (a) can be automatically computed from a large pool of random initial models, as in Fig.6. In contrast to Fig.11, the results are stable for different initializations as long as the set of initial randomly sampled lines is large enough (*e.g.* 500 lines). Parameters h and α control sparsity of the results (a-c).

Sec.3. The only changing components 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 ϕ 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 Simple synthetic examples (lines, circles, etc.)

Throughout this paper we used many illustrative examples of multi-line fitting. Below we detail the corresponding setup and discuss some additional synthetic tests with simple geometric models. Our energy ($\star'$) was motivated by applications in vision that involve images (Sections 5.1.2–5.3), but synthetic examples with simple models help to understand our energy, our algorithm, and their relation to standard methods.

Line fitting. Data points are sampled i.i.d. from a ground-truth set of line segments (*e.g.* Fig.6), under reasonably similar noise; outliers are sampled uniformly. Since the data is i.i.d. we set $V_{pq} = 0$ in ($\star'$) and use the greedy algorithm from Section 2.4. We also use fixed per-label costs as in (30). Keeping per-label costs independent of θ simplifies the re-estimation of θ itself.

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 σ^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 σ based on a prior. The data cost for a 2D point $x_p = (x_p^{\text{x}}, x_p^{\text{y}})$ is computed w.r.t.

orthogonal distance

$$D_p(\,l\,) \;=\; -\log\Big(\tfrac{1}{\sqrt{2\pi}\sigma}\exp\Big(-\tfrac{(ax_p^{\mathrm{x}}+bx_p^{\mathrm{y}}+c)^2}{2\sigma^2}\Big)\Big)\,. \qquad (33)$$

Besides the greedy algorithm for ($\star'$) without smoothness, we also tested α-expansion for high-order label cost potentials (Section 2.1). Not surprisingly, the greedy algorithm was by far the best algorithm when smooth costs are not involved. Greedy gives similar energies to α-expansion but is 5–20 times faster.

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 (33) and performing re-estimation.

Note that (33) does not correspond to a well-defined probability 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 it is common to fit full lines to data that was actually generated from line *intervals*, *e.g.* [31,64]. 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 misspecification, causing discrepancy between the energy being optimized versus the optimal solution from a generative viewpoint. Surprisingly, [31] 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 13 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 13 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 μ interpolating a and b. The probability of a data point appearing at position x is thus

$$\Pr(x\,|\,a,b,\sigma^2) = \int_0^1 \mathcal{N}\big(x\,|\,(1-t)a+tb,\sigma^2\big)\,\mathrm{d}t. \qquad (34)$$

In two dimensions, the above integral evaluates to

$$\tfrac{1}{4\pi\sigma^2\|a-b\|}\cdot\exp\Big(-\Big(\tfrac{x^{\mathrm{x}}(b^{\mathrm{y}}-a^{\mathrm{x}})-x^{\mathrm{y}}(b^{\mathrm{x}}-a^{\mathrm{y}})+a^{\mathrm{y}}b^{\mathrm{x}}-a^{\mathrm{x}}b^{\mathrm{y}}}{\sqrt{2}\sigma\|a-b\|}\Big)^2\Big)$$
$$\cdot\Big(\mathrm{erf}\Big(\tfrac{(x-b)\cdot(a-b)}{\sqrt{2}\sigma\|a-b\|}\Big)-\mathrm{erf}\Big(\tfrac{(x-a)\cdot(a-b)}{\sqrt{2}\sigma\|a-b\|}\Big)\Big) \qquad (35)$$

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

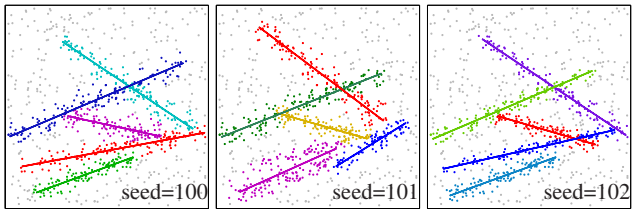


Fig. 13 We can also fit line *intervals* to the raw data in Figure 6. The three results above were each computed from a different set $\mathcal{L}$ of random initial proposals. See Section 5.1 for details.

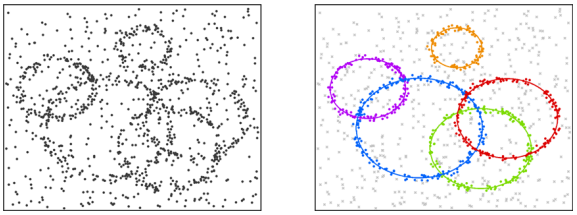


Fig. 14 For multi-model fitting, each label can represent a specific model from any family (Gaussians, lines, circles...). Above shows circle-fitting by minimizing geometric error of points.

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

$$E(X_l;a,b,\sigma) = -\sum_p \log\Pr(x_p\,|\,a,b,\sigma^2). \qquad (36)$$

Circle fitting. Figure 14 shows a typical circle-fitting result. Our circle parameters are center-point a, radius r, and variance σ^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 σ based on some prior. We find ML estimators numerically, much like for line intervals.

5.1.2 Homography Estimation

Energy ($\star'$) can be used to automatically detect multiple homographies in uncalibrated wide-base stereo image pairs. Our setup follows [31], so we give only a brief outline.

The input comprises two (static) images related by a fundamental matrix. We first detect SIFT features [43] 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) [28] of a match w.r.t. each candidate homography. Our set of neighbors $pq\in\mathcal{N}$ 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-Marquardt [28]. Figures 2c and 19 show representative results.

Fig. 15 Unsupervised segmentation by clustering simultaneously over pixels and color space using Gaussian Mixtures (color images) and non-parametric histograms (gray-scale images). Notice we find coarser clustering on baseball than Zabih & Kolmogorov [62] without over-smoothing. For segmentation, our energy is closer to Zhu & Yuille [63] but our algorithm is more powerful than region-competition.

5.1.3 Rigid Motion Estimation

The general setup follows [31,42] and is essentially the same as for homography estimation, except now 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 [28].

Again, SIFT matches work as data points. Initial proposals are generated by randomly sampling eight matching pairs. Fundamental matrices [28] are computed by minimizing the non-linear SSD error using Levenberg-Marquardt. Data costs measure the squared Sampson's distance [28] of a match with respect to each candidate fundamental matrix. Figures 1(c) and 21 show representative results.

5.2 Image Segmentation

Our goal is to automatically partition an image into some small number of regular segments with consistent appearance. In contrast to *superpixels*, our segments can be of any size and need not be contiguous. We propose to label the image using the following form of energy ($\star'$)

$$E(f,M) = \underbrace{\sum_{l\in\mathcal{L}} \sum_{p:f_p=l} -\log P(I_p \mid M_l)}_{\text{segment appearance}} + \underbrace{\lambda \sum_{pq\in\mathcal{N}} [f_p \neq f_q]}_{\text{segments' boundaries}} + \underbrace{\sum_{l\in\mathcal{L}} h_l \cdot \delta_l(f)}_{\text{segments' labels}} \tag{37}$$

where parameter M_l describes probability distribution associated with label l. For example, if values I_p are image intensities/colors[6] then vector M_l could represent an intensity histogram or parameters of some family of distributions.

[6] In general, I_p could represent any feature at pixel p, *e.g.* texture.

In what sense does segmentation energy (37) correspond to the goals proclaimed at the beginning of the previous paragraph? The third term sums penalties h_l for each label (model M_l) used in the image. This directly encourages a small number of segments. The second term is a standard expression for regularity of segment boundaries.

The information theory helps to show how the first term in (37) yields segments with *consistent appearance*. Indeed, following Kraft-McMillan theorem [44], any probability distribution $P(I \mid M)$ corresponds to some coding scheme for storing image intensities. Moreover, $-\log P(I_p \mid M)$ is the number of bits required to represent any given intensity I_p using coding scheme $P(I \mid M)$. Therefore,

$$\sum_{p\in S} -\log P(I_p \mid M)$$

is the number of bits required to describe the appearance of any segment $S \subset \mathcal{P}$ using coding scheme M. When optimizing over distribution M, the expression above yields the shortest possible description of segment S, that is

$$|S| \cdot H(I \mid S) \;=\; \inf_M \sum_{p\in S} -\log P(I_p \mid M)$$

where $H(I|S)$ is the entropy of intensities in segment S. Thus, optimization over all distribution models M makes the first term of energy (37) equal

$$\sum_{l\in\mathcal{L}} |S_l| \cdot H(I \mid S_l)$$

where $S_l = \{p : f_p = l\}$ is a segment with label l. This quantity can be further optimized over segmentation (labeling) f. It achieves its minimum for any segmentation with constant intensity segments where $H(I|S) = 0$. Such segments can be connected or disconnected. The size of the segments is also irrelevant. For example, single pixel segments

are optimal for the quantity above. Alternatively, segments could be connected components of the same intensity pixels. More generally, low values of the quantity above correspond to segments with low variability of intensity, that is, segments with consistent or homogeneous appearance.

In our segmentation experiments based on energy (37) the appearance models M_l are 256-dimensional histograms for greyscale images, and Gaussian mixtures in RGB space for color images. Initial proposals for models M_l were generated by sampling small patches of the image, just like in [63,62]. Similarly to [63,62] we iterated segmentation and model re-estimation steps to optimize our energy over f and M. We did not use segmentation-specific heuristics such as merging or splitting the histograms. Figure 8 shows running-time performance of our *coordinate descent* approach using α-expansions to optimize (37) over f, as in Section 2.

Our results in Figures 3 and 15 show how energy (37) balances regularity and homogeneity of segments. It is particularly instructional to compare image segmentation results in Figure 3(b)-(c). The result in (b) uses only spatial regularization as in energy (2), see [62]. This approach over-smoothes the segments even when the weight of the regularization term is too small to merge all "zebra" parts. The label costs term in (37) allows to obtain "zebra" (c) without over-smoothing. In this case we do not depend on the spatial regularization to merge all "zebra" parts and smoothing weight λ can be significantly reduced.

The label costs term in (37) could be used to obtain segments with certain preferred appearance by assigning penalties h_l depending on M_l. Also note that a general version of our label costs term in ($\star$) uses subsets of labels. This allows interesting new ideas for segmentation, as recently demonstrated in [39] in the context of object recognition.

It should be emphasized that we are not first to suggest energies with label costs for segmentation. A large amount of related work on image segmentation is based on *minimum description length* (MDL) principle [44] which provides information theoretic foundation for regularization energies like (37). The MDL principle was first proposed for unsupervised segmentation by Leclerc [41]. As further detailed in our section 5.3.1 on lossless compression, specific technical realization of MDL principle in [41] is distinct from ours. Leclerc derives energies somewhat different from (37) and optimizes them using continuation technique similar to *graduated nonconvexity* [7]. Further more, to simplify optimization [41] makes approximations, *e.g.* (2), that effectively ignore the label costs term.

Zhu & Yuille [63] used a continuous image segmentation energy inspired by MDL ideas of Leclerc. Specific formulation in [63] is much closer to ours and their functional is a continuous analogue of (37). They developed a *region competition* algorithm based on local contour evolution and explicit merging of adjacent regions to address the

mean-shift [14] optimization of energy (38)

Fig. 16 Comparing *mean-shift* results (a,c) versus optimization of energy (38) using UFL heuristics (b,d).

label cost term. A subsequent algorithm by Brox & Weickert [12] uses level sets to recursively partition the domain until it no longer pays to add regions (labels). Ben-Ayed & Mitiche [2] use multi-level sets to optimize an MDL-like region merging prior. Our work is first to demonstrate applications of powerful α-expansion approach to MDL-based image segmentation using energy (37).

To conclude this section we show some alternative ways of using label cost energies in segmentation. Clustering of image pixels represented by points (p, I_p) in $X \times Y \times Color$ space was popularized for image segmentation by the *mean-shift* algorithm [14]. Section 4 may suggest that label cost energies can be used in this segmentation framework as a regularization-based alternative to mean-shift. For example, Fig.16 compares mean-shift and clustering using energy

$$E(f, M) = \sum_{p \in \mathcal{P}} -\log P(p, I_p \mid M_{f_p}) + \sum_{l \in \mathcal{L}} h_l \cdot \delta_l(f) \qquad (38)$$

where distributions M_l were fixed-covariance Gaussians. Optimization was done using fast UFL heuristics from Sec.2. More sophisticated energy formulations are also possible.

5.3 Image Compression

Image compression is another application for label cost energy ($\star$). We separately consider *lossless* and *lossy* compression. It should be emphasized that we show straightforward coding schemes based on ($\star$) only to demonstrate the general idea. The main goal of our compression examples is to illustrate energy ($\star$) and minimization techniques from section 2. More sophisticated coding schemes using ($\star$) are possible, but they are beyond the scope of this paper.

5.3.1 Lossless Compression

Our approach to lossless compression uses energy (37) and is technically identical to our segmentation approach in section 5.2. The context of image compression, however, requires some specific interpretation for the terms in (37). In this section we further develop our information theoretic interpretation of this energy. In particular, this is necessary for section 5.3.2 on lossy compression .

On a conceptual level, we follow the same MDL principle as Leclerc [41]. However, our *descriptive language* is different and it leads to energies distinct from those in [41]. First, instead of *chain codes* describing boundaries, we describe interior regions of segments traversing all image pixels in a raster-scan order. Second, Leclerc describes an image as a combination of white noise with some piece-wise constant or piece-wise smooth function. Instead, we describe an image as a collection of segments with arbitrary *coding schemes*. Finally, Leclerc's goal is a piece-wise constant or piece-wise smooth restoration. Our goal is to find optimal segments and coding schemes describing images exactly with the minimum number of bits.

To better motivate our approach, we will first make several informal observations explaining why some appropriate segmentation may help to get a shorter description of an image. First of all, segments with consistent colors require fewer bits to represent their intensities. Second, segments with coherent boundaries require fewer bits to describe switches between coding schemes as image pixels are scanned in a predefined order. Third, smaller number of segments requites fewer coding schemes. Therefore, a small number of spatially coherent and color-consistent segments may help with compression. Note that the same segmentation criteria motivated energy (37) in section 5.2.

More formally, we will show that expression (37) corresponds to the total number of bits required to represent an image. We assume that each intensity is recorded in a raster-scan order using coding scheme M_l corresponding to pixel's label $l = f_p$. We already showed in section 5.2 that the first term in (37) is the number of bits required to store intensities using coding schemes M_l. More bits are also necessary to store all coding schemes them self. Clearly, the third term in (37) can represent these extra bits. In general, different coding schemes may vary in the number of bits required to store them. For example, Gaussian mixture models with larger number of components use more space. In this case, parameter h_l will depend on specific model M_l. In simpler examples, models M_l may take the same number of bits d. In this case parameter $h_l = d$ is a constant.

It remains to see that some additional bits (*e.g.* b) are required to indicate a coding scheme change when image traversal crosses a segment boundary. The second term[7] in (37) can represent such bits. In case if pixels are traversed row-by-row, one should set $\lambda = b$ and use horizontal 2-neighbor system $\mathcal{N}$. For column-by-column traversal, we need vertical 2-neighborhood. In case we want to estimate the number of bits for "model switching" without committing to one specific traversal direction, it makes sense to average over two options. In this case one should use 4-neighborhood with $\lambda = b/2$. It is also possible to account for any diagonal traversal of an image by adding the corresponding direction into the neighborhood system $\mathcal{N}$. Interestingly, Cauchy-Crofton formula [9] suggests the following information-theoretic interpretation of the geometric length of the segmentation boundary: it is a rotationally invariant measure of the expected number of "model switching" bits assuming that the direction of image traversal is chosen at a random angle.

In order to illustrate one specific example, assume that descriptions of all models M_l require the same number of bits d and that we can use either row-by-row or column-by-column traversals. Then, the average number of bits to represent image I segmented according to labeling f is

$$B(f, M \mid I) = \underbrace{\sum_{p \in \mathcal{P}} -\log P(I_p \mid M_{f_p})}_{\text{compressed intensities}} + \underbrace{\frac{b}{2} \sum_{pq \in \mathcal{N}} [f_p \neq f_q]}_{\text{coding scheme switches}} + \underbrace{d \sum_{l \in \mathcal{L}} \delta_l(f)}_{\text{coding schemes description}} \tag{39}$$

where $\mathcal{N}$ is a 4-neighborhood. This is a special case of energy (37). Minimizing (39) for given I over segmentation f and coding schemes M yields a solution for the *minimum description length* lossless compression of image I.

5.3.2 Lossy Compression and RD Optimization

We generalize ideas from the previous section to further illustrate applications for label costs functionals like ($\star$). This section follows Shannon's *rate-distortion* (RD) optimization approach to lossy image compression, see [27,47].

Minimization of energy (39) gives the optimal number of bits for storing image I without loss of information. The RD approach is to find some image $\bar{I}$ sufficiently close to I that compresses better than I. Formally, this problem can be written as the following constrained optimization

$$\min_{f, M, \bar{I}} \underbrace{B(f, M \mid \bar{I})}_{\text{compression rate}} \qquad s.t. \qquad \underbrace{\sum_{p \in \mathcal{P}} \left(I_p - \bar{I}_p\right)^2}_{\text{distortion measure}} \leq \epsilon$$

[7] Note that the second term in (37) is standard *piece-wise constant* (a.k.a. Potts) model of spatial smoothness in MRF literature.

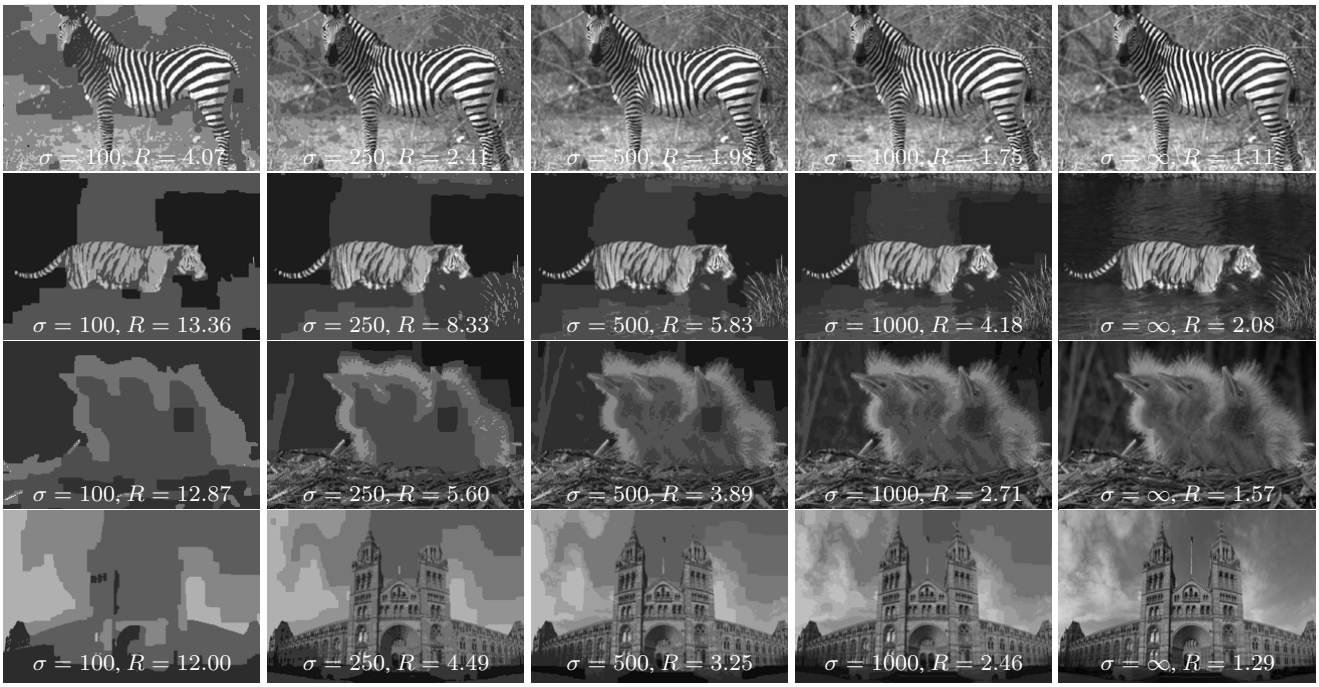


Fig. 17 Optimal "distortion" images $\bar{I}$ for various values of parameter σ in (40). These images $\bar{I}$ correspond to the best compression rate among all images with the same distortion measure. Since optimal $\bar{I} = I$ for $\sigma = \infty$, the right column shows the original images (lossless compression). Values of R show compression ratios: $R = \frac{|\mathcal{P}| \cdot H(U)}{arg\min_{f,M} B(f,M|\bar{I})}$ where $H(U)$ is the entropy of the uniform distribution over the range of intensities.

where ϵ fixes the distortion level. If $\epsilon = 0$ then $\bar{I} = I$ and the problem reduces to lossless compression (39). Higher levels of distortion $\epsilon > 0$ give solutions $\bar{I}$ with better compression. Note that besides the squared-error distortion measure above, Hamming distortion measure is also common in RD.

Following the discrete version of Lagrange method [20], the constrained problem above is closely related to unconstrained optimization of *generalized Lagrange function*

$$E(f, M, \bar{I}) = \sigma \underbrace{\sum_{p \in \mathcal{P}} \left(I_p - \bar{I}_p\right)^2}_{\text{distortion measure}} + \underbrace{B(f, M \mid \bar{I})}_{\text{compression rate}} \tag{40}$$

with parameter σ instead of ϵ. In particular, an optimal solution for (40) for any fixed value of Lagrange multiplier σ also solves the constrained problem above for some ϵ. Following the discussion after Theorem 1 in [20], the unconstrained minimum solution $(f^*, M^*, \bar{I}^*)$ for energy (40) achieves the lowest compression rate which is possible without exceeding this solution's distortion measure[8]. Figure 17 shows the balance of compression and distortion for the optimal solutions $\bar{I}$ corresponding to different σ in energy (40).

Similarly to other model fitting problems in this paper, we optimize (40) in a coordinate descent fashion iterating f, M and $\bar{I}$ optimization steps. We initialize $\bar{I} = I$. Note that f and M steps are analogous to (39) since $\bar{I}$ is fixed. Optimization over $\bar{I}$ for fixed f and M requires some clarification. Since variables $\bar{I}_p$ appear only in the unary terms in (40), optimization over $\bar{I}_p$ can be done very efficiently. For example, one can compute look-up tables $T(\cdot \mid l)$ for every currently supported model M_l (label l)

$$T(x \mid l) = \arg\min_i \Big(\sigma \left(x - i\right)^2 - \log P(i \mid M_l) \Big)$$

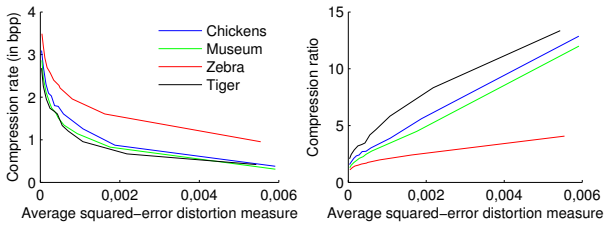
giving optimal value $\bar{I}_p = T(I_p \mid f_p)$ for any pixel p.

Figure 17 presents the results obtained in this fashion. All intensities are normalized to the range $[0, 1]$. We used ad-hoc values for parameters $b = 12$ and $d = 1000$ and varied the value of σ. For each optimal "distorted" image $\bar{I}$ we show the corresponding compression ratio R which is the ratio of the the length of the single-model uniform encoding of I to the optimal encoding length for $\bar{I}$.

Figure 18 shows *rate-distortion* and *ratio-distortion* functions which are common compression analysis tools ever since Shannon introduced RD approach in the 1940s. Each $\bar{I}$ obtained for certain σ in (40) corresponds to some optimal compression rate $B(f, M \mid \bar{I})/|\mathcal{P}|$ and average distortion measure $\sum_{p \in \mathcal{P}} \left(I_p - \bar{I}_p\right)^2 / |\mathcal{P}|$. Rate-distortion function in Fig.18(a) plots these values on the vertical and horizontal axis for different solutions $\bar{I}$ as σ varies. Similarly, Fig.18(b) plots compression ratio R versus distortion.

We should emphasize again that our compression results only illustrate the spectrum of applications for label costs energies ($\star$). More advanced description languages can fol-

[8] A similar discussion also appears after Proposition 1(b) in [36].



(a) rate-distortion function (b) ratio-distortion function

Fig. 18 Compression analysis plots for the results in Fig.17. The leftmost points correspond to lossless compression ($\sigma = \infty$).

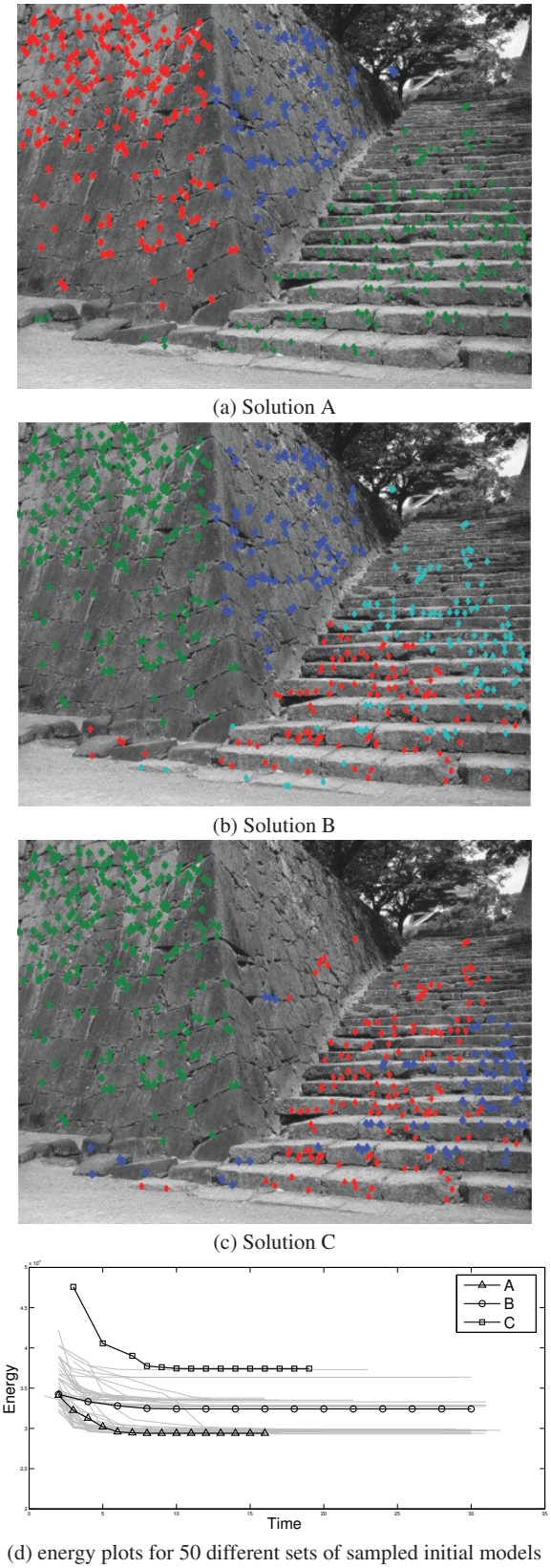
low the basic ideas in this section. Label costs naturally represent the length of coding schemes in any language.

6 Empirical Performance of Algorithms for ($\star'$)

This section presents an empirical comparison for several algorithmic variants to minimizing energy ($\star'$) where both smooth costs and label costs are present. In particular, we compare algorithms from section 2 and several algorithms originally designed for spatial regularization functional (2) which can be applied to ($\star'$) using some *merging heuristics* as in [31]. Our goal is to compare running times and energy values obtained on real examples in the context of geometric model fitting described in Sec.5.1.

Figure 19 illustrates our first homography fitting example (see Sec.5.1.2). The curves in (d) show how the energy ($\star'$) decreases in 50 different tests running PEARL with the extended α-expansion algorithm from section 2. Each test depends on some initial set of randomly samples models. The algorithm can converge to different solutions illustrated in Figure 19(a-c). Better results as in (a) correspond to solutions with lower energy values, and worse results as in (c) correspond to poor energy values. The black curve in Fig.20 is the average of 50 curves in Fig.19(d). This section uses such average curves to compare different combinatorial algorithms for minimizing label cost energies. In addition to homography fitting results in Fig.20, we also use two rigid motion estimation examples (see Sec.5.1.3) to compare similarly obtained average performance curves in Fig.21.

Now we briefly review combinatorial algorithms compared in this section. In contrast to other tested methods, the extended α-expansion algorithm from section 2 directly addresses label costs in ($\star$) without any extra heuristics. We test two versions of the algorithm: α+ (basic) consistently iterates expansion steps over all labels, and α++ (adaptive) removes labels corresponding to empty expansions until the "last" iteration validating local minima with respect to all labels. Both versions have the same optimality guarantees (see Sec.2). Our empirical results in Figures 20 and 21 suggest that α+ and α++ find solutions with comparable energy values. The adaptive method α++ converges faster.



(a) Solution A

(b) Solution B

(c) Solution C

(d) energy plots for 50 different sets of sampled initial models

Fig. 19 Homography fitting example ("Stairs"). Different runs of the algorithm (PEARL with α++) in (d) converge to solutions with different energy values depending on a specific initial collection of randomly sampled models. As shown in (a-c), lower energy solutions correspond to better practical results.

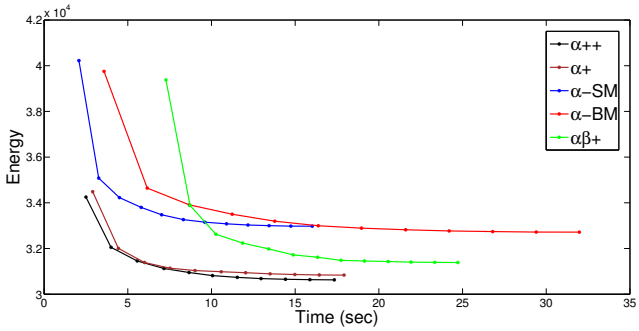


Fig. 20 Homography fitting example ("Stairs") for different algorithms minimizing energy ($\star$): α++ and α+ are two versions of extended α-expansion from Sec.2; $\alpha\beta$+ is a straightforward modification of the standard $\alpha\beta$-swap [11]; α-SM and α-BM are standard α-expansions with different merging heuristics [31]. The plots show values of energy ($\star'$) obtained after each iteration of segmentation and re-estimation, see Sec.3. As in PEARL [31], the labels are initialized by randomly sampling 1000 models. Each plot above is obtained by averaging energy curves for 50 different initializations as in Fig.19(d).

Other tested methods are based on standard algorithms for energy (2) adapted to label cost in ($\star$) using some heuristics. For example, [31] uses basic α-expansion [11] for the first two terms in ($\star$) and adds a separate merging step to account for the label costs. Each merging step tries to replace some pair of labels A and B in the current solution with one label C. Two segments $\mathcal{A} = \{p : f_p = A\}$ and $\mathcal{B} = \{p : f_p = B\}$ are merged if and only if assigning some label C to combined segment $\mathcal{A} \cup \mathcal{B}$ lowers overall energy ($\star$). Note that merging decreases the second and the third terms in ($\star$) but it can increase the first (data) term. Iterating standard α-expansions with merging steps is guaranteed to decrease energy ($\star$) after each iteration. Note that separate merging steps for minimizing MDL-based functionals like ($\star$) were also used in [41,63] in the context of *continuation* methods and *variational* approaches.

We tested two merging heuristics [31]: α-SM (simple merge) tries to merge two segments using $C = A$ or $C = B$, and α-BM (best merge) tries the optimal label C for two current segments $\mathcal{A} = \{p : f_p = A\}$ and $\mathcal{B} = \{p : f_p = B\}$

$$C^* = \arg\min_C \sum_{p \in \mathcal{A} \cup \mathcal{B}} D_p(C).$$

Due to extra optimization procedure α-BM is slower than α-SM but it generates lower energy values, see Figs.20,21.

We also note that the standard $\alpha\beta$-swap algorithm [11] was originally designed for smoothness energy ($\star$) but can be easily extended to label cost energy ($\star$). At each step the swap algorithm works with two fixed labels A and B and a region $\mathcal{A} \cup \mathcal{B}$. Only two trivial outcomes of a swap move change the label costs: when all nodes in $\mathcal{A} \cup \mathcal{B}$ are assigned either label A or B. The standard swap method does not account for the label cost term in ($\star$). Yet, it is easy to compare the outcome of an optimal $\alpha\beta$-swap move with two trivial solutions and choose one with the lowest value of energy ($\star$). We use symbol $\alpha\beta$+ to refer to this algorithm and its empirical results in Figure 20.

While discrete energy ($\star$) could be addressed by many combinatorial optimization techniques (*e.g.* [26,41,35]) or their modifications, our empirical evaluation is focused on graph cut methods that we consider more promising due to optimality guarantees associated with them. The experiments in Figure 20 show that α++, an adaptive version of extended α-expansion in Sec.2, generated better quality solutions faster than other methods. Standard α-expansion with a "best merge" heuristic α-BM [31] obtained better energy values in Fig.21 but it was also much slower. Comparing α-BM with α-SM ("simple merge" version of the same algorithm) suggests that α-BM benefits from adaptive new model proposals. In fact, α-BM is the only method in our tests that used adaptively generated new proposals in addition to basic model re-estimation. Note that rigid motion models in Fig.21 have higher dimensionality than (planar) homographies in Figs.19-20. Generating label proposals adaptively could be a practical mechanism improving exploration of larger label spaces of higher-dimensions.

Our general practical observation is that often all tested algorithms α++, α+, $\alpha\beta$+, α-SM, α-BM generate comparable results. In most cases, however, it is easier to use α++ as it is fast, robust, and does not rely on extra merging heuristics. In higher dimensional model-fitting problems the combination of PEARL and α++ may further benefit from additional application-specific mechanisms adaptively generating new model proposals.

7 Discussion

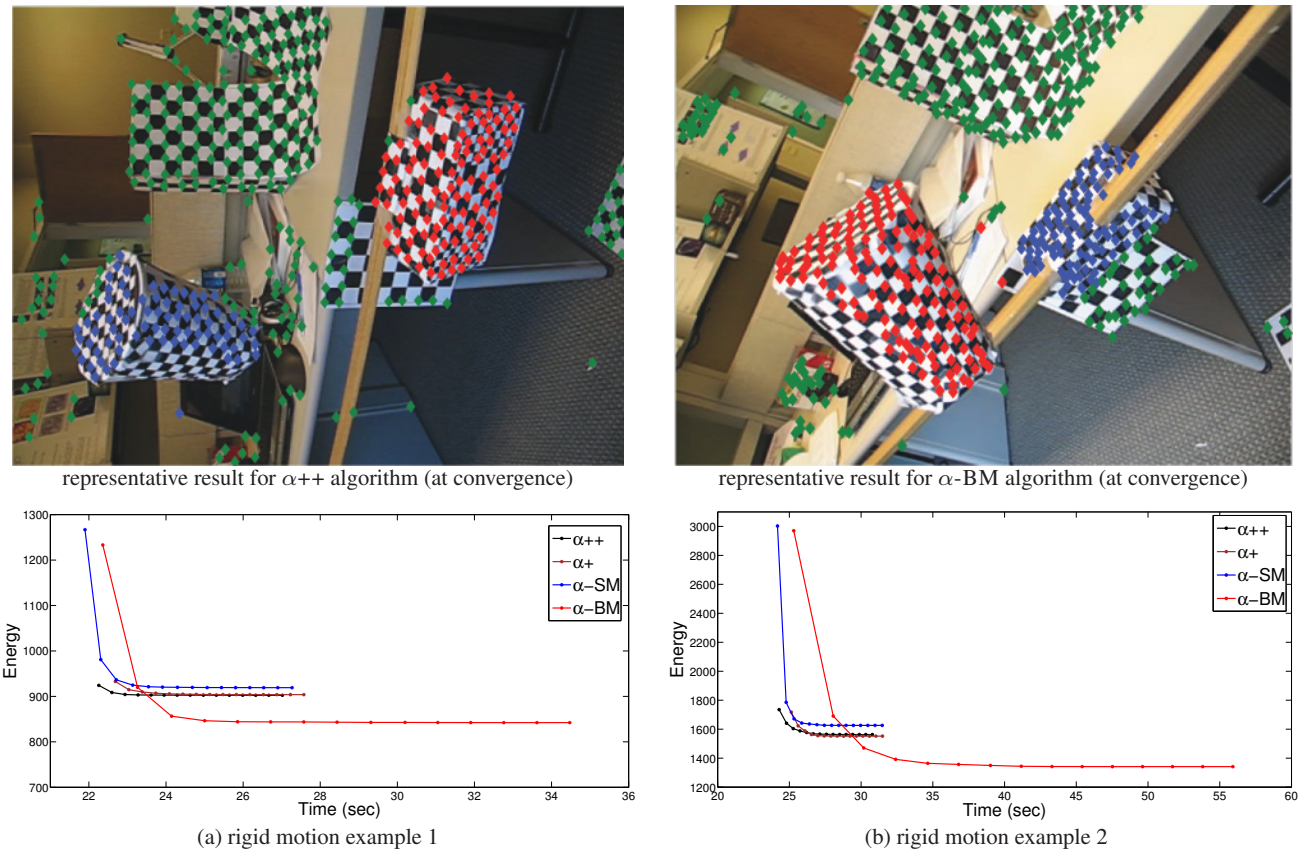
The potential applications of our algorithm are nearly as broad as for α-expansion. Our new algorithm can be applied whenever observations are known *a priori* to be positively correlated, for example in space or in time, whereas classical mixture model algorithms (Section 4) are largely designed for i.i.d. data.

Our C++ code and MATLAB wrapper are available at `http://vision.csd.uwo.ca/code/`. Besides minimizing general energy ($\star$), 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 2.4), 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.[9]

Our new α-expansion code optionally uses a simple strategy to invest expansions mainly on 'successful' labels. This is

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



representative result for α++ algorithm (at convergence)

representative result for α-BM algorithm (at convergence)

(a) rigid motion example 1

(b) rigid motion example 2

Fig. 21 Rigid motion estimation examples (Vidal's data set [56]) comparing different algorithms minimizing energy ($\star$): α++, α+, α-SM, α-BM. Algorithm $\alpha\beta$+ generated solutions with similar energy values but it was much slower than the other methods. Thus, we chose not to show its energy curves for the rigid motion estimation examples above.

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 α-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 α-expansion by 40–60%, but this is still relatively fast for such difficult energies [52]. This slowdown may be because the Boykov-Kolmogorov maxflow algorithm [10] 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-occurrence potential first proposed in [18]. 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, beach) unless the local evidence is strong enough. With respect to object recognition/segmentation with co-occurrence, similar costs were independently developed by Ladický *et al.* [39]. We foresee further applications for high-order label costs in multi-motion and multi-homography estimation.

Relation to Ladický *et al.* [39]. The application in [39] is object recognition with co-occurrance statistics. They are motivated by the principle of *parsimony:* if several segmentations explain the image equally well, then the one that requires the fewest object labels should be preferred. They develop an extension to α-expansion that is equivalent to ours, but they also consider energies outside the class of co-occurrence potentials (subset costs) that we defined earlier in [18]. However, their class of energies is not submodular with respect to expansion and so they apply a heuristic with no guarantee of finding an optimal expansion move for energies outside our class.

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

$$\sum_{P \subseteq \mathcal{P}} \sum_{L \subseteq \mathcal{L}} h_L^P \cdot \delta_L(f_P) \tag{41}$$

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^α 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 [33]**.** 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 $\gamma_\alpha \leq \gamma_{\max}$ for all $\alpha \in \mathcal{L}$. This potential encodes a label-specific reward $\gamma_{\max} - \gamma_\alpha$ for clique P taking label α 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 γ_α).

Let $\bar{\alpha}$ denote the set of all labels except α, *i.e.* the set $\mathcal{L} \setminus \{\alpha\}$. A regional label subset cost over clique P can encode the P^n Potts potential in energy ($\star$) as follows:

1. Set cost $h^P_{\bar{\alpha}} := \gamma_{\max} - \gamma_\alpha$ 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 $h^P_{\bar{\alpha}}$ is non-negative by definition of $\psi_P(\cdot)$, thus a P^n Potts potential can be expressed as a sum of high-order label costs.

The P^n Potts potential and its robust generalization [34] 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.

Learning label costs. Our paper studied label costs in an unsupervised setting where parameters are chosen based on information criteria or tuned manually. It is important to note that energy ($\star$) and the α-expansion-based inference algorithm can be used in supervised settings as well. The label cost terms are included in energy ($\star$) linearly and can thus be learned by max-margin methods [54,57]. This approach was recently used for CRF learning, *e.g.* [53].

Acknowledgements We would like to thank Fredrik Kahl for referring us to the works of Li and Vidal, 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.

A - Optimality Results

Proof of Theorem 1. The proof idea follows Theorem 6.1 of [11]. Let us fix some $\alpha \in \mathcal{L}$ and define

$$\mathcal{P}_\alpha \stackrel{\text{def}}{=} \{\, p \in \mathcal{P} : f^*_p = \alpha \,\}. \tag{42}$$

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

$$f^\alpha_p = \begin{cases} \alpha & \text{if } p \in \mathcal{P}_\alpha \\ \hat{f}_p & \text{otherwise.} \end{cases} \tag{43}$$

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

$$E(\hat{f}) \leq E(f^\alpha). \tag{44}$$

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 $\mathcal{P}_\alpha$:

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

The following facts now hold:

$$E(f^\alpha)|_{\mathcal{I}^\alpha} = E(f^*)|_{\mathcal{I}^\alpha} \tag{45}$$

$$E(f^\alpha)|_{\mathcal{O}^\alpha} = E(\hat{f}\,)|_{\mathcal{O}^\alpha} \tag{46}$$

$$E(f^\alpha)|_{\mathcal{B}^\alpha} \leq cE(f^*)|_{\mathcal{B}^\alpha}. \tag{47}$$

Inequality (47) follows from the fact that $V(f^\alpha_p, f^\alpha_q) \leq cV(f^*_p, f^*_q)$ for any $pq \in \mathcal{B}^\alpha$.

Let E_H denote the label cost terms of energy E. Using (45), (46) and (47) we can rewrite (44) as

$$E(\hat{f}\,)|_{\mathcal{I}^\alpha} + E(\hat{f}\,)|_{\mathcal{B}^\alpha} + E_H(\hat{f}\,) \tag{48}$$

$$\leq E(f^\alpha)|_{\mathcal{I}^\alpha} + E(f^\alpha)|_{\mathcal{B}^\alpha} + E_H(f^\alpha) \tag{49}$$

$$\leq E(f^*)|_{\mathcal{I}^\alpha} + cE(f^*)|_{\mathcal{B}^\alpha} + E_H(f^\alpha) \tag{50}$$

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

$$E_H(f^\alpha) \leq E_H(\hat{f}) + \sum_{\substack{L \subseteq \mathcal{L} \setminus \hat{\mathcal{L}} \\ \alpha \in L}} h_L \tag{51}$$

where $\hat{\mathcal{L}}$ contains only the unique labels in $\hat{f}$. We also let $\mathcal{L}^*$ denote the unique labels in f^*.

To bound the total energy we sum expressions (48) and (50) over all labels $\alpha \in \mathcal{L}^*$ to arrive at the following:

$$\sum_{\alpha \in \mathcal{L}^*} \Big(E(\hat{f}\,)|_{\mathcal{I}^\alpha} + E(\hat{f}\,)|_{\mathcal{B}^\alpha} \Big) \tag{52}$$

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

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 (52), 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 (52). Therefore equation (52) can be rewritten as

$$\begin{aligned} E(\hat{f}) \leq\ & E(f^*) + (2c-1)E_V(f^*) - E(\hat{f})|_{\mathcal{B}} \\ & + E_H(\hat{f}) - E_H(f^*) + \sum_{L \subseteq \mathcal{L} \setminus \hat{\mathcal{L}}} h_L |L \cap \mathcal{L}^*|. \end{aligned} \tag{53}$$

Observe that the second line of (53) involving label costs is equal to

$$\sum_{\substack{L\subseteq\mathcal{L}\setminus\mathcal{L}^*\\ L\cap\hat{\mathcal{L}}\neq\emptyset}} h_L \;+\; \sum_{\substack{L\subseteq\mathcal{L}\setminus\hat{\mathcal{L}}\\ L\cap\mathcal{L}^*\neq\emptyset}} h_L\big(|L\cap\mathcal{L}^*|-1\big)\,. \tag{54}$$

The right-hand sum includes label costs that f^* pays but that $\hat{f}$ does not. Expression (54) can be bounded by

$$\leq \sum_{\substack{L\subseteq\mathcal{L}\setminus\mathcal{L}^*\\ L\cap\hat{\mathcal{L}}\neq\emptyset}} h_L \;+\; d\cdot\sum_{\substack{L\subseteq\mathcal{L}\setminus\hat{\mathcal{L}}\\ L\cap\mathcal{L}^*\neq\emptyset}} h_L, \quad \text{where } d=\max_{\substack{L\subset\mathcal{L}\\ h_L>0}}|L|-1 \tag{55}$$

$$\leq \sum_{L\subseteq\mathcal{L}\setminus\mathcal{L}^*} h_L \;+\; dE_H(f^*) \tag{56}$$

where d is understood to be zero if all $h_L=0$. Combining (53) with (56) and using the fact that $dE_H(f^*)\geq 0$ we can simplify the bound as

$$E(\hat{f})\leq\; E(f^*) \;+\; (2c-1)E_V(f^*) \;+\; dE_H(f^*) \;+\; \sum_{L\subseteq\mathcal{L}\setminus\mathcal{L}^*} h_L \tag{57}$$

$$\leq (2c+d)E(f^*) \;+\sum_{L\subset\mathcal{L}} h_L. \tag{58}$$

We have derived *a posteriori* bounds (53) and (57) with respect to any particular $\hat{f}$ and f^*. Assuming $D_p\geq 0$ we have *a priori* bound (58). ■

B - Equivalence of Label Costs and Sparsity Prior

Proof of Theorem 2. Assuming $h=\log\frac{1}{\gamma}$, the negative logarithm of distribution (27) gives the following posterior energy

$$\begin{aligned}E'(f;\theta,\omega) &= -\sum_{p\in\mathcal{P}}\log\left(\omega_{f_p}\cdot\Pr(x_p\,|\,\theta_{f_p})\right)+\sum_{l\in\mathcal{L}}h\cdot[\omega_l>\varepsilon]\\ &= -\sum_{l\in\mathcal{L}}k_l\log\omega_{f_p}-\sum_{p\in\mathcal{P}}\log\Pr(x_p\,|\,\theta_{f_p})+\sum_{l\in\mathcal{L}}h\cdot[\omega_l>\varepsilon]\end{aligned} \tag{59}$$

where $k_l=\#\{p: f_p=l\}$ is the number of pixels assigned to label l. We will show that for $\varepsilon\leq\gamma^{|\mathcal{L}|}/\,|\mathcal{P}|^{|\mathcal{P}|}$ posterior energy E' in (59) has the same minimum as label cost energy E from (30)

$$E(f;\theta,\omega)=-\sum_{l\in\mathcal{L}}k_l\log\omega_{f_p}-\sum_{p\in\mathcal{P}}\log\Pr(x_p\,|\,\theta_{f_p})\;+\;\sum_{l\in\mathcal{L}}h\cdot\delta_l(f) \tag{60}$$

that differs from (59) in the last term.

Assume that the minimum of E in (60) is achieved at $\hat{f},\hat{\omega},\hat{\theta}$. It is easy to check that at any optimal solution of E we have $\omega_l=k_l/\,|\mathcal{P}|$. Since $\varepsilon<1/|\mathcal{P}|$ then $[\hat{\omega}_l>\varepsilon]=\delta_l(\hat{f})$. Thus, $E'(\hat{f},\hat{\omega},\hat{\theta})=E(\hat{f},\hat{\omega},\hat{\theta})$ and

$$min_{f,\omega,\theta}E'\leq min_{f,\omega,\theta}E.$$

It remains to show that

$$min_{f,\omega,\theta}E'\geq min_{f,\omega,\theta}E$$

as this implies that $\hat{f},\hat{\omega},\hat{\theta}$ is a global minimizer for both E and E'.

Assume that f',ω',θ' is a global minimum of E'. Let k'_l be the number of pixels assigned label l in labeling f'. To complete the proof we will show that $E'(f',\omega',\theta')\geq E(f',\omega'',\theta')$, where $\omega''_l=k'_l/\,|\mathcal{P}|$.

Note that $k'_l=0$ implies $\delta_l(f')=0$. Thus, the last term of E is not greater than the corresponding term in energy E'. Without loss of generality let us assume that $k'_l>0,\forall l\in\mathcal{L}$. The only difference between terms of E and E' may appear if some label l with nonzero support $k'_l>0$ has the corresponding weight ω'_l that is less than ε. Let $\mathcal{Z}$ be the set of such labels. Then the difference between E and E' can be written in the following form:

$$E'(f',\omega',\theta')-E(f',\omega'',\theta')=-\sum_{l\in\mathcal{L}}k'_l\log\omega'_l+h\,|\mathcal{L}/\mathcal{Z}|$$

$$+\sum_{l\in\mathcal{L}}k'_l\log\omega''_l-h\,|\mathcal{L}|\geq-\sum_{l\in\mathcal{Z}}k'_l\log\omega'_l+\sum_{l\in\mathcal{L}}k'_l\log\frac{k'_l}{|\mathcal{P}|}-h\,|\mathcal{Z}|$$

$$\geq-\log\varepsilon\sum_{l\in\mathcal{Z}}k'_l+\sum_{l\in\mathcal{L}}k'_l\log\frac{k'_l}{|\mathcal{P}|}-h\,|\mathcal{Z}|$$

$$\geq-\log\varepsilon-|\mathcal{P}|\log|\mathcal{P}|-h\,|\mathcal{L}|\,.$$

This difference is not less than zero if $\varepsilon\leq\gamma^{|\mathcal{L}|}/\,|\mathcal{P}|^{|\mathcal{P}|}$. ■

C - Implementation details for EM

Below we detail our implementation of EM algorithm for maximizing likelihood (19) and posterior (20). The results of this implementation were presented in Figures 10, 11, 12 from Section 4 in the context of GMM estimation and multi-line fitting. Note that our EM implementation for posterior (20) is similar to the EM algorithm in [22].

E-step: As in standard EM we compute probabilities for points p to be in each class l that are often called *responsibilities*

$$g_{pl}=\frac{\omega_l\cdot\exp\left(-D_p(\,l\,)\right)}{\sum_s\omega_s\cdot\exp\left(-D_p(\,s\,)\right)}$$

where data fit is $D_p(\,l\,)=-\log Pr(x_p|\theta_l)$. In case of GMM one should use the standard Normal distribution. In case of line fitting we use $D_p(\,l\,)$ as in formula (33) assuming that inliers for each line have Gaussian orthogonal errors.

M-step: To re-estimate each Gaussian model's parameters μ_l and Σ_l in GMM, we use the standard weighted MLE formula, e.g see [6]. We use g_{pl} as a weight for each point p. To re-estimate each line's parameters a_l, b_l, and c_l in multi-line fitting, we use a closed form solution for weighted orthogonal regression. Re-estimation of noise level σ is analogous to re-estimation of variance for 1-D gaussian:

$$\sigma_l=\sqrt{\frac{\sum_p g_{pl}\left(a_l x^{\mathrm{x}}_p+b_l x^{\mathrm{y}}_p+c_l\right)^2}{\sum_p g_{pl}}}.$$

Optimal mixture weights ω_l can be obtained analytically by minimizing (19) or (20) under constraint $\sum_{l\in\mathcal{L}}\omega_l=1,\;\;\omega_l\geq 0$. In case of likelihood (19) one gets the standard formula for EM

$$\omega_l=\frac{\sum_p g_{pl}}{|\mathcal{P}|}.$$

In case of posterior (20) including additional Dirichlet prior (21) for weights ω_l, the optimal solution is

$$\omega_l=\frac{\sum_p g_{pl}+\alpha-1}{|\mathcal{P}|+|\mathcal{L}|\cdot(\alpha-1)}. \tag{61}$$

Interestingly, the EM algorithm for (19) and (20) differs only in the corresponding formulas for re-estimating mixture weights ω.

Note that Dirichlet prior is a proper (integrable) distribution only for $\alpha > 0$. But it does not generate sufficiently sparse MAP solutions for (20) even with very small positive values of α, see Fig.12b. As discussed in [22], one can use $\alpha < 0$ giving much stronger sparsity, see Fig.12a,c. Technically, the EM algorithm above still works since it does not compute normalization constant or re-estimate parameter α.

Sometimes equation (61) produces negative weight $\omega_l < 0$. In this case the extremum point is outside the valid domain for constrained optimization and, consequently, the maximum value is achieved at the border of the domain where Dirichlet-based posterior (20) has infinite value. Similarly to [22], we solve this problem in the following way. We drop the mixture components with negative weights and renormalize other weights ω_l to fit their sum to 1. Note that this heuristic produces a "jumps" of the posterior function (since it goes to $+\infty$ when any weight goes to zero) making it difficult to compare solutions with different number of components.

Since Dirichlet-based posterior (20) automatically controls sparsity of the solution, we initialize this version of EM with a large number of randomly sampled models. This approach to minimizing (20) is robust to initialization and avoids local minima [22]. This is also similar to how PEARL [31] avoids local minima for ($\star'$).

References

1. H. Akaike. A new look at statistical model identification. *IEEE Trans. on Automatic Control*, 19:716–723, 1974.
2. I. B. Ayed and A. Mitiche. A Region Merging Prior for Variational Level Set Image Segmentation. *IEEE Trans. on Image Processing (TIP)*, 17(12):2301–2311, 2008.
3. D. A. Babayev. Comments on the note of Frieze. *Mathematical Programming*, 7(1):249–252, December 1974.
4. O. Barinova, V. Lempitsky, and P. Kohli. On the Detection of Multiple Object Instances using Hough Transforms. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2010.
5. S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *International Conf. on Computer Vision (ICCV)*, 1999.
6. C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, August 2006.
7. A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press: Cambridge, MA, 1987.
8. E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225, 2002.
9. Y. Boykov and V. Kolmogorov. Computing Geodesics and Minimal Surfaces via Graph Cuts. In *International Conf. on Computer Vision (ICCV)*, 2003.
10. Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(9):1124–1137, 2004.
11. Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(11):1222–1239, 2001.
12. T. Brox and J. Weickert. Level set based segmentation of multiple objects. In *Pattern Recognition*, volume 3175 of *LNCS*, pages 415–423, 2004.
13. K. P. Burnham and D. R. Anderson. *Model Selection and Multimodel Inference*. Springer, 2002.
14. D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 24(5):603–619, May 2002.
15. 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.
16. G. Cornuejols, G. L. Nemhauser, and L. A. Wolsey. The Uncapacitated Facility Location Problem. Technical Report 605, Op. Research, Cornell University, August 1983.
17. 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.
18. A. Delong, A. Osokin, H. Isack, and Y. Boykov. Fast Approximate Energy Minimization with Label Costs. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2010.
19. 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.
20. H. Everett. Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources. *Operations Research*, 11(3):399–417, May-June 1963.
21. U. Feige. A Threshold of $\ln n$ for Approximating Set Cover. *Journal of the ACM*, 45(4):634–652, 1998.
22. M. A. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(3):381–396, 2002.
23. 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.
24. D. Freedman and P. Drineas. Energy minimization via graph cuts: settling what is possible. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2005.
25. A. M. Frieze. A cost function property for plant location problems. *Mathematical Programming*, 7(1):245–248, December 1974.
26. S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.
27. A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 2001.
28. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
29. D. S. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming*, 22(1):148–162, 1982.
30. D. Hoiem, C. Rother, and J. Winn. 3D LayoutCRF for Multi-View Object Class Recognition and Segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.
31. H. N. Isack and Y. Boykov. Energy-based Geometric Multi-Model Fitting. *International Journal on Computer Vision (IJCV)*, 97(2):123–147, April 2012. earlier version: H.Isack MS thesis, UWO, 2009.
32. J. Kleinberg and E. Tardos. Approximation Algorithms for Classification Problems with Pairwise Relationships: Metric Labeling and Markov Random Fields. *Journal of the ACM*, 49(5), 2002.
33. P. Kohli, M. P. Kumar, and P. H. S. Torr. $\mathcal{P}^3$ & Beyond: Solving Energies with Higher Order Cliques. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.
34. P. Kohli, L. Ladický, and P. H. S. Torr. Robust Higher Order Potentials for Enforcing Label Consistency. *International Journal on Computer Vision (IJCV)*, 82(3):302–324, 2009.
35. V. Kolmogorov. Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(10):1568–1583, October 2006.
36. V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. In *International Conf. on Computer Vision (ICCV)*, 2007.
37. V. Kolmogorov and R. Zabih. What Energy Functions Can Be Optimized via Graph Cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(2):147–159, 2004.
38. A. A. Kuehn and M. J. Hamburger. A Heuristic Program for Locating Warehouses. *Management Science*, 9(4):643–666, 1963.
39. L. Ladicky, C. Russell, P. Kohli, and P. Torr. Graph Cut based Inference with Co-occurrence Statistics. In *European Conf. on Computer Vision (ECCV)*, September 2010.

40. N. Lazic, I. Givoni, B. Frey, and P. Aarabi. FLoSS: Facility Location for Subspace Segmentation. In *International Conf. on Computer Vision (ICCV)*, 2009.
41. Y. G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision (IJCV)*, 3(1):73–102, May 1989.
42. H. Li. Two-view Motion Segmentation from Linear Programming Relaxation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.
43. D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal on Computer Vision (IJCV)*, 60:91–110, 2004.
44. D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
45. T. Mitchell and J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.
46. G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions – I. *Mathematical Programming*, 14(1):265–294, 1978.
47. A. Ortega and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine*, 15(6):23–50, September 1998.
48. C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive Foreground Extraction using Iterated Graph Cuts. In *ACM SIGGRAPH*, 2004.
49. D. B. Shmoys, E. Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract). In *ACM Symposium on Theory of Computing (STOC)*, pages 265–274, 1998.
50. M. Sun. A Tabu Search Heuristic for the Uncapacitated Facility Location Problem. In *Metaheuristic Optimization via Memory and Evolution*, volume 30, pages 191–211. Springer US, 2005.
51. K. K. Sung and T. Poggio. Example based learning for view-based human face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 20:39–51, 1995.
52. 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 Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(6):1068–1080, June 2008.
53. M. Szummer, P. Kohli, and D. Hoiem. Learning CRFs using Graph Cuts. In *European Conf. on Computer Vision (ECCV)*, 2008.
54. B. Taskar, V. Chatalbashev, and D. Koller. Learning Associative Markov Networks. In *International Conf. on Machine Learning (ICML)*, 2004.
55. P. H. S. Torr. Geometric Motion Segmentation and Model Selection. *Philosophical Trans. of the Royal Society A*, pages 1321–1340, 1998.
56. R. Tron and R. Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.
57. I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6(2):1453–1484, 2006.
58. N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. SMEM Algorithm for Mixture Models. *Neural Computation*, 12(9):2109–2128, 2000.
59. T. Werner. High-arity Interactions, Polyhedral Relaxations, and Cutting Plane Algorithm for Soft Constraint Optimisation (MAP-MRF). In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2008.
60. O. J. Woodford, C. Rother, and V. Kolmogorov. A Global Perspective on MAP Inference for Low-Level Vision. In *International Conf. on Computer Vision (ICCV)*, October 2009.
61. J. Yuan and Y. Boykov. TV-Based Multi-Label Image Segmentation with Label Cost Prior. In *British Machine Vision Conference (BMVC)*, Sept 2010.
62. R. Zabih and V. Kolmogorov. Spatially Coherent Clustering with Graph Cuts. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2004.
63. S. C. Zhu and A. L. Yuille. Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 18(9):884–900, 1996.
64. M. Zuliani, C. S. Kenney, and B. S. Manjunath. The multi-RANSAC algorithm and its application to detect planar homographies. In *International Conf. on Image Processing (ICIP)*, 2005.