International Journal of Computer Vision (IJCV) **August 2018 (accepted)**
https://doi.org/10.1007/s11263-018-1115-1

Kernel Cuts: Kernel and Spectral Clustering meet Regularization

Meng Tang · Dmitrii Marin · Ismail Ben Ayed · Yuri Boykov

Received: date / Accepted: date

Abstract This work bridges the gap between two popular methodologies for data partitioning: *kernel clustering* and *regularization*-based segmentation. While addressing closely related practical problems, these general methodologies may seem very different based on how they are covered in the literature. The differences may show up in motivation, formulation, and optimization, *e.g.* spectral relaxation vs max-flow. We explain how regularization and kernel clustering can work together and why this is useful. Our joint energy combines standard regularization, *e.g.* MRF potentials, and kernel clustering criteria like normalized cut. Complementarity of such terms is demonstrated in many applications using our bound optimization *Kernel Cut* algorithm for the joint energy (code is publicly available). While detailing combinatorial move-making, our main focus are new linear *kernel* and *spectral* bounds for kernel clustering criteria allowing their integration with any regularization objectives with existing discrete or continuous solvers.

Keywords Segmentation · Markov Random Fields · Spectral Clustering · Kernel Methods · Bound Optimization

1 Introduction: Terminology and Motivation

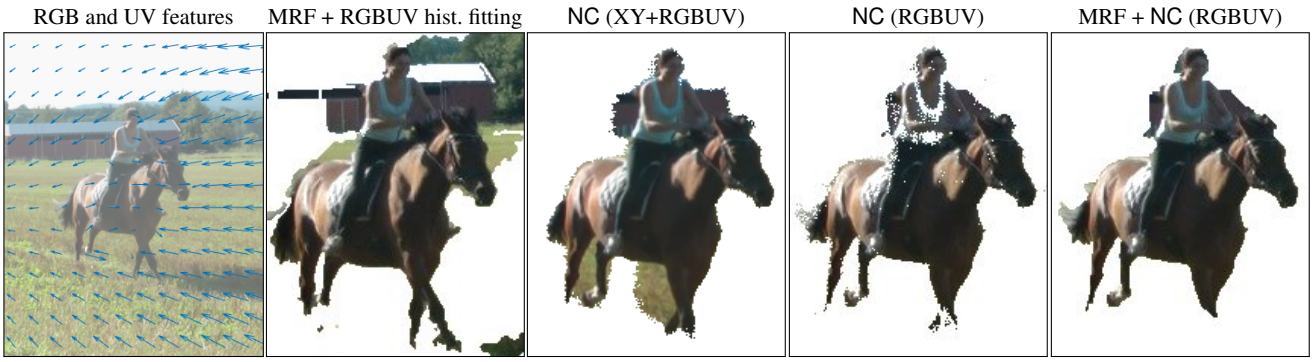
While independently developed as different methodologies, standard regularization and kernel clustering techniques are based on objective functions with many complementary properties. Our goal is to combine these functions into a joint objective or *energy* applicable to image segmentation or general clustering problems. On the one hand, we show that common regularization methods can use extra terms like *normalized cut* (NC) [94] to enforce balanced partitioning of arbitrary high-dimensional image features, *e.g.* a combination of color, texture, depth, or motion, where *model-fitting* [114, 90] fails, compare Fig.1(b)(e). On the other hand, standard clustering applications can benefit from an inclusion of basic pairwise or higher-order regularization constraints, *e.g.* edge alignment [22, 14], bin-consistency [54], label cost [33]. Regularization and kernel clustering could not be combined before due to optimization difficulties [60].

On a surface, even the formulations of kernel clustering and regularization-based segmentation may seem significantly different. While the general terms *clustering* and *segmentation* are largely synonyms, the latter is more common for images where data points are intensities, colors, or higher dimensional features $I_p \in \mathcal{R}^N$ sampled at regularly placed pixels $p \in \mathcal{R}^M$. For example, the image in Fig.1(a) combines colors and motion vectors into RGBUV features $I_p \in \mathcal{R}^5$ on grid points $p \in \mathcal{R}^2$. The pixels' locations are important. Many *regularization* methods for image segmentation treat I_p as a function $I : \mathcal{R}^M \to \mathcal{R}^N$ and process domain $\mathcal{R}^M$ (locations) and range $\mathcal{R}^N$ (features) in very different ways. For example, MRF [41] and variational techniques [76] use pixel locations for geometrically motivated segments' shape priors, while pixel features are used in segments' appearance likelihood models [22, 14, 15, 86], *e.g.* Fig.1(b).

In contrast, *clustering* typically assumes arbitrary data points I_p with non-informative indices p. General clustering techniques [38, 105, 3], *e.g. K-means* or spectral methods, apply to images [94, 1] by combining pixel locations with colors or other features into data points I_p in $\mathcal{R}^{M+N}$. For example, the result in Fig.1(c) uses $\mathcal{R}^7$ points combining locations XY and RGBUV values. Without the locations the result is spatially noisy (d). We focus on a well-known general group of *kernel clustering* methods [48, 94, 6, 35].

Meng Tang · Dmitrii Marin · Yuri Boykov
Computer Science, University of Waterloo, Canada
E-mail: m62tang@uwaterloo.ca · dmitrii.a.marin@gmail.com · yboykov@uwaterloo.ca

Ismail Ben Ayed
ETS Montreal, Canada
E-mail: ismail.benayed@etsmtl.ca



(a) image + optic flow (input data) (b) **grab-cut** (weak local minima) (c) **spectral clustering** (weak edge alignment) (d) **spectral clustering** (irregular boundary) (e) **kernel cut** (our approach)

Fig. 1: Segmentation of 5D image data (a). For higher-dimensional features, regularized model-fitting [114,33] becomes sensitive to local minima, *e.g. grabcut* [90] fitting RGBUV histograms (b). Spectral clustering like *normalized cut* (NC) [94] is scalable to high dimensional features, but it is known for splitting regions (c) or lack of regularity (d). Our *kernel cut* (e) combines *kernel clustering* over arbitrary features with standard regularization in the image domain, see energy (1).

variable	**our definition / representation**	**range**	alternative *relaxed* representation
S_p	segment (label) assignment for given pixel $p \in \Omega$	$\{1,\dots,K\}$	vector $[0,1]^K$ in Δ^K p-th row of *assignment matrix* S
S_c	$(S_p \,\vert\, p \in c)$ - labeling of pixels in *factor* $c \subseteq \Omega$	$\{1,\dots,K\}^{\vert c\vert}$	subset of rows of *assignment matrix* S
S	$(S_p \,\vert\, p \in \Omega)$ - segmentation or labeling of all points	$\{1,\dots,K\}^{\vert\Omega\vert}$	an *assignment matrix* $[0,1]^{\vert\Omega\vert\times K}$
S_p^k	indicator for "p is in segment k", *i.e.* $S_p^k \equiv [S_p = k]$	$\{0,1\}$	*assignment matrix* element in $[0,1]$
S^k	k-th **segment**, that is, subset $\{p \subset \Omega \,\vert\, S_p = k\}$ **or** the corresponding **indicator vector** $(S_p^k \,\vert\, p \in \Omega)$	$\mathcal{P}(\Omega)$ **or** $\{0,1\}^{\vert\Omega\vert}$	vector $[0,1]^{\vert\Omega\vert}$ k-th column of *assignment matrix* S
$S^{k\prime}$	transpose of vector S^k, *i.e.* $S^{k\prime} \equiv (S^k)^T$	$\{0,1\}^{\vert\Omega\vert}$	(transposed) vector $[0,1]^{\vert\Omega\vert}$

Table 1: Our notation for segmentation of points $p \in \Omega$ uses discrete labels and binary indicators (the first three columns). Without much ambiguity, segment S^k could mean both a subset of Ω or its indicator vector, *i.e.* S^k is either an element of *power set* $\mathcal{P}(\Omega)$ or a vector $\{0,1\}^{\vert\Omega\vert}$. While unnecessary for most of the technical results in this paper, in the context of *relaxation* methods it is easy to switch to an alternative representation (the last column) where segment S^k becomes a relaxed vector $[0,1]^{\vert\Omega\vert}$. This is consistent with a common (relaxed) *assignment matrix* representation of segmentation S where integer label S_p becomes a vector on probability simplex Δ^K specifying pixel's support/distribution over K labels.

Some differences in formulations of kernel clustering and regularization methods are not essential and easily resolve with proper notation working as a common platform for both (Sec.1.1, Tab.1). Our notation presents spectral clustering as a high-order term in a joint energy making similarities and differences more transparent. Once notation is established, we present our joint energy (1) combining kernel clustering and regularization terms, give some specific basic examples (Tab.2), and further motivate our approach. Later background section reviews standard (MRF) regularization and kernel clustering objectives in details and technical sections explain how to optimize the joint energy using new linear bounds for the high-order kernel clustering term.

1.1 Notation and conventions

We use notation applicable to either image segmentation or general data clustering. Let Ω be a set of pixels, voxels, or any other points p. For example, for 2D images Ω could be a subset of regularly spaced points in $\mathcal{R}^2$. Set Ω could also represent data points indices. We assume that every $p \in \Omega$ comes with an observed feature $I_p \in \mathcal{R}^N$. For example, I_p could be a greyscale intensity in $\mathcal{R}^1$, an RGB color in $\mathcal{R}^3$, or RGBUV features in $\mathcal{R}^5$ as in Fig.1(a), *et cetera*. If needed, feature I_p could also include the pixel's location.

Our notation describing segmentation of Ω is summarized in the first three columns of Table 1. We use the fol-

	regularization terms $\sum_{c\in\mathcal{F}} E_c(S_c)$	**kernel clustering terms** $E_A(S)$
some standard "graph cut" criteria	$\sum_{pq} w_{pq}\cdot[S_p \neq S_q] \quad \equiv \quad \frac{1}{2}\sum_k S^{k\prime}\mathcal{W}(\mathbf{1}-S^k)$	$\sum_k \frac{S^{k\prime}A(\mathbf{1}-S^k)}{\lvert S^k\rvert}$ or $\sum_k \frac{S^{k\prime}A(\mathbf{1}-S^k)}{d'S^k}$
	multi-way cut, a.k.a. Potts model [16], see (2)	*average* and *normalized cuts* [94], see (38) and (39)

Table 2: Examples of *"graph cut"* criteria appearing in the contexts of (MRF) *regularization* and *kernel clustering* that can be used in joint energy (1) simultaneously. The cut cost, *i.e.* the sum of edge weights w_{pq} or affinities A_{pq} between the segments, can be represented via matrices $\mathcal{W}=[w_{pq}]$ or $A=[A_{pq}]$, and segment indicators S^k, see Tab.1. The right column differs only by normalization over segment *cardinality* $|S^k|$ or *weighted cardinality* $d'S^k$ where $d := A\mathbf{1}$ are *node degrees*.

lowing standard notation: $\{\cdot|\cdot\}$ stands for sets or subsets, $(\cdot|\cdot)$ stands for ordered collections or vectors, and $[\cdot]$ is used in the context of intervals, matrices, or *Iverson brackets*[1]. Note that this paper uses upper case letters for both vectors and matrices, but some vectors are denoted by lower case letters.

Our notation in Table 1 is somewhat superfluous, but it gives flexibility needed for uniting diverse methodologies for segmentation and clustering covered in Section 2. We equivalently represent segmentation of Ω either as a *labeling* $S := (S_p|p\in\Omega)$ combining integer point labels $1\le S_p\le K$ or as a partitioning $\{S^k\}$ of set Ω into K non-overlapping subsets or segments $S^k := \{p\in\Omega|S_p=k\}$. As a minor abuse of notation, S^k will also be a set indicator vector $\{0,1\}^{|\Omega|}$. Exact interpretation of S^k is clear from the context. Since our bounds are also useful for relaxation methods, we may discuss *relaxed* segment support vectors S^k in $[0,1]^{|\Omega|}$.

1.2 Our approach summary

We combine standard kernel (pairwise) clustering criteria such as *Average Association* (AA) or *Normalized Cut* (NC) [94] and common regularization functionals such as MRF potentials [41,64]. The general form of our joint energy is

$$E(S) = E_A(S) + \gamma \sum_{c\in\mathcal{F}} E_c(S_c) \tag{1}$$

where the first term is some kernel clustering objective based on data *affinity matrix* or *kernel* $A := [A_{pq}]$ with elements $A_{pq} := A(I_p, I_q)$ defined by some similarity function $A(\cdot,\cdot)$. The second term in (1) is a general formulation of MRF *potentials* [16,54,33]. Table 2 previews basic examples of the terms in joint energy (1) using different "graph cut" criteria.

Constant γ in (1) is a relative weight of the (MRF) regularization term. Subset $c\subseteq\Omega$ represents a *factor* often consisting of nearby pixels. Factor labels $S_c := (S_p|p\in c)$ is a *restriction* of labeling S to c. Potentials $E_c(S_c)$ for a given set of factors $\mathcal{F}$ represent various unary, second, or higher order constraints, where factor size $|c|$ defines the order. The left column in Table 2 is an example of the second-order *Potts model* that can be equivalently written as a quadratic function. Factor features $\{I_p|p\in c\}$ often work as parameters for potentials E_c. For example, $w_{pq}=w(I_p,I_q)$ is a common way to set pairwise penalties in Tab.2 (left column). Section 2.1 reviews several standard MRF potentials.

Typical kernel clustering methods encourage balanced segments using ratio-based objectives E_A as in Tab.2 (right column). Due to normalization, such objectives can be seen as high-order potentials of order $|\Omega|$ that are difficult to optimize. Sections 2.2, 2.3 review popular kernel clustering criteria and standard approximate optimization methods.

In order to optimize the combination of kernel clustering term E_A with regularization constraints in energy (1), we propose two unary (linear) bounds for E_A. Such bounds are easy to integrate into many existing regularization solvers as outlined in Figure 2. In general, the second term in (1) could be any discrete or continuous objective with a good solver. We focus on discrete (MRF/CRF) regularization potentials in (1) only to be specific and because the code for the corresponding solvers is widely available. The following two subsections summarize the motivation and the main technical contributions of this paper.

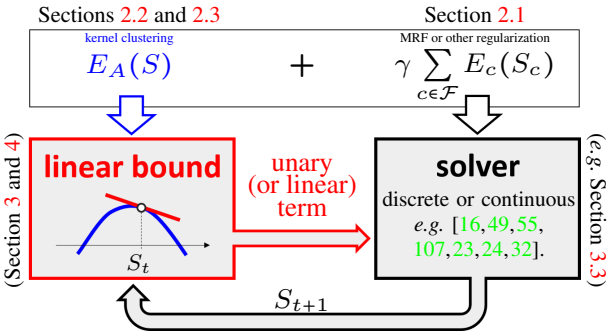


Fig. 2: Our *Kernel Cut* approach to minimizing energy (1). Standard (MRF) regularization solvers can easily integrate our linear *kernel* or *spectral* bounds for the clustering term (Sec.3,4) producing an iterative *bound optimization* for (1).

[1] Iverson brackets $[\cdot]$ enclosing a logical proposition, e.g. $[S_p=k]$, return 1 or 0 depending on true or false value of this proposition.

1.2.1 Motivation and Related work

Due to significant differences in their existing optimization methods, kernel clustering (*e.g.* NC) and regularization methods (*e.g.* MRF) are used separately in unsupervised or weakly-supervised applications of vision and learning. They have complementary strengths and weaknesses.

For example, NC optimizes a balanced kernel clustering criterion based on a kernel (affinities) between any high-dimensional features [94,68,4]. In contrast, regularization methods for unsupervised or weakly-supervised image segmentation typically combine constraints on segments shapes with *probabilistic K-means* [53] or explicit *model fitting* over segments features [25,114,90,33]. Fitting parametric models seems viable when data in each segment supports a simple model, *e.g.* Gaussian [25] or line/plane [33]. But, if segment's data is arbitrarily complex, the corresponding model should be sufficiently general in order to represent such complexities. Thus, image segmentation of generic objects requires fitting models like histograms or GMMs [114,90]. This results in over-fitting, see Fig.1(b). Indeed, we show that such over-fitting happens even for low dimensional color features [98], see Fig.3(b,e) and Fig.4(b). Our joint energy (1) allows to combine regularization of segments shapes with unsupervised kernel-based clustering of arbitrarily complex segments features. In general, kernel-based clustering methods are a prevalent choice in the learning community as model fitting (*e.g.* EM) becomes intractable in high dimensions. Section 5.2 shows potent segmentation results for basic examples of energy (1) with features like RGBXY (color + location), RGBD (color + depth), RGBUV (color + motion) where regularized model-fitting methods fail.

Standard applications of kernel clustering methods can also benefit from regularization constraints [111,40,26]. For example, NC approach to image segmentation is known for weak alignment to contrast boundaries [4], see Fig.1(cd). Adding the standard contrast-sensitive Potts (regularization) term [16,14] offers a principled solution, see Fig.1(e). We also show benefits from combining NC with higher-order constraints, such as sparsity or label costs [33]. For example, P^n-Potts regularization [54] can enforce tag-consistency in the context of image database clustering. Section 5.1 shows many proof-of-the-concept examples.

Kernel clustering vs. Potts model: Kernel clustering objectives E_A (Sections 2.2, 2.3) in our joint energy (1) can be juxtaposed with the most basic MRF regularizer, the Potts model (Section 2.1), *e.g.* compare two columns in Tab.2. Kernel clustering and Potts regularization minimize the sum of weighted edges between segments on a given graph. Both corresponding objectives are often called "pairwise" or "cuts". The main difference is that clustering criteria E_A normalize the sum of edge weights to encourage balanced partitioning, while the Potts model minimizes the sum "as is" to reduce segmentation boundary length. Due to normalization, E_A is a hard-to-optimize high-order term in energy (1).

Both kernel clustering and Potts model objectives are defined by the graph connectivity and/or the corresponding edge weights or affinities, *e.g.* w_{pq} or A_{pq} in Sections 2.1 and 2.2. It is usual to set the neighborhood and edge weights based on specific features, criteria, and application. For instance, Potts model over nearest-neighbor pixel grid defines first-order geometric shape priors [15], while an example of larger connectivity Potts is *dense CRF* [58,103]. All Potts models lack balancing. Their minimization results in a trivial solution unless there are some additional constraints, *e.g.* volumetric or data likelihood terms (Sec.2.1).

Kernel clustering criteria E_A normally use dense graphs. But unlike dense CRF or any other Potts model, the corresponding ratio-based objectives are designed for *unsupervised* balanced partitioning that does not require any known or estimated data likelihood models.

1.2.2 Main contributions

Our energy (1) combines standard concepts in unsupervised learning with regularization methodologies common in computer vision. Previous efforts [60] combining kernel clustering (*e.g.* NC) with the Potts model significantly altered the latter to make it fit the standard trace-based formulation of NC, see Sec.2.4. In contrast, we propose a general *majorize-minimize* optimization principle directly integrating our new unary/linear bounds for kernel clustering objectives E_A into existing powerful solvers for Potts or other regularization models. Examples of such solvers are combinatorial [16, 49], LP relaxation [55,107], mean field approximation [58], or TV-based [23,24,32] methods.

Our preliminary results appear in [97] and [100]. The main contributions of our work are summarized below:

- We propose a general multi-label segmentation or clustering energy (1) combining kernel clustering (*e.g.* NC) with second or higher-order regularization (*e.g.* MRF). The clustering term can enforce balanced partitioning of observed features and MRF or other terms can enforce regularization constraints. In particular, including balanced kernel clustering term is a robust well-motivated alternative to model-fitting terms [114,90], which fail on higher dimensional image features.
- We use a *concave relaxation* to derive two types of unary (linear) upper bounds for several classes of kernel clustering criteria E_A. The two types are *kernel bound* (exact) and *spectral bound*[2] (approximate). Interestingly, optimizing our linear bounds for $E_A(S)$ (no other terms) over discrete segmentation variables $S^k \in \{0,1\}^{|\Omega|}$ is

[2] Here *spectral bound* means *spectral auxiliary function* in the context of optimization, not to be confused with bounds on eigenvalues.

equivalent to iterative *kernel K-means* or K-means discretization heuristic in spectral relaxation methods.
- Our unary/linear bounds for E_A give solvable *auxiliary functions* for joint energy (1) as long as its second term has a solver that can integrate extra unary/linear potentials, see Fig.2. For example, the second term can be any regularization potentials solvable by discrete (*e.g.* message passing, relaxations, mean-field approximations) or continuous (*e.g.* convex, primal-dual) algorithms. In the context of standard pairwise and higher-order MRF potentials we demonstrate move-making algorithms generalizing α-expansion and $\alpha\beta$-swap moves to energy (1).
- As our experiments show, typical applications of kernel clustering (*e.g.* NC) can benefit from extra MRF constraints. MRF segmentation also benefits from kernel clustering terms encouraging balanced partitioning of object features. In particular, NC+MRF framework scales to object segmentation with higher-dimensional image features (*e.g.* RGBXY, RGBD, RGBM) where standard regularization methods with model-fitting [114,90,33] fail.

The rest of the paper is organized as follows. Background Section 2 starts from reviewing standard (MRF) regularization models for segmentation. Due to importance for our work, Section 2 also covers the basics of clustering from K-means to its powerful kernel-based generalizations, including normalized cut (NC). The main technical Sections 3 and 4 present our kernel and spectral bounds for standard kernel clustering objectives E_A. They also discuss combinatorial move making graph cut algorithms using such unary/linear bounds for optimizing joint energy (1) combining E_A with MRF regularization constraints. Section 5 presents many experiments where either standard kernel clustering methods benefit from additional MRF constraints or common applications of MRF benefit from an additional kernel clustering term for various high-dimensional image features.

2 Background on Regularization and Clustering

2.1 Overview of MRF regularization

Probably the most basic MRF regularization potential corresponds to the pairwise (second-order) Potts model [16] used for segmentation boundary smoothness and **edge alignment**

$$\sum_{c\in\mathcal{F}} E_c(S_c) \;=\; \sum_{pq\in\mathcal{N}} w_{pq}\cdot[S_p\neq S_q] \;\approx\; \|\partial S\| \tag{2}$$

where a set of pairwise factors $\mathcal{F}=\mathcal{N}$ includes *edges* $c=\{pq\}$ between pairs of neighboring nodes and $[\cdot]$ are *Iverson brackets*. Weight w_{pq} is a discontinuity penalty between p and q. It could be a constant or may be set by a decreasing function of intensity difference I_p-I_q attracting the segmentation boundary to image contrast edges [14]. This is similar to the image-based boundary length in geodesic contours [22,15].

A useful **bin consistency** constraint enforced by the P^n-Potts model [54] is defined over an arbitrary collection of high-order factors $\mathcal{F}$. Factors $c\in\mathcal{F}$ correspond to predefined subsets of nodes such as *superpixels* [54] or *bins* of pixels with the same color/feature [85,98]. The model penalizes inconsistency in segmentation of each factor

$$\sum_{c\in\mathcal{F}} E_c(S_c) \;=\; \sum_{c\in\mathcal{F}} \min\{T, |c|-|S_c|^*\} \tag{3}$$

where T is some threshold and $|S_c|^* := \max_k |S^k\cap c|$ is the cardinality of the largest segment inside c. Potential (3) has its lowest value (zero) when all nodes in each factor are within the same segment.

Standard **label cost** [33] is a sparsity potential defined for a single high-order factor $c=\Omega$. In its simplest form it penalizes the number of distinct segments (labels) in S

$$E_\Omega(S) \;=\; \sum_k h_k\cdot[|S^k|>0] \tag{4}$$

where h_k could be a constant or a cost for each specific label.

Potentials (2), (3), (4) are only a few examples of regularization terms widely used in combination with powerful discrete solvers like graph cut [16], belief propagation [109], TRWS [56], LP relaxation [107,52], or continuous methods [23,24,32].

Image segmentation methods often combine regularization with a **likelihood term** integrating segments/objects color models. For example, [14,13] used graph cuts to combine second-order edge alignment (2) with a unary (first-order) appearance term

$$-\sum_k \sum_{p\in S^k} \log P^k(I_p) \tag{5}$$

where $\{P^k\}$ are given probability distributions. Unary terms like (5) are easy to integrate into any of the solvers above.

If unknown, parameters of the models $\{P^k\}$ in a regularization energy including (5) are often estimated by iteratively minimizing the energy with respect to S and model parameters [114,25,5,90,33]. In presence of variable model parameters, (5) can be seen as a *maximum likelihood* (ML) model-fitting term or a *probabilistic K-means* clustering objective [53]. The next section reviews K-means and other standard clustering methods.

2.2 Overview of K-means and clustering

Many clustering methods are based on K-means (KM). The most basic iterative KM algorithm [39] can be described as

the *block-coordinate descent* for the following *mixed* objective

$$F(S,m) \;:=\; \sum_k \sum_{p\in S^k} \|I_p - m_k\|^2 \tag{6}$$

combining discrete variables $S = \{S^k\}_{k=1}^K$ with continuous variables $m = \{m_k\}_{k=1}^K$ representing cluster "centers". Norm $\|.\|$ denotes the Euclidean metric. For any given S the optimal centers $\arg\min_m F(S,m)$ are the means

$$\mu_{S^k} \;:=\; \frac{\sum_{q\in S^k} I_q}{|S^k|} \tag{7}$$

where $|S^k|$ is the segment's cardinality. Assuming current segments S_t^k the update operation giving $\arg\min_S F(S,\mu_{S_t})$

$$\left(\begin{array}{c}\textbf{basic KM}\\ \textbf{procedure}\end{array}\right) \qquad S_p \;\leftarrow\; \arg\min_k \|I_p - \mu_{S_t^k}\| \tag{8}$$

defines the next solution S_{t+1} as per standard K-means algorithm. This greedy descent technique converges only to a local minimum of KM objective (6), which is known to be NP hard to optimize. There are also other approximation methods. Below we review the properties of KM objective (6) independently of optimization.

The optimal centers m_k in (7) allow to represent (6) via an equivalent objective of a single argument S

$$\sum_k \sum_{p\in S^k} \|I_p - \mu_{S^k}\|^2 \;\equiv\; \sum_k |S^k|\cdot var(S^k). \tag{9}$$

The sum of squared distances between data points $\{I_p | p \in S^k\}$ and mean μ_{S^k} normalized by $|S^k|$ gives the *sample variance* denoted by $var(S^k)$. Formulation (9) presents the basic KM objective as a standard *variance criterion* for clustering. That is, K-means attempts to find K compact clusters with small variance.

K-means can also be presented as a "pairwise" or kernel clustering criteria with Euclidean affinities. The *sample variance* can be expressed as the sum of distances between all pairs of the points. For example, plugging (7) into (9) reduces this KM objective to

$$\sum_k \frac{\sum_{pq\in S^k} \|I_p - I_q\|^2}{2\,|S^k|}. \tag{10}$$

Taking the square in the denominator transforms (10) to another equivalent KM energy with Euclidean dot-product affinities

$$\stackrel{c}{=} \; -\sum_k \frac{\sum_{pq\in S^k} \langle I_p, I_q\rangle}{|S^k|}. \tag{11}$$

Note that we use $\stackrel{c}{=}$ and $\stackrel{c}{\approx}$ for "up to additive constant" relations.

Alternatively, K-means clustering can be seen as Gaussian model fitting. Formula (5) for normal distributions with variable means m_k and some fixed variance

$$-\sum_k \sum_{p\in S^k} \log N(I_p|m_k) \tag{12}$$

equals objective (6) up to a constant.

Various extensions of objectives (6), (9), (10), (11), or (12) lead to many powerful clustering methods such as kernel K-means, average association, and Normalized Cut, see Table 3.

2.2.1 *Probabilistic K-means (*pKM*) and model fitting*

One way to generalize K-means is to replace squared Euclidean distance in (6) by other *distortion* measures $\|\|_d$ leading to a general *distortion energy* commonly used for clustering

$$\sum_k \sum_{p\in S^k} \|I_p - m_k\|_d. \tag{13}$$

The optimal value of parameter m_k may no longer correspond to a *mean*. For example, the optimal m_k for non-squared L_2 metric is a *geometric median*. For exponential distortions the optimal m_k may correspond to *modes* [92, 21], see [99, Appendix B].

A seemingly different way to generalize K-means is to treat both means and covariance matrices for the normal distributions in (12) as variables. This corresponds to the standard *elliptic K-means* [95,91,33]. In this case variable model parameters $\theta_k = \{m_k, \Sigma_k\}$ and data points I_p are not in the same space. Yet, it is still possible to present elliptic K-means as distortion clustering (13) with "distortion" between I_p and θ_k defined by an operator $\|\cdot - \cdot\|_d$ corresponding to a likelihood function

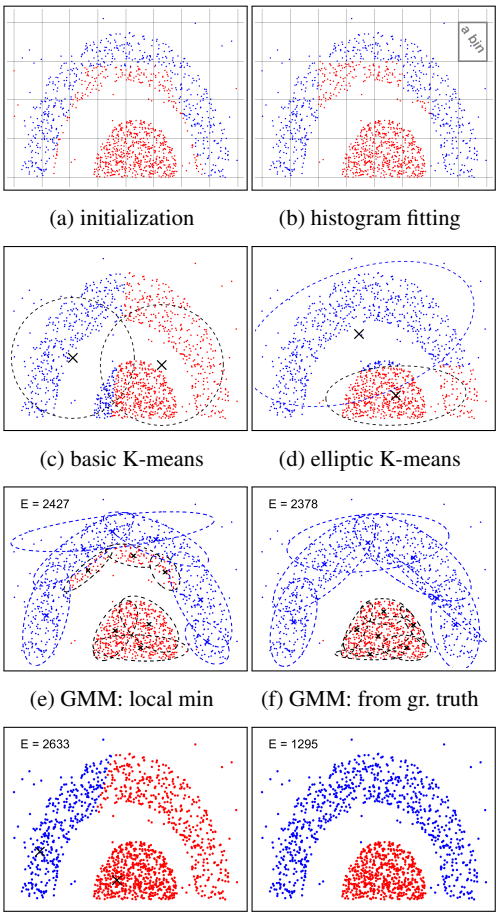
$$\|I_p - \theta_k\|_d := -\log N(I_p|\theta_k).$$

Similar distortion measures can be defined for arbitrary probability distributions with any variable parameters θ_k. Then, distortion clustering (13) generalizes to ML model fitting objective

$$\sum_k \sum_{p\in S^k} \|I_p - \theta_k\|_d \;\equiv\; -\sum_k \sum_{p\in S^k} \log P(I_p|\theta_k) \tag{14}$$

which is (5) with explicit model parameters θ_k. This formulation suggests *probabilistic K-means*[3] (pKM) as a good idiomatic name for ML model fitting or distortion clustering (13), even though the corresponding parameters θ_k are not "means", in general.

[3] The name *probabilistic K-means* in the general clustering context was coined by [53]. They formulated (14) after representing distortion energy (13) as ML fitting of Gibbs models $\frac{1}{Z_d} e^{-\|x-m\|_d}$ for arbitrary integrable metrics.



(a) initialization (b) histogram fitting

(c) basic K-means (d) elliptic K-means

(e) GMM: local min (f) GMM: from gr. truth

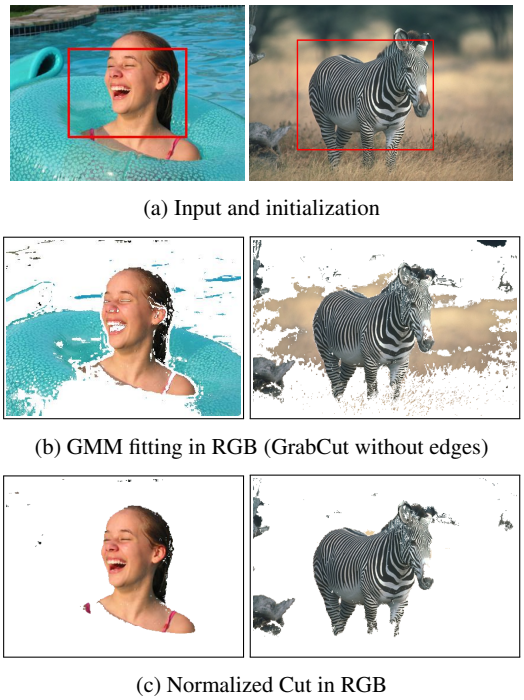
(g) K-modes ~ *mean-shift* (h) kernel K-means

Fig. 3: *Model fitting (*pKM*)* (14) *vs kernel K-means (*kKM*)* (22). Histogram fitting converges in one step assigning initially dominant bin label (a) to all points in the bin (b): energy (14,15) is minimal at any volume-balanced solution with one label inside each bin [53]. Basic and elliptic K-means (one mode GMM) under-fit the data (c,d). Six mode GMMs over-fit (e) as in (b). GMMs have local minima issues; ground-truth initialization (f) yields lower energy (14,15). Kernel K-means (21,22) with Gaussian kernel k in (h) outperforms pKM with distortion $\|\|_k$ in (g) related to K-modes or mean-shift (*weak* kKM, see Sec.2.2.3).

Probabilistic K-means (14) is used in image segmentation with models such as elliptic Gaussians [95,91,33], gamma/exponential [5], or other generative models [73]. Zhu-Yuille [114] and GrabCut [90] use pKM with highly descriptive probability models such as GMM or histograms. Information theoretic analysis in [53] shows that in this case pKM objective (14) reduces to the standard *entropy criterion* for clustering

$$\sum_k |S^k| \cdot H(S^k) \tag{15}$$

where $H(S^k)$ is the distribution entropy for $\{I_p | p \in S^k\}$.



(a) Input and initialization

(b) GMM fitting in RGB (GrabCut without edges)

(c) Normalized Cut in RGB

Fig. 4: Without edge alignment (2) GMM-fitting [90] shows stronger data over-fit compared to kernel clustering [94].

Intuitively, minimization of the entropy criterion (15) favors clusters with tight or "peaked" distributions. This criterion is widely used in categorical clustering [65] and decision trees [18,66] where the entropy evaluates histograms over "naturally" discrete features. However, the entropy criterion with either discrete histograms or continuous GMM densities has limitations in the context of *continuous* feature spaces, see [99, Appendix C]. Iterative fitting of descriptive models is highly sensitive to local minima [98,96] and easily over-fits even low dimentional features in $\mathcal{R}^2$ (Fig.3b,e) or in $\mathcal{R}^3$ (RGB colors, Fig.4b). This may explain why this approach to clustering is not too common in the learning community. As proposed in (1), instead of entropy criterion we will combine MRF regularization with general kernel clustering objectives E_A widely used for balanced partitioning of arbitrary high-dimensional features [94].

2.2.2 Kernel K-means and related "pairwise" clustering

This section reviews pairwise extensions of K-means (11) such as *kernel K-means* (kKM) and related kernel clustering criteria. In machine learning, kKM is a well established data clustering technique [101,74,42,35,27,50] that can identify non-linearly separable structures. In contrast to pKM based on complex models, kKM corresponds to complex (nonlinear) mappings

$$\phi : \mathcal{R}^N \to \mathcal{H}$$

A. **basic K-means** (KM) (*e.g.* [39])	
$\sum_k \sum_{p\in S^k} \|I_p - \mu_{S^k}\|^2$	**Variance criterion**
$= \sum_k \frac{\sum_{pq\in S^k} \|I_p - I_q\|^2}{2\|S^k\|}$	$\sum_k \|S^k\| \cdot var(S^k)$
$\stackrel{c}{=} -\sum_k \frac{\sum_{pq\in S^k} \langle I_p, I_q\rangle}{\|S^k\|}$	
$\stackrel{c}{=} -\sum_k \sum_{p\in S^k} \ln \mathcal{N}(I_p\|\mu_{S^k})$	

more complex probability models — more complex data representation

B. **probabilistic K-means** (pKM)	C. **kernel K-means** (kKM)
equivalent energy formulations:	*equivalent energy formulations:*
$\sum_k \sum_{p\in S^k} \|I_p - \theta_k\|_d = -\sum_k \sum_{p\in S^k} \ln \mathcal{P}(I_p\|\theta_k)$	$\sum_k \sum_{p\in S^k} \|\phi(I_p) - \mu_{S^k}\|^2 = \sum_k \frac{\sum_{pq\in S^k} \|I_p - I_q\|_k^2}{2\|S^k\|} \stackrel{c}{=} -\sum_k \frac{\sum_{pq\in S^k} k(I_p, I_q)}{\|S^k\|}$
related examples:	*related examples:*
elliptic K-means [95,91]	Average Association or Distortion [89]
geometric model fitting [33]	Average Cut [94]
K-modes [92] or mean-shift [29] (weak kKM)	Normalized Cut [94,35] (weighted kKM)
Entropy criterion $\sum_k \|S^k\| \cdot H(S^k)$ [114,90] for highly descriptive models (GMMs, histograms)	Gini criterion $\sum_k \|S^k\| \cdot G(S^k)$ [18,97] for small-width normalized kernels [69]

Table 3: *K-means and related clustering criteria:* Basic K-means (A) minimizes clusters variances. It works as Gaussian model fitting. Fitting more complex models like elliptic Gaussians [95,91,33], exponential distributions [5], GMM or histograms [114,90] corresponds to *probabilistic K-means* [53] in (B). Kernel clustering via *kernel K-means* (C) using more complex data representation.

embedding data $\{I_p | p \in \Omega\} \subset \mathcal{R}^N$ as points $\phi_p \equiv \phi(I_p)$ in a higher-dimensional Hilbert space $\mathcal{H}$. The original non-linear problem can often be solved by simple linear separators of the embedded points $\{\phi_p | p \in \Omega\} \subset \mathcal{H}$. Kernel K-means corresponds to the basic K-means (6) in the embedding space

$$F(S,m) = \sum_k \sum_{p\in S^k} \|\phi_p - m_k\|^2. \tag{16}$$

Optimal segment centers m_k corresponding to the means

$$\mu_{S^k} = \frac{\sum_{q\in S^k} \phi_q}{|S^k|}. \tag{17}$$

reduce (16) to kKM energy of the single variable S similar to (9)

$$F(S) = \sum_k \sum_{p\in S^k} \|\phi_p - \mu_{S^k}\|^2. \tag{18}$$

Similarly to (10) and (11) one can write kernel clustering criteria equivalent to (18) based on Euclidean distances $\|\phi(I_p) - \phi(I_q)\|$ or inner products $\langle \phi(I_p), \phi(I_q)\rangle$, which are commonly represented via *kernel function* $k(x,y)$

$$k(x,y) := \langle \phi(x), \phi(y)\rangle. \tag{19}$$

The (non-linear) kernel function $k(x,y)$ corresponds to the inner product in $\mathcal{H}$. It also defines *Hilbertian metric*[4]

$$\begin{aligned} \|x-y\|_k^2 &:= \|\phi(x) - \phi(y)\|^2 \\ &\equiv k(x,x) + k(y,y) - 2k(x,y) \end{aligned} \tag{20}$$

isometric to the Euclidean metric in the embedding space. Then, pairwise formulations (10) and (11) for K-means in the embedding space (18) can be written for the original data points using isometric kernel distance $\|\|_k^2$ in (20)

$$F(S) \equiv \sum_k \frac{\sum_{pq\in S^k} \|I_p - I_q\|_k^2}{2|S^k|} \tag{21}$$

or using kernel function k in (19)

$$F(S) \stackrel{c}{=} -\sum_k \frac{\sum_{pq\in S^k} k(I_p, I_q)}{|S^k|}. \tag{22}$$

The definition of kernel k in (19) requires embedding ϕ. Since pairwise objectives (21) and (22) are defined for any kernel function in the original data space, it is possible to formulate kKM by directly specifying an affinity function or kernel $k(x,y)$ rather than embedding $\phi(x)$. This is typical for kKM explaining why the method is called *kernel K-means* rather than *embedding K-means*[5].

[4] These can be isometrically embedded into a Hilbert space [46].

[5] This could be a name for some clustering techniques constructing explicit embeddings [8,112] instead of working with pairwise affinities/kernels.

Given embedding ϕ, kernel function k defined by (19) is *positive semi-definite* (p.s.d), that is $k(x,y) \geq 0$ for any x, y. Moreover, *Mercer's theorem* [75] states that p.s.d. condition for any given kernel $k(x,y)$ is sufficient to guarantee that $k(x,y)$ is an inner product in some Hilbert space. That is, it guarantees existence of some embedding $\phi(x)$ such that (19) is satisfied. Therefore, kKM objectives (18), (21), (22) are equivalently defined either by embeddings ϕ or p.s.d. kernels k. Thus, kernels are commonly assumed p.s.d. However, as discussed later, kernel clustering objective (22) is also used with non p.s.d. affinities.

To optimize kKM objectives (18), (21), (22) one can use the basic KM procedure (8) iteratively minimizing mixed objective (16) explicitly using embedding ϕ

$$\left(\begin{array}{c}\textbf{explicit kKM}\\ \textbf{procedure}\end{array}\right) \qquad S_p \;\leftarrow\; \arg\min_k \|\phi_p - \mu_{S_t^k}\| \tag{23}$$

where $\mu_{S_t^k}$ is the mean (17) for current segment S_t^k. Equivalently, this procedure can use kernel k instead of ϕ. Indeed, as in Section 8.2.2 of [93], the square of the objective in (23) is

$$\cancel{\|\phi_p\|^2} - 2\phi_p{}'\mu_{S_t^k} + \|\mu_{S_t^k}\|^2 \;=\; -2\frac{\phi_p{}'\phi S_t^k}{|S_t^k|} + \frac{S_t^{k\prime}\phi'\phi S_t^k}{|S_t^k|^2}$$

where we use segment S^k as an indicator vector, embedding ϕ as an *embedding matrix* $\phi := [\phi_p]$ where points $\phi_p \equiv \phi(I_p)$ are columns, and $'$ denotes the transpose. Since the crossed term is a constant at p, the right hand side gives an equivalent objective for computing S_p in (23). Using *kernel matrix* $\mathcal{K} := \phi'\phi$ and indicator vector $\mathbf{1}_p$ for element p we get

$$\left(\begin{array}{c}\textbf{implicit}\\ \textbf{kKM}\\ \textbf{procedure}\end{array}\right) \; S_p \leftarrow \arg\min_k \frac{S_t^{k\prime}\mathcal{K}S_t^k}{|S_t^k|^2} - 2\frac{\mathbf{1}_p'\mathcal{K}S_t^k}{|S_t^k|} \tag{24}$$

where the kernel matrix is directly determined by kernel k

$$\mathcal{K}_{pq} \equiv \phi_p'\phi_q = \langle\phi_p, \phi_q\rangle = k(I_p, I_q).$$

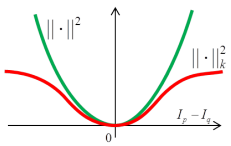
Approach (24) has quadratic complexity $\mathcal{O}(|\Omega|^2)$ iterations. But, it avoids explicit high-dimensional embeddings ϕ_p in (23) replacing them by kernel k in all computations, *a.k.a.* the *kernel trick*.

Note that the implicit kKM procedure (24) is guaranteed to decrease pairwise kKM objectives (21) or (22) only for p.s.d. kernels. Indeed, equation (24) is derived from the standard greedy K-means procedure in the embedding space (23) assuming kernel (19). The backward reduction of (24) to (23) can be done only for p.s.d. kernels k when Mercer's theorem guarantees existence of some embedding ϕ such that $k(I_p, I_q) = \langle\phi(I_p), \phi(I_q)\rangle$.

Pairwise energy (21) helps to explain the positive result for kKM with common Gaussian kernel $k = \exp\frac{-(I_p-I_q)^2}{2\sigma^2}$ in Fig.3(h). Gaussian kernel distance (red plot below)

$$\|I_p - I_q\|_k^2 \propto 1 - k(I_p, I_q) = 1 - \exp\frac{-(I_p-I_q)^2}{2\sigma^2} \tag{25}$$

is a "robust" version of Euclidean metric (green) in basic K-means (10). Thus, Gaussian kKM finds clusters with small local variances, Fig.3(h). In contrast, basic K-means (c) tries to find good clusters with small global variances, which is impossible for non-compact clusters.



Average association (AA) or distortion (AD): Equivalent pairwise objectives (21) and (22) suggest natural extensions of kKM. For example, one can replace Hilbertian metric $\|\|_k^2$ in (21) by an arbitrary zero-diagonal distortion matrix $D = [D_{pq}]$ generating *average distortion* (AD) energy

$$E_{ad}(S) := \sum_k \frac{\sum_{pq\in S^k} D_{pq}}{2|S^k|} \tag{26}$$

reducing to kKM energy (21) for $D_{pq} = \|I_p - I_q\|_k^2$. Similarly, p.s.d. kernel k in (22) can be replaced by an arbitrary pairwise similarity or affinity matrix $A = [A_{pq}]$ defining standard *average association* (AA) energy

$$E_{aa}(S) := -\sum_k \frac{\sum_{pq\in S^k} A_{pq}}{|S^k|} \tag{27}$$

reducing to kKM objective (22) for $A_{pq} = k(I_p, I_q)$. We will also use association between any two segments S^i and S^j

$$assoc(S^i, S^j) \;:=\; \sum_{p\in S^i, q\in S^j} A_{pq} \;\equiv\; S^{i\prime}AS^j \tag{28}$$

allowing to rewrite AA energy (27) as

$$E_{aa}(S) \;\equiv\; -\sum_k \frac{assoc(S^k, S^k)}{|S^k|} \;\equiv\; -\sum_k \frac{S^{k\prime}AS^k}{\mathbf{1}'S^k} \tag{29}$$

The matrix expressions in (28) and (29) represent segments S^k as indicator vectors such that $S_p^k = 1$ iff $S_p = k$ and symbol $'$ means a transpose. Matrix notation as in (29) will be used for all kernel clustering objectives in this paper.

kKM algorithm (24) is not guaranteed to decrease (27) for improper (non p.s.d.) kernel matrix $\mathcal{K} = A$, but general AA and AD energies could be useful despite optimization issues. However, [89] showed that dropping metric and proper kernel assumptions are not essential; there exist p.s.d. kernels with kKM energies equivalent (up to constant) to AD (26) and AA (27) for arbitrary associations A and zero-diagonal distortions D, see Fig. 5.

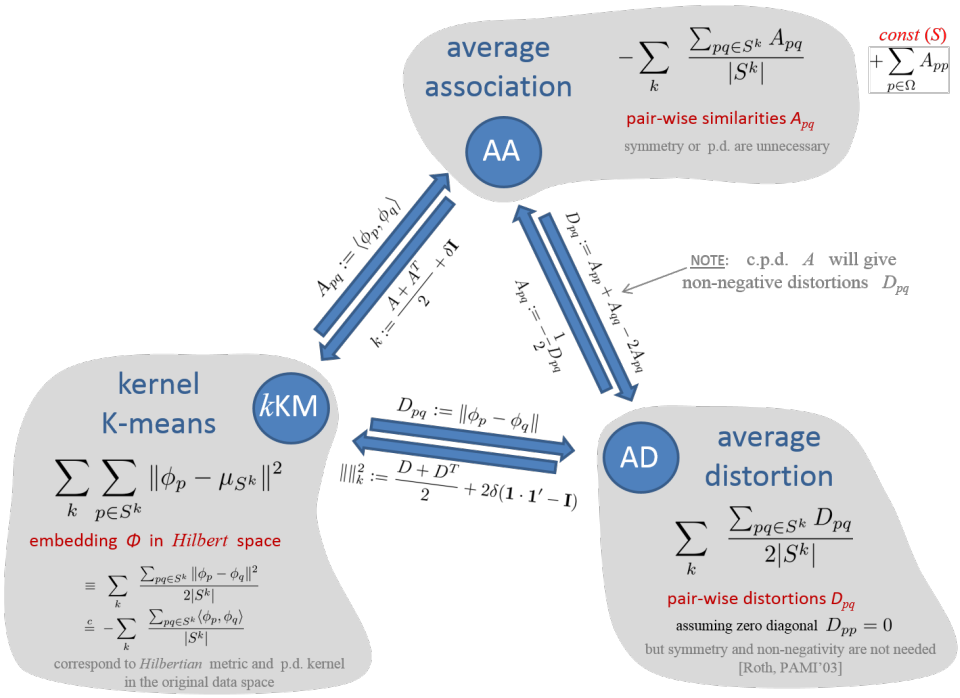


Fig. 5: Equivalence of kernel clustering methods: *kernel K-means* (kKM), *average distortion* (AD), *average association* (AA) based on Roth et al. [89], see (30), (31). Equivalence of these methods in the general *weighted* case is discussed in [99, Appendix A, Fig.33].

For example, for any affinity A in (27) the *diagonal shift* trick of Roth et al. [89] generates the "kernel matrix"

$$\mathcal{K} = \frac{A + A'}{2} + \delta \cdot \mathbf{I}. \tag{30}$$

For sufficiently large scalar δ matrix $\mathcal{K}$ is positive definite yielding a proper discrete kernel $k(I_p, I_q) \equiv \mathcal{K}_{pq}$

$$k(I_p, I_q) : \chi \times \chi \to \mathcal{R}$$

for finite set $\chi = \{I_p | p \in \Omega\}$. It is easy to check that kKM energy (22) with kernel $k \equiv \mathcal{K}$ in (30) is equivalent to AA energy (27) with affinity A, up to a constant. Indeed, for any indicator $X \in \{0,1\}^{|\Omega|}$ we have $X'X = \mathbf{1}'X$ implying

$$\frac{X'\mathcal{K}X}{\mathbf{1}'X} = \frac{X'AX}{2(\mathbf{1}'X)} + \frac{X'A'X}{2(\mathbf{1}'X)} + \delta\frac{X'X}{\mathbf{1}'X} = \frac{X'AX}{\mathbf{1}'X} + \delta.$$

Also, Section 4.1 uses eigen decomposition of $\mathcal{K}$ to construct an explicit finite-dimensional Euclidean embedding[6] $\phi_p \in \mathcal{R}^{|\Omega|}$ satisfying isometry (20) for any p.d. discrete kernel $k \equiv \mathcal{K}$. Minimizing kKM energy (18) over such embedding isometric to $\mathcal{K}$ in (30) is equivalent to optimizing (22) and, therefore, (27).

[6] Mercer's theorem is a similar eigen decomposition for continuous p.d. kernels $k(x, y)$ giving infinite-dimensional Hilbert embedding $\phi(x)$. Discrete kernel embedding $\phi_p \equiv \phi(I_p)$ in Sec. 4.1 (60) has finite dimension $|\Omega|$, which is still much higher than the dimension of points I_p, *e.g.* $\mathcal{R}^3$ for colors. Sec. 4.1 also shows lower dimensional embeddings $\tilde{\phi}_p$ approximating isometry (20).

Since average distortion energy (26) for arbitrary D is equivalent to average association for $A = -\frac{D}{2}$, it can also be converted to kKM with a proper kernel [89]. Using the corresponding kernel matrix (30) and (20) it is easy to derive Hilbertian distortion (metric) equivalent to original distortions D

$$\|I_p - I_q\|_k^2 := \frac{D + D'}{2} + 2\delta(\mathbf{1} \cdot \mathbf{1}' - \mathbf{I}). \tag{31}$$

For simplicity and without loss of generality, the rest of the paper assumes symmetric affinities $A = A'$ since non-symmetric ones can be equivalently replaced by $\frac{A+A'}{2}$. However, we do not assume positive definiteness and discuss diagonal shifts, if needed.

Weighted kKM and weighted AA: Weighted K-means [39] is a common extension of KM techniques incorporating some given point weights $w = \{w_p | p \in \Omega\}$. In the context of embedded points ϕ_p it corresponds to *weighted* kKM iteratively minimizing the weighted version of the mixed objective in (16)

$$F^w(S, m) := \sum_k \sum_{p \in S^k} w_p \|\phi_p - m_k\|^2. \tag{32}$$

Optimal segment centers m_k are now weighted means

$$\mu^w_{S^k} = \frac{\sum_{q \in S^k} w_q \phi_q}{\sum_{q \in S^k} w_q} \equiv \frac{\phi W S^k}{w' S^k} \tag{33}$$

where the matrix formulation has weights represented by column vector $w \in \mathcal{R}^{|\Omega|}$ and diagonal matrix $W := diag(w)$. Assuming a finite dimensional data embedding $\phi_p \in \mathcal{R}^m$ this formulation uses *embedding matrix* $\phi := [\phi_p]$ with column vectors ϕ_p. This notation implies two simple identities used in (33)

$$\sum_{q \in S^k} w_q \equiv w' S^k \quad \text{and} \quad \sum_{q \in S^k} w_q \phi_p \equiv \phi W S^k. \tag{34}$$

Inserting weighted means (33) into mixed objective (32) produces a pairwise energy formulation for weighted kKM similar to (22)

$$\begin{aligned} F^w(S) &:= \sum_k \sum_{p \in S^k} w_p \|\phi_p - \mu^w_{S^k}\|^2 && (35) \\ &\stackrel{c}{=} -\sum_k \frac{\sum_{pq \in S^k} w_p w_q \mathcal{K}_{pq}}{\sum_{p \in S^k} w_p} && (36) \\ &\equiv -\sum_k \frac{S^{k\prime} W \mathcal{K} W S^k}{w' S^k} \end{aligned}$$

where p.s.d kernel matrix $\mathcal{K} = \phi'\phi$ corresponds to the dot products in the embedding space, *i.e.* $\mathcal{K}_{pq} = \phi'_p \phi_q$.

Replacing the p.s.d. kernel with an arbitrary affinity matrix A defines a weighted AA objective generalizing (27) and (29)

$$E^w_{aa}(S) := -\sum_k \frac{S^{k\prime} W A W S^k}{w' S^k}. \tag{37}$$

Weighted AD can also be defined. Equivalence of kKM, AA, and AD in the general *weighted* case is discussed in [99, Appendix A].

Other kernel clustering criteria: Besides AA there are many other standard kernel clustering criteria defined by affinity matrices $A = [A_{pq}]$. For example, Average Cut (AC)

$$\begin{aligned} E_{ac}(S) &:= \sum_k \frac{assoc(S^k, \bar{S}^k)}{|S^k|} \equiv \sum_k \frac{S^{k\prime} A(\mathbf{1} - S^k)}{\mathbf{1}' S^k} \\ &= \sum_k \frac{S^{k\prime}(D - A) S^k}{\mathbf{1}' S^k} \end{aligned} \tag{38}$$

where $D := diag(d)$ is a *degree matrix* defined by node degrees vector $d := A\mathbf{1}$. The formulation on the last line (38) comes from the following identity valid for Boolean $X \in \{0,1\}^{|\Omega|}$

$$X'DX = X'd.$$

Normalized Cut (NC) [94] in (39) is another well-known kernel clustering criterion. Due to popularity of NC we discuss it and its relation to other kernel clustering criteria in a dedicated Section 2.3.

Kernel selection issues: One of the practically important problems in kernel clustering is selection of the kernel or its bandwidth. It is known [69] that for a common class of kernels (*e.g.* popular Gaussian kernel), NC (41) and AC (38), AA (29) and kKM (22) have various *density biases*. In particular, AA and kKM with a small bandwidth isolate density modes [94] while AC and NC separate isolated data points [69]. Zelnik-Manor and Perona [113] discuss other related biases in NC. In practice the bandwidth choice is a trade-off between the prominence of the density biases for small bandwidths and lack of non-linear separation for large bandwidths. Instead of fitting a single bandwidth value, one can employ adaptive weights [69] or adaptive kernel bandwidths [113,69], *e.g.* on K-nearest neighbor (KNN) graphs, to correct the density biases while keeping non-linearity of the decision boundary. Interestingly, in this case objectives NC, AC, kKM and AA become equivalent [69].

2.2.3 *Pairwise vs. pointwise distortions*

Equivalence of kKM to pairwise distortion criterion in (26) helps to juxtapose *kernel K-means* with *probabilistic K-means* (Sec.2.2.1) from one more point of view. Both methods generalize the basic K-means (6), (10) by replacing the Euclidean metric with a more general distortion measure $\|\|_d$. While pKM uses "pointwise" formulation (13) where $\|\|_d$ measures distortion between a point and a model, kKM uses "pairwise" formulation (21) where $\|\|_d = \|\|^2_k$ measures distortion between pairs of points.

These two different formulations are equivalent for Euclidean distortion (*i.e.* basic K-means), but the pairwise approach is strictly stronger than the pointwise version using the same Hilbertian distortion $\|\|_d = \|\|^2_k$ in non-Euclidean cases [99, Appendix B]. The corresponding pointwise approach is often called *weak kernel K-means*. Interestingly, weak kKM with standard Gaussian kernel can be seen as *K-modes* [92], see Fig. 3(g), which is closely related to popular *mean-shift* clustering [29], see [99, Appendix B]. An extended version of Table 3 including weighted KM and weak kKM is given in [99, Fig.34].

2.3 NC objective and its relation to AA, AC, and kKM

Section 2.2.2 has already discussed kKM and many related *kernel clustering* criteria based on specified affinities $A = [A_{pq}]$. This section is focused on a related kernel clustering method, Normalized Cut (NC) [94]. Shi and Malik [94] also popularized kernel clustering optimization via *spectral relaxation*, which is different from iterative K-means algorithms (23) (24). Note that there are many other popular optimization methods for different clustering energies using pairwise affinities [30,51,106,57,47], which are outside the scope of this work.

The Normalized Cut (NC) objective [94] is defined as

$$E_{nc}(S) := -\sum_k \frac{assoc(S^k, S^k)}{assoc(\Omega, S^k)} \equiv -\sum_k \frac{S^{k\prime} A S^k}{\mathbf{1}' A S^k} \stackrel{c}{=} \sum_k \frac{S^{k\prime} A(\mathbf{1} - S^k)}{\mathbf{1}' A S^k} \quad (39)$$

where *association* (28) is defined by a given *affinity matrix* A. The matrix formulations above shows that the difference between NC and AA (29) or AC (38) is in the normalization. In fact, normalization by $\mathbf{1}'AS^k$ is chosen specifically to make *normalized average association* equivalent to *normalized cut*, the last two expressions in (39). In contrast, AA and AC are distinct objectives normalized by $\mathbf{1}'S^k \equiv |S^k|$, which is k-th segment's size or cardinality. NC (39) normalizes by weighted cardinality. Indeed, using $d := A'\mathbf{1}$

$$\mathbf{1}'AS^k \equiv d'S^k \equiv \sum_{p \in S^k} d_p$$

where weights $d = \{d_p | p \in \Omega\}$ are *node degrees*

$$d_p := \sum_{q \in \Omega} A_{pq}. \quad (40)$$

For shortness, NC objective will be formatted like (29)

$$E_{nc}(S) \equiv -\sum_k \frac{S^{k\prime} A S^k}{d' S^k}. \quad (41)$$

It is known [69] that affinities such that $d_p \approx const$, *e.g.* on K-nearest neighbor (KNN) graphs, remove various *density biases* in kernel clustering. In this case objectives NC (41), AC (38), and AA (29) become equivalent. More generally, Bach & Jordan [6], Dhillon et al. [35] showed that NC objective can always be reduced to weighted AA or kKM with specific weights and affinities.

Our matrix notation makes equivalence between NC (41) and weighted AA (37) straightforward. Indeed, objective (41) with A coincides with (37) for weights w and affinity $\tilde{A}$

$$w = d = A'\mathbf{1} \quad \text{and} \quad \tilde{A} = W^{-1} A W^{-1}. \quad (42)$$

The weighted version of kKM procedure (24) [99, Appendix A] minimizes weighted AA (37) only for p.s.d. affinities, but positive definiteness of A is not critical. For example, an extension of the *diagonal shift* (30) [89] can convert NC (41) with arbitrary (symmetric) A to an equivalent NC objective with p.s.d. affinity

$$\mathcal{K} = A + \delta \cdot D \quad (43)$$

using *degree matrix* $D := diag(d) \equiv W$ and sufficiently large δ. Indeed, for indicators $X \in \{0,1\}^{|\Omega|}$ we have $X'DX = d'X$ and

$$\frac{X'\mathcal{K}X}{d'X} = \frac{X'AX}{d'X} + \delta \frac{X'DX}{d'X} = \frac{X'AX}{d'X} + \delta.$$

Positive definite $\mathcal{K}$ (43) implies positive definite affinity (42) of weighted AA

$$\tilde{\mathcal{K}} = D^{-1}\mathcal{K}D^{-1} = D^{-1}AD^{-1} + \delta D^{-1}. \quad (44)$$

The weighted version of kKM procedure (24) for this p.d. kernel [36] greedily optimizes NC objective (41) for any (symmetric) A.

2.4 Optimization methods for kernel clustering

As discussed in the previous section, both NC and AC can be reduced to weighted AA. Thus, all of these objectives can be optimized by basic kernel K-means procedure (8,24) or its weighted variant. However, there are many other standard methods for approximate optimization of NP-hard kernel clustering energies.

Spectral relaxation: Shi, Malik, and Yu [94, 110] popularized *spectral relaxation* methods in the context of normalized cuts. Such methods also apply to AA and other problems [94]. For example, similarly to [110] one can rewrite AA energy (27) as

$$E_{aa}(S) = -\operatorname{tr}(Z'AZ) \quad \text{for} \quad Z := \left[\ldots, \frac{S^k}{\sqrt{|S^k|}}, \ldots\right]$$

where Z is a $|\Omega| \times K$ matrix of normalized indicator vectors S^k. Orthogonality $(S^i)'S^j = 0$ implies $Z'Z = I_K$ where I_K is an identity matrix of size $K \times K$. Minimization of the *trace* energy above with relaxed Z constrained to a "unit sphere" $Z'Z = I_K$ is a simple representative example of *spectral relaxation* in the context of AA. This relaxed trace optimization is a generalization of *Rayleigh quotient* problem that has an exact closed form solution in terms of K largest eigenvectors for matrix A. This approach extends to general multi-label weighted AA and related graph clustering problems, *e.g.* AC and NC [94, 110]. The main computational difficulties for spectral relaxation methods are explicit eigen decomposition for large matrices and integrality gap - there is a final heuristics-based discretization step for extracting an integer solution for the original combinatorial problem from an optimal relaxed solution. For example, one basic discretization heuristic is to run K-means over the row-vectors of the optimal relaxed Z.

Fuzzy kernel k-means: Buhmann et al. [48, 89] address the general AD and AA energies via mean-field approximation. They derive an iterative algorithm that can be seen as a soft or *fuzzy* version[7] of kKM procedure (24). In particular, at current segments S^k_t they compute unary "potentials"

$$U^k_{p,t} = \frac{S^{k\prime}_t \mathcal{K} S^k_t}{|S^k_t|^2} - 2\frac{\mathbf{1}_p{}' \mathcal{K} S^k_t}{|S^k_t|} \quad (45)$$

[7] Similar to fuzzy K-means in [67, 38, 88] if extended to kKM.

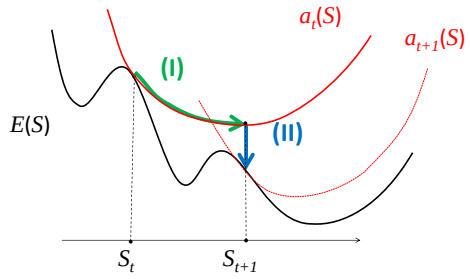


Fig. 6: *Illustration of the general bound optimization procedure:* Iteration t of optimizing function $E(S)$ using auxiliary functions (bounds) $a_t(S)$. *Step I* minimizes $a_t(S)$. *Step II* computes the next bound $a_{t+1}(S)$.

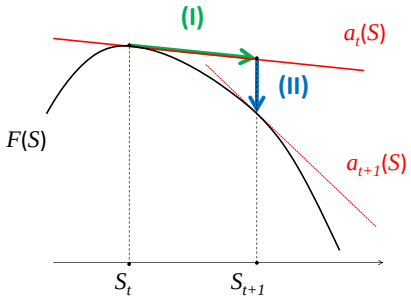


Fig. 7: *K-means as linear bound optimization:* As obvious from (51), the bound $a_t(S)$ in Theorem 1 is a unary function of S. KM procedures (23,24) correspond to optimization of linear auxiliary functions $a_t(S)$ for KM objectives. Optimum S_{t+1} is finite since optimization is over $S^k \in \{0,1\}^{|\Omega|}$.

where $U^k_{p,t}$ is a penalty for assigning label k to pixel p identical to the expression evaluated in (24). But, instead of updating point labels according to the lowest penalty $S_{p,t+1} = \arg\min_k U^k_{p,t}$ as in (24), the updates in [48] use *soft-min* operation based on *temperature* parameter T

$$S^k_{p,t+1} = \frac{\exp\left(\frac{-U^k_{p,t}}{T}\right)}{\sum_l \exp\left(\frac{-U^l_{p,t}}{T}\right)} \tag{46}$$

where soft assignments $S^k_p \in [0,1]$ define probability distributions $(S^k_p | 1 \leq k \leq K) \in \Delta^K$ over labels, see the "alternative" side of Table 1. As $T \to 0$ soft-min (46) converges to binary indicators $S^k_p \in \{0,1\}$ and distributions over labels become vertices of simplex Δ^K. That is, soft-min (46) reduces to "hard-min" $S_{p,t+1} = \arg\min_k U^k_{p,t}$ in (24).

Handling extra constraints: Some efforts to combine kernel clustering and regularization were made before us. For example, to combine kKM or NC objectives with Potts regularization [60] normalizes the corresponding pairwise constraints by cluster sizes. This alters the Potts model to fit the problem to a standard trace-based formulation. In contrast, we address joint optimization via new bounds for the kernel clustering terms.

Adding non-homogeneous linear constraints into spectral relaxation techniques also requires approximations [111] or model modifications [108]. Exact optimization for the relaxed quadratic ratios (including NC) with arbitrary linear equality constraints is possible by solving a sequence of spectral problems [40]. To incorporate *must-link* and *cannot-link* constraints, [26] reformulated normalized cut and solved a different eigen problem .

3 Kernel Bounds

Our bound optimization approach allows to combine many standard kernel clustering objectives and any regularization terms with existing solvers. We interpret kernel clustering objectives as high-order energy terms and approximate them by linear upper bounds during optimization.

First, we review the general bound optimization principle and present basic K-means as an example. Section 3.2 derives kernel bounds for standard kernel clustering objectives. Without loss of generality, we assume symmetric affinities $A = A'$ since non-symmetric ones can be equivalently replaced by $\frac{A+A'}{2}$, *e.g.* see (30) in Sec.2.2.2. Positive definiteness of A is not assumed and *diagonal shifts* are discussed when needed. Move-making bound optimization for energy (1) is discussed in Section 3.3.

3.1 Bound optimization and K-means

In general, bound optimizers are iterative algorithms that optimize *auxiliary functions* (upper bounds) for a given energy $E(S)$ assuming that these auxiliary functions are more tractable than the original difficult optimization problem [61, 77, 10, 96]. Let t be a current iteration index. Then $a_t(S)$ is an auxiliary function of $E(S)$ at current solution S_t if

$$E(S) \leq a_t(S) \quad \forall S \tag{47a}$$
$$E(S_t) = a_t(S_t). \tag{47b}$$

The auxiliary function is minimized at each iteration t (Fig. 6)

$$S_{t+1} = \arg\min_S a_t(S). \tag{48}$$

objective $E_A(S)$	**matrix formulation** $e(X)$ in $\sum_k e(S^k)$	**concave relaxation** $\widehat{e}(X)$ (53)	$\mathcal{K}$ **and** w **in Lemma 1**	**kernel bound** for $E_A(S)$ at S_t
AA (29)	$-\frac{X'AX}{\mathbf{1}'X}$	$-\frac{X'(\delta\mathbf{I}+A)X}{\mathbf{1}'X}$	$\mathcal{K}=\delta\mathbf{I}+A, \quad w=\mathbf{1}$	$\sum_k \nabla\widehat{e}(S_t^k)' S^k + const$ for $\nabla\widehat{e}$ in (54)
AC (38)	$\frac{X'(D-A)X}{\mathbf{1}'X}$	$-\frac{X'(\delta\mathbf{I}+A-D)X}{\mathbf{1}'X}$	$\mathcal{K}=\delta\mathbf{I}+A-D, \quad w=\mathbf{1}$	
NC (41)	$-\frac{X'AX}{d'X}$	$-\frac{X'(\delta D+A)X}{d'X}$	$\mathcal{K}=\delta D+A, \quad w=d$	

Table 4: *Kernel bounds* for different kernel clustering objectives $E_A(S)$. The second column shows formulations of these objectives $E_A(S) \equiv \sum_k e(S^k)$ using functions e over segment indicator vectors $S^k \in \{0,1\}^{|\Omega|}$. The last column gives a unary (linear) upper bound for $E_A(S)$ at S_t based on the first-order Taylor approximation of *concave relaxation* function $\widehat{e}: \mathcal{R}^\Omega \to \mathcal{R}^1$ (53).

This procedure iteratively decreases function $E(S)$ since

$$E(S_{t+1}) \leq a_t(S_{t+1}) \leq a_t(S_t) = E(S_t).$$

We show that standard KM procedures (23), (24) correspond to bound optimization for K-means objective (18). Note that variables m_k in mixed objective $F(S,m)$ (16) can be seen as relaxations of segment means μ_{S^k} (17) in single-variable KM objective $F(S)$ (18) since

$$\mu_{S^k} = \arg\min_{m_k} \sum_{p \in S^k} \|\phi_p - m_k\|^2$$

$$\text{and} \quad F(S) = \min_m F(S,m). \tag{49}$$

Theorem 1 (bound for KM). *Standard iterative K-means procedures (23,24) are bound optimization methods for K-means objectives $F(S)$ (18,22) using auxiliary function*

$$a_t(S) = F(S,\mu_t) \tag{50}$$

at any current segmentation $S_t = \{S_t^k\}$ with means $\mu_t = \{\mu_{S_t^k}\}$.

Proof. Equation (49) implies $a_t(S) \geq F(S)$. Since $a_t(S_t) = F(S_t)$ then $a_t(S)$ is an auxiliary function for $F(S)$. Re-segmentation step (23) gives optimal segments S_{t+1} minimizing the bound $a_t(S)$. The re-centering step minimizing $F(S_{t+1},m)$ for fixed segments gives means μ_{t+1} defining bound $a_{t+1}(S)$ for the next iteration. These re-segmentation (I) and re-centering (II) steps are illustrated in Figs. 6,7. □

Theorem 1 could be generalized to *probabilistic K-means* [53] by stating that block-coordinate descent for distortion clustering or ML model fitting (14) is a bound optimization [96,97]. Theorem 1 can also be extended to pairwise and weighted versions of KM. For example, one straightforward extension is to show that $F^w(S,\mu_t^w)$ (32) with weighted means $\mu_t^w = \{\mu_{S_t^k}^w\}$ (33) is a bound for weighted KM objective $F^w(S)$ (35) [99, Th.6]. Then, some bound for pairwise wkKM energy (36) can also be derived [99, Cor.1]. It follows that bounds can be deduced for many kernel clustering criteria using their reductions to various forms of kKM reviewed in Sec.2.2.2 or 2.3.

Alternatively, the next Section 3.2 follows a more direct and intuitive approach to deriving kernel clustering bounds motivated by the following simple observation. Note that function $a_t(S)$ in Theorem 1 is *unary* with respect to S. Indeed, functions $F(S,m)$ (16) or $F^w(S,m)$ (32) can be written in the form

$$F(S,m) \equiv \sum_k \sum_p \|\phi_p - m_k\|^2 \, S_p^k \tag{51}$$

$$F^w(S,m) \equiv \sum_k \sum_p w_p \|\phi_p - m_k\|^2 \, S_p^k \tag{52}$$

highlighting the sum of unary terms for variables S_p^k. Thus, bounds for KM or weighted KM objectives are *modular* (linear) function of S. This simple technical fact has useful implications that were previously overlooked. For example,

- in the context of bound optimization, KM can be integrated with many regularization potentials whose existing solvers can work with extra unary (linear) terms
- assuming real-valued relaxation of indicators S^k, linearity of upper bound $a_t(S)$ (50) implies that the bounded function $F(S) \in \mathcal{C}^1$ (22) is *concave*, see Fig.7.

In Section 3.2 we confirm that many standard kernel clustering objectives in Sections 2.2.2 and 2.3 have *concave relaxations*. Thus, their linear upper bounds easily follow from the corresponding first-order Taylor expansions, see Figure 7 and Table 4.

3.2 Kernel Bounds for AA, AC, and NC

The next lemma helps to find linear bounds for clustering terms AA, AC, or NC in Table 4 (Theorem 2) and bounds for our joint energy (1) in Corolary 1.

Lemma 1 (concave relaxation). *Consider function* $\widehat{e}: \mathcal{R}^{\Omega} \to \mathcal{R}^1$ *defined by matrix* $\mathcal{K}$ *and vector* w *as*

$$\widehat{e}(X) \;:=\; -\frac{X'\mathcal{K}X}{w'X}. \tag{53}$$

Function $\widehat{e}(X)$ *is concave over region* $w'X > 0$ *assuming (symmetric) matrix* $\mathcal{K}$ *is positive semi-definite (see Fig. 8).*

Proof. Lemma 1 follows from the following expression for the Hessian of function $\widehat{e}$ for symmetric $\mathcal{K}$

$$\begin{aligned}\frac{\nabla\nabla\widehat{e}}{2} &= -\frac{\mathcal{K}}{w'X} + \frac{\mathcal{K}Xw' + wX'\mathcal{K}}{(w'X)^2} - \frac{wX'\mathcal{K}Xw'}{(w'X)^3} \\ &\equiv -\frac{1}{w'X}\left(\mathbf{I} - \frac{Xw'}{w'X}\right)' \mathcal{K}\left(\mathbf{I} - \frac{Xw'}{w'X}\right)\end{aligned}$$

which is negative semi-definite for $w'X > 0$ for p.s.d. $\mathcal{K}$. □

The first-order Taylor expansion at current solution X_t

$$T_t(X) \;:=\; \widehat{e}(X_t) \;+\; \nabla\widehat{e}(X_t)'\,(X - X_t)$$

is a bound for the concave function $\widehat{e}(X)$ (53). Its gradient[8]

$$\nabla\widehat{e}(X_t) \;=\; w\,\frac{X_t'\mathcal{K}X_t}{(w'X_t)^2} \;-\; \mathcal{K}X_t\frac{2}{w'X_t} \tag{54}$$

gives linear bound $T_t(X)$ for concave function $\widehat{e}(X)$ at X_t

$$T_t(X) \;\equiv\; \nabla\widehat{e}(X_t)'\,X. \tag{55}$$

As shown in the second column of Table 4, common kernel clustering objectives defined by affinity matrix A such as AA (29), AC (38), and NC (41) have the form

$$E_A(S) \;=\; \sum_k e(S^k)$$

with function $e(X)$ as in (53) from Lemma 1. However, arbitrary affinity A may not correspond to a positive semi-definite $\mathcal{K}$ in (53) and $e(X)$ may not be concave for $X \in \mathcal{R}^{|\Omega|}$. However, the *diagonal shift* trick [89] in (30) works here too. The third column in Table 4 shows concave function $\widehat{e}(X)$ that equals $e(X)$ for any non-zero Boolean $X \in \{0,1\}^{|\Omega|}$, up to a constant. Indeed, for AA

$$\widehat{e}(X) \;= -\frac{X'(\delta\mathbf{I} + A)X}{\mathbf{1}'X} \;= -\frac{X'AX}{\mathbf{1}'X} - \delta \;\overset{c}{=}\; e(X)$$

[8] Function $\widehat{e}$ and gradient $\nabla\widehat{e}$ are defined only at non-zero indicators X_t where $w'X_t > 0$. We can formally extend $\widehat{e}$ to $X = \mathbf{0}$ and make the bound T_t work for $\widehat{e}$ at $X_t = \mathbf{0}$ with some *supergradient*. However, $X_t = 0$ is not a problem in practice since it corresponds to an empty segment.

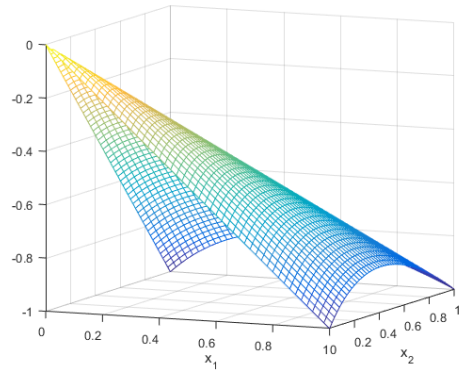


Fig. 8: Example: concave function $\widehat{e}(X) = -\frac{X'X}{\mathbf{1}'X}$ for $X \in [0,1]^2$. Note that convexity/concavity of similar rational functions with quadratic enumerator and linear denominator is known in other optimization areas, *e.g.* [12, p.72] states convexity of $\frac{x^2}{y}$ for $y > 0$ and [7, exercise 3.14] states convexity of $\frac{(w'X)^2}{w'X}$ for $w'X > 0$.

since $X'X = \mathbf{1}'X$ for Boolean X. Clearly, $\delta\mathbf{I} + A$ is p.s.d. for sufficiently large δ and Lemma 1 implies that the first-order Taylor expansion $T_t(X)$ (55) is a linear bound for concave function $\widehat{e}(X)$. Equivalence between e and $\widehat{e}$ over Booleans allows to use $T_t(X)$ as a bound for e when optimizing over indicators X. Function $\widehat{e}: \mathcal{R}^{|\Omega|} \to \mathcal{R}^1$ can be described as a *concave relaxation* of the high-order pseudo-boolean function $e: \{0,1\}^{|\Omega|} \to \mathcal{R}^1$.

Concave relaxation $\widehat{e}$ for AC in Table 4 follows from the same diagonal shift $\delta\mathbf{I}$ as above. But NC requires diagonal shift δD with degree matrix $D = diag(d)$ as in (43). Indeed,

$$\widehat{e}(X) \;= -\frac{X'(\delta D + A)X}{d'X} \;= -\frac{X'AX}{d'X} - \delta \;\overset{c}{=}\; e(X) \tag{56}$$

since $X'DX \equiv X'diag(d)X = d'X$ for any Boolean X. Clearly, $\delta D + A$ is p.s.d. for sufficiently large δ assuming $d_p > 0$ for all $p \in \Omega$. Concave relaxations and the corresponding Taylor-based bounds for $E_A(S)$ in Table 4 imply the following theorem.

Theorem 2 (kernel bound for E_A). *For (symmetric) affinity matrix* A *and current solution* S_t *the following is a unary (linear) bound for any kernel clustering energy* $E_A(S)$ *in Table 4*

$$a_t(S) \;=\; \sum_k \nabla\widehat{e}(S_t^k)'\,S^k \tag{57}$$

where $\widehat{e}$ *and* $\nabla\widehat{e}$ *are defined in (53), (54) and* δ *is large enough so that the corresponding* $\mathcal{K}$ *in Table 4 is positive semi-definite.*

Similarly to Theorem 1, optimization of our linear kernel bound in Theorem 2 can be related to kKM updates (24).

Indeed, (57) can be written in the form

$$a_t(S) \equiv \sum_k \sum_p \mathbf{1}'_p \nabla \widehat{e}(S_t^k) S_p^k = \sum_p \left(\sum_k \mathbf{1}'_p \nabla \widehat{e}(S_t^k) S_p^k \right)$$

that breakes into the sum of linear terms for each p

$$\sum_k \mathbf{1}'_p \nabla \widehat{e}(S_t^k) S_p^k \,. \tag{58}$$

Each of these can be optimized independently over probability simplex $\sum_k S_p^k = 1$. The optimal solution for (58) is always at one of K corners of the simplex corresponding to label k with the lowest potential $\mathbf{1}'_p \nabla \widehat{e}(S_t^k)$. For example, assuming AA objective (29) with $w = \mathbf{1}$ and $A = \mathcal{K}$, then (54) implies optimal k as in the "hard" kKM update (24). Interestingly, combining (58) with an "entropy barrier" pushing the solution away from the simplex corners

$$\sum_k \mathbf{1}'_p \nabla \widehat{e}(S_t^k) S_p^k + T \cdot \sum_k S_p^k \log S_p^k$$

results in the optimal "soft" kKM update (46) as in [48]. In their mean-field approach, the objective above comes as KL divergence between Gibbs distributions for the exact and approximate AA energies. Note $\mathbf{1}'_p \nabla \widehat{e}(S_t^k) \equiv U_{p,t}^k$, see (45).

For the joint energy (1) combining kernel clutsering and regularization terms we can use the following bounds.

Corollary 1 (kernel bound for (1)). *For any (symmetric) affinity matrix A and any current solution S_t the following is an auxiliary function for energy (1) with any clustering term $E_A(S)$ from Tab.4*

$$a_t(S) = \sum_k \nabla \widehat{e}(S_t^k)' S^k + \gamma \sum_{c \in \mathcal{F}} E_c(S_c) \tag{59}$$

where $\widehat{e}$ and $\nabla \widehat{e}$ are defined in (53), (54) and δ is large enough so that the corresponding $\mathcal{K}$ in Table 4 is positive semi-definite.

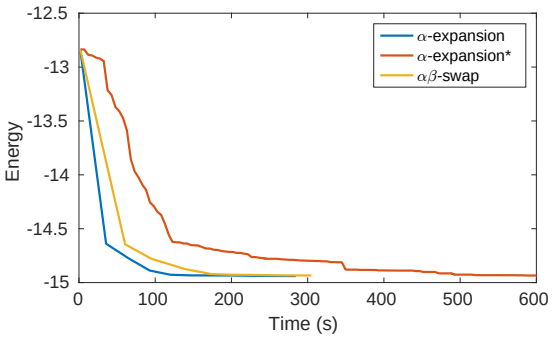
3.3 Move-making algorithms

Combination (59) of regularization potentials with a unary (linear) bound $\sum_k \nabla \widehat{e}(S_t^k)' S^k$ for high-order term $E_A(S)$ can be optimized with many standard discrete or continuous multi-label methods including graph cuts [16,49], message passing [55], LP relaxations [107], or well-known continuous convex formulations [23,24,32]. We focus on MRF regularizers (see Sec.2.1) commonly addressed by graph cuts [16]. We discuss some details of kernel bound optimization technique using such methods.

Step I of the bound optimization algorithm (Fig.6) using auxiliary function $a_t(S)$ (59) for energy $E(S)$ (1) with regularization potentials reviewed in Sec.2.1 can be done via move-making methods [16,54,33]. Step II requires re-evaluation of the first term in (59), *i.e.* the kernel bound for E_A. Estimation of gradients $\nabla \widehat{e}(S_t^k)$ in (54) has complexity $O(K|\Omega|^2)$.

Algorithm 1: α-Expansion for Kernel Cut

Input : Affinity matrix $\mathcal{A}$ of size $|\Omega| \times |\Omega|$;
Initial labeling $S_0^1, ..., S_0^K$
Output: $S^1, ..., S^K$: partition of the set Ω
Find p.s.d. matrix $\mathcal{K}$ as in Table 4. Set $t := 0$;
while *not converged* **do**
  Set $a_t(S)$ to be kernel bound (59) at current partition S_t;
  for *each label* $\alpha \in \mathcal{L} = \{1, ..., K\}$ **do**
    Find $S_t := \arg\min a_t(S)$ within one α expansion of S_t;
  end
  Set $t := t + 1$;
end



(a) Versions of Kernel Cut

Compare	against	# of wins	p-value†
α-expansion	$\alpha\beta$-swap	135/200	10^{-6}
α-expansion	α-expansion*	182/200‡	10^{-34}‡

† The probability to exceed the given number of wins by random chance.
‡ The algorithm stopped due to time limit (may cause incorrect number of wins).

(b) BSDS500 training dataset

Fig. 9: Typical energy evolution wrt different moves and frequency of bound updates. α-*expansion* updates the bound after a round of expansions, α-*expansion** updates the bound after each expansion move. Initialization is a regular 5×5 grid of patches.

Even though the global optimum of a_t at step I (Fig.6) is not guaranteed for general potentials E_c, it suffices to decrease the bound in order to decrease the energy, *i.e.* (47a) and (47b) imply

$$a_t(S_{t+1}) \le a_t(S_t) \quad \Rightarrow \quad E(S_{t+1}) \le E(S_t).$$

For example, Algorithm 1 shows a version of our kernel cut algorithm using α-expansion [16] for decreasing bound $a_t(S)$ in (59). Other moves are also possible, for example $\alpha\beta$-swap.

In general, *tighter* bounds work better. Thus, we do not run iterative move-making algorithms for bound a_t until con-

vergence before re-estimating a_{t+1}. Instead, one can reestimate the bound either after each move or after a certain number of moves. One should decide the order of iterative move making and bound evaluation. In the case of α-expansion, there are at least three options: updating the bound after a single expansion step, or after a single expansion loop, or after the convergence of α-expansion. More frequent bound recalculation slows down the algorithm, but makes the bound tighter. The particular choice generally depends on the trade-off between the speed and solution quality. However, in our experiments more frequent update does not always improve the energy, see Fig.9. We recommend updating the bound after a single loop of expansions, see Alg.1. We also evaluated a swap move version of our kernel cut method with bound re-estimation after a complete $\alpha\beta$-swaps loop, see Fig.9.

4 Data Embeddings and Spectral Bounds

This section shows a different bound optimization approach to kernel clustering, see Table 5 and Theorem 3, and joint regularization energy (1), see Corollary 2. In contrast to the bounds explicitly using affinity A or kernel matrices $\mathcal{K}$ in Sec.3.2, the new approach is based on explicit use of isometric data embeddings ϕ, see Sec. 2.2.2. While the general Mercer theorem guarantees existence of such possibly infinite dimensional Hilbert space embedding, we show finite dimensional Euclidean embedding

$$\phi := [\phi_p] \qquad \text{where} \quad \{\phi_p | p \in \Omega\} \subset \mathcal{R}^{|\Omega|}$$

with exact isometry (19,20) to kernels $\mathcal{K}$ in Table 4 and lower dimensional embeddings

$$\tilde{\phi} := [\tilde{\phi}_p] \qquad \text{where} \quad \{\tilde{\phi}_p | p \in \Omega\} \subset \mathcal{R}^{m} \quad \text{for} \quad m \leq |\Omega|$$

that can approximate the same isometry with any accuracy. The embeddings use eigen decompositions of the kernels.

Explicit embeddings allow to formulate exact or approximate *spectral bounds* for standard kernel clustering objectives like AA, AC, NC. This approach is very closely related to spectral relaxation, see Sec. 4.3. For example, optimization of our approximate spectral bounds for $m = K$ is similar to standard discretization heuristics using K-means over eigenvectors [94]. Our bound optimization framework provides justification for such heuristics. Moreover, our spectral bounds also allow to optimize joint energy (1) combing kernel clustering objectives with common regularization terms.

Spectral bound is a useful alternative to kernel bound in Sec. 3.2. Their complexity and other numerical properties are different. In particular, spectral bound optimization with lower dimensional Euclidean embeddings $\tilde{\phi}$ for $m \ll |\Omega|$ is often less sensitive to local minima. This may lead to better solutions, even though such embeddings $\tilde{\phi}$ are only approximately isometric to given pairwise affinities. For $m = |\Omega|$,

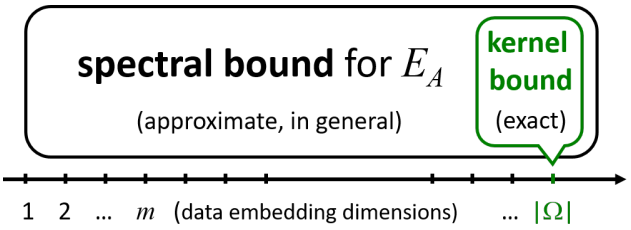


Fig. 10: Interpreting our linear bounds for E_A term in (1) via K-means: optimization of the *spectral* bound (alone) is equivalent to K-means algorithm over approximately isometric data embeddings in $\mathcal{R}^m$ for $m \leq |\Omega|$, see Sec. 4. As m approaches $|\Omega|$, the isometry becomes more accurate and our approximate spectral bound for E_A reduces to the exact *kernel* bound. While relations between E_A and K-means were known for $m = K$ [94] (as a heuristic, see Sec.4.3) and $m = |\Omega|$ [89,6,35] (as energy equivalence), we establish it in a new bound optimization context essential for our work.

the spectral bound is mathematically equivalent to the kernel bound, but their numerical representations are different. Figure 10 summarizes the relationship between our (*kernel* and *spectral*) bounds for kernel clustering objective $E_A(S)$.

4.1 Exact and approximate embeddings ϕ for kKM

This section uses some standard methodology [31] to build the finite-dimensional embedding $\phi_p \equiv \phi(I_p)$ with exact or approximate isometry (19,20) to any given positive definite kernel k over finite data set $\{I_p | p \in \Omega\}$. As discussed in Sec. 2.2.2, kKM and other kernel clustering methods are typically defined by affinities/kernels k and energy (22) rather than by high-dimensional embeddings ϕ with basic KM formulation (18). Nevertheless, data embeddings ϕ_p could be useful and some clustering techniques explicitly construct them [94,80,89,8,6,112]. In particular, if dimensionality of the embedding space is relatively low then the basic iterative KM procedure (23) minimizing (18) could be more efficient than its kernel variant (24) for quadratic formulation (22). Even when working with a given kernel k it may be algorithmically beneficial to build the corresponding isometric embedding ϕ. Below we discuss finite-dimensional Euclidean embeddings in $\mathcal{R}^m$ ($m \leq |\Omega|$) allowing to approximate standard kernel clustering via basic KM.

First, we show an exact Euclidean embedding isometric to a given kernel. Any finite data set $\{I_p | p \in \Omega\}$ and any given kernel k define a positive definite *kernel matrix*[9]

$$\mathcal{K}_{pq} = k(I_p, I_q)$$

[9] If k is given as a continuous kernel $k(x, y) : \mathcal{R}^N \times \mathcal{R}^N \to \mathcal{R}$ matrix $\mathcal{K}$ is its *restriction* to finite data set $\{I_p | p \in \Omega\} \subset \mathcal{R}^N$.

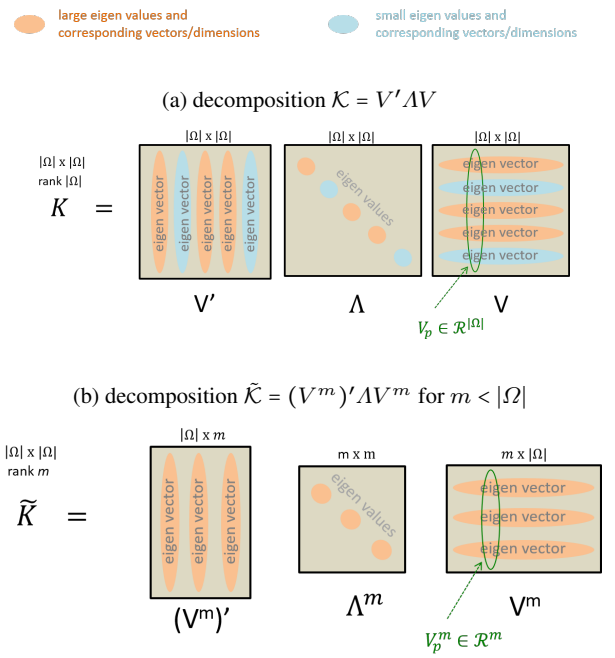


Fig. 11: Eigen decompositions for kernel matrix $\mathcal{K}$ (a) and its rank m approximation $\tilde{\mathcal{K}}$ (b) minimizing Frobenius errors (61) [31]. Decompositions (a,b) give explicit embeddings (60,63) isometric to the kernels, as in the Mercer theorem. One specific example for the Gaussian kernel is in Fig.12.

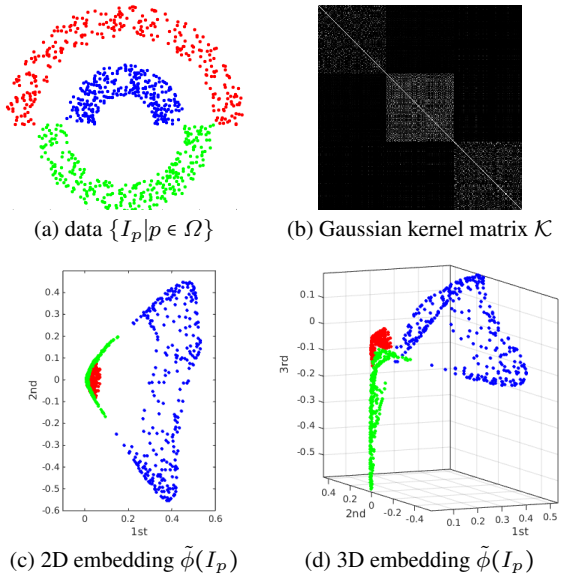


Fig. 12: Low-dimensional Euclidean embeddings (63) for $m = 2$ and $m = 3$ in (c,d) are approximately isometric to a given affinity matrix (b) over the data points in (a). The approximation error (62) decreases for larger m. While generated by standard MDS methodology [31], it is intuitive to call embeddings ϕ in (60) and (63) as (exact or approximate) *isometry eigenmap* or *eigen isomap*.

of size $|\Omega| \times |\Omega|$. The *eigen decomposition* of this matrix

$$\mathcal{K} = V'\Lambda V$$

involves diagonal matrix Λ with non-negative eigenvalues and orthogonal matrix V whose rows are eigenvectors, see Fig.11(a). Non-negativity of the eigenvalues is important for obtaining decomposition $\Lambda = \sqrt{\Lambda} \cdot \sqrt{\Lambda}$ allowing us to define the following Euclidean space embedding

$$\phi_p := \sqrt{\Lambda} V_p \quad \in \mathcal{R}^{|\Omega|} \tag{60}$$

where V_p are column of V, see Fig.11(a). This embedding satisfies isometry (19,20) since

$$\langle \phi_p, \phi_q \rangle = (\sqrt{\Lambda} V_p)'(\sqrt{\Lambda} V_q) = \mathcal{K}_{pq} = k(I_p, I_q).$$

Note that (60) defines a simple finite dimensional embedding $\phi_p \equiv \phi(I_p)$ only for subset of points $\{I_p | p \in \Omega\}$ in $\mathcal{R}^N$ based on a discrete kernel, *i.e.* matrix $\mathcal{K}_{pq}$. In contrast, Mercer's theorem should produce a more general infinite dimensional Hilbert embedding $\phi(x)$ for any $x \in \mathcal{R}^N$ by extending the eigen decomposition to continuous kernels $k(x, y)$. In either case, however, the embedding space dimensionality is much higher than the original data space. For example, ϕ_p in (60) has dimension $|\Omega|$, which is much larger than the dimension of data I_p, *e.g.* 3 for RGB colors.

Embedding (60) satisfying isometry (19,20) is not unique. For example, any decomposition $\mathcal{K} = G'G$, *e.g.* Cholesky [43], defines a mapping $\phi_p^G := G_p$ with desired properties. Also, rotational matrices R generate a class of isometric embeddings $\phi_p^R := R\phi_p$.

It is easy to build lower dimensional embeddings by weakening the exact isometry requirements (19,20) following the standard *multi-dimensional scaling* (MDS) methodology [31], as detailed below. Consider a given rank $m < |\Omega|$ approximation $\tilde{\mathcal{K}}$ for kernel matrix $\mathcal{K}$ minimizing Frobenius norm errors [31]

$$\|\mathcal{K} - \tilde{\mathcal{K}}\|_F \;:=\; \sum_{pq \in \Omega} (\mathcal{K}_{pq} - \tilde{\mathcal{K}}_{pq})^2. \tag{61}$$

It is well known [31,43] that the minimum Frobenius error is achieved by

$$\tilde{\mathcal{K}} = (V^m)'\Lambda^m V^m$$

where V^m is a submatrix of V including m rows corresponding to the largest m eignenvalues of $\mathcal{K}$ and Λ^m is the diagonal matrix of these eigenvalues, see Fig.11(b). The corresponding minimum Frobenius error is given by the norm of zeroed out eigenvalues

$$\|\mathcal{K} - \tilde{\mathcal{K}}\|_F \;=\; \sqrt{\lambda_{m+1}^2 + \cdots + \lambda_{|\Omega|}^2}. \tag{62}$$

It is easy to check that lower dimensional embedding

$$\tilde{\phi}_p := \sqrt{\Lambda^m} V_p^m \qquad \in \mathcal{R}^m \tag{63}$$

is isometric with respect to approximating kernel $\tilde{\mathcal{K}}$, that is

$$\langle \tilde{\phi}_p, \tilde{\phi}_q \rangle \;=\; \tilde{\mathcal{K}}_{pq} \;\approx\; \mathcal{K}_{pq}. \tag{64}$$

Fig. 12 shows examples of low-dimensional approximate isometry embeddings (63) for a Gaussian kernel. Note that $\tilde{\phi}_p \in \mathcal{R}^m$ (63) can be obtained from $\phi_p \in \mathcal{R}^{|\Omega|}$ (60) by selecting coordinates corresponding to dimensions of the largest m eigenvalues.

According to (62) lower dimensional embedding $\tilde{\phi}_p$ in (63) is nearly-isometric to kernel matrix $\mathcal{K}$ if the ignored dimensions have sufficiently small eigenvalues. Then (63) may allow efficient approximation of kernel K-means. For example, if sufficiently many eigenvalues are close to zero then a small rank m approximation $\tilde{\mathcal{K}}$ will be sufficiently accurate. In this case, we can use a basic iterative K-means procedure directly in $\mathcal{R}^m$ with $O(|\Omega|m)$ complexity of each iteration. In contrast, each iteration of the standard kernel K-means (22) is $O(|\Omega|^2)$ in general[10].

There is a different way to justify approximate low-dimensional embedding $\tilde{\phi}_p$ ignoring small eigenvalue dimensions in ϕ_p. The objective in (22) for exact kernel $\mathcal{K}$ is equivalent to the basic K-means (16) over points ϕ_p (60). The latter can be shown to be equivalent to (probabilistic) K-means (13) over columns V_p in orthonormal matrix V using weighted distortion measure

$$\|V_p - \mu\|_\Lambda^2 := \sum_{i=1}^{|\Omega|} \lambda_i (V_p[i] - \mu[i])^2 = \|\phi_p - \sqrt{\Lambda}\mu\|^2$$

where index $[i]$ specifies coordinates of the column vectors. Thus, a good approximation is achieved when ignoring coordinates for small enough eigenvalues contributing low weight in the distortion above. This is equivalent to K-means (16) over points (63).

4.2 Spectral Bounds for AA, AC, and NC

The last Section showed that kKM clustering with given p.s.d. kernel $\mathcal{K}$ can be approximated by basic KM over low-dimensional Euclidean embedding $\tilde{\phi} \in \mathcal{R}^m$ (63) with approximate isometry to $\mathcal{K}$ (64). Below we use equivalence of standard kernel clustering criteria to kKM, as discussed in Sections 2.2.2 and 2.3, to derive the corresponding low-dimensional embeddings for AA, AC, NC. Then, equivalence of KM to bound optimization (Theorem 1) allows to formulate our approximate *spectral bounds* for the kernel clustering and joint energy (1). The results of this Section are summarized in Table 5. For simplicity, assume symmetric affinity matrix A. If not, equivalently replace A by $\frac{A+A'}{2}$.

Average association (AA): Diagonal shift $\mathcal{K} = \delta\mathbf{I} + A$ in (30) converts AA (29) with A to equivalent kKM (22) with p.d. kernel $\mathcal{K}$. We seek rank-m approximation $\tilde{\mathcal{K}}$ minimizing Frobenius error $\|\mathcal{K} - \tilde{\mathcal{K}}\|_F$. Provided eigen decomposition $A = V'\Lambda V$, equation (63) gives low-dimensional embedding (also in Tab. 5)

$$\tilde{\phi}_p \;=\; \sqrt{\delta\mathbf{I}^m + \Lambda^m} V_p^m \tag{65}$$

corresponding to optimal approximation kernel $\tilde{\mathcal{K}}$. It follows that KM (23) over this embedding approximates AA objective (22). Note that the eigenvectors (rows of matrix V, Fig. 11) also solve the spectral relaxation for AA in Tab. 6. However, ad hoc discretization by KM over points V_p^K may differ from the result for points (65).

Average cut (AC): As follows from objective (38) and diagonal shift (30) [89], average cut clustering for affinity A is equivalent to minimizing kKM objective with kernel $\mathcal{K} = \delta\mathbf{I} + A - D$ where D is a diagonal matrix of node degrees $d_p = \sum_q A_{pq}$. Diagonal shift $\delta\mathbf{I}$ is needed to guarantee positive definiteness of the kernel. Eigen decomposition for $D - A = V'\Lambda V$ implies $\mathcal{K} = V'(\delta\mathbf{I} - \Lambda)V$. Then, (63) implies rank-m approximate isometry embedding (also in Tab. 5)

$$\tilde{\phi}_p \;=\; \sqrt{\delta\mathbf{I}^m - \Lambda^m} V_p^m \tag{66}$$

using the same eigenvectors (rows of V) that solve AC's spectral relaxation in Tab. 6. However, standard discretization heuristic using KM over $\tilde{\phi}_p = V_p^K$ may differ from the results for our approximate isometry embedding $\tilde{\phi}_p$ (66) due to different weighting.

Normalized cut (NC): According to [35] and a simple derivation in Sec.2.3 normalized cut for affinity A is equivalent to weighted kKM with kernel $\mathcal{K} = \delta D^{-1} + D^{-1}AD^{-1}$ (44) and node weights $w_p = d_p$ based on their degree. Weighted kKM (36) can be interpreted as KM in the embedding space with weights w_p for each point ϕ_p as in (32,33). The only issue is computing m-dimensional embeddings approximately isometric to $\mathcal{K}$. Note that previously discussed solution $\tilde{\phi}$ in (63) uses eigen decomposition of matrix $\mathcal{K}$ to minimize the sum of quadratic errors between $\mathcal{K}_{pq}$ and approximating kernel $\tilde{\mathcal{K}}_{pq} = \langle \tilde{\phi}_p, \tilde{\phi}_q \rangle$. This solution may still be acceptable, but in the context of weighted points it seems natural to minimize an alternative approximation measure taking w_p into account. For example, we can find rank-m approximate affinity matrix $\tilde{\mathcal{K}}$ minimizing the sum of weighted squared errors

$$\sum_{pq \in \Omega} w_p w_q (\mathcal{K}_{pq} - \tilde{\mathcal{K}}_{pq})^2 = \|D^{\frac{1}{2}}(\mathcal{K} - \tilde{\mathcal{K}})D^{\frac{1}{2}}\|_F. \tag{67}$$

Substituting $\mathcal{K} = \delta D^{-1} + D^{-1}AD^{-1}$ gives an equivalent objective

$$\|D^{-\frac{1}{2}}(\delta D + A)D^{-\frac{1}{2}} - D^{\frac{1}{2}}\tilde{\mathcal{K}}D^{\frac{1}{2}}\|_F.$$

[10] Without *KNN* or other special kernel accelerations.

objective $E_A(S)$	**matrix formulation** $e(X)$ in $\sum_k e(S^k)$	**equivalent** kKM (22,36) as in [89,36]	**eigen decomposition** $V'\Lambda V = \dots$	**embedding in** $\mathcal{R}^m$, $m \le \lvert\Omega\rvert$ with approx. isometry (64)	**spectral bound** for $E_A(S)$ at S_t
AA (29)	$-\frac{X'AX}{\mathbf{1}'X}$	$\mathcal{K} = \delta\mathbf{I} + A$	A	$\tilde{\phi}_p = \sqrt{\delta\mathbf{I}^m + \Lambda^m}V_p^m$ (65)	$F(S,\mu_t)$ (51) for points $\tilde{\phi}_p$
AC (38)	$\frac{X'(D-A)X}{\mathbf{1}'X}$	$\mathcal{K} = \delta\mathbf{I} + A - D$	$D - A$	$\tilde{\phi}_p = \sqrt{\delta\mathbf{I}^m - \Lambda^m}V_p^m$ (66)	
NC (41)	$-\frac{X'AX}{d'X}$	$\mathcal{K} = \delta D^{-1} + D^{-1}AD^{-1}$ weighted, $w_p = d_p$	$D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$	$\tilde{\phi}_p = \sqrt{\frac{\delta\mathbf{I}^m + \Lambda^m}{d_p}}V_p^m$ (70)	$F^w(S,\mu_t^w)$ (52) for points $\tilde{\phi}_p$

Table 5: *Spectral bounds* for objectives $E_A(S)$. The third column shows p.d. kernel matrices $\mathcal{K}$ for the equivalent kKM energy (22). Eigen decomposition for matrices in the forth column defines our Euclidean embedding $\tilde{\phi}_p \in \mathcal{R}^m$ (fifth column) isometric to $\mathcal{K}$ (64). Thus, K-means over $\tilde{\phi}_p$ approximates kKM (22). Bounds for KM (last column) follow from Th. 1 where $\mu_t = \{\mu_{S_t^k}\}$ are means (17) and $\mu_t^w = \{\mu_{S_t^k}^w\}$ are weighted means (33). Functions $F(S,m)$ and $F^w(S,m)$ are modular (linear) w.r.t. S, see (51,52).

	spectral relaxation [94]	**common discretization heuristic [105]**			(embedding & K-means)
AA	$A\mathbf{u} = \lambda\mathbf{u}$	$\tilde{\phi}_p := U_p^K$	$\equiv$	$V_p^K \Leftarrow V'\Lambda V = A$	
AC	$(D-A)\mathbf{u} = \lambda\mathbf{u}$	$\tilde{\phi}_p := U_p^K$	$\equiv$	$V_p^K \Leftarrow V'\Lambda V = D - A$	
NC	$(D-A)\mathbf{u} = \lambda D\mathbf{u}$	$\tilde{\phi}_p := U_p^K$	$\equiv$	$[V^K D^{-\frac{1}{2}}]_p^{rn} \Leftarrow V'\Lambda V = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$	

Table 6: *Spectral relaxation* and discretization heuristics for objectives for kernel clustering objectives $E_A(S)$ for affinity A. The corresponding *degree* matrix D is diagonal with elements $d_p := \sum_q A_{pq}$. To extract integer labeling from the relaxed solutions produced by the eigen systems (second column), spectral methods often apply basic KM to some ad hoc data embedding $\tilde{\phi}$ (last column) based on the first K unit eigenvectors $\mathbf{u}$, the rows of matrix U^K. While our main text discusses some variants, the most basic idea [94,105] is to use the columns of U^K as embedding $\tilde{\phi}_p$. For easier comparison, the last column also shows equivalent representations of this embedding based on the same eigen decompositions $V'\Lambda V$ as those used for our isometry eigenmaps in Tab. 5. In contrast, our embeddings are derived from justified approximations of the original non-relaxed AA, AC, or NC objectives. Note that NC corresponds to a *weighted* case of K-means with data point weights $w_p = d_p$ [6,35], see (42) in Section 2.3.

Consider rank-m matrix $\tilde{M} := D^{\frac{1}{2}}\tilde{\mathcal{K}}D^{\frac{1}{2}}$ as a new minimization variable. Its optimal value $(V^m)'(\delta\mathbf{I}^m + \Lambda^m)V^m$ follows from $D^{-\frac{1}{2}}(\delta D + A)D^{-\frac{1}{2}} = V'(\delta\mathbf{I} + \Lambda)V$ for eigen decomposition

$$D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \equiv V'\Lambda V. \tag{68}$$

Thus, optimal rank-m approximation kernel $\tilde{\mathcal{K}}$ is

$$\tilde{\mathcal{K}} = D^{-\frac{1}{2}}(V^m)'(\delta\mathbf{I}^m + \Lambda^m)V^m D^{-\frac{1}{2}}. \tag{69}$$

It is easy to check that m-dimensional embedding (also in Tab. 5)

$$\tilde{\phi}_p = \sqrt{\frac{\delta\mathbf{I}^m + \Lambda^m}{d_p}}V_p^m \tag{70}$$

is isometric to kernel $\tilde{\mathcal{K}}$, that is $\langle\tilde{\phi}_p, \tilde{\phi}_q\rangle = \tilde{\mathcal{K}}_{pq}$. Therefore, weighted KM (32) over low-dimensional embedding $\tilde{\phi}_p$ (70) with weights $w_p = d_p$ approximates NC objective (41).

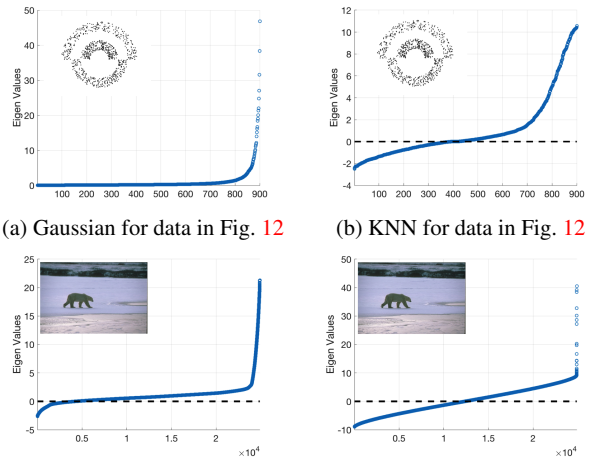
Summary: The ideas above can be summarized as follows. Assume AA, AC, or NC objectives $E_A(S)$ with (symmetric) A. The third column in Table 5 shows kernels $\mathcal{K}$ for equivalent kKM objectives $F(S)$ (22,36). Following *eigenmap* approach (Fig.11), we find rank-m approximate kernel $\tilde{\mathcal{K}} \approx \mathcal{K}$ minimizing Frobenius error $\|\tilde{\mathcal{K}} - \mathcal{K}\|_F$ (61) or its weighted version (67) and deduce embeddings $\tilde{\phi}_p \in \mathcal{R}^m$ (65), (66), (70) satisfying isometry

$$\tilde{\phi}_p'\tilde{\phi}_q = \tilde{\mathcal{K}}_{pq} \approx \mathcal{K}_{pq}.$$

Basic K-means objective $\tilde{F}(S,m)$ (16,32) for $\{\tilde{\phi}_p\}$ is equivalent to kKM energy $\tilde{F}(S)$ (22,36) for kernel $\tilde{\mathcal{K}} \approx \mathcal{K}$ and, therefore, approximates the original kernel clustering objective

$$\tilde{F}(S,\mu_S) \overset{c}{=} \tilde{F}(S) \approx F(S) \overset{c}{=} E_A(S).$$

Theorem 1 gives unary (linear) bound $\tilde{F}(S,\mu_t)$ (51,52) for objective $\tilde{F}(S)$ (16,32). We refer to $\tilde{F}(S,\mu_t)$ as a *spectral auxiliary function* for approximate optimization of $E_A(S)$



(a) Gaussian for data in Fig. 12 (b) KNN for data in Fig. 12

(c) mPb kernel [4] for image (d) KNN kernel for image

Fig. 13: Spectrum of eigenvalues of typical kernel matrices for synthetic data (top row) or real image color (bottom row). This helps us to select approximate embedding so as to have small approximation error (62). For example, with fixed width gaussian kernel in (a), it suffices to select a few top eigenvectors since the remaining eigenvalues are negligible. Note that the spectrum elevates with increasing diagonal shift δ in (65). In principle, we can find the optimal shift for a given number of dimensions m to minimize approximation error.

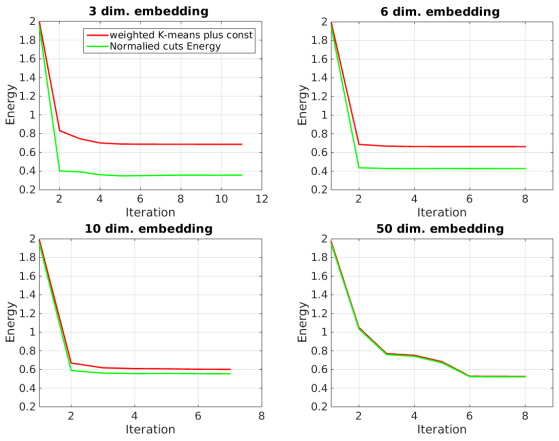


Fig. 14: For data and affinity matrix in Fig. 12, we run weighted K-means with our approximate embedding. The approximation errors $\|\mathcal{K} - \tilde{\mathcal{K}}\|_F^2 / \|\mathcal{K}\|_F^2$ for 3, 6, 10 and 50 dim. embedding are 58%, 41%, 27% and 3% respectively. We compute weighted K-means energy (up to a const) and normalized cuts energy for solution obtained at each iteration. We observed that normalized cuts energy indeed tends to decrease during iterations of K-means. Even 10 dim. embedding gives good alignment between K-means energy and normalized cuts energy. Higher dimensional embedding gives better energy approximation, but not necessarily better solution with lower energy.

(last column in Table 5). We will also simply call $\tilde{F}(S, \mu_t)$ a *spectral bound* for E_A, not to be confused with a similar term used for matrix eigenvalues.

Theorem 3 (spectral bound for E_A). *For (symmetric) affinity matrix A assume sufficiently large diagonal shift δ generating p.s.d. kernel $\mathcal{K}$ as in Table 5. Then, auxiliary function*

$$\tilde{a}_t(S) \;=\; \tilde{F}(S, \mu_t) \tag{71}$$

using $\tilde{F}(S, m)$ (51,52) with embedding $\{\tilde{\phi}_p\} \subset \mathcal{R}^m$ in Tab. 5 is a unary (linear) bound for K-means energy $\tilde{F}(S)$ (22,36) approximating objective $E_A(S)$ as $m \to |\Omega|$.

For $m = |\Omega|$ the spectral bounds (Tab.5) are algebraically equivalent to our kernel bounds (Tab.4) since $\tilde{\mathcal{K}} = \mathcal{K}$, see (62). Yet, their numerical representation is different. For $m < |\Omega|$ we obtain a range of approximate spectral bounds since $\tilde{\mathcal{K}} \approx \mathcal{K}$ and $\tilde{F} \approx F$. Figure 10 summarizes the relation between our spectral and kernel bounds for E_A.

Interestingly, Section 4.3 shows that optimization of our spectral bounds for $m = K$ is algorithmically similar to the common K-means discretization heuristic in spectral relaxation solutions for kernel clustering. Thus, our spectral bound optimization can be seen as a principled formulation justifying this heuristic post-processing step.

Similarly to *kernel bound* in Section 3.2, *spectral bound* is useful for optimizing joint energy (1). We can iteratively minimize energy $E(S)$ in (1) by applying bound optimization approach to its *spectral approximation*

$$\tilde{E}(S) \;=\; \tilde{F}(S) + \;\gamma\; \sum_{c \in \mathcal{F}} E_c(S_c) \tag{72}$$

or its *weighted spectral approximation*

$$\tilde{E}(S) \;=\; \tilde{F}^w(S) + \;\gamma\; \sum_{c \in \mathcal{F}} E_c(S_c). \tag{73}$$

Corollary 2 (spectral bound for (1)). *For any (symmetric) affinity matrix A assume sufficiently large diagonal shift δ generating p.s.d. kernel $\mathcal{K}$ as in Table 5. Then, auxiliary function*

$$\tilde{a}_t(S) \;=\; \tilde{F}(S, \mu_t) \;+\; \gamma \sum_{c \in \mathcal{F}} E_c(S_c) \tag{74}$$

using $\tilde{F}(S, m)$ (51,52) with embedding $\{\tilde{\phi}_p\} \subset \mathcal{R}^m$ in Tab. 5 is a bound for joint energy (72,73) approximating (1) as $m \to |\Omega|$.

Approximation quality (62) depends on omitted eigenvalues λ_i for $i > m$. Representative examples in Fig.13 show that relatively few eigenvalues may dominate the others. Thus, practically good approximation with small m

Algorithm 2: α-Expansion for Spectral Cut

Input : Affinity matrix $\mathcal{A}$ of size $|\Omega| \times |\Omega|$;
Initial labeling $S_0^1, ..., S_0^K$
Output: $S^1, ..., S^K$: partition of the set Ω
Find top m eigenvalues/vectors Λ^m, V^m for a matrix in the 4^{th} col. of Tab. 5 ;
Compute embedding $\{\tilde{\phi}_p\} \subset \mathcal{R}^m$ for some δ and set $t := 0$;
while *not converged* **do**
  Set $\tilde{a}_t(S)$ to be spectral bound (74) at current partition S_t;
  for *each label* $\alpha \in \mathcal{L} = \{1, ..., K\}$ **do**
    Find $S_t := \arg\min \tilde{a}_t(S)$ within one α expansion of S_t;
  end
  Set $t := t + 1$;
end

is possible. Larger m are computationally expensive since more eigenvalues/vectors are needed. Interestingly, smaller m may give better optimization since K-means in higher-dimensional spaces may be more sensitive to local minima. Thus, spectral bound optimization for smaller m may find solutions with lower energy, see Fig.14, even though the quality of approximation is better for larger m.

Similarly to the kernel bound algorithms discussed in Section 3.3 one can optimize the approximate spectral bound (74) for energy (1) using standard algorithms for regularization. This follows from the fact that the first term in (74) is unary (linear). Algorithm 2 shows a representative (approximate) bound optimization technique for (1) using move-making algorithms [17]. Note that for $\gamma = 0$ (no regularization terms) our bound optimization Algorithm 2 reduces to basic K-means over approximate isometry embeddings $\{\tilde{\phi}_p\} \subset \mathcal{R}^m$ similar but not identical to common discretization heuristics in spectral relaxation methods.

Some extensions for optimization ideas in Sec. 3 and 4 are discussed in [99]. For example, diagonal shift δ can be used to reduce Frobenius error (62). We also discuss *pseudo-bounds* [96].

4.3 Relation to spectral clustering

Our approximation of kernel clustering such as NC via basic KM over low dimensional embeddings $\tilde{\phi}_p$ is closely related to popular spectral clustering algorithms [94, 80, 8] using eigen decomposition for various combinations of kernel, affinity, distortion, laplacian, or other matrices. Other methods also build low-dimensional Euclidean embeddings [80, 8, 112] for basic KM using motivation different from isometry and approximation errors with respect to given affinities. We are mainly interested in discussing relations to spectral methods approximately optimizing kernel clustering criteria such as AA, AC, and NC [94].

Many spectral relaxation methods also use various eigen decompositions to build explicit data embeddings followed by basic K-means. In particular, the smallest or largest eigenvectors for the (generalized) eigenvalue problems in Table 6 give well-known exact solutions for the relaxed problems. In contrast to our approach, however, the final K-means stage in spectral methods is often presented without justification [94, 105, 4] as a heuristic for quantizing the relaxed continuous solutions into a discrete labeling. It is commonly understood that

> *"... there is nothing principled about using the K-means algorithm in this step"* (Sec. 8.4 in [105])

or that

> *"... K-means introduces additional unwarranted assumptions."* (Sec. 4 in [110])

Also, typical spectral methods use K eigenvectors solving the relaxed K-cluster problems followed by KM quantization. In contrast, we choose the number of eigenvectors m based on Frobenius error for isometry approximation (62). Thus, the number m is independent from the predefined number of clusters.

Below we juxtapose our approximate isometry low dimensional embeddings in Table 5 with embeddings used for ad-hoc discretization by the standard spectral relaxation methods in Table 6. While such embeddings are similar, they are not identical. Thus, our Frobenius error argument offers a justification and minor corrections for KM heuristics in spectral methods, even though the corresponding methodologies are unrelated. More importantly, our bound formulation allows integration of kernel clustering with additional regularization constraints (1).

Embeddings in spectral methods for NC: Despite similarity, there are differences between our low-dimensional embedding (70) provably approximating kernel $\mathcal{K} = \delta D^{-1} + D^{-1}AD^{-1}$ for the kKM formulation of NC [6, 35] and common ad-hoc embeddings used for KM discretization step in the spectral relaxation methods. For example, one such discretization heuristic [94, 105] uses embedding $\tilde{\phi}_p$ (right column in Tab. 6) defined by the columns of matrix U^K whose rows are the K top (unit) eigenvectors of the standard eigen system (left column). It is easy to verify that the rows of matrix $VD^{-\frac{1}{2}}$ are non-unit eigenvectors for the generalized eigen system for NC. The following relationship

$$\tilde{\phi}_p \;=\; U^K \;\equiv\; [V^K D^{-\frac{1}{2}}]^{rn}$$

where operator $[\cdot]^{rn}$ normalizes matrix rows, demonstrates certain differences between ad hoc embeddings used by many spectral relaxation methods in their heuristic K-means discretization step and justified approximation embedding (70) in Tab. 5. Note that our formulation scales each embedding dimension, *i.e.* rows in matrix $V^K D^{-\frac{1}{2}}$, according to eigenvalues instead of normalizing these rows to unit length.

There are other common variants of embeddings for the K-means discretization step in spectral relaxation approaches to the normalized cut. For example, [9,68,4] use

$$\tilde{\phi}_p \;=\; [\Lambda^{-\frac{1}{2}}U]_p^K$$

for discretization of the relaxed NC solution. The motivation comes from the physics-based *mass-spring system* interpretation [9] of the generalized eigenvalue system.

Some spectral relaxation methods motivate their discretization procedure differently. For example, [110,6] find the closest integer solution to a *subspace* of equivalent solutions for their particular very similar relaxations of NC based on the same eigen decomposition (68) that we used above. Yu and Shi [110] represent the subspace via matrix

$$X' \;\equiv\; [\sqrt{\Lambda^m}V^m D^{-\frac{1}{2}}]^{cn}$$

where columns differ from our embedding $\tilde{\phi}(I_p)$ in (70) only by normalization. Theorem 1 by Bach and Jordan [6] equivalently reformulates the distance between the subspace and integer labelings via a *weighted* K-means objective for embedding

$$\tilde{\phi}_p \;=\; \sqrt{\frac{1}{d_p}}V_p^m \tag{75}$$

and weights $w_p = d_p$. This embedding is different from (70) only by eigenvalue scaling.

Interestingly, a footnote in [6] states that NC objective (41) is equivalent to weighted KM objective (32) for exact isometry embedding

$$\phi_p \;=\; \frac{1}{d_p}G_p \qquad \in \mathcal{R}^{|\Omega|} \tag{76}$$

based on any decomposition $A \equiv G'G$. For example, our exact isometry map (70) for $m = |\Omega|$ and $G = \sqrt{\Lambda}VD^{\frac{1}{2}}$ is a special case. While [6] reduce NC to K-means[11], their low-dimensional embedding $\tilde{\phi}$ (75) is derived to approximate the subspace of relaxed NC solutions. In contrast, low-dimensional embedding (70) approximates the exact esometry map ϕ ignoring relaxed solutions. It is not obvious if decomposition $A \equiv G'G$ for the exact embedding (76) can be used to find any approximate lower-dimensional embeddings like (70).

[11] KM procedure (23) (weighted version) is not practical for objective (32) for points ϕ_p in $\mathcal{R}^{|\Omega|}$. Instead, Dhillon et al. [35] later suggested *pairwise* KM procedure (24) (weighted version) using kernel $\mathcal{K}_{pq} \equiv \langle \phi_p, \phi_q \rangle$.

5 Experiments

This section is divided into two parts. The first part (Sec.5.1) shows the benefits of extra MRF regularization for kernel & spectral clustering, e.g. normalized cut. We consider pairwise Potts, label cost and robust bin consistency term, as discussed in Sec.2.1. We compare to spectral clustering [94, 68] and kernel K-means [35], which can be seen as degenerated versions for spectral and kernel cuts (respectively) without MRF terms. We show that MRF helps kernel & spectral clustering in segmentation and image clustering. In the second part (Sec.5.2) we replace the log-likelihoods in model-fitting methods, *e.g.* GrabCut [90], by kernel clustering term, e.g. AA and NC. This is particularly advantageous for high dimension features (location, depth, motion).

Implementation details: For segmentation, our kernel cut method uses either Gaussian kernels of fixed bandwidth σ or common KNN kernels with adaptive bandwidth *e.g.* see [113,11] and [69]. Pixel features I_p can be concatenation of LAB (color), XY (location) and M (motion or optical flow) [20]. We choose 400 neighbors and randomly sample 50 neighbors for each pixel. Sampling does not degrade our segmentation but expedites bound evaluation. We also use popular *mPb* contour based affinities [4]. The window radius is set to 5 pixels.

Another detail to mention is diagonal shift of the kernel matrix. It is necessary to give PSD matrix so that our bounds hold. However, in practice, we find the energies to decrease at each iteration even without any diagonal shift for some kernels that are not necessarily PSD, e.g. KNN kernel. As such, we choose not to add any diagonal shift in our experiments bellow. Also adding too large a diagonal shift may lead to poor local minima in kernel K-means algorithm, as discussed in [36].

For regularization in (2) we use standard contrast-sensitive penalty $w_{pq} = \frac{1}{d_{pq}} e^{-0.5\|I_p - I_q\|_2^2/\eta}$ [14] where η is the average of $\|I_p - I_q\|^2$ over a 8-connected neighborhood and d_{pq} is the distance between pixels p and q in the image plane. We set $w_{pq} = \frac{1}{d_{pq}}$ for length regularization.

We compare kernel clustering term $E_A(S)$ in (1) with a standard model-fitting term (5) using histogram-based probability model, as is common in Grabcut approach [104,62]. We tried various bin size for spatial and depth channels.

With fixed width Gaussian kernel, the time complexity of the naive implementation of kernel bound evaluation in (59) is $O(|\Omega|^2)$. The bottleneck is the evaluation of $\mathcal{K}X_t$ and ${X_t}'\mathcal{K}X_t$ in derivative $\nabla\hat{e}(X_t)$ (54). In this case, we resort to fast approximate dense filtering method in [84], which takes $O(|\Omega|)$ time. Also notice that the time complexity of the approach in [84] grows exponentially with data dimension N. A better approach for high-dimensional dense filtering is proposed in [2], which is of time $O(|\Omega| \times N)$. We use [84] for low-dimensional color spaces.

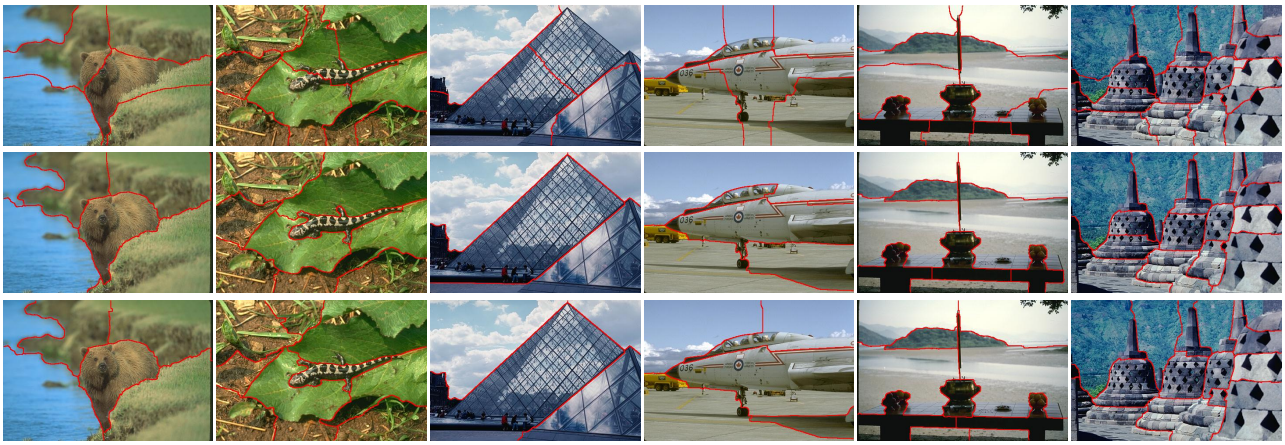


Fig. 15: Sample results on BSDS500. Top row: spectral clustering. Middle & Bottom rows: our Kernel & Spectral Cuts.

5.1 MRF helps Kernel & Spectral Clustering

Here we add MRF regulation terms to typical normalized cut applications, such as unsupervised multi-label segmentation [4] and image clustering [28]. Our kernel and spectral cuts are used to optimize the joint energy of normalized cut and MRF (1) or (73).

5.1.1 Normalized Cut with Potts Regularization

Spectral clustering [94] typically solves a (generalized) eigen problem, followed by simple clustering method such as K-means on the eigenvectors. However, it is known that such paradigm results in undesirable segmentation in large uniform regions [4,68], see examples in Fig. 15. Obviously such edge mis-alignment can be penalized by contrast-sensitive Potts term. Our spectral and kernel cuts get better segmentation boundaries. As is in [35] we use spectral initialization.

Tab.7 gives quantitative results on BSDS500 datasal. Number of ground truth segments is provided to each method. Kernel and spectral cuts give better covering, PRI (probabilistic rand index) and VOI (variation of information) than spectral clustering. Fig.15 gives sample results. Kernel K-means [35] gives results similar to spectral clustering and hence are not shown.

5.1.2 Normalized Cuts with Label Cost [33]

Unlike spectral clustering, our kernel and spectral cuts do not need the number of segments beforehand. We use kernel cut to optimize a combination of the normalized cut, Potts model and label costs terms. The label cost (4) penalizes each label by constant h_k. The energy is minimized by α-expansion and $\alpha\beta$-swap moves in Sec.3.3. We sample initial models from patches, as in [33]. Results with different label cost are shown in Fig.16. Due to sparsity prior, our kernel and spectral cuts automatically prune *weak* models and determine the number of segments, yet yield regularized segmentation. We use *KNN* affinity for normalized cut and mPb [4] based Potts regularization.

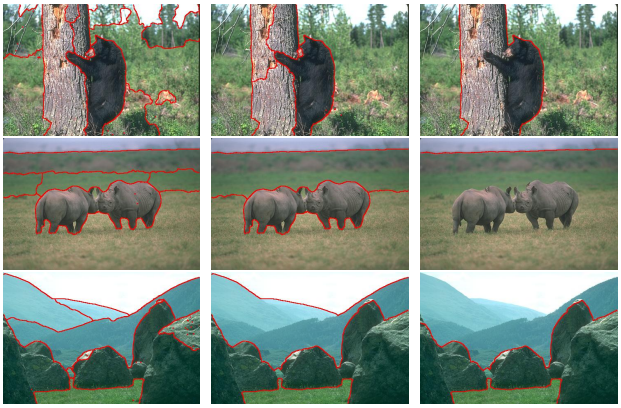


Fig. 16: Segmentation using our kernel cut with label cost. We experiment with increasing value of label cost h_k for each label (from left to right)

method	Covering	PRI	VOI
Spectral Clustering	0.34	0.76	2.76
Our Kernel Cut	0.41	**0.78**	2.44
Our Spectral Cut	**0.42**	**0.78**	**2.34**

Table 7: Results of spectral clustering (K-means on eigenvectors) and our Kernel Cut & Spectral Cuts on BSDS500 dataset. For this experiment mPb-based kernel is used [4].

5.1.3 Normalized Cut with High-Order Consistency

It is common that images come with multiple tags, such as those in Flickr platform or the LabelMe dataset [83]. We study how to utilize tag-based group prior for image clus-

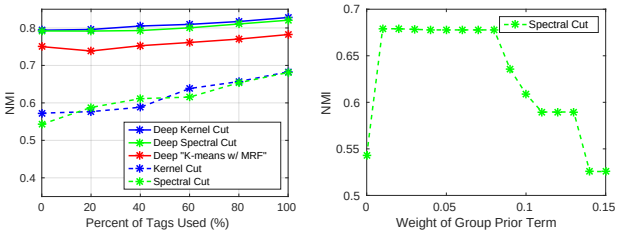


Fig. 17: Incorporating group prior achieves better NMI for image clustering. Here we use tags-based group prior. Our method achieved better NMI when more images are tagged. The right plot shows how the weight of bin consistency term affects our method.

tering [28] enforced as a high-order consistency potential common in MRF-based image segmentation [54,85,98].

We experiment on the LabelMe dataset [83] which contains 2,600 images of 8 scene categories (coast, mountain, forest, open country, street, inside city, tall buildings and highways). We use the same GIST feature, affinity matrix and group prior as used in [28]. We found the group prior to be noisy. The dominant category in each group occupies only 60%-90% of the group. The high-order consistency term is defined on each group. For each group, we introduce an energy term that is akin to the *robust* P^n-Potts [54], which can be exactly minimized within a single $\alpha\beta$-swap or α-expansion move. Notice that here we have to use robust consistency potential instead of rigid ones.

Our kernel cut minimizes NC plus the *robust* P^n-Potts term. Spectral cut minimizes energy of (72). Normalized mutual information (NMI) is used as the measure of clustering quality. Perfect clustering with respect to ground truth has NMI value of 1.

Spectral clustering and kernel K-means [35] give NMI value of 0.542 and 0.572 respectively. Our kernel cut and spectral cut significantly boost the NMI to 0.683 and 0.681. Fig. 17 shows the results with respect to different amount of image tags used. The left most points correspond to the case when no group prior is given. We optimize over the weight of high order consistency term, see Fig.17. Note that it's not the case the larger the weight the better since the grouping prior is noisy.

We also utilize deep features, which are 4096 dimensional fc7 layer from AlexNet [59]. We either run plain K-means, or construct a KNN kernel on deep features. These algorithms are denoted as deep K-means, deep spectral cut or deep kernel cut in Fig. 17. Incorporating group prior indeed improved clustering. The best NMI of 0.83 is achieved by our kernel cut and spectral cut for KNN kernel on deep features.

5.2 Kernel & Spectral Clustering helps MRF

In typical MRF applications we replace the log-likelihood terms by average association or normalized cut. We evaluate our Kernel Cut (fixed width kernel or KNN) in the context of interactive segmentation, and compare with the commonly used GrabCut algorithm [90]. In Sec. 5.2.1, we show that our kernel cut is less sensitive to choice of regularization weight γ. We further report results on the GrabCut dataset of 50 images and the Berkeley dataset in Sec. 5.2.2. We experiment with both (i) contrast-sensitive edge regularization, (ii) length regularization and (iii) color clustering (i.e., no regularization) so as to assess to what extent the algorithms benefit from regularization.

From Sec. 5.2.3 to Sec. 5.2.6, we also report segmentation results of our kernel cut with high-dimensional features I_p, including location, texture, depth, and motion respectively.

5.2.1 Robustness to regularization weight

We first run all algorithms without smoothness. Then, we experiment with several values of γ for the contrast-sensitive edge term. In the experiments of Fig. 18 (a) and (b), we used the yellow boxes as initialization. For a clear interpretation of the results, we did not use any additional hard constraint. In Fig. 18, "KernelCut-KNN-AA" means Kernel Cut with KNN kernel for average association (AA). Without smoothness, our Kernel Cut yields much better results than Grab Cut. Regularization significantly benefited the latter, as the decreasing blue curve in (a) indicates. For instance, in the case of the zebra image, model fitting yields a plausible segmentation when assisted with a strong regularization. However, in the presence of noisy edges and clutter, as is the case of the chair image in (b), regularization does not help as much. Note that for small regularization weights γ our method is substantially better than model fitting. Also, our method is less dependent on regularization weight and does not require fine tuning of γ.

5.2.2 Segmentation on GrabCut & Berkeley datasets.

First, we report results on the GrabCut database (50 images) using the bounding boxes provided in [63]. For each image the error is the percentage of mis-labeled pixels. We compute the average error over the dataset.

We experiment with four variants of our Kernel Cut, depending on whether to use fixed width Gaussian kernel or KNN kernel, and also the choice of normalized cut or average association term. We test different smoothness weights and plot the error curves[12] in Fig.19. Table 8 reports the best

[12] The smoothness weights for different energies are not directly comparable; Fig. 19 shows all the curves for better visualization.

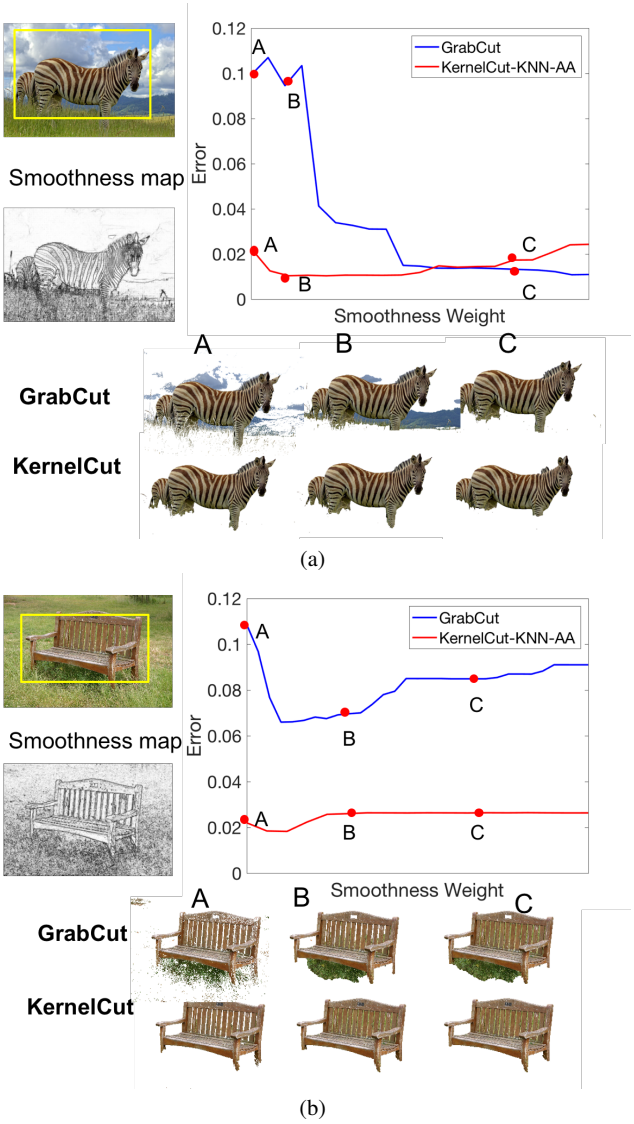


Fig. 18: Illustration of robustness to smoothness weight.

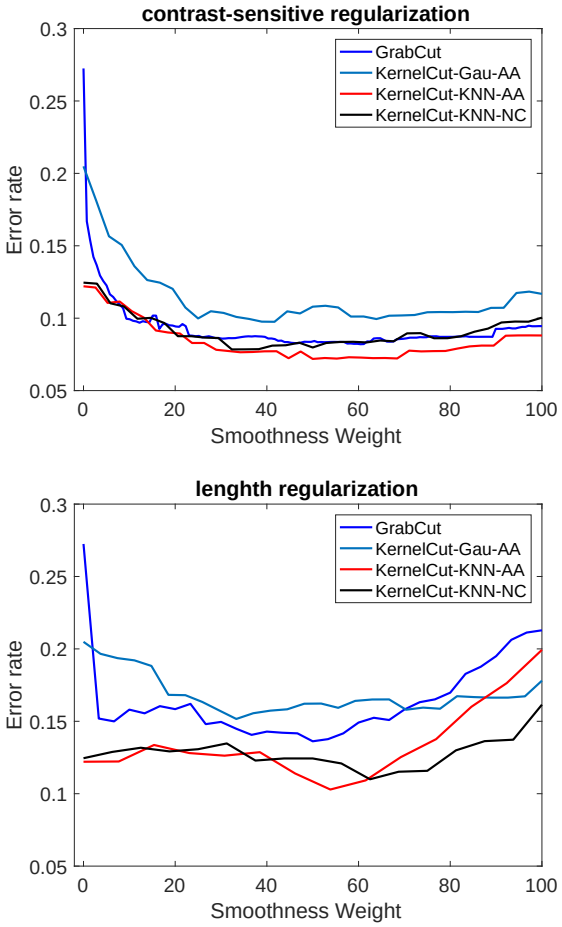


Fig. 19: Average error vs. regularization weights for different variants of our KernelCut on the GrabCut dataset.

boundary smoothness	color clustering term			
	GrabCut	KernelCut -Gau-AA	KernelCut -Gau-NC	KernelCut -KNN-AA
none	27.2	20.4	17.6	**12.2**
Euclidean length	13.6	15.1	16.0	**10.2**
contrast-sensitive	8.2	9.7	13.8	**7.1**

Table 8: Box-based interactive segmentation (Fig.21). Error rates (%) are averaged over 50 images in GrabCut dataset. KernelCut-Gau-NC means KernelCut for fixed width Gaussian kernel based normalized cut objective.

error for each method. For contrast-sensitive regularization GrabCut gets good results (8.2%). However, without edges (Euclidean or no regularization) GrabCut gives much higher errors (13.6% and 27.2%). In contrast, KernelCut-KNN-AA (Kernel Cut with adaptive KNN kernel for AA) gets only 12.2% doing a better job in color clustering without any help from the edges. In case of contrast-sensitive regularization, our method outperformed GrabCut (7.1% vs. 8.2%) but both methods benefit from strong edges in the GrabCut dataset. Fig .20 shows that our Kernel Cut is also robust to the hyper-parameter, i.e. K for nearest neighbours, unlike GrabCut.

Figure 21 gives some results. The top row shows a failure case for GrabCut where the solution aligns with strong edges. The second row shows a challenging image where our KernelCut-KNN-AA works well. The third and fourth rows show failure cases for Kernel Cut with fixed-width Gaussian kernel due to Breiman's bias [69] separating uniform color segments; see green bush and black suit. Adaptive kernel (KNN) addresses this bias.

We also tested seeds-based segmentation on a different database [71] with ground truth, see Tab.9 and Fig.22.

5.2.3 Segmentation of similar appearance objects

Even though objects may have similar appearances or look similar to the background (*e.g.* the top row in Fig.24), we

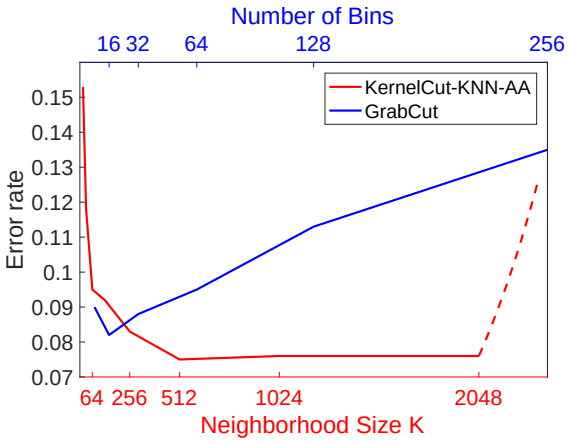


Fig. 20: Our method aKKM is robust to choice of K while GrabCut is sensitive to bin size for histograms.

assume that the objects of interest are compact and have different locations. This assumption motivates using XY coordinates of pixels as extra features for distinguishing similar or camouflaged objects. XY features have also been used in [81] to build space-variant color distribution. However, such distribution used in MRF-MAP inference [81] would still over-fit the data [97]. Let $I_p \in \mathcal{R}^5$ be the augmented color-location features $I_p = [l_p, a_p, b_p, \beta x_p, \beta y_p]$ at pixel p where $[l_p, a_p, b_p]$ is its color, $[x_p, y_p]$ are its image coordinates, and β is a scaling parameter. Note that the edge-based Potts model [14] also uses the XY information. Location features in the clustering and regularization terms have complementary effect: the former solves appearance camouflage while the latter gets edge alignment.

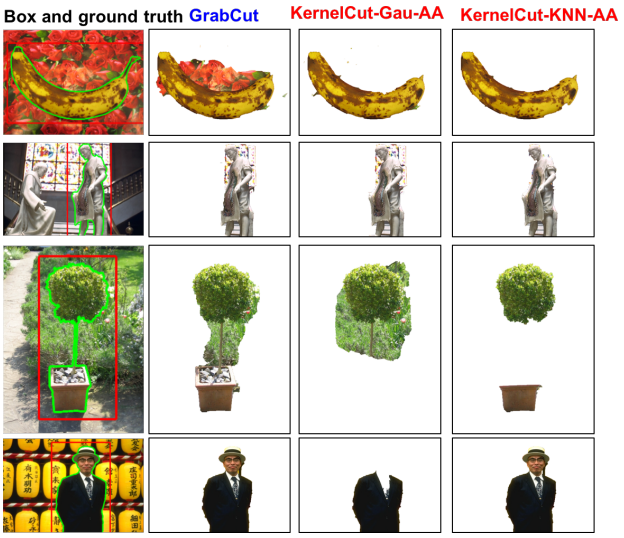


Fig. 21: Sample results for GrabCut and our kernel cut with fixed width Gaussian or adaptive width KNN kernel, see Tab.8.

boundary smoothness	color clustering term		
	BJ	GrabCut	KernelCut -KNN-AA
none	12.4	12.4	**7.6**
contrast-sensitive	3.2	3.7	**2.8**

Table 9: Seeds-based interactive segmentation (Fig.22). Error rates (%) are averaged over 82 images from Berkeley database. Methods get the same seeds entered by four users. We removed 18 images with multiple nearly-identical objects (see Fig.24) from 100 image subset in [71]. (GrabCut and KernelCut-KNN-AA give 3.8 and 3.0 errors on the whole database.)

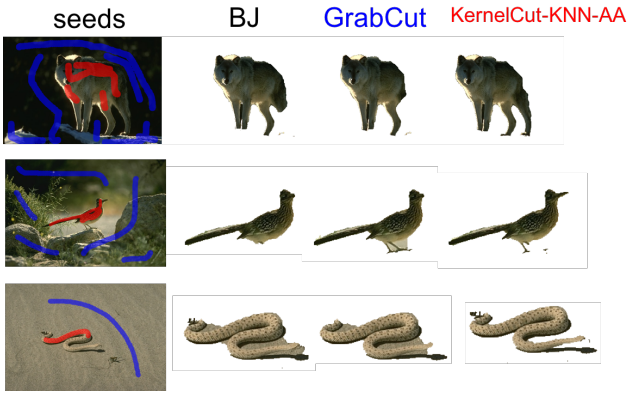
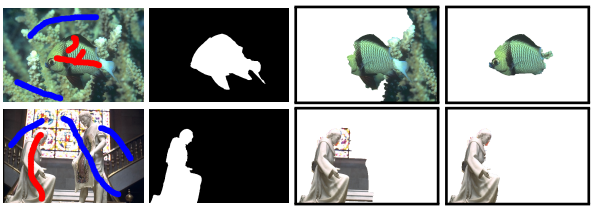


Fig. 22: Sample results for BJ [14], GrabCut [90], and our kernel cut for adaptive KNN kernel, see Tab.9.

Fig. 23: Visualization of a pixel's K-Nearest-Neighbours for RGB feature (left) or RGBXY feature (right).

We test the effect of adding XY into feature space for GrabCut and Kernel Cut. We try various β for Kernel Cut. Fig.23 shows the effect of different β on KNNs of a pixel. For histogram-based GrabCut we change spatial bin size for the XY channel, ranging from 30 pixels to the image size. We report quantitative results on 18 images with similar objects and camouflage from the Berkeley database [70]. Seeds are used here. Fig. 25 shows average errors for multi-object dataset, see example segmentations in Fig. 24.

Fig. 26 gives multi-label segmentation of similar objects in one image with seeds using our algorithm. We optimize kernel bound with move-making for NC and smoothness



(a) seeds (b) ground truth (c) GrabCut (d)Kernel Cut

Fig. 24: Sample results using RGBXY+XY.

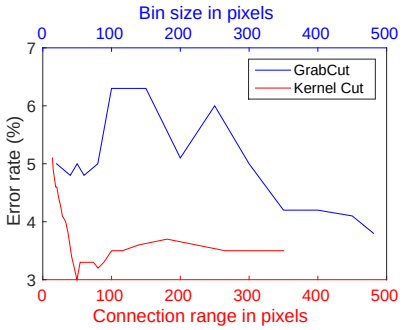
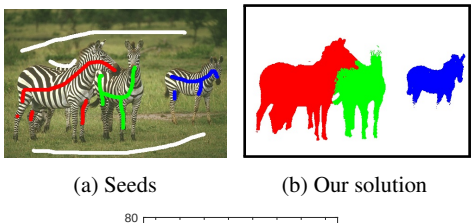
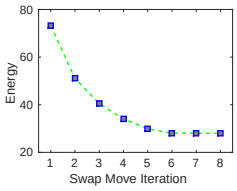


Fig. 25: Error on Multi-objects dataset. We vary spatial bin-size for GrabCut and weight β in $[l, a, b, \beta X, \beta Y]$ for Kernel Cut. The connection range is the average geometric distance between a pixel and its k^{th} nearest neighbor. **The right-most point of the curves corresponds to the absence of XY features.** GrabCut does not benefit from XY features. Kernel Cut achieves the best error rate of 2.9% with connection range of 50 pixels.



(a) Seeds (b) Our solution



(c) Energy minimization for NC plus smoothness

Fig. 26: Multi-label segmentation for similar objects.

term combination, as discussed in Sec. 3.2. Fig. 26 (c) shows energy convergence.

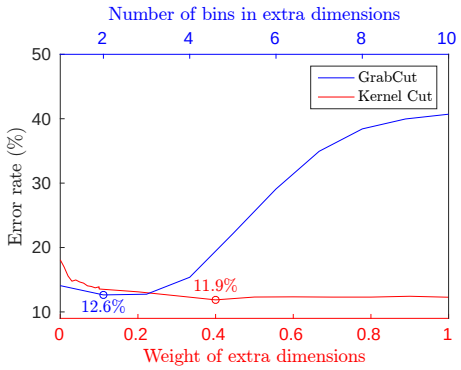


Fig. 27: The average errors of GrabCut and Kernel Cut methods for texture segmentation over 50 desaturated images from GrabCut database [90]. We optimize GrabCut with respect to smoothness weight and bin sizes in the intensity dimension. We optimize the result of Kernel Cut with respect to smoothness weight.

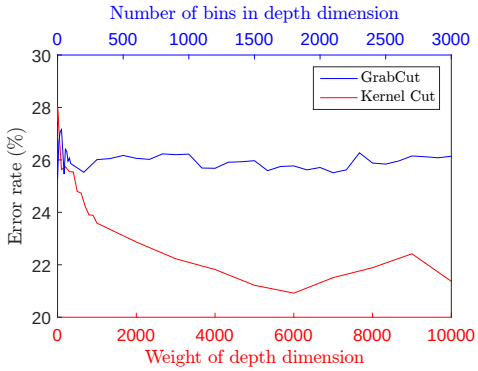


Fig. 28: The average errors of GrabCut and Kernel Cut methods over 64 images selected from NYUv2 database [78].

5.2.4 Texture segmentation

The goal of this experiment is to demonstrate scalability of our methods to highly dimensional data. First, desaturated images from GrabCut database [90] are convolved with 48 filters from [102]. This yields a 48-dimensional descriptor for each pixel. Secondly, these descriptors are clustered into 32 textons by K-means. Thirdly, for each pixel we build a 32-dimensional normalized histogram of textons in 5×5 vicinity of the pixel. Then the gray-scale intensity[13] of a pixel is augmented by the corresponding texton histogram scaled by a factor w. Finally, resulting 33-dimensional feature vectors are used for segmentation. We show the result of Kernel Cut with respect to w in Fig.27. We compare our

[13] We found that for the GrabCut database adding texture features to RGB does not improve the results.

results with GrabCut with various bin sizes for texture features.

5.2.5 Interactive RGBD Images Segmentation

Depth sensor are widely used in vision for 3D modelling [37, 79], semantic segmentation [34,45,78,87], motion flow [44]. We selected 64 indoor RGBD images from semantic segmentation database NYUv2 [78] and provided bounding boxes and ground truth. In contrast to [90], the prepared dataset consists of low-quality images: there are camera motion artifacts, underexposed and overexposed regions. Such artifacts make color-based segmentation harder.

We compare GrabCut to Kernel Cut over joint features $I_p = [L_p, a_p, b_p, \beta D_p]$ as in Sec.5.2.3. Figs. 28 and 29 show the error statistics and segmentation examples. While Kernel Cut takes advantage of the additional channel, GrabCut fails to improve.

5.2.6 Motion segmentation

Besides the location and depth features, we also test segmentation with motion features. Figs. 30, 31 and 32 compare motion segmentations using different feature spaces: RGB, XY, M (optical flow) and their combinations (RGBM or RGBXY or RGBXYM). Abbreviation **+XY** means Potts regularization. We apply kernel cut (Alg.1) to the combination of NC with the Potts term.

Challenging video examples: For videos in FBMS-59 dataset [19], our algorithm runs on individual frames instead of 3D volume. Segmentation of previous frame initializes the next frame. The strokes are provided *only* for the first frame. We use the optical flow algorithm in [20] to generate M features. Selected frames are shown in Figs. 30 and 31. Instead of tracks from all frames in [82], our segmentation of each frame uses only motion estimation between two consecutive frames. Our approach jointly optimizes normalized cut and Potts model. In contrast, [82] first clusters semi-dense tracks via spectral clustering [19] and then obtains dense segmentation via regularization.

Kitti segmentation example: We also experiment with Kitti dataset [72]. Fig.32 shows the multi-label segmentation using either color information RGB+XY (first row) or motion MXY+XY (second row). The ground-truth motion field works as M channel. Note that the motion field is known only for approximately 20% of the pixels. To build an affinity graph, we construct a *KNN* graph from pixels that have motion information. The regularization over 8-neighborhood on the pixel grid interpolates the segmentation labels during the optimization procedure.

Acknowledgements We greatly thank Carl Olsson (Lund University, Sweden) for hours of stimulating discussions, as well as for detailed feedback and valuable recommendations at different stages of our work. We appreciate his tremendous patience when our thoughts were much more confusing than they might be now. Ivan Stelmakh (PhysTech, Russia) also gave helpful feedback on our draft and caught several errors. Anders Eriksson (Lund University, Sweden) helped with related work on NC with constraints. We also thank Jianbo Shi (UPenn, USA) for his excellent spectral-relaxation optimization code for NC.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Ssstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(11), 2274–2282 (2012) 1
2. Adams, A., Baek, J., Davis, M.A.: Fast high-dimensional filtering using the permutohedral lattice. Computer Graphics Forum **29**(2), 753–762 (2010) 23
3. Aggarwal, C.C., Reddy, C.K. (eds.): Data Clustering: Algorithms and Applications. Chapman & Hall / CRC (2014) 1
4. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33**(5), 898–916 (2011) 4, 21, 22, 23, 24
5. Ayed, I.B., Mitiche, A., Belhadj, Z.: Multiregion level set partitioning of synthetic aperture radar images. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **27**(5), 793—800 (2005) 5, 7, 8
6. Bach, F., Jordan, M.: Learning spectral clustering. Advances in Neural Information Processing Systems **16**, 305–312 (2003) 1, 12, 17, 20, 22, 23
7. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming: Theory and Algorithms. Wiley (2006) 15
8. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation **15**(6), 1373–1396 (2003) 8, 17, 22
9. Belongie, S., Malik, J.: Finding boundaries in natural images: A new method using point descriptors and area completion. In: Proceedings of the European Conference on Computer Vision (ECCV) (1998) 23
10. Ben Ayed, I., Gorelick, L., Boykov, Y.: Auxiliary cuts for general classes of higher order functionals. In: IEEE conference on Computer Vision and Pattern Recognition (CVPR), pp. 1304–1311. Portland, Oregon (2013). URL http://www.csd.uwo.ca/~yuri/Abstracts/cvpr13-auxcut-abs.shtml 13
11. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006) 23
12. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004) 15
13. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient N-D image segmentation. International Journal of Computer Vision (IJCV) **70**(2), 109–131 (2006) 5
14. Boykov, Y., Jolly, M.P.: *Interactive graph cuts* for optimal boundary & region segmentation of objects in N-D images. In: ICCV, vol. I, pp. 105–112 (2001) 1, 4, 5, 23, 27
15. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: International Conference on Computer Vision, vol. I, pp. 26–33 (2003) 1, 4, 5
16. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE transactions on Pattern Analysis and Machine Intelligence **23**(11), 1222–1239 (2001) 3, 4, 5, 16
17. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. Pattern Analysis and Machine Intelligence, IEEE Transactions on **23**(11), 1222–1239 (2001) 22
18. Breiman, L.: Technical note: Some properties of splitting criteria. Machine Learning **24**(1), 41–47 (1996) 7, 8

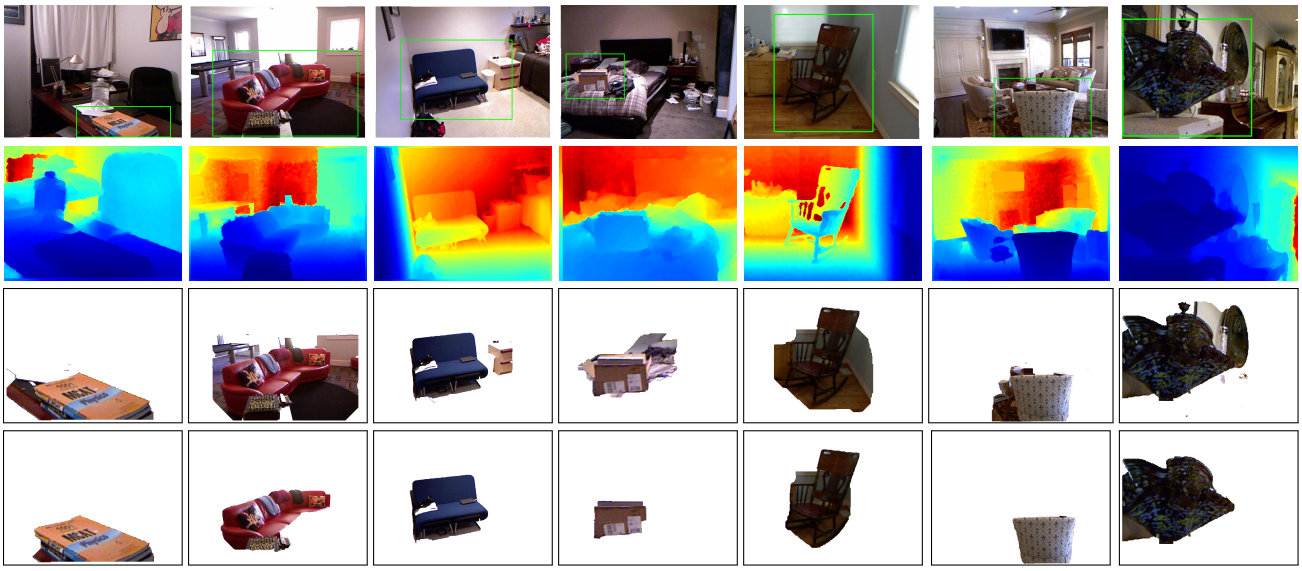
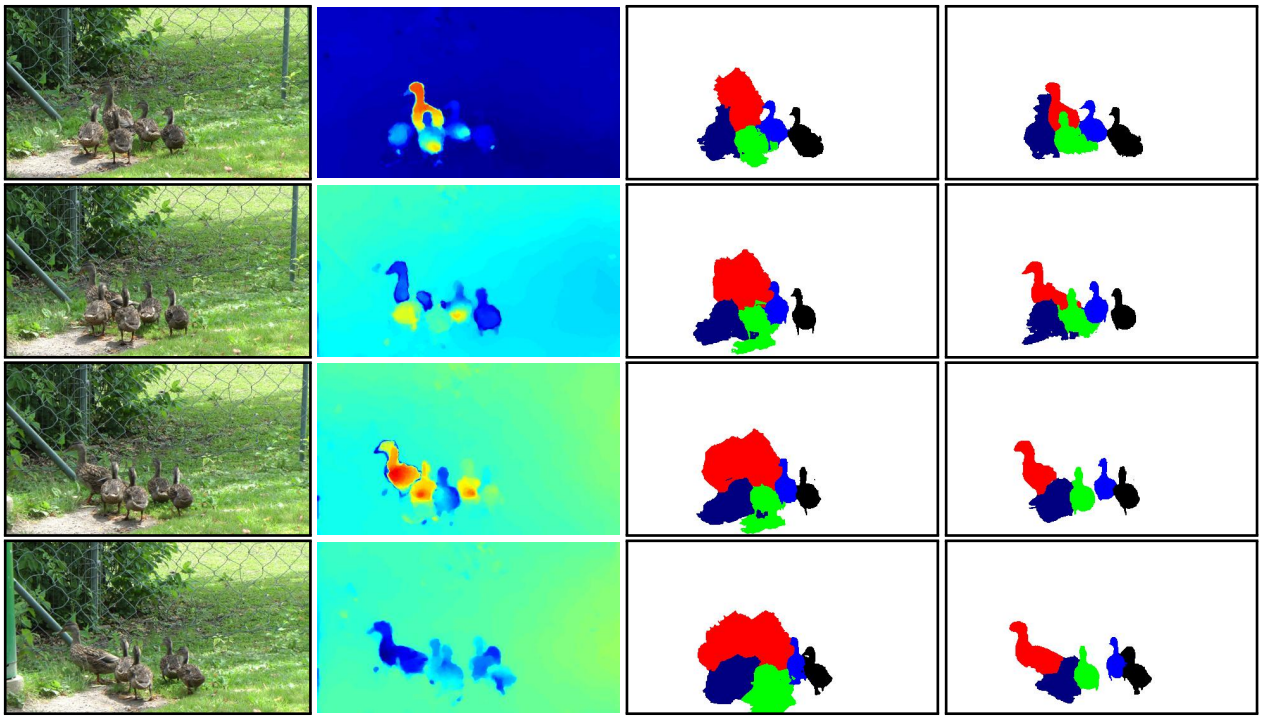


Fig. 29: RGBD+XY examples. The first two rows show original images wit bounding box and color-coded depth channel. The third row shows the results of Grabcut, the forth row shows the results of Kernel Cut. The parameters of the methods were independently selected to minimize their average error rates over the database.

19. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: Computer Vision–ECCV 2010, pp. 282–295. Springer (2010) 29, 31
20. Brox, T., Malik, J.: Large displacement optical flow: descriptor matching in variational motion estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(3), 500–513 (2011) 23, 29, 31
21. Carreira-Perpinan, M.A., Wang, W.: The K-Modes Algorithm for Clustering. In: arXiv:1304.6478v1 [cs.LG] (2013) 6
22. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. International Journal of Computer Vision **22**(1), 61–79 (1997) 1, 5
23. Chambolle, A.: An algorithm for total variation minimization and applications. Journal of Mathematical imaging and vision **20**(1-2), 89–97 (2004) 3, 4, 5, 16
24. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision **40**(1), 120–145 (2011) 3, 4, 5, 16
25. Chan, T., Vese, L.: Active contours without edges. IEEE Trans. Image Processing **10**(2), 266–277 (2001) 4, 5
26. Chew, S.E., Cahill, N.D.: Semi-supervised normalized cuts for image segmentation. In: The IEEE International Conference on Computer Vision (ICCV) (2015) 4, 13
27. Chitta, R., Jin, R., Havens, T., Jain, A.: Scalable kernel clustering: Approximate kernel k-means. In: KDD, pp. 895–903 (2011) 7
28. Collins, M.D., Liu, J., Xu, J., Mukherjee, L., Singh, V.: Spectral clustering with a convex regularizer on millions of images. In: Computer Vision–ECCV 2014, pp. 282–298. Springer (2014) 24, 25
29. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **24**(5), 603–619 (2002) 8, 11
30. Cox, I., Rao, S., Zhong, Y.: "Ratio Regions": A Technique for Image Segmentation. In: International Conference on Pattern Recognition (ICPR), pp. 557–564 (1996) 11
31. Cox, T., Cox, M.: Multidimensional scaling. CRC Press (2000) 17, 18
32. Cremers, D., Rousson, M., Deriche, R.: A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. International journal of computer vision **72**(2), 195–215 (2007) 3, 4, 5, 16
33. Delong, A., Osokin, A., Isack, H., Boykov, Y.: Fast Approximate Energy Minization with Label Costs. Int. J. of Computer Vision (IJCV) **96**(1), 1–27 (2012) 1, 2, 3, 4, 5, 6, 7, 8, 16, 24
34. Deng, Z., Todorovic, S., Latecki, L.J.: Semantic segmentation of rgbd images with mutex constraints. In: International Conference on Computer Vision (ICCV). Santiago, Chile (2015) 29
35. Dhillon, I., Guan, Y., Kulis, B.: Kernel k-means, spectral clustering and normalized cuts. In: KDD (2004) 1, 7, 8, 12, 17, 19, 20, 22, 23, 24, 25
36. Dhillon, I., Guan, Y., Kulis, B.: Weighted graph cuts without eigenvectors: A multilevel approach. IEEE Transactions on Pattern Analysis and Machine Learning (PAMI) **29**(11), 1944–1957 (2007) 12, 20, 23
37. Dou, M., Taylor, J., Fuchs, H., Fitzgibbon, A., Izadi, S.: 3d scanning deformable objects with a single rgbd sensor. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 493–501 (2015) 29
38. Duda, R., Hart, P., Stork, D.: Pattern classification. John Wiley & Sons (2001) 1, 12
39. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons (2001) 5, 8, 10
40. Eriksson, A., Olsson, C., Kahl, F.: Normalized cuts revisited: A reformulation for segmentation with linear grouping constraints. Journal of Mathematical Imaging and Vision **39**(1), 45–61 (2011) 4, 13
41. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE transactions on Pattern Analysis and Machine Intelligence **6**, 721–741 (1984) 1, 3
42. Girolami, M.: Mercer kernel-based clustering in feature space. IEEE Trans. Neural Networks **13**(3), 780–784 (2002) 7
43. Golub, G.H., Van Loan, C.F.: Matrix computations, vol. 3. JHU Press (2012) 18

(a) frames (b) optical flow [20] (c) **M+XY** (d) **RGB+XY** (e) **RGBM+XY**

Fig. 30: Motion segmentation using our framework for the sequence *horses01* in FBMS-59 dataset [19]. Motion feature alone (**M+XY** in (c)) is not sufficient to obtain fine segmentation. Our framework successfully utilize motion feature (optical flow) to separate the horse from the barn, which have similar appearances. See supplementary material for results on the video.



(a) frames (b) optical flow [20] (c) **RGBXY+XY** (d) **RGBXYM+XY**

Fig. 31: Multi-label motion segmentation using our framework for the sequence *ducks01* in FBMS-59 dataset [19]. This video is challenging since the ducks have similar appearances and even spatially overlap with each other. However, different ducks come with different motions, which helps our framework to better separate individual ducks. See supplementary materials.

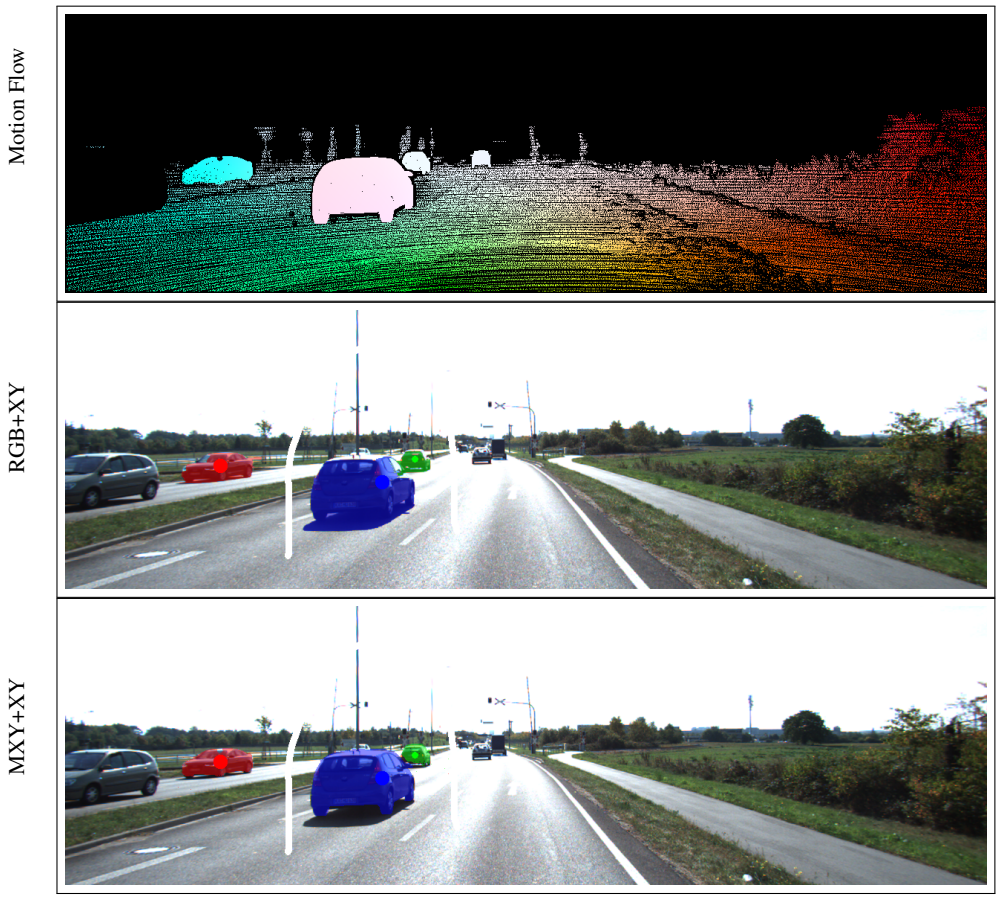


Fig. 32: Motion segmentation for image 000079_10 from *KITTI* [72] dataset. The first row shows the motion flow. Black color codes the pixels that do not have motion information. The second row shows color-based segmentation. The third row shows motion based segmentation with location features. We also tried M+XY segmentation, but it does not work as well as MXY+XY above. The results for RGBMXY+XY were not significantly different from MXY+XY.

44. Gottfried, J.M., Fehr, J., Garbe, C.S.: Computing range flow from multi-modal kinect data. In: Advances in Visual Computing, pp. 758–767. Springer (2011) 29
45. Gulshan, V., Lempitsky, V., Zisserman, A.: Humanising grabcut: Learning to segment humans using the kinect. In: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, pp. 1127–1133. IEEE (2011) 29
46. Hein, M., Lal, T.N., Bousquet, O.: Hilbertian metrics on probability measures and their application in svm's. Pattern Recognition **LNCS 3175**, 270—277 (2004) 8
47. Hochbaum, D.S.: Polynomial time algorithms for ratio regions and a variant of normalized cut. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(5), 889–898 (2010) 11
48. Hofmann, T., Buhmann, J.: Pairwise data clustering by deterministic annealing. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **19**(1), 1–14 (1997) 1, 12, 13, 16
49. Ishikawa, H.: Exact optimization for Markov Random Fields with convex priors. IEEE transactions on Pattern Analysis and Machine Intelligence **25**(10), 1333–1336 (2003) 3, 4, 16
50. Jayasumana, S., Hartley, R., Salzmann, M., Li, H., Harandi, M.: Kernel methods on riemannian manifolds with gaussian rbf kernels. IEEE Trans. Pattern Anal. Mach. Intell. **In press** (2015) 7
51. Jermyn, I., Ishikawa, H.: Globally optimal regions and boundaries as minimum ratio weight cycles. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **23**(10), 1075–1088 (2001) 11
52. Kappes, J.H., Andres, B., Hamprecht, F.A., Schnörr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B.X., Kröger, T., Lellmann, J., et al.: A comparative study of modern inference techniques for structured discrete energy minimization problems. International Journal of Computer Vision **115**(2), 155–184 (2015) 5
53. Kearns, M., Mansour, Y., Ng, A.: An Information-Theoretic Analysis of Hard and Soft Assignment Methods for Clustering. In: Conf. on Uncertainty in Artificial Intelligence (UAI) (1997) 4, 5, 6, 7, 8, 14
54. Kohli, P., Torr, P.H., et al.: Robust higher order potentials for enforcing label consistency. International Journal of Computer Vision **82**(3), 302–324 (2009) 1, 3, 4, 5, 16, 25
55. Kolmogorov, V.: Convergent Tree-Reweighted Message Passing for Energy Minimization. IEEE transactions on Pattern Analysis and Machine Intelligence **28**(10), 1568–1583 (2006) 3, 4, 16
56. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. Pattern Analysis and Machine Intelligence, IEEE Transactions on **28**(10), 1568–1583 (2006) 5
57. Kolmogorov, V., Boykov, Y., Rother, C.: Applications of parametric maxflow in computer vision. In: IEEE International Conference on Computer Vision (ICCV) (2007) 11

58. Krahenbuhl, P., Koltun, V.: Efficient inference in fully connected CRFs with Gaussian edge potentials. In: NIPS (2011) 4
59. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105 (2012) 25
60. Kulis, B., Basu, S., Dhillon, I., Mooney, R.: Semi-supervised graph clustering: a kernel approach. Machine Learning **74**(1), 1–22 (2009) 1, 4, 13
61. Lange, K., Hunter, D.R., Yang, I.: Optimization transfer using surrogate objective functions. Journal of Computational and Graphical Statistics **9**(1), 1–20 (2000) 13
62. Lempitsky, V., Blake, A., Rother, C.: Image segmentation by branch-and-mincut. In: ECCV (2008) 23
63. Lempitsky, V., Kohli, P., Rother, C., Sharp, T.: Image segmentation with a bounding box prior. In: Int. Conference on Computer Vision (ICCV), pp. 277–284 (2009) 25
64. Li, S.: Markov Random Field Modeling in Image Analysis, 3nd edn. Springer-Verlag (2009) 3
65. Li, T., Ma, S., Ogihara, M.: Entropy-based criterion in categorical clustering. In: Int. Conf. on M. Learning (2004) 7
66. Louppe, G., Wehenkel, L., Sutera, A., Geurts, P.: Understanding variable importances in forests of randomized trees. In: NIPS, pp. 431–439 (2013) 7
67. MacKay, D.J.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press (2003) 12
68. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. International journal of computer vision **43**(1), 7–27 (2001) 4, 23, 24
69. Marin, D., Tang, M., Ayed, I.B., Boykov, Y.: Kernel clustering: density biases and solutions (2018). DOI 10.1109/TPAMI.2017.2780166 8, 11, 12, 23, 26
70. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, vol. 2, pp. 416–423. IEEE (2001) 27
71. McGuinness, K., Oconnor, N.E.: A comparative evaluation of interactive segmentation algorithms. Pattern Recognition **43**(2), 434–444 (2010) 26, 27
72. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Conference on Computer Vision and PatternRecognition (CVPR) (2015) 29, 32
73. Mitiche, A., Ayed, I.B.: Variational and Level Set Methods in Image Segmentation. Springer (2010) 7
74. Muller, K., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B.: An introduction to kernel-based learning algorithms. IEEE Trans. Neural Networks **12**(2), 181–201 (2001) 7
75. Müller, K.R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks **12**(2), 181–201 (2001) 9
76. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Comm. Pure Appl. Math. **42**, 577–685 (1989) 1
77. Narasimhan, M., Bilmes, J.A.: A submodular-supermodular procedure with applications to discriminative structure learning. In: UAI, pp. 404–412 (2005) 13
78. Nathan Silberman Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012) 28, 29
79. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on, pp. 127–136. IEEE (2011) 29
80. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Advances in neural information processing systems (NIPS), vol. 2, pp. 849–856 (2002) 17, 22
81. Nieuwenhuis, C., Cremers, D.: Spatially varying color distributions for interactive multilabel segmentation. IEEE transactions on pattern analysis and machine intelligence **35**(5), 1234–1247 (2013) 27
82. Ochs, P., Brox, T.: Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In: Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 1583–1590. IEEE (2011) 29
83. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International journal of computer vision **42**(3), 145–175 (2001) 24, 25
84. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. In: Computer Vision–ECCV 2006, pp. 568–580. Springer (2006) 23
85. Park, K., Gould, S.: On learning higher-order consistency potentials for multi-class pixel labeling. In: ECCV (2012) 5, 25
86. Pock, T., Chambolle, A., Cremers, D., Bischof, H.: A convex relaxation approach for computing minimal partitions. In: IEEE conference on Computer Vision and Pattern Recognition (CVPR) (2009) 1
87. Ren, X., Bo, L., Fox, D.: Rgb-(d) scene labeling: Features and algorithms. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 2759–2766. IEEE (2012) 29
88. Rose, K.: Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. Proceedings of the IEEE **86**(11), 2210–2239 (1998) 12
89. Roth, V., Laub, J., Kawanabe, M., Buhmann, J.: Optimal cluster preserving embedding of nonmetric proximity data. IEEE Trans. Pattern Anal. Mach. Intell. **25**(12), 1540—1551 (2003) 8, 9, 10, 12, 15, 17, 19, 20
90. Rother, C., Kolmogorov, V., Blake, A.: Grabcut - interactive foreground extraction using iterated graph cuts. In: ACM trans. on Graphics (SIGGRAPH) (2004) 1, 2, 4, 5, 7, 8, 23, 25, 27, 28, 29
91. Rousson, M., R., D.: A variational framework for active and adaptative segmentation of vector valued images. In: Workshop on Motion and Video Computing (2002) 6, 7, 8
92. Salah, M.B., Mitiche, A., Ayed, I.B.: Effective level set image segmentation with a kernel induced data term. IEEE Transactions on Image Processing **19**(1), 220–232 (2010) 6, 8, 11
93. Shawe-Tayler, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004) 9
94. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**, 888–905 (2000) 1, 2, 3, 4, 7, 8, 11, 12, 17, 20, 22, 23, 24
95. Sung, K.K., Poggio, T.: Example based learning for viewbased human face detection. IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI) **20**, 39–51 (1995) 6, 7, 8
96. Tang, M., Ayed, I.B., Boykov, Y.: Pseudo-bound optimization for binary energies. In: European Conference on Computer Vision (ECCV), pp. 691–707 (2014) 7, 13, 14, 22
97. Tang, M., Ayed, I.B., Marin, D., Boykov, Y.: Secrets of grabcut and kernel k-means. In: International Conference on Computer Vision (ICCV). Santiago, Chile (2015) 4, 8, 14, 27
98. Tang, M., Gorelick, L., Veksler, O., Boykov, Y.: Grabcut in one cut. In: International Conference on Computer Vision (ICCV). Sydney, Australia (2013) 4, 5, 7, 25
99. Tang, M., Marin, D., Ayed, I.B., Boykov, Y.: Kernel Cuts: MRF meets kernel and spectral clustering. In: arXiv:1506.07439 (2016) 6, 7, 10, 11, 12, 14, 22
100. Tang, M., Marin, D., Ayed, I.B., Boykov, Y.: Normalized Cut meets MRF. In: European Conference on Computer Vision (ECCV). Amsterdam, Netherlands (2016) 4
101. Vapnik, V.: Statistical Learning Theory. Wiley (1998) 7
102. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. International Journal of Computer Vision **62**(1-2), 61–81 (2005) 28

103. Veksler, O.: Efficient graph cut optimisation for full crfs with quantized edges. In: arXiv:1809.04995 (2018) 4
104. Vicente, S., Kolmogorov, V., Rother, C.: Joint optimization of segmentation and appearance models. In: International Conf. on Computer Vision (ICCV) (2009) 23
105. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and computing **17**(4), 395–416 (2007) 1, 20, 22
106. Wang, S., Siskind, J.M.: Image segmentation with ratio cut. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **25**(6), 675–690 (2003) 11
107. Werner, T.: A linear programming approach to max-sum problem: A review. Pattern Analysis and Machine Intelligence, IEEE Transactions on **29**(7), 1165–1179 (2007) 3, 4, 5, 16
108. Xu, L., Li, W., Schuurmans, D.: Fast normalized cut with linear constraints. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 2866–2873 (2009) 13
109. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Constructing free-energy approximations and generalized belief propagation algorithms. IEEE Transactions on Information Theory **51**(7), 2282–2312 (2005) 5
110. Yu, S., Shi, J.: Multiclass spectral clustering. In: International Conference on Computer Vision (ICCV) (2003) 12, 22, 23
111. Yu, S.X., Shi, J.: Segmentation given partial grouping constraints. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **26**(2), 173–183 (2004) 4, 13
112. Yu, Y., Fang, C., Liao, Z.: Piecewise flat embedding for image segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1368–1376 (2015) 8, 17, 22
113. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: Advances in NIPS, pp. 1601–1608 (2004) 11, 23
114. Zhu, S.C., Yuille, A.: Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence **18**(9), 884–900 (1996) 1, 2, 4, 5, 7, 8