Kernel Cuts: MRF meets Kernel & Spectral Clustering

Meng Tang∗ Ismail Ben Ayed† Dmitrii Marin∗ Yuri Boykov∗

∗Computer Science, University of Western Ontario, Canada
†École de Technologie Supérieure, University of Quebec, Canada
mtang73@csd.uwo.ca ismail.benayed@etsmtl.ca dmitrii.a.marin@gmail.com yuri@csd.uwo.ca

Abstract

The log-likelihood energy term in popular model-fitting segmentation methods, e.g. [104, 24, 85, 33], is presented as a generalized “probabilistic” K-means energy [54] for color space clustering. This interpretations reveals some limitations, e.g. over-fitting. We propose an alternative approach to color clustering using kernel K-means with well-known properties such as non-linear separation and scalability to higher-dimensional feature spaces. Similarly to log-likelihoods, our kernel energy term for color space clustering can be combined with image grid regularization, e.g. boundary smoothness, and minimized using bound-optimization and max-flow algorithm. In contrast to fitting histograms or GMMs [104, 85], at each step our algorithm uses Parzen densities for segment appearance. In contrast to implicit entropy minimization [33], our approach is related to Gini and average association criteria. Using Nash theorem our analysis suggests principled adaptive kernel selection strategies to counter Breiman bias in these criteria. Our general kernel-based approach improves sensitivity to color in synthetic and real images, and opens the door for many extensions and applications.

1. Introduction and Motivation

Many standard segmentation methods combine regularization in the image domain with a likelihood term integrating color appearance models [11, 104, 24, 13, 85]. These appearance models are often treated as variables. They can be estimated jointly with segmentation by minimizing energies like

\[-\sum_{i=0}^{K-1} \sum_{p \in S^i} \log P^i(I_p) + ||\partial S||\]  \hspace{1cm} (1)

where segmentation \(\{S^i\}\) is defined by integer variables \(S_p\) such that \(S^i = \{ p : S_p = i \}\), models \(P = \{P^i\}\) are probability distributions of a given class, and \(||\partial S||\) is the segmentation boundary length in Euclidean or some contrast sensitive image-weighted metric. This popular approach to unsupervised [104, 24] or supervised [85] seg-

Figure 1: Failure of iterative GMM fitting without image smoothness or hard constraints. Our proposed kernel K-means & adaptive kernel K-means find better color clusters.
A. basic K-means  
(e.g. [24])

\[
\sum_{p \in S} \|I_p - \mu_s\|^2 + \sum_{p \in S} \|I_p - \mu_k\|^2
= \frac{\sum_{p \in S} \|I_p - \mu_s\|^2}{2|S|} + \frac{\sum_{p \in \bar{S}} \|I_p - \mu_k\|^2}{2|S|} = |S| \cdot var(S) + |S| \cdot var(S)
\]

\[
\sum_{p \in S} \ln N(I_p | \mu_s) - \sum_{p \in \bar{S}} \ln N(I_p | \mu_k)
\]

B. probabilistic K-means  
(e.g. [104, 90, 86, 85, 33])

(i) equivalent energy formulations:

\[
\sum_{p \in S} \|I_p - \theta_s\|d + \sum_{p \in \bar{S}} \|I_p - \theta_k\|d
= - \sum_{p \in S} \ln P(h(I_p | \theta_s)) - \sum_{p \in \bar{S}} \ln P(h(I_p | \theta_k))
\approx |S| \cdot H(S | h) + |\bar{S}| \cdot H(\bar{S} | h)
\]

(ii) example: descriptive models (histograms or GMM) yield high-order log-likelihood energy

\[
- \sum_{p \in S} \ln \mathcal{P}_h(I_p | S) - \sum_{p \in \bar{S}} \ln \mathcal{P}_h(I_p | \bar{S})
= \frac{\sum_{p \in S} \|I_p - \theta_s\|^2}{2|S|} + \frac{\sum_{p \in \bar{S}} \|I_p - \theta_k\|^2}{2|\bar{S}|} = \frac{\sum_{p \in S} k(I_p | \theta_s)}{|S|} - \frac{\sum_{p \in \bar{S}} k(I_p | \theta_k)}{|\bar{S}|}
\]

(iii) bound optimization: auxiliary function at \(S\)

\[
A_1(S) = -\sum_{p \in S} \ln \mathcal{P}_h(I_p | S) - \sum_{p \in \bar{S}} \ln \mathcal{P}_h(I_p | \bar{S})
= |S| \cdot H(S | h) + |\bar{S}| \cdot H(\bar{S} | h)
\]

\[
C. kernel K-means  
(ours)
\]

(i) equivalent energy formulations:

\[
\sum_{p \in S} \| \phi(I_p) - \mu_s \|^2 + \sum_{p \in \bar{S}} \| \phi(I_p) - \mu_k \|^2
= \frac{\sum_{p \in S} \| \phi(I_p) - \mu_s \|^2}{2|S|} + \frac{\sum_{p \in \bar{S}} \| \phi(I_p) - \mu_k \|^2}{2|\bar{S}|} = \frac{\sum_{p \in S} k(I_p | \theta_s)}{|S|} - \frac{\sum_{p \in \bar{S}} k(I_p | \theta_k)}{|\bar{S}|}
\]

(ii) example: normalized kernels (Gaussians) yield high-order Parzen density energy

\[
- \sum_{p \in S} \mathcal{P}_k(I_p | S) - \sum_{p \in \bar{S}} \mathcal{P}_k(I_p | \bar{S})
\approx \frac{\sum_{p \in S} k(I_p | \theta_s)}{|S|} - \frac{\sum_{p \in \bar{S}} k(I_p | \theta_k)}{|\bar{S}|}
\]

(iii) bound optimization: auxiliary function at \(S\)

\[
A_1(S) \approx -\sum_{p \in S} \mathcal{P}_k(I_p | S) - 2 \sum_{p \in \bar{S}} \mathcal{P}_k(I_p | \bar{S}) + |S| \cdot \left[ G(S | h) - G(\bar{S} | h) \right]
\]

Table 1: K-means terms for color clustering that can be combined with Potts model for the segmentation boundary, e.g. (1). Basic K-means (A) corresponds to Gaussian model fitting minimizing cluster variances. Fitting more complex models like elliptic Gaussian [90, 86, 33], gamma/exponential distributions [3], GMM or histograms [104, 85] corresponds to probabilistic K-means (B), see [54]. Instead, we propose kernel K-means approach (C) using more complex data representation.

multi-label extension is deferred to Sec. 5.

1.1. Probabilistic K-means (pKM)

The connection of the likelihood term in (1) to K-means clustering is obvious in the context of Chan-Vese approach [24] where probability models \(P\) are Gaussian with fixed variances. In this case, the likelihoods in (1) reduce to \( \sum_{p \in S} \|I_p - \mu_s\|^2 + \sum_{p \in \bar{S}} \|I_p - \mu_k\|^2, \) \( \text{(2)} \) the sum of squared errors from each cluster mean. This is the standard K-means objective for clustering, Table 1A.

\[1\] We use \( \leq \) and \( \approx \) for “up to additive constant” relations.

\[2\] Optimal bandwidth for accurate Parzen density estimation is near data resolution [98]. Such kernel width is too small for good clustering, Sec. 4.1.

One way to generalize Chan-Vese’s color clustering is to replace squared Euclidean distance in (2) by other distortion measures \( \| \cdot \|_d \) corresponding to a general distortion energy

\[
\sum_{p \in S} \|I_p - \mu_s\|_d + \sum_{p \in \bar{S}} \|I_p - \mu_k\|_d, \; \text{(3)}
\]

which is very common in clustering. In this case, the optimal value of parameter \( \mu \) may no longer correspond to a mean. For example, optimal \( \mu \) for \( L_1 \) metric is a median and metric in (12) gives a mode [87, 20].

A seemingly different way to generalize Chan-Vese’s color clustering (2) is to treat both means and covariance matrices for the Gaussian models as variables. Then the likelihood term in (1) correspond to the standard elliptic K-means energy [90, 86, 33]. In this case optimized parameters \( \theta = \{ \mu, \Sigma \} \) are not even in the same space as data
points \( I_p \). But, it is still possible to define the distance between point \( I_p \) and model \( \theta \) as distortion

\[
\|I_p - \theta\|_d := -\log P(I_p|\theta).
\]

Elliptic K-means and distortion energy (3) are examples of a general class of probabilistic K-means methods [54] in Table 1B. This generalization of K-means corresponds to fitting arbitrary probability models, not necessarily Gaussian, extending energy (2) to

\[
-\sum_{p \in S} \log P(I_p|\theta_a) - \sum_{p \in S} \log P(I_p|\theta_b) \quad (4)
\]

where \( \theta_a \) and \( \theta_b \) are ML model parameters for each segment.

The name probabilistic K-means for energy (4) in the general clustering context was coined by [54]. They formulated (4) after representing distortion energy (3) as ML fitting of Gibbs models \( \frac{1}{Z} e^{-\|x-\mu\|^2} \) for integrable metrics.

In computer vision, energy (4) with any probability models is known as the log-likelihood term for clustering colors, geometric or higher-level features \( \{I_p\} \). Since data points correspond to pixels \( p \), segmentation energies like (1) often combine probabilistic K-means (4) for features \( I_p \) with image domain regularization, e.g. Potts smoothness. Typical models for (4) in vision are elliptic Gaussians [90, 86, 33], gamma/exponential [3], or other generative models [72]. As discussed, the corresponding parameters \( \theta \) or \( \mu \) are often different from the basic “mean”. Yet, probabilistic K-means is a good idiomatic name for general clustering energy (4).

Zhu-Yuille [104] and GrabCut [85] popularized fitting highly descriptive probability models (GMM or histograms) for color clustering in segmentation energies like (1). Assuming \( P(\theta) \equiv P(\cdot|\theta) \) is a continuous density of a sufficiently descriptive class (e.g. GMM), information theoretic analysis in [54] shows that probabilistic K-means energy (4) reduces to standard entropy criterion for clustering

\[
\theta \approx |S| \cdot H(S) + |S| \cdot H(S).
\]

\footnote{Outside of vision, general distortion clustering (3) can use its probabilistic Gibbs representation in [54] to integrate Potts-like prior, see [5].}

Indeed, for any function \( f(x) \) Monte-Carlo estimation gives

\[
\sum_{p \in S} f(I_p) \approx |S| \cdot \int f(x) d\mu(x) \equiv |S| \cdot \langle f, d\mu \rangle
\]

where \( d\mu \) is a “true” density for intensities in \( S \) and \( \langle \cdot \rangle \) is a dot product. If \( f = -\log P(\theta_a) \) and \( d\mu \approx P(\theta_a) \) then (4) implies (5) for differential entropy \( H(S) := H(P(\theta_a)) \). For histograms \( P_h(S) \equiv P_h(\cdot|S) \) entropy-based interpretation (5) of (4) is exact for discrete entropy \( H(S) := -\sum_x P_h(x|S) \cdot \log P_h(x|S) \equiv -\langle P_h(S), \log P_h(S) \rangle \).

Intuitively, minimization of the entropy criterion (5) fa-
vors clusters with tight or “peaked” distributions. This criterion is widely used in categorical clustering [65] or decision trees [17, 66] where the entropy evaluates histograms over “naturally” discrete features. Our paper demonstrates that the entropy criterion with either discrete histograms or continuous GMM densities has important limitations in the context of continuous color spaces.

In case of histograms, the key problem for color space clustering is illustrated in Fig. 2. Once continuous color space is broken into bins, the notion of proximity between the colors in the nearby bins is lost. Since bin permutations do not change the histogram entropy, criterion (5) cannot distinguish the quality of clusterings A and B in Fig. 2; some permutation of bins can make B look very similar to A.

In case of continuous GMM densities, the problem of entropy criterion (5) is quite different. In general, continuous density estimators commonly use Gaussian kernels, which preserve the notion of continuity in the color space. Indeed, the (differential) entropy for any reasonable continuous density estimate will see a significant difference between the clusters in A and B, see Figure 2.

We observe that the main issue for entropy criterion (5) with GMM densities is related to optimization problems. In this case high-order energies (5) or (4) require joint optimization of discrete variables $S_p$ and a large number of additional continuous parameters for optimum GMM density $P(\cdot | \theta_S)$. That is, the use of complex parametric probability models leads to complex high-order mixed objective functions. Typical block coordinate descent methods [104, 85] iterating optimization of $S$ and $\theta$ are sensitive to local minima, see Figures 1 and 3(e). Better solutions like Figure 3(f) have lower energy, but they can not be easily found unless initialization is very good.

These problems of probabilistic K-means with histograms or GMM in color spaces may explain why descriptive model fitting is not common in the learning community for clustering high-dimensional continuous spaces. Instead of probabilistic K-means they often use a different extension of K-means, that is kernel K-means in Table 1C.

1.2. Towards Kernel K-means (%K\#)

We propose kernel K-means energy to replace the standard likelihood term (4) in common regularization functionals for segmentation (1). In machine learning, kernel K-means (%K\#) is a well established data clustering technique [92, 74, 42, 36, 27, 51], which can identify complex structures that are non-linearly separable in the input space. In contrast to probabilistic K-means using complex models, see Tab.1, this approach maps data points $\{I_p | p \in \Omega\} \subset \mathcal{R}^N$ into a higher-dimensional Hilbert space using a complex (nonlinear) mapping

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

defining data embedding $\phi_p \equiv \phi(I_p) \in \mathcal{H}$. The original non-linear problem often can be solved by simple linear separators of the embedded points $\{\phi_p | p \in \Omega\}$ in $\mathcal{H}$.

Given data $\{I_p | p \in \Omega\}$ and embedding function $\phi$ kernel K-means corresponds to the basic K-means in the embedding space over points $\phi_p \equiv \phi(I_p)$. In case of two clusters (segments) $S$ and $\bar{S}$ this gives energy

$$E_k(S) := \sum_{p \in S} \|\phi(I_p) - \mu_S\|^2 + \sum_{p \in \bar{S}} \|\phi(I_p) - \mu_{\bar{S}}\|^2. \quad (6)$$

where $\|\cdot\|$ denotes the Euclidean norm, $\mu_S$ is the mean of segment $S$ in the new space

$$\mu_S = \frac{\sum_{q \in S} \phi(I_q)}{|S|} \quad (7)$$

and $|S|$ denotes the cardinality of segment $S$. Plugging $\mu_S$ and $\mu_{\bar{S}}$ into (6) gives equivalent formulations of this criterion using solely pairwise distances $\|\phi(I_p) - \phi(I_q)\|$ or dot products $\langle \phi(I_p), \phi(I_q) \rangle$ in the embedding space. Such equivalent pairwise energies are now discussed in detail.

It is a common practice to use kernel function $k(x, y)$ directly defining the dot product

$$\langle \phi(x), \phi(y) \rangle := k(x, y) \quad (8)$$

and distance

$$\|\phi(x) - \phi(y)\|^2 \equiv k(x, x) + k(y, y) - 2k(x, y) \equiv \|x - y\|_k^2. \quad (9)$$

in the embedding space. Mercer’s theorem [75] states that any continuous positive semi-definite (p.s.d.) kernel $k(x, y)$ corresponds to a dot product in some high-dimensional Hilbert space. The use of such kernels (a.k.a. kernel trick) helps to avoid explicit high-dimensional embedding $\phi(x)$.

For example, rewriting K-means energy (6) with pairwise distances $\|\phi(I_p) - \phi(I_q)\|^2$ in the embedding space implies one of the equivalent $k$KM formulations in Tab.1C(i)

$$E_k(S) \equiv \sum_{p,q \in S} \|I_p - I_q\|_k^2 \sum_{p,q \in \bar{S}} \|I_p - I_q\|_k^2 \quad (10)$$

with isometric kernel distance $\|\cdot\|_k^2$ as in (9). This Hilbertian metric\footnote{Such metrics can be isometrically embedded into a Hilbert space [46].} replaces Euclidean metric inside the basic K-means formula in the middle of Tab.1A. Plugging (9) into (10) yields another equivalent (up to a constant) energy formulation for $k$KM directly using kernel $k$ without any explicit reference to embedding $\phi(x)$

$$E_k(S) \equiv - \sum_{p,q \in S} k(I_p, I_q) \frac{|S|}{|S|} - \sum_{p,q \in \bar{S}} k(I_p, I_q) \frac{|\bar{S}|}{|\bar{S}|}. \quad (11)$$
Kernel K-means energy (10) can explain the positive result for the standard Gaussian kernel $k = \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right)$ in Fig.3(h). Gaussian kernel distance (red plot below)

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

(12)

is a “robust” version of Euclidean metric in basic K-means (green). Thus, Gaussian kernel K-means finds clusters with small local variances, which is impossible for non-compact clusters.

Focusing on pairwise clustering criteria (10) and (11) raises a natural question: why should distortion measure $\|x - y\|_k^2$ and similarity function $k(x,y)$ be restricted to Hilbertian metrics and p.s.d. kernels. That is, why should one worry that there is some embedding $\phi$ with an equivalent clustering energy (6). There are two complimentary answers. On the one hand, the standard $k$KM algorithm [74, 42] directly corresponds to the basic iterative K-means procedure for (6) even though it avoids explicit embedding $\phi$ due to pairwise formulation with kernels, see Sec.2. Thus, $k$KM algorithm convergence is not guaranteed for improper metrics or kernels. On the other hand, pairwise energies (10) and (11) for proper $k$ are general enough. The next subsection shows that generalizations to arbitrary distortion and similarity measures are not essential as there exist p.s.d. kernels with equivalent (up to constant) $k$KM energies.

**Average distortion (AD) or association (AA):** Equivalent formulations for energy $E_k$ in (10) and (11) suggest natural extensions of kernel K-means. For example, dropping p.s.d. assumption for kernels one can replace $k$ in (11) with arbitrary pairwise similarities or affinities $A = [A_{pq}]$ defining standard average association (AA) energy

$$E_A(S) := -\frac{\sum_{pq\in S} A_{pq}}{|S|} - \frac{\sum_{pq\in S} A_{pq}}{|S|}. \quad (13)$$

Kernel K-means energy (11) is a special case of (13) for $A_{pq} = k(I_p, I_q)$. Similarly, Hilbertian metric $\|\|_k^2$ in (10) can be replaced by arbitrary zero-diagonal distortion matrix $D = [D_{pq}]$ generating average distortion (AD) energy

$$E_D(S) := -\frac{\sum_{pq\in S} D_{pq}}{2|S|} - \frac{\sum_{pq\in S} D_{pq}}{2|S|}. \quad (14)$$

Kernel K-means energy (10) is a special case of (14) for $D_{pq} = \|I_p - I_q\|_k^2$. Below we discuss equivalence relationships between $k$KM, AD, and AA energies. Figure 4 illustrates these relations in a more general weighted case. Despite dropping metric and proper kernel assumptions, average distortion (14) and average association (13) clustering criteria can be reduced to $k$KM for arbitrary associations $A$ and any zero-diagonal distortions $D$. For example, for any given matrix $A$ in (13) consider “kernel matrix”

$$K := A + A^T + \lambda \cdot I. \quad (15)$$

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

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

for finite set $\chi = \{ I_p | p \in \Omega \}$. It is easy to check that $k$KM energy (11) with kernel $k \equiv K$ in (15) is equivalent to AA energy (13) with affinity $A$, up to a constant. Moreover, eigen decomposition $K = V^T \Lambda V$ gives an explicit finite-dimensional Euclidean embedding $\phi$ satisfying isometry (9)

$$\phi(I_p) = \sqrt{\Lambda} V_p \in \mathcal{R}^{|\Omega|} \quad (16)$$

where $V_p$ is a column of eigen vectors matrix $V$ in Fig.29(a) and $\Lambda$ is a diagonal matrix of eigen values. Indeed, as in (8)

$$\langle \phi(I_p), \phi(I_q) \rangle = (\sqrt{\Lambda} V_p)^T (\sqrt{\Lambda} V_q) = K_{pq} \equiv k(I_p, I_q).$$

Non-negativity of eigen values for $K$ is critical for decomposition $\Lambda = \sqrt{\Lambda} \cdot \sqrt{\Lambda}$. K-means for embedded points (16) minimizing $k$KM energy (6) is equivalent to (11) and (13).

Since average distortion energy (14) for arbitrary $D$ is equivalent to average association for $A = - \frac{D}{2}$, it can also be converted to $k$KM as above. Using the corresponding kernel matrix (15) and (9) it is easy to derive Hilbertian distortion (metric) equivalent to original distortions $D$

$$\|I_p - I_q\|_k^2 := \frac{D + D^T}{2} + 2\lambda (ee^T - I) \quad (17)$$

where $ee^T$ is a matrix where all entries are 1.

The above reduction of AD and AA to $k$KM summarizes the technical results in Roth et al. [84]. Dhillon et al. [36, 37] also reduce normalized cut [89] to a weighted version of $k$KM. In Section 3 we present a simpler proof showing that normalized cuts is a special case of weighted AA directly from the definition in [89]. Our Figure 4 outlines equivalence relations between $k$KM, AD and AA in the general case of weighted data points. These reductions to $k$KM allow to apply our bound and pseudo-bound optimization approach (Sec.2) to arbitrary weighted AD and AA criteria including normalized cuts. In particular, they can be combined with standard geometric and MRF-based regularizers and constraints. For simplicity, most of the paper

---

5 Mercer’s theorem for continuous p.d. kernels $k(x,y)$ uses a similar eigen decomposition giving infinite-dimensional Hilbert embedding $\phi(x)$. Our discrete kernel embedding $\phi(I_p)$ has finite dimension $|\Omega|$, which is still much higher than the dimension of points $I_p$, e.g. $\mathcal{R}^5$ for colors. Besides high-dimensional embedding (16) one can also compute a lower dimensional version $\hat{\phi}(I_p)$ approximating isometry (9), see Appendix D.
is presented in the context of non-weighted kernel K-means energies $E_k$ defined in (6-11).

**Pairwise vs. pointwise distortions:** Equivalence of $k\text{KM}$ to pairwise distortion criteria (14) helps to juxtapose kernel K-means with probabilistic K-means (Sec.1.1) from a point of view complementary to Table 1. Both methods generalize the basic K-means (2) by replacing Euclidean metric with a more general distortion measure $\|d\|$. But, $k\text{KM}$ uses “pairwise” formulation (10) where $\|d\|_d = \|\overline{\phi}_d\|_k^2$ measures distortion between pairs of points, while pKM uses “pointwise” formulation (3) where $\|d\|_d$ measures distortion between a point and a model. Even though these formulations are equivalent for Euclidean distortion (i.e. basic K-means), they are very different in general, see Figure 5.

For example, $k\text{KM}$ energy (10) for data points $\{I_p|p \in \Omega\}$ is defined by a finite set of pairwise distortions represented by matrix $D_{p,q} = \|I_p - I_q\|_d$ in (14). In contrast, pKM energy (3) uses pointwise distortion $\|I_p - \theta\|$ where the second argument is typically a continuous model parameter. While parameter $\theta$ is often in the same space as data points, it does not have to be in general, for instance, elliptic K-means uses Gaussians with $\theta = (\mu, \Sigma)$. Thus, general pointwise distortions for pKM correspond to likelihoods $\|I_p - \theta\|_d = -\ln P(I_p|\theta)$ in (4) where points $I_p$ and model parameters $\theta$ could be in different spaces.

In summary, pKM methods are essentially ML model-fitting techniques jointly optimizing clustering $S$ and explicit parameters $\theta$ using pointwise distortions (3) or likelihoods (4) representing distances between data points and models. In contrast, $k\text{KM}$ methods use pairwise distortions (10) or (14) minimizing distances between pairs of points within each cluster. Yet, $k\text{KM}$ is equivalent to basic K-means (6) implicitly optimizing cluster means $\mu$ in some isometric high-dimensional embedding space.

**Weak kernel K-means:** For Hilbertian distortions $\|d\|_d = \|\overline{\phi}_d\|_k^2$ with p.s.d. kernels we can show that pairwise $k\text{KM}$ approach (10) is “stronger” than a pointwise pKM approach (3) using the same metric. In this case pKM can be called weak kernel K-means, see Figure 5. Equivalent $k\text{KM}$ formulation (6) guarantees more complex decision boundaries, see Fig.3(h), compared to pKM energy (3)

$$\sum_{p \in S} \|I_p - \mu_S\|_k^2 + \sum_{p \in S} \|I_p - \mu_S\|_k^2$$

$$= \sum_{p \in S} \|\phi(I_p) - \phi(\mu_S)\|^2 + \sum_{p \in S} \|\phi(I_p) - \phi(\mu_S)\|^2$$

with isometric kernel distance $\|\cdot\|_k$ and explicit $\mu$ in the original space, Fig.3(g). Indeed, any $\mu$ in the original space corresponds to some search point $\phi(\mu)$ in the high-dimensional embedding space, while the opposite is false. Thus, optimization of (10) and (6) has larger search space than (18). It is also easy to check that energy (18) is an upper bound for (10) at any $S$. For this reason we refer to distortion energy (18) with kernel distance $\|\cdot\|_k$ and explicit $\mu$ in the original space as weak kernel K-means. Pointwise energy (18) is an example of pKM (3), while pairwise energy (10) with the same kernel metric is the regular $k\text{KM}$. Note that weak kernel K-means (18) for Gaussian kernel corresponds to $K$-modes, see Fig.3(g) and Sec.1.3.

Figure 5 illustrates relations between kernel K-means (10) and probabilistic K-means (3,4). It includes a few examples discussed earlier and more examples from Sec.1.3.

### 1.3. Previous kernel methods in segmentation

One of our goals is to combine kernel K-means clustering in the feature (color) space with standard constraints and regularization techniques (geometric or MRF) in the spacial domain. Figure 5 and Table 2 illustrate the general context of this work helping to relate our kernel approach to prior optimization-based segmentation and clustering methods.

**Pairwise clustering, normalized cuts, etc.:** Section 1.2 has already discussed strong relations between $k\text{KM}$ and general pairwise clustering criteria such as average distortion (14), average association (13), and normalized cuts. Figure 4 summarizes these relations for the general case of weighted points. Relations of kernel-based clustering to many common binary segmentation energies with a single ratio term [30, 52, 97, 59, 47] are not clear and they are left outside the scope of this work.

There are several existing methods for approximate optimization of NP-hard pairwise clustering energies discussed in this paper. Shi, Malik, and Yu [89, 101] popularized spectral relaxation methods in the context of normalized cuts. Such methods also apply to AA and other problems [89]. E.g., similarly to [101] one can rewrite AA energy (13) as

$$E_A(S) = -\text{tr}(Z^T AZ)$$

for $Z := \begin{bmatrix} S & S \\ \sqrt{|S|} & \sqrt{|S|} \end{bmatrix}$

where $Z$ is a $\Omega \times 2$ matrix of normalized indicator vectors $S$ and $S$. Orthogonality $S^T S = 0$ implies $Z^T Z = I_2$ where $I_2$ is an identity matrix of size $2 \times 2$. Minimization of the trace energy above with relaxed $Z$ constrained to a unit sphere $Z^T Z = I_2$ 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 two largest eigenvectors for matrix $A$. This approach extends to general multi-label weighted AA and related graph clustering problems, e.g., normalized cuts and ratio cuts [89, 101]. The main computational difficulties for spectral methods are explicit eigen decomposition for huge matrices and integrity gap due to heuristics for extracting integer solutions for the original combinatorial problem.

Buhmann et al. [48, 84] address the general AD and AA energies via mean field approximation and deterministic an-
Figure 4: Equivalence of clustering methods: kernel K-means (kKM), average distortion (AD), average association (AA). We present the general case of weighted points. Typically, kKM is associated with positive-definite (p.d.) or conditionally positive-definite (c.p.d.) kernels [88] and Hilbertian distortions [46]. The above formulations of AA and AD make no assumptions for association matrix A and distortions D except for zero diagonal in D. Then, equivalence of AD and AA objectives (up to a constant) is straightforward. Roth et al. [84] reduce non-weighted case of AD to kKM. For arbitrary D they derive Hilbertian distortion \( \|D\|_2 \) with an equivalent AD energy (up to a constant) and explicitly construct the corresponding embedding \( \phi \). We show Hilbertian metric \( \|D\|_2 \) for the general weighted case of AD, see \( AD \rightarrow kKM \) above. Dhillon et al. [36, 37] prove that normalized cuts is a special case of weighted kKM and construct the corresponding p.d. kernel. Sec.3 shows a simpler reduction of normalized cuts to weighted AA. Similarly to [84], an equivalent p.d. kernel could be constructed for any association matrix A, see \( AA \rightarrow kKM \) above and Sec.3. Note that the formulas for A-equivalent p.d. kernel and D-equivalent Hilbertian metric require some sufficiently large scalar \( \lambda \). Roth et al. [84] relate proper \( \lambda \) to the smallest eigenvalue of \( A = D - D^T \). Our weighted formulation requires the eigenvalue of \( \text{diag}(W) - \frac{1}{2} \cdot A \cdot \text{diag}(W) \). Applying pseudo-bound optimization [91] to kKM can bypass an expensive computation of eigenvalues. Indeed, term \( \lambda \cdot \text{diag}(W) \) generates a monotone perturbation for our kKM bound. The corresponding pseudo-bounds for all \( \lambda \) can be efficiently explored by parametric max-flow in [91]. For the algorithm to work it is enough that some (unknown) \( \lambda \) gives a proper bound.

nealing. It can be seen as a fuzzy version\(^6\) of kernel K-means algorithm. In the context of normalized cuts Dhillon et al. [36, 37] use spectral relaxation to initialize kKM algorithm.

One of our contributions is a formulation of the standard iterative kKM algorithm as a bound optimizer for the general pairwise clustering energies. In particular, this allows pseudo-bound optimization [91]. Moreover, our bound-based framework can easily combine pairwise clustering criteria with standard MRF and geometric regularization.

In the context of image segmentation pairwise clustering is typically applied to RGBXY space using spectral relaxation, see Tab.2. Unification of the color space and the image domain is a standard way to enforce segmentation boundary smoothness in this framework. Common MRF or geometric energies typically combine unary data (color) and arbitrary regularization terms in a linear fashion as in (1). In contrast, it is not easy to integrate specific constraints on geometry, shape, or sparsity into spectral methods for pairwise clustering. For example, to combine kKM 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, which is

\(^6\)Fuzzy version of K-means in Duda et al. [39] generalizes to kKM.
Figure 5: Clustering with (A) pointwise and (B) pairwise distortion energies generalizing (3) and (14) for points with weights $W = \{ w_p \}$. Pointwise distortion relates a data point and a model as log-likelihood function $\| I_p - \theta_s \|_d = -\ln P(I_p|\theta)$. Pairwise distortion is defined by matrix $D_{pq} = \| I_p - I_q \|_d$. Weighted AD or AA for arbitrary metrics are equivalent to weighted $k$KM, see Figure 4. As shown in [84, 36] average cut, normalized cut [89], and spectral ratio cut [23] are examples of (weighted) $k$KM. Our (pseudo-) bound optimization for $k$KM extends to any weighted pairwise distortion energy (B).

Table 2: Standard image segmentation methods and our approach. RGB is the space of observed features (colors) $I_p$ at each pixel $p$ in XY - the spacial domain or image grid. Standard regularization methods for XY integrate geometric priors (segmentation boundary length, curvature, shape prior). Clustering in unified RGBXY space complicates the use of specific geometric priors. Joining regularization in XY with kernel approach to RGB allows to combine the benefits of well-known geometric or MRF methods and powerful kernel clustering techniques extendable to high-dimensional features $I_p$. analogous to RGBXY clustering in the context of segmentation. In contrast, our bound formulation allows to combine $k$KM clustering term with the exact Potts and other priors.

We propose a proper bound (surrogate function) for $k$KM that allows pseudo-bound optimization [91]. Without computing expensive eigen decomposition, the method applies to general pairwise clustering energies AD and AA. Such energies for RGB (feature) space can be combined with any standard (e.g. MRF) or geometric regularization in XY domain and label cost sparsity terms [33].

**K-modes and mean-shift:** Weak kernel K-means using unary formulation (3) with kernel distance $\| \|_k$ and explicit
optimization of \( \mu \) in the original data space is closely related to \( K \)-modes approach to clustering continuous [20] or discrete [49, 25] data. For example, for Gaussian kernel distance \( \| \|_k \) energy (3) becomes

\[
- \sum_{p \in S} e^{-\frac{\| p - \bar{\mu} \|^2}{2\sigma^2}} - \sum_{p \in S} e^{-\frac{\| p - \bar{\mu} \|^2}{2\sigma^2}}
\]

which can be rewritten using Parzen densities as

\[
= -|S| \cdot \mathcal{P}_k(\mu_S|S) - |S| \cdot \mathcal{P}_k(\bar{\mu}_S|S).
\] (19)

Clearly, optimal \( \mu_S \) and \( \bar{\mu}_S \) are Parzen density modes in each segment. \( K \)-modes objective (19) can be seen as an energy-based formulation of mean-shift clustering [26, 29] with a fixed number of clusters. Formal objective allows to combine color clustering via (19) with geometric regularization in the image domain [87]. If needed, the number of clusters (modes) can be regularized by adding label cost [33]. In contrast, mean-shift segmentation [29] clusters RGBXY space combining color and spatial information. The number of clusters is indirectly controlled by the bandwidth. Appendix C extends our \( K \)-modes/mean-shift discussion.

**Parzen-based entropy clustering:** non-parametric kernel (Parzen) densities can be used to approximate entropy criterion (5), e.g. [55, 56], even though their mutual information narrative may seem different. This approximation for color clustering can be combined with geometric spatial regularization. The corresponding complex optimization problem is addressed by gradient descent via level sets [56], which is slow and prone to local minima, or by computationally expensive quadratic bounds [55]. It would be interesting to compare with our kernel \( K \)-means method, but their code or results on public data-sets are not available.

### 1.4. Summary of contributions

We propose kernel \( K \)-means as feature/color clustering criteria in combination with standard regularizers in the image domain. This combination is possible due to our bound formulation for \( kKM \) allowing to incorporate standard regularization algorithms such as max-flow. Our general framework applies to multi-label segmentation (supervised or non-supervised). Our approach is a new extension of \( K \)-means for color-based segmentation [24] different from probabilistic \( K \)-means [104, 85]. As special cases, our \( kKM \) color clustering term includes average distortion, average association, normalized cuts and other pairwise clustering criteria (Sec.1.3) previously used for segmentation in a different setting [89, 48].

Our bound formulation for \( kKM \) also allows pseudo-bound optimization [91]. Besides regularized segmentation framework above, this approach also applies to standard pairwise clustering, including normalized cuts. Pseudo-bound approach can improve standard iterative \( kKM \) algorithms and may compete with spectral relaxation methods for general (weighted) AD and AA problems while avoiding expensive eigen decompositions. More experiments should fully evaluate the potential of this optimization approach to general graph clustering energies.

Our \( kKM \) approach to color clustering has several advantages over standard probabilistic \( K \)-means methods [104, 85] based on histograms or GMM (Sec.1.1). In contrast to histograms, kernels preserve color space continuity without breaking it into unrelated bins, see Figure 2. Unlike GMM, our use of non-parametric kernel densities avoids mixed optimization over a large number of additional model-fitting variables. This reduces sensitivity to local minima, as illustrated in Figure 3(e,h).

For high-dimensional applications, kernel methods are a prevalent choice in the learning community as EM becomes intractable for high dimensions. Thus, our segmentation method extends to higher dimensional feature spaces.

Our approach allows arbitrary kernels or distortion measures. As a proof-of-the-concept, this paper focuses on the most standard Gaussian and 0–1 (KNN) kernels and discusses bandwidth selection strategies (adaptive and non-adaptive). We analyze the extreme bandwidth cases (Sec.4). It is known that for wide kernels (approaching data range) \( kKM \) converges to basic \( K \)-means, which has bias to equal size clusters [54, 12]. It has been observed empirically that small-width kernels (approaching data resolution) show bias to compact dense clusters [89]. We provide a theoretical explanation for this bias by connecting \( kKM \) energy for small bandwidth with the Gini criterion for clustering (37). We analytically prove the bias to compact dense clusters for the continuous case of Gini, see Theorem 1, extending the previous result for histograms by Breiman [17].

We also propose a class of adaptive bandwidth strategies. We use Nash theorem to obtain such strategies from density transformations (Nash embedding). That is, adaptive kernels work as “density equalization” techniques. Our experiments show state-of-the-art performance even with the most straightforward Gaussian kernels. Other kernels, weighted points, learning techniques, and applications to higher-dimensional features are left to future work.

\( kKM \) methods could be expensive for large image analysis problems. General kernels (e.g. Gaussian) imply quadratic complexity of each iteration due to high connectivity of the graph. We propose an efficient parallel implementation for our bound optimization framework that applies to general (e.g. adaptive) kernels.

### 2. Bound Optimization

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, 91]. \( A_i(S) \) is an auxiliary function of \( E(S) \) at current
The Potts model \[60\]

S (point re-assignment) globally optimizes auxiliary function \(k\).

For example, image-domain regularization and (II) with standard techniques. In fact, the following standard up-bound optimization interpretation allows to combine \(k\) to arbitrary affinities \(A_{pq}\) and propose a pseudo-bound optimization technique that can handle the most general class of pairwise clustering energies, including various popular criteria such as normalized cuts. Unlike spectral methods, our pseudo-bound framework removes the need for expensive eigen-value computations.

Proposition 1. The following is an auxiliary function for the \(k\)KM formulation in (11):

\[
E_k(S) = -\frac{\sum_{pq \in S} k_{pq}}{|S|} - \frac{\sum_{pq \in \mathcal{E}} k_{pq}}{|S|} \leq A_t(S) \quad \text{where}
\]

\[
A_t(S) = -2 \sum_{p \in S} \frac{\sum_{q \in S_i} k_{pq}}{|S_i|} - 2 \sum_{p \in S} \frac{\sum_{q \in S_j} k_{pq}}{|S_j|} + |S| \frac{\sum_{pq \in S_k} k_{pq}}{|S_k|^2} + \bar{|S|} \sum_{pq \in S_k} k_{pq}.
\]

\[E_k(S) \leq A_t(S) \leq A_t(S) = E(S).
\]

Proof. See Appendix A.

Proposition 2. The kernel K-means algorithm is a bound-optimization technique. In fact, the following standard updates [37] optimize \(A_t(S)\) at each iteration \(t\):

(I): For each point \(p\) and cluster \(S^i\), compute the distance between \(p\) and the mean of \(S^i\) in feature space:

\[
\text{dist}(p, S^i) = \left\| \phi(I_p) - \frac{\sum_{q \in S^i} \phi(I_q)}{|S^i|} \right\|^2.
\]

(II): Assign each point to the closest cluster:

\[
p \in S^j \text{ if } j = \arg \min_i \text{ dist}(p, S^i).
\]

In appendix A we clarify how these two standard steps in each iteration of the \(k\)KM algorithm yield a bound optimizer (see illustration below). In fact, step II (point re-assignment) globally optimizes auxiliary function \(A_t(S)\). Step I (distance re-computation) gives new auxiliary function \(A_{t+1}(S)\) at \(S_{t+1}\). It is not clear how to combine steps (I) and (II) with standard image-domain regularization. For example, to combine \(k\)KM with the Potts model [60]

solution \(S_t\) (\(t\) is the iteration number) if:

\[
E(S) \leq A_t(S) \quad \forall S \quad (20a)
\]

\[
E(S_t) = A_t(S_t) \quad (20b)
\]

To minimize \(E(S)\), we iteratively minimize an auxiliary function at each iteration \(t\): \(S_{t+1} = \arg \min S A_t(S)\). It is easy to show that such an iterative procedure decreases original function \(E(S)\) at each step:

\[
E(S_{t+1}) \leq A_t(S_{t+1}) \leq A_t(S_t) = E(S_t).
\]

For example, iterative optimization in standard GrabCut algorithm [85] was shown to be an optimizer of a cross-entropy bound [91], see Table 1B(iii).

Image segmentation functional: We propose to minimize the following high-order functional for image segmentation, which combines image-plane regularization with the pairwise clustering energy \(E_k(S)\) in (11):

\[
E(S) = E_k(S) + \beta R(S)
\]

where \(\beta\) is a (positive) scalar and \(R(S)\) is any functional with an efficient optimizer, e.g. a submodular boundary regularization term optimizable by max-flow methods

\[
R(S) = \sum_{\{p,q\} \in \mathcal{N}} l_{pq}[s_p \neq s_q]
\]

where \([\cdot]\) are Iverson bracket symbols and \(\mathcal{N}\) is the set of neighboring pixels. Pairwise weights \(l_{pq}\) are evaluated by the spatial distance and color contrast between pixels \(p\) and \(q\) as in [13].

From proposition 1, it follows that \(A_t(S) + \beta R(S)\) is an auxiliary function of high-order segmentation functional \(E(S)\) in (24). Furthermore, this auxiliary function is a combination of unary (modular) term \(A_t(S)\) and a submodular term \(R(S)\). Therefore, at each iteration of step II for our bound optimization algorithm, the global optimum of the bound can be obtained efficiently in low-order polynomial time by solving an equivalent max-flow problem [14]. Step I of our algorithm requires evaluation of the auxiliary function (21), which has quadratic complexity (\(O(N^2)\)). Section 6 discusses efficient implementation of step I.

3. Generalization to weighted AA

For completeness, this section considers \(K\) label case for a generalization of the formulation in (13) to weighted average association for given \(W := \{w_p|p \in \Omega\}\)

\[
E_W^{\text{AA}} = -\sum_{k=0}^{K-1} \sum_{p,q \in S^k} w_p w_q A_{pq}.
\]

The study of this energy is motivated by the works of Roth et al. [84] and Dhillon et al. [36]. We show how to convert general energy (26) to weighted \(k\)KM naturally extending the arguments for reducing non-weighted AA to \(k\)KM
in [84]. Moreover, we provide a bound for arbitrary affinities $A$ enabling pseudo-bound optimization [91]. One of the motivations for studying (26) is the fact that normalized cuts [89, 101] is a special case of weighted AA, as shown later in this section. While our derivation is inspired by the well-known reduction of normalized cuts to weighted $k$KM in [36, 37], we propose an equivalence argument much simpler than their connection via trace optimization.

First, we reduce weighted AA to weighted $k$KM. This is trivial if $A$ is a psd. Indeed, it is easy to check that (26) is equivalent to a weighted version of $k$KM (see Fig. 4)

\[ \sum_k \sum_{p \in S^k} w_p \| \phi(I_p) - \mu_{S^k} \|^2 \]  
\(27\)

where

\[ \mu_{S^k} := \frac{\sum_{p \in S^k} w_p \phi(I_p)}{\langle S^k, W \rangle} \]  
\(28\)

is a weighted mean of segment $S^k$ in the embedding space. Indeed, plugging (28) into (27) gives (26) up to a constant.

For the reduction of energy (26) to weighted $k$KM, we reduce weighted $k$KM to weighted AA and next weighted AA to weighted $k$KM via pseudo-bound. First, we reduce weighted $k$KM to weighted AA. Note that [84] use Prop.3 in building psd kernels in the simpler case of (26) when $W = I$.

A weighted version of the average distortion energy (14) is equivalent to (26) for $A = \frac{D}{D^2}$. Therefore, weighted AD can also be converted to $k$KM as above, see Fig. 4.

Reduction of weighted AA or AD to $k$KM using equivalent psd kernels in (29) allows to write down an explicit bound for these energies. However, the use of (29) requires proper $\lambda$ based on eigen values. Section 3.1 shows a pseudo-bound approach as in [91] that efficiently optimizes our energies without an expensive eigen decomposition.

**Normalized cuts is an example of weighed AA:** Dhillion et al. [36, 37] established the link between normalized cuts and weighted $k$KM showing that both problems can be transformed into a trace optimization for specific choices of weights and affinities. However, their proof is quite lengthy and algebraically involved. Below we demonstrate these links in a much simpler way using a trivial conversion of normalized cuts energy to weighted AA objective directly based on their definitions in [89, 101] and (26).

Consider the following affinity matrix

\[ \tilde{A} = diag(W) \cdot A \cdot diag(W) \]  
\(31\)

where diagonal elements $\frac{1}{w_p}$ for matrix $diag(W)$ are defined by weights

\[ w_p = \sum_{q \in \Omega} A_{pq}. \]

Indeed, plugging these $\tilde{A}$ and $w$ into weighted AA energy (26) immediately gives the well-known normalized cuts cost function [89, 101] with affinity matrix $A$

\[ -\sum_k \sum_{p,q \in S^k} A_{pq} \frac{\sum_{p,q \in S^k} A_{pq}}{\sum_{p,q \in S^k} A_{pq}} assoc(S^k, S^k) = -\sum_k assoc(S^k, S^k). \]  
\(32\)

In case of Gaussian kernel affinities $A$ as used in [89] both $A$ and $\tilde{A}$ are positive definite matrices, which proves that (26) and, consequently, (32) are examples of $k$KM energy (27).

In a more general case, the connection to $k$KM follows from the equivalent psd kernel (29) discussed earlier.

### 3.1. Pseudo-bound optimization (no eigen values)

In this section, we propose a pseudo-bound optimization technique for energy (26). Pseudo bounds [91] can be used for avoiding weak local minima and to alleviate the need for computing eigen decompositions for large matrices. For simplicity, we focus on the binary case $K = 2$, but we will discuss extensions to multi-label optimization at the end.

Let us recall the following definition introduced in [91].

**Definition 1.** (Pseudo-bound [91]) Given an energy $E(S)$, current solution $S_t$ and a parameter $\lambda \in \mathbb{R}$, functional $F_t(S, \lambda)$ is a pseudo-bound for energy $E(S)$ if there exists at least one $X$ such that $F_t(S, X)$ is an auxiliary function of $E(S)$ at current solution $S_t$.

Instead of using an auxiliary function, one can optimize a family of pseudo-bounds that includes at least one proper

\[ k' = diag(W) \cdot A \cdot diag(W) + \lambda \cdot diag(W). \]  
\(33\)

To make $k'$ psd one needs a different $\lambda$ than the one that works for our kernel in (29).
bound. This guarantees that original functional $E(S)$ decreases when the best solution is selected among the global minima for the whole family [91]. In the meanwhile, such pseudo-bounds may approximate $E(S)$ better than a single auxiliary function, even though they come from the same class of sub-modular (globally optimizable) functionals. The experiments of [91] confirmed that pseudo-bounds significantly outperform the optimization quality obtained by a single auxiliary function in the context of several high-order segmentation functionals, e.g., entropy [94], Bhattacharyya measure [8] and KL divergence [9]. If the pseudo-bounds are monotone w.r.t. parameter $\lambda$, we can find all global minima for the whole family in polynomial time via parametric max-flow algorithm [59]. This practical consideration is important when building a pseudo-bound family. For instance, [91] built pseudo-bounds by simply adding monotone unary term $\lambda A$ to a given auxiliary function. Here we follow a different approach for (26). As we will see next, our approach removes the need for computing expensive eigen-value decompositions.

Proposition 4. The following function is a pseudo-bound over parameter $\lambda$ for the general pairwise clustering formulation (26) ($K = 2$) with arbitrary matrix $A$

$$F_t(S, \lambda) = -\sum_{p \in S} \sum_{q \in S_t} w_p w_q \frac{(A_{pq} + A_{qp})}{\langle S_t, W \rangle} - \sum_{p \in S} \sum_{q \in S_t} w_p w_q \frac{(A_{pq} + A_{qp})}{\langle S_t, W \rangle} + \langle S, W \rangle \sum_{p \in S_t} \sum_{q \in S_t} w_p w_q \frac{(A_{pq} + A_{qp})}{2(\langle S_t, W \rangle)^2}$$

Furthermore, for any $\lambda \geq -\lambda_0(A^W)$ functional $F_t(S, \lambda)$ is an auxiliary function for weighted AA energy (26).

Proof. See appendix A.

Note that pseudo-bound (33) for $\lambda = 0$ gives the bound in Proposition 1 in case of psd $A$ such that $\lambda_0(A^W) \geq 0$.

Proposition 4 provides a pseudo-bound for the binary case of weighted AA. Pseudo bound $F_t(S, \lambda)$ consists of unary terms monotonic w.r.t. $\lambda$. Therefore, any combination of $F_t(S, \lambda)$ with sub-modular regularization terms can be minimized for all $\lambda$ in polynomial time via parametric max-flow [59]. This removes the need for evaluating explicitly $\lambda_0(A^W)$ avoiding eigen-value decompositions. Pseudo-bound $F_t(S, \lambda)$ can also be used for weighted AA without any regularization. In this case neither max-flow nor monotonicity are needed. Indeed, a unary potential for each pixel in (33) changes its sign only once as parameter $\lambda$ increases. It is enough to sort all critical values of $\lambda$ changing at least one pixel in order to traverse all (at most $1 + |\Omega|$) distinct solutions for pseudo bound (33) in a linear time.

Detailed evaluation of pseudo-bound optimization based on (33) will be reported later. As a proof-of-the-concept, we included Figure 6 demonstrating only one synthetic binary clustering example. It uses standard normalized cut (weighted AA) energy (32) with Gaussian affinity and no additional regularization terms. Basic bound optimization ($\lambda = 0$ in (33)) converges to a bad local minimum, while pseudo-bound optimization over all $\lambda$ achieves a much better solution with lower energy. This toy example also shows that our pseudo-bound approach to pairwise clustering may compete with standard spectral relaxation methods [89].

Examples in later sections use bound (21) for color space clustering combined with spatial domain regularization.

4. Parzen Analysis and Bandwidth Selection

This section discusses connections of $k$KM energy to Parzen densities providing probabilistic interpretations for our pairwise clustering approach. In particular, this section gives insights on bandwidth selection. We discuss extreme...
cases and analyze adaptive strategies. For simplicity, we mainly focus on Gaussian kernels, even though the analysis applies to other types of positive normalized kernels.

Note that standard Parzen density estimate for the distribution of data points within segment $S$ can be expressed using normalized Gaussian kernels. For Gaussian kernels of large bandwidth $\sigma$ like mixing the desired probability model the data range and small kernels approaching the data resolution. It is easy to see that $k$KM energy (11) is exactly the following high-order Parzen density energy

$$E_k(S) \approx -\frac{1}{2} \sum_{p \in S} \sum_{q \in S} \left( I_p - I_q \right)^2 |S|.$$  

It is easy to see that $k$KM energy (11) is exactly the following high-order Parzen density energy

$$E_k(S) \approx -\sum_{p \in S} P_k(I_p | S) - \sum_{p \in S} P_k(I_p | S).$$  

This probabilistic interpretation of $k$KM gives an additional point of view for comparing it with pKM clustering with log-likelihood energy (4). Instead of parametric ML models $k$KM uses Parzen density estimates. Another difference is absence of the log in (35). Omitting the log reduces the weight of low probability points, that is, outliers. In contrast, log-likelihoods in (4) are highly sensitive to outliers.

To address this problem, pKM methods often use heuristics like mixing the desired probability model $P$ with a uniform distribution, e.g. $P(\cdot | \theta) := \epsilon + (1 - \epsilon)P(\cdot | \theta)$.

4.1. Extreme bandwidth cases

Parzen energy (35) is also useful for analyzing two extreme cases of kernel bandwidth: large kernels approaching the data range and small kernels approaching the data resolution. This section analyses these two extreme cases.

Large bandwidth and basic K-means: Consider Gaussian kernels of large bandwidth $\sigma$ approaching the data range. In this case Gaussian kernels $k$ in (34) can be approximated (up to a scalar) by Taylor expansion $1 - \frac{\|I_x - I_y\|^2}{2\sigma^2}$.

Then, Parzen density energy (35) becomes (up to a constant)

$$\frac{1}{2} \sum_{pq \in S} \|I_p - I_q\|^2 |S| + \frac{1}{2} \sum_{pq \in S} \|I_p - I_q\|^2 |S|$$

which is proportional to the pairwise formulation for the basic K-means or variance criteria in Tab. 1A with Euclidean metric $\|\|$.

That is, $k$KM for large bandwidth Gaussian kernels reduces to the basic K-means in the original data space instead of the high-dimensional embedding space.

In particular, this proves that as the bandwidth gets too large $k$KM looses its ability to find non-linear separation of the clusters. This also emphasizes the well-known bias of basic K-means to equal size clusters [54, 12].

Small bandwidth and Gini criterion: Very different properties could be shown for the opposite extreme case of small bandwidth approaching data resolution. It is easy to approximate Parzen formulation of $k$KM energy (35) as

$$E_k(S) \approx -\|S\| \cdot \langle P_k(S), d_s \rangle - \|S\| \cdot \langle P_k(S), d_s \rangle$$

where $P_k(S)$ is kernel-based density (34) and $d_s$ is a “true” density for intensities in $S$. Approximation (36) follows directly from the same Monte-Carlo estimation argument given earlier below Eq. (5) in Sec.1.1 with the only difference being $f = -P_k(S)$ instead of $-\log P(\theta S)$.

If kernels have small bandwidth optimal for accurate Parzen density estimation we get $P_k(S) \approx d_s$ further reducing (36) to approximation

$$\approx -\|S\| \cdot \langle d_s, d_s \rangle - \|S\| \cdot \langle d_s, d_s \rangle$$

that proves the following property.

Property 1. Assume small bandwidth Gaussian kernels optimal for accurate Parzen density estimation. Then kernel K-means energy (35) can be approximated by the standard Gini criterion for clustering [17]:

$$E_G(S) := |S| \cdot G(S) + |S| \cdot G(S)$$

where $G(S)$ is the Gini impurity for the data points in $S$

$$G(S) := 1 - \langle d_s, d_s \rangle \equiv 1 - \int x d_s(x) dx.$$  

Similarly to entropy, Gini impurity $G(S)$ can be viewed as a measure of sparsity or “peakness” for continuous or discrete distributions. Both Gini and entropy clustering criteria are widely used for decision trees [17, 66].

In this discrete context Breiman [17] analyzed theoretical properties of Gini criterion (37) for the case of histograms $P_h$ where $G(S) = 1 - \sum x P_h(x | S)^2$. He proved that the minimum of the Gini criterion is achieved by sending all data points within the highest-probability bin to one cluster and the remaining data points to the other cluster, see the color encoded illustration above. We extend Breiman’s result to the continuous Gini criterion (37)-(38).

Theorem 1. (Gini Bias) Let $d_\Omega$ be a continuous probability density function over domain $\Omega \subseteq \mathbb{R}^n$ defining conditional density $d_s(x) := d_\Omega(x | x \in S)$ for any non-empty subset $S \subset \Omega$. Then, continuous version of Gini clustering criterion (37) achieves its optimal value at the partition of $\Omega$ into regions $S$ and $S = \Omega \setminus S$ such that $S = \arg \max_x d_s(x)$.

Proof. See Appendix B and Proposition 5.

The bias to small dense clusters is practically noticeable for small bandwidth kernels, see Fig.8(d). Similar empirical bias to tight clusters was also observed in the context of

Bandwidth near inter-point distances avoids density oversmoothing.
average association (11) in [89]. As kernel gets wider the continuous Parzen density (34) no longer approximates the true distribution \( d_p \) and Gini criterion (37) is no longer valid as an approximation for \( k\text{KM} \) energy (35). In practice, Gini bias gradually disappears as bandwidth gets wider. This also agrees with the observations for wider kernel in average association [89]. As discussed earlier, in the opposite extreme case when bandwidth gets very large (approaching data range) \( k\text{KM} \) converges to basic K-means or variance criterion, which has very different properties. Thus, kernel K-means properties strongly depend on the bandwidth.

The extreme cases for kernel K-means, i.e. Gini and variance criteria, are useful to know when selecting kernels. Variance criteria for clustering has bias to equal cardinality segments [54, 12]. In contrast, Gini criteria has bias to small dense clusters (Theorem 1). To avoid these biases kernel-K-means should use kernels of width that is neither too small nor too large. Our experiments compare different strategies with fixed and adaptive-width kernels (Sec.4.2). Equivalence of kernel-K-means to many standard clustering criteria such as average distortion, average association, normalized cuts, etc. (see Sec.1.2) also suggest kernel selection strategies based on practices in the prior literature.

4.2. Adaptive kernels via Nash embedding and KNN

As discussed in Sec.4.1, kernel width should neither be too small nor too large. We propose adaptive kernels designed to equalize the density in highly dense regions in the color space. The following equation interprets adaptive Gaussian kernels via Riemannian distances in the color space (left picture in Fig.7).

\[
k_p(I_p, I_q) = e^{-\frac{||I_p - I_q||^2}{2\sigma^2}} = e^{-\frac{(I_p - I_q)^T \Sigma^{-1}(I_p - I_q)}{2}}.
\]

According to Nash embedding theorem [76], this Riemannian color space can be isometrically embedded into a Euclidean space, so that the last expression above is equal to

\[
\begin{align*}
&= e^{-\frac{||I_p' - I_q'||^2}{2}} = k(I_p', I_q')
\end{align*}
\]

where \( k \) is a fixed-width Gaussian kernel in the new transformed space (right picture in Fig.7). Thus, non-normalized Gaussian kernels of adaptive width \( \sigma_p \) (or covariance matrix \( \Sigma_p \), in general) define some color space transformation, Nash embedding, that locally stretches or shrinks the space. After this transformation, clustering is done using a fixed (homogeneous) Gaussian kernel of constant width.

Figure 7 helps to illustrate how Nash embedding changes the color space density. The number of points in a unit (Euclidean) ball neighborhood in the transformed space is equal to the number of points in the corresponding unit (Riemannian) ball in the original space:

\[
K = d' \cdot V_1 = d \cdot V_\sigma
\]

where \( d \) and \( d' \) are local densities in the original and transformed spaces. Thus, kernel width \( \sigma_p \) can be selected adaptively based on any desired transformation of density \( d'(d) \) according to formula

\[
\sigma_p \sim \sqrt{\frac{d'(d_p)}{d_p}}
\]

(39)

where \( d_p := d(I_p) \) is an observed local density for points in the color space. This local density can be evaluated using any common estimator, e.g. Parzen approach gives

\[
d(I_p) \sim \sum_q \frac{1}{\Delta q} e^{-\frac{||I_p - I_q||^2}{2\Delta^2}}
\]

(40)

where \( \Delta \) could be adaptive or fixed \( \Delta = \text{const} \), according to any standard technique for density estimation [10].

To address Breiman bias one can use density equalizing transforms \( d'(d) = \text{const} \) or \( d' = \frac{1}{2} \log(1 + ad) \), which even up the highly dense parts of the color space. Formula (39) works for any target transform \( d'(d) \). Once adaptive kernels \( \sigma_p \) are chosen, Nash theory also allows to obtain empirical scatter plots \( d'(d_p)|\Omega := \{(d'(I_p), d(I_p))|p \in \Omega\} \), for example, to compare it with the selected “theoretical” plot \( d'(d) \). Estimates \( d(I_p) \) are provided in (40) and the density estimate for Nash embedding are

\[
d'(I_p') \sim \sum_q e^{-\frac{||I_p' - I_q'||^2}{2\sigma^2}} = \sum_q e^{-\frac{||I_p - I_q||^2}{2\sigma^2}} .
\]

Note the difference between empirical density estimates for \( d' \) in (41) and \( d \) in (40): the former uses the sum of non-normalized kernels of selected adaptive width \( \sigma_q \) in (39).
and the latter is the sum of normalized kernels of width $\Delta_d$ based on chosen density estimator. While parameter $\sigma_d$ directly controls the density transformation, $\Delta$ plays a fairly minor role concerning the quality of estimating density $d$.

Figure 8(e) illustrates the empirical density mapping $d'(d)_{\Omega}$ induced by adaptive kernels (39) for $d'(d) = \text{const}$. Notice a density-equalization effect within high density areas in the color space addressing the Breiman bias.

The const density mapping can be approximated using KNN graph. To be specific, the symmetric KNN kernel in this paper is defined as follows:

$$k_{pq} = \begin{cases} \frac{1}{|KNN(f_p)|} & \text{if } f_p \in KNN(f_q), \\ \frac{1}{|KNN(f_q)|} & \text{if } f_q \in KNN(f_p) \end{cases}$$

where $KNN(f_p)$ is a set of K nearest neighbors of $f_p$. The affinity between $f_p$ and $f_q$ achieves maximum value of 2 if they are mutually each other’s KNNs.

5. Generalization to multi-label problems

It is straightforward to rewrite our unary $k$KM bound (21) and pseudo-bound (33) for the general multi-label $k$KM energy, e.g. (26). Each corresponding pair of terms for $S$ and $S^*$ has to be written as a sum of the terms for $K$ segments $S^K$. Combination of such unary/linear bounds with arbitrary regularizers can be optimized with any standard discrete or continuous multi-label methods including graph cuts [15, 50], message passing [58], LP relaxations [99], or well-known continuous convex formulations [21, 22, 32]. For illustration, our experiments use multi-label applications with MRF regularizers commonly optimized by standard move-making graph cut methods [15]. We discuss some details of such multi-label algorithms for energies with $k$KM clustering terms.

Algorithm 1: $\alpha$-Expansion for Kernel Cut

Input: Affinity Matrix $A$ of size $|\Omega| \times |\Omega|$; initial labeling $S_{0}^{1}, \ldots, S_{0}^{K}$

Output: $S_{t}^{1}, \ldots, S_{t}^{K}$; partition of the set $\Omega$

1. Set $t := 0$

2. while not converged do

   a. Set $\tilde{A}_{t}(S)$ to be bound (44) for energy (43) at current partition $S_t$

   b. for each label $\alpha \in L = \{1, \ldots, K\}$ do

      i. Find $S_t^{\alpha} := \arg\min \tilde{A}_t(S)$ within one $\alpha$-expansion of $S_t$

   end

3. Set $t := t + 1$

5.1. Move-making algorithms

Assume that $E_k$ and $A_t$ are multi-label versions of $k$KM energy (11) and bound (21), correspondingly. A combination of $k$KM energy with many standard MRF potentials $R$

$$\bar{E} = E_k + \beta R$$

can be straightforwardly optimized by the standard expansion moves and swap moves [16] using bound

$$\bar{A}_t = A_t + \beta R.$$  

Bound (44) is optimizable since $A_t$ is a simple unary term. Even though the global optimum of this bound is not guaranteed for general multi-label problems, it suffices to decrease the bound in order to decrease energy (43), i.e.

$$\bar{A}_t(S_{t+1}) \leq \bar{A}_t(S_t) \Rightarrow \bar{E}(S_{t+1}) \leq \bar{E}(S_t).$$

In general, tighter bounds should work better. Thus, we do not run iterative move-making algorithms for bound $A_t$ until convergence before re-estimating $A_{t+1}$. Instead, one can reestimate the bound either after each move or after a certain number of moves. Algorithm 1 shows the $\alpha$-expansion version of the proposed method for energy (43).

5.2. Pseudo-bound monotonicity issues

As mentioned earlier, the unary pseudo-bound $F_1$ in (33) can also be extended to the general multi-label case. However, it is easy to check that monotonicity of bound (33) w.r.t. parameter $\lambda$ is no longer guaranteed. For additional regularization terms $R$ it is still possible to use $\alpha$-expansion [15] or other algorithms, but the number of distinct solutions for all $\lambda$ can not be linearly bounded. Yet, some easy fixes are possible. For example, restricted expansion moves can be limited to “monotone” subsets of pixels with either positive or negative unary potential with respect to $\lambda$.

Without regularization terms, monotonicity of the parametric unary pseudo-bound (33) is not essential. Similarly to the binary case discussed earlier, it is easy to check that the optimal label for each pixel changes at most $K - 1$ times as $\lambda$ increases. Critical values of $\lambda$ for all pixels can be

10The optimal value of a unary potential for each pixels is the lower envelope of $K$ linear functions of $\lambda$, which has at most $K - 1$ breakpoints.
High-dimensional dense filtering is proposed in [1], which is notice that the time complexity of the approach in [82] grows 6. Experiments for pseudo bound (33) can be explored in a linear time. sorted so that all (at most 1) \((K - 1)\) distinct solutions for pseudo bound (33) can be explored in a linear time.

6. Experiments

Efficient evaluation of bounds \(A_t(S)\): As mentioned in Sec. 2, deriving the coefficients of upper bound \(A_t(S)\) in (21) for each iteration is of \(O(|\Omega|^2)\) time complexity. Here we give a feeble way of bound evaluation. Notice that the most expensive part of deriving the linear bound \(A_t(S)\) is to compute \(\sum_{q \in S_t} k_{pq}\) and \(\sum_{q \in S_t} k_{pq}\) for each \(p \in \Omega\).

For computing \(\sum_{q \in S_t} k_{pq}\), in the case of fixed width Gaussian kernel \(k_{pq}\), we resort to fast approximate dense filtering method in [82], which takes \(O(|\Omega|)\) time. Also notice that the time complexity of the approach in [82] grows exponentially with data dimension \(D\). A better approach for high-dimensional dense filtering is proposed in [1], which is of time \(O(|\Omega| \times D)\). We stick to [82] for low-dimensional color space in our experiments. For general kernel (adaptive width Gaussian kernel), we developed the GPU framework in Fig. 9. Evaluation of surrogates is conducted on GPU, while the CPU optimizes the bounds, updates solutions and checks convergence. Using this framework, we achieved empirical speedup close to the number of cores in GPU.

For \(KNN\) graph (42) approximating const density mapping, as discussed in Sec. 4.2, the evaluation of bounds is of \(O(K \times |\Omega|)\). Sampling of such \(KNN\) graph can further accelerate without degrading clustering.

Implementation details: LAB color space is used for all algorithms. For GrabCut, we used histogram-based probability model, as is common in the literature [94, 63]. For contrast-sensitive regularization, we use standard penalty \(w_{pq} = \frac{1}{d_{pq}} e^{-0.5||I_p - I_q||^2/\beta}\), where \(\beta\) 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 choose sampling rate as half of kernel width \(\sigma\) for Bilateral Grid in [82].

We evaluated our Kernel Cut (fixed width kernel or \(KNN\)) in the context of interactive segmentation, and compared with the commonly used GrabCut algorithm [85]. 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. We further report results on the GrabCut dataset of 50 images and the Berkeley dataset. We also report results of segmentation with high-dimensional features, including texture, location, motion and depth. Kernel Cut is compared to spectral clustering [89] as well, which can’t jointly optimize pairwise clustering term and MRF regularization term.

6.1. Robustness to regularization weight

We first run all algorithms without smoothness. Then, we experiment with several values of \(\lambda\) for the contrast-sensitive edge term. In the experiments of Fig. 10 (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. Without smoothness, our Kernel Cut yielded much better results than model fitting. Regularization significantly benefited the latter, as the decreasing blue curve in (a) indicates. For instance, in the case of the zebra image, model fitting yielded 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 did not help as much. Notice that, for small regularization weights, our method is substantially better than model fitting. Also, notice the performance of our method is less dependent on regularization weight; therefore, it does not require fine tuning of \(\lambda\).

6.2. Segmentation on GrabCut & Berkeley datasets.

First, we report results on the GrabCut database (50 images) using the bounding boxes provided in [64]. For each image the error is the percentage of mis-labeled pixels. We compute the average error over the dataset.

We test different smoothness weights and plot the error curves in Fig. 11. Table 3 reports the best 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, aKKM (Kernel Cut with adaptive kernel) 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. 12 shows that our Kernel Cut is also robust to the hyper-parameter, i.e. \(K\) for nearest neighbours, unlike GrabCut.

\textsuperscript{11}The smoothness weights for different energies are not directly comparable; Fig. 11 shows all the curves for better visualization.
M. Tang, I. Ben Ayed, D. Marin, Y. Boykov. In arXiv:1506.07439v2 [cs.CV], June, 2016 p.17

Figure 10: Illustration of robustness to smoothness weight.

Figure 11: Average error vs. regularization weights for different algorithms on the GrabCut dataset.

Figure 12: Our method aKKM is robust to choice of $K$ while GrabCut is sensitive to bin size for histograms.

Figure 13: Sample results for GrabCut and our kernel methods with fixed & adaptive widths (KKM, aKKM), see Tab.3.

Table 3: Box-based interactive segmentation (Fig.13). Error rates (%) are averaged over 50 images in GrabCut dataset.

| boundary smoothness | color clustering term |
|---------------------|-----------------------|
| GrabCut             | KKM                   |
| none                | 27.2                  |
| Euclidean length    | 13.6                  |
| contrast-sensitive  | 8.2                   |
| KKM                 | 20.4                  |
| wKKM                | 17.6                  |
| aKKM                | 12.2                  |
| KKM                 | 15.1                  |
| aKKM                | 16.0                  |
| KKM                 | 9.7                   |
| aKKM                | 13.8                  |
| KKM                 | 7.1                   |

Table 4: Seeds-based interactive segmentation (Fig.14). 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.16) from 100 image subset in [70]. (GrabCut and aKKM give 3.8 and 3.0 errors on the whole database.)

| boundary smoothness | color clustering term |
|---------------------|-----------------------|
| BJ                  | GrabCut               |
| none                | 12.4                  |
| contrast-sensitive  | 3.2                   |
| GrabCut             | 12.4                  |
| aKKM                | 7.6                   |
| aKKM                | 3.7                   |
| BJ                  | 2.8                   |

6.3. Segmentation of similar appearance objects

We apply our Kernel Cut to separating similar objects within an image. This is different from co-segmentation [53, 95, 73] that separates similar objects for image col-
selections. We test the effect of extra XY feature space for GrabCut and Kernel Cut. We try various γ in \( f_p = [L_p, a_p, b_p, \gamma X, \gamma Y] \) for Kernel Cut. To get practical sense of γ, which is a relative scale, we embed KNN graph on image, as shown in Fig. 15. We measure the average distance of pixels to its \( k^{th} \) nearest neighbor. The connection range obtained for each γ is in pixels. For histogram based GrabCut we experiment with different spatial bin size for XY channel, ranging from 30 pixels to image size.

Visualization of pixels’ KNNs are shown in Fig. 15 in image domain. Such KNN graph from RGBXY space (left of Fig. 15) suits our needs for separating similar objects and is cheap to construct.

We report quantitative results on a multi-objects and camouflage database we collected. The 18 selected images are from the Berkeley segmentation database [69] and we generate strokes for selecting one object among multiple similar objects, see example images and segmentations in Fig. 16. Fig. 17 shows the average error rates for GrabCut and Kernel Cut. Fig. 18 gives multi-label segmentation of similar objects in one image with seeds using our algorithm. We use swap-move for multi-label weighted AA energy (26) that corresponds to NC plus smoothness term, as discussed in Sec. 5. Fig. 18 (c) shows energy convergence.

6.4. Texture segmentation

The goal of this experiment is to demonstrate scalability of our method to highly dimensional data. First, desaturated images from GrabCut database [85] are convolved with 48 filters from [93]. 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 \(^{12}\) of a pixel is augmented by the corresponding texton histogram.

\(^{12}\)We found that for the GrabCut database adding texture features to RGB does not improve the results.
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. 19. We compare our results with GrabCut method where we vary the bin size for texture features.

### 6.5. Segmentation with depth features

Over a last few years depth sensors have been widely and successfully used for various computer vision problems such as 3D modeling [38, 78], semantic segmentation [35, 45, 77, 83], motion flow estimation [44], etc. This section describes a few experiments, that use an additional depth channel for interactive segmentation.

In order to evaluate the methods (GrabCut and Kernel Cut) with respect to performance on RGBD data an appropriate database is required. We manually selected 64 images from NYUv2 database [77] of densely labeled indoor RGBD scenes. The database was initially used for semantic segmentation. For the selected images we provide bounding boxes and ground truth for interactive binary segmentation. In contrast to GrabCut database, the prepared dataset consist of low-quality images: there are camera motion artifacts, underexposed and overexposed regions, out of focus images. These artifacts make the color-based segmentation problem harder.

Both tested methods require defining several parameters. All of them have smoothness weight \( \lambda \) for MRF regularization. In addition, GrabCut is parameterized by the size of histogram bins, while Kernel Cut requires to set number \( K \) of nearest neighbors. We used the same \( K \) as in the previous sections for Kernel Cut. We optimize the results of GrabCut over the size of color bins and \( \lambda \). We also optimize the results of Kernel Cut over \( \lambda \).

One of the objectives of the experiment is to study the relation of the performance of the methods and the “importance” of D-channel. In case of GrabCut the measure of D-channel importance is the size of the bins. There are two extremes. First, if all pixels fall into the same bin w.r.t. D-channel the method will not be able to differentiate pixels with different depths. That corresponds to the lowest importance. On the other end, all pixels with different depth would be treated independently. That corresponds to highest importance of D-channel. In case of Kernel Cut the importance is defined by scaling factor \( \gamma \), which is used in the definition of the feature vector of a pixel: \([R, G, B, \gamma D]\).

The average error statistics for GrabCut and Kernel Cut with respect to importance of the depth dimension are shown in Fig. 20. The examples of segmentation are shown in Fig. 21. The optimal value of \( \gamma \) suggests that for that particular database D-channel contains strictly more information than RGB channels. On the other hand, GrabCut failed to improve given the additional channel.

### 6.6. Motion segmentation

This subsection shows two applications using our proposed framework for motion segmentation in binary and multi-label settings. In addition to location and depth features used above, we show segmentation results for motion. Figs. 26, 27 and 28 compare motion segmentations using different feature spaces: RGB features, XY feature, M feature (optical flow) and their combinations (RGBM or RGBXYM or RGBXYM). Here we use Kernel Cut for the normalized cut energy combined with MRF regularization.
Figure 21: RGBD+XY examples. The first two rows show original images with bounding box and color-coded depth channel. The third row shows the results of Grabcut, the fourth row shows the results of Kernel Cut. Parameters of the methods were independently selected to minimize average error rate over the database. The parameters of the algorithms were selected to minimize the average error over the dataset.

| method                | Covering | PRI   | VOI |
|-----------------------|----------|-------|-----|
| Spectral Clustering   | 0.34     | 0.76  | 2.76|
| Our Kernel Cuts       | 0.41     | 0.78  | 2.44|
| Our Spectral Cuts     | 0.42     | 0.78  | 2.34|

Table 5: Results of spectral clustering (K-means on eigenvectors) and our Kernel Cuts & Spectral Cuts on BSDS500 dataset. For this experiment mPb-based kernel is used.

Motion segmentation example 1: For videos in FBMS-59 dataset [18], our algorithm runs on individual frames, instead of 3D volume. Segmentation of previous frame is taken as initialization for the next frame and strokes are provided only for the first frame. We used recent optical flow algorithm in [19] to generate M features. Results for selected frames are in Figs. 26 and 27. Instead of tracks from all frames in [80], our segmentation of each frame uses only motion estimation between two consecutive frames. Our approach jointly optimizes feature space clustering and XY regularization (normalized cut + Potts), while [80] first clusters semi-dense tracks using normalized cuts [18] and then obtains dense image segmentation using regularization.

Motion segmentation example 2: We also experiment with Kitti dataset [71]. Fig. 28 shows the multi-label segmentation using either color information RGB+XY (the first row) or motion MXY+XY (the second row). We use the provided ground-truth motion field as a source of M channel. Note that the motion field is known only for approximately 20% of 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 naturally interpolates the segmentation labels during the optimization procedure.

6.7. Spectral Clustering vs. Kernel & Spectral Cuts

Spectral clustering [89] 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 [2, 68], see examples in Fig. 22. To overcome such edge misalignment, our kernel cuts jointly optimize normalized cuts energy and edge contrast based smoothness term. As such, kernel cuts achieves better segmentation boundaries. Fig. 22 shows sample results. We also show our Spectral Cuts here. As is shown in Tab. 5, kernel cuts and spectral cuts give better covering, PRI (probabilistic rand index) and VOI (variation of information) over spectral clustering on BSDS500 dataset. Number of segments in groundtruth is provided to each method.

6.8. Normalized Cuts plus Label Costs

Here we use kernel cuts to optimize a joint energy of normalized cuts (equivalently normalized association), Potts regularization term and label costs term, which penalizes
Figure 22: Sample results on BSDS500. Top row: spectral clustering. Middle & Bottom rows: our Kernel & Spectral Cuts.

Figure 23: From left to right label cost $\delta$ is increased.

Each label by a constant $\delta$. The energy is minimized by our proposed $\alpha$ expansion and $\alpha - \beta$ swap moves in Sec. 5. The initial models are sampled from patches, as is in [33]. Sample results with different label cost is shown in Fig. 23. Due to sparsity prior for normalized association, our kernel cuts automatically determine the number of segments and yield regularized segmentation. For simplicity $K$NN kernel in color space is used in this experiment.

### 6.9. Semi-supervised Image Clustering

To show that our kernel cuts and spectral cuts can benefit semi-supervised image clustering, we run experiments from a small ImageNet [34] dataset of 15K images. Ten classes are selected from the ImageNet. Here the spatial pyramid kernel [62] on SIFT descriptors [67] is used as pairwise affinities and our kernel cuts minimize normalized energy plus regularization term. We used pyramid level of 4. Semi-supervision is of the form of must-link pairwise constraints [60] and is encoded into our energy as submodular MRF regularization term. NMI (normalized mutual information) is used as measure for quality of clustering. NMI of value 1 means perfect clustering with respect to groundtruth.

Spectral clustering gives NMI of 0.208, while kernel k-means [36] achieves 0.255. Our kernel cuts and spectral cuts significantly boost the NMI value with such semi-supervision, see Fig. 24. Notice that our kernel cuts (spectral cuts) degenerates to kernel k-means (spectral clustering) when no constraints are given.

We also incorporate group prior for image clustering. We experiment on the LabelMe dataset [81] which contains 2600 images of 8 scene categories (coast, mountain, forest, open country, street, inside city, tall buildings and highways). GIST feature and Gaussian kernel is used to generate the kernel. We used the group prior introduced in [28] that is based on image tags. The group prior is noisy and the dominant category in each group occupies 60%-90% of the group. For each group we introduce an energy term that is akin to the robust $P^n$ potts [57], which can be exactly minimized in $\alpha - \beta$ swap or $\alpha$ expansion by graph cuts. The high-order consistency term is defined on each group.

Spectral clustering and kernel K-means [36] give NMI value of 0.542 and 0.572 respectively. Our kernel cuts and spectral cuts are able to incorporate the group prior and
and only if:
\[(11):\]
larger the weight the better since the grouping prior is noisy. Note that it’s not the case the larger the weight the better since the grouping prior is noisy.

**APPENDIX A**

In this appendix we prove Proposition 1 in the case of a weighted (more general) version of the kKM formulation in (11):
\[
E_k^W(S) = -\sum_{p \in S} w_p \langle \phi(I_p) - c_1 \rangle^2 - \sum_{p \in S} w_p \langle \phi(I_p) - c_2 \rangle^2 \tag{A-1}
\]
where \(W := [w_p]\) denotes a vector of some given positive weights \(w_p\). We also prove proposition 4.

**Proof of proposition 1:** Let us consider the following function for a given \(S\):
\[
F_k^W(c_1, c_2) = \sum_{p \in S} w_p \langle \phi(I_p) - c_1 \rangle^2 + \sum_{p \in S} w_p \langle \phi(I_p) - c_2 \rangle^2 \tag{A-2}
\]
This function is convex w.r.t variables \(c_1\) and \(c_2\). Therefore its global minimas are reached at stationary points \(c^*_1\) and \(c^*_2\) verifying \(\frac{\partial F_k^W}{\partial c_1} = 0\) and \(\frac{\partial F_k^W}{\partial c_2} = 0\). It is easy to check these necessary conditions for a minimum of \(F_k^W\) are verified if and only if:
\[
c^*_1 = \frac{\sum_{q \in S} w_q \phi(I_q)}{\langle S, W \rangle} \tag{A-3}
\]
\[
c^*_2 = \frac{\sum_{q \in S} w_q \phi(I_q)}{\langle S, W \rangle}
\]
where \(\langle S, W \rangle = \sum_{p \in S} w_p\). Now notice the following:
\[
F_k^W(c^*_1, c^*_2) = C(\Omega) + E_k^W(S) \leq F_k^W(c_1, c_2) \forall c_1, c_2 \tag{A-4}
\]
where \(C(\Omega) = \sum_{p \in S} w_p k_{pp}\) is a constant independent of segmentation \(S\). The inequality in (A-4) is due to the fact that solution \(\{c^*_1, c^*_2\}\) is the global minimum for \(F_k^W(c_1, c_2)\). The equality in the first line comes from expanding the Euclidean distances in \(F_k^W(c^*_1, c^*_2)\) and replacing dot products \(\langle \phi(I_p), \phi(I_q) \rangle\) by kernels \(k_{pq}\). This yields the following bounds on the weighted kKM formulation in (A-1):
\[
E_k^W(S) \leq F_k^W(c_1, c_2) - C(\Omega) \forall c_1, c_2 \tag{A-5}
\]
Given current solution \(S_t\), we apply inequality (A-5) to \(c_1 = \mu_s = \frac{\sum_{q \in S_t} w_q \phi(I_q)}{\langle S_t, W \rangle}\) and \(c_2 = \mu_s = \frac{\sum_{q \in S_t} w_q \phi(I_q)}{\langle S_t, W \rangle} + 2\sum (\Omega) + 2\). It is easy to verify that this bound is equal to \(E_k^W\) at \(S_t\). This proves proposition 1. Furthermore, from the second and third lines of (A-6), one can easily see that the standard point re-assignment step in kernel k-means (Step II in proposition 2) optimizes globally the auxiliary function.

**Proof of proposition 4:** It suffices to see the following:
\[
E_k^W(S) = \frac{1}{2} A + \lambda \cdot diag(W), \text{ we have } E_k^W(S) = E_k^W(S) + 2\lambda \tag{A-7}
\]
Figure 26: Motion segmentation using our framework for the sequence horses01 in FBMS-59 dataset [18]. 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.
Figure 27: Multi-label motion segmentation using our framework for the sequence ducks01 in FBMS-59 dataset [18]. This video is challenging since the ducks here 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.
Figure 28: Motion segmentation for image 000079_10 from KITTI [71] 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.

Then, because $k$ is psd $\forall \lambda \geq -\lambda_0(A^W)$, we can apply the bound in (A-6) to the $E_k^W$ that appears in (A-7). This yields:

$$E_k^W(S) \leq F_t(S, \lambda) \quad \forall \lambda \geq -\lambda_0(A^W) \quad (A-8)$$

It is easy to check that $E_A^W(S_t) \leq F_t(S_t, \lambda) \quad \forall \lambda$.

**APPENDIX B (proof of Gini Bias Theorem 1)**

Let $d_\Omega$ be a continuous probability density function over domain $\Omega \subseteq \mathcal{R}^n$ defining conditional density

$$d_\Omega(x) := d_\Omega(x| x \in S) \quad (B-1)$$

for any non-empty subset $S \subset \Omega$ and expectation

$$E_z := \int z(x) d_\Omega(x) dx$$

for any function $z : \Omega \rightarrow \mathcal{R}^1$.

Suppose $\Omega$ is partitioned into two sets $S$ and $\bar{S}$ such that $S \cup \bar{S} = \Omega$ and $S \cap \bar{S} = \emptyset$. Note that $S$ here and in the statement of Theorem 1 is not a discrete set of observations, which is what $S$ means in the rest of the paper. Theorem 1 states a property of a fully continuous version of Gini criterion (37) that follows from an additional application of
Monte-Carlo estimation allowing to replace discrete set cardinality $|S|$ by probability $w$ of a continuous subset $S$

$$w := \int_S d_{\Omega}(x)dx = \int d_{\Omega}(x) \cdot [x \in S]dx = E[x \in S].$$

Then, minimization of $E_C(S)$ in (37) corresponds to maximization of the following objective function

$$L(S) := w \int d_a^2(x)dx + (1 - w) \int d_a^2(x)dx.$$  \hspace{1cm} (B-2)

Note that conditional density $d_a$ in (B-1) can be written as

$$d_a(x) = d_{\Omega}(x) \cdot \left[ \frac{x \in S}{w} \right].$$  \hspace{1cm} (B-3)

where $[\cdot]$ is an indicator function. Eqs. (B-3) and (B-2) give

$$L(S) = \frac{1}{w} \int d_a^2(x)[x \in S]dx + \frac{1}{1 - w} \int d_a^2(x)[x \in S]dx.$$

Introducing notation

$$I := [x \in S] \quad \text{and} \quad F := d_{\Omega}(x)$$

allows to further rewrite objective function $L(S)$ as

$$L(S) = \frac{EF}{EI} + \frac{EF(1 - I)}{1 - EI}. $$

Without loss of generality assume that $\frac{EF(1 - I)}{1 - EI} \leq \frac{EF}{EI}$ (the opposite case would yield a similar result). We now need the following lemma.

**Lemma 1.** Let $a, b, c, d$ be some positive numbers, then

$$\frac{a}{b} \leq \frac{c}{d} \implies \frac{a + c}{b + d} \leq \frac{c}{d}. $$

*Proof.* Use reduction to a common denominator. \hfill \Box

Lemma 1 implies inequality

$$\frac{EF(1 - I)}{1 - EI} \leq \frac{EF}{EI} \leq \frac{EF}{EI}$$  \hspace{1cm} (B-4)

which is needed to prove the Proposition below.

**Proposition 5.** (*Gini-bias*) Assume that subset $S_\varepsilon \subset \Omega$ is

$$S_\varepsilon := \{ x : d_\Omega(x) \geq \sup_x d_\Omega(x) - \varepsilon \}.$$  \hspace{1cm} (B-5)

Then

$$\sup L(S) = \lim_{\varepsilon \to 0} L(S_\varepsilon) = Ed_\Omega + \sup_x d_\Omega(x).$$  \hspace{1cm} (B-6)

*Proof.* Due to monotonicity of expectation we have

$$\frac{EF}{EI} \leq \frac{E(I \sup_x d_\Omega(x))}{EI} = \sup_x d_\Omega(x).$$  \hspace{1cm} (B-7)

Then (B-4) and (B-7) imply

$$L(S) = \frac{EF}{EI} + \frac{EF(1 - I)}{1 - EI} \leq \sup_x d_\Omega(x) + EF.$$  \hspace{1cm} (B-8)

That is, the right part of (B-6) is an upper bound for $L(S)$.

Let $I_\varepsilon := \{ x \in S_\varepsilon \}$. It is easy to check that

$$\lim_{\varepsilon \to 0} \frac{EF(1 - I_\varepsilon)}{1 - EI_\varepsilon} = EF.$$  \hspace{1cm} (B-9)

Definition (B-5) also implies

$$\lim_{\varepsilon \to 0} \frac{EFL_\varepsilon}{EI_\varepsilon} = \sup_x d_\Omega(x).$$  \hspace{1cm} (B-10)

Finally, the limits in (B-9) and (B-10) imply

$$\lim_{\varepsilon \to 0} L(S_\varepsilon) = \lim_{\varepsilon \to 0} \frac{EF(1 - I_\varepsilon)}{1 - EI_\varepsilon} + \lim_{\varepsilon \to 0} \frac{EFL_\varepsilon}{EI_\varepsilon} = Ed_\Omega + \sup_x d_\Omega(x).$$

This equality and bound (B-8) prove (B-6). \hfill \Box

**APPENDIX C (K-modes and mean shift)**

Here we discuss some extra points about K-modes and standard mean-shift [41, 26, 29]. First of all, note that K-modes energy (19) follows from a weak kKM approach (Sec.1.2) for arbitrary positive normalized kernels. Such kernels define different Parzen densities, but they all lead to energy (19) where optimal $\mu_a$ and $\mu_b$ are modes of the corresponding densities. Different kernels give different modes.

Many optimization methods can be used for K-modes energy. For example, it is possible to use iterative (block-coordinate descent) approach typical of K-means methods: one step reclusters points and the other step locally refines the modes, e.g. using mean-shift operation [87]. For better optimization, local refinement of the mode in each cluster can be replaces by the best mode search tracing all points within each cluster using mean-shift. RANSAC-like sampling procedure can be used for some compromise between speed and quality. It is also possible to use exhaustive search for the strongest mode in each cluster over observed discrete features and then locally refine each cluster’s mode with mean-shift.
It is also interesting that discrete version of K-modes for histograms [49, 25] define mode \( \mu = (\mu_1, ..., \mu_j, ...) \) as a combination of marginal modes for all attributes or dimensions \( j \). Implicitly, they use distortion \( \| \|_k \) for discrete kernel \( k(x, y) = \sum_j [x^j - y^j] \) where \([\cdot]\) are Iverson brackets. Marginal modes could be useful for aggregating sparse high-dimensional data.

Analogously, we can also define a continuous kernel for marginal modes as

\[
k(x, y) = \sum_j e^{-\frac{(x^j - y^j)^2}{2z_j^2}} . \tag{C-1}
\]

Note that this is different from the standard Gaussian kernel

\[
e^{-\frac{(x - y)^2}{2x^2}} = \prod_j e^{-\frac{(x^j - y^j)^2}{2z_j^2}},
\]

which leads to regular modes energy (19). It is easy to check that kernel (C-1) corresponds to weak \( k \)KM energy

\[
- \sum_j \sum_{p \in S} e^{-\frac{(1 - \mu p)^2}{2z_j^2}} - \sum_j \sum_{p \in S} e^{-\frac{(1 - \mu p)^2}{2z_j^2}}
\]

\[
= -|S| \cdot \sum_j \mathcal{P}_k^j(\mu_p | S) - |S| \cdot \sum_j \mathcal{P}_k^j(\mu_p | S)
\]

where \( \mathcal{P}_k^j \) is a marginal Parzen density for dimension \( j \).

APPENDIX D (Explicit data embedding \( \phi \))

This section uses some standard methodology [31] to build finite-dimensional embedding \( \phi(I_p) \) with exact or approximate isometry (9) to any given positive definite kernel \( k \) over finite data set \( \{I_p | p \in \Omega \} \). As discussion in Sec.1.2, many \( k \)KM methods prefer to work directly with kernels \( k \) and energy (11) in order to avoid formulation (6) with explicit high-dimensional embedding \( \phi(I_p) \). Indeed, the Mercer theorem justifies such kernel trick by establishing conceptual equivalence between p.d. kernels and Hilbert embeddings. Nevertheless, data embeddings \( \phi(I_p) \) could be useful and many clustering techniques explicitly construct them [89, 79, 84, 6, 4, 103]. In particular, if dimensionality of the embedding space is relatively low than basic iterative K-means procedure minimizing (6) could be more efficient than optimization methods for the equivalent quadratic formulation (11). Even when working with a given kernel \( k \) it may be algorithmically helpful to build the corresponding isometric embedding \( \phi \). Below we discuss embeddings in \( \mathcal{R}^m \) (\( m \leq |\Omega| \)) allowing to approximate several standard pairwise clustering criteria via basic K-means.

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\(^{13}\)

\[
K_{pq} = k(I_p, I_q)
\]

of size \( |\Omega| \times |\Omega| \). The eigen decomposition of this matrix

\[
K = V^T \Lambda V
\]

involves diagonal matrix \( \Lambda \) with non-negative eigen values and orthogonal matrix \( V \) whose rows are eigen vectors, see Fig.29(a). Non-negativity of the eigen values is important for obtaining decomposition \( \Lambda = \sqrt{\Lambda} \cdot \sqrt{\Lambda} \) allowing to define the following Euclidean space embedding

\[
\phi(I_p) := \sqrt{\Lambda} V_p \quad \in \mathcal{R}^{|\Omega|} \tag{D-1}
\]

where \( V_p \) are column of \( V \), see Fig.29(a). This embedding satisfies isometry (9) and (8) since

\[
\langle \phi(I_p), \phi(I_q) \rangle = (\sqrt{\Lambda} V_p)^T (\sqrt{\Lambda} V_q) = K_{pq} = k(I_p, I_q).
\]

\(^{13}\)If \( k \) is given as a continuous kernel \( k(x, y) : \mathcal{R}^N \times \mathcal{R}^N \rightarrow \mathcal{R} \) matrix \( K \) can be seen as its restriction to finite data set \( \{I_p | p \in \Omega \} \subseteq \mathcal{R}^N \).
Note that (D-1) defines a simple finite dimensional embedding \( \phi(I_p) \) only for subset of points \( \{I_p \mid p \in \Omega \} \) in \( \mathbb{R}^N \) based on a discrete kernel, \( K_{pq} \), \( \mathcal{K} \). In contrast, Mercer’s theorem should produce a more general infinite dimensional Hilbert embedding \( \phi(x) \) for any \( x \in \mathbb{R}^N \) 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, \( \phi(I_p) \) in (D-1) has dimension \( |\Omega| \), which is much larger than the dimension of points \( I_p \), e.g. 3 for colors in RGB.

The embedding (D-1) satisfying isometry (9) and (8) is not unique. For example, any decomposition \( \mathcal{K} = G^T G \), e.g. Cholesky [43], defines a mapping \( \phi_G(I_p) := G_p \) with desired properties. Also, rotational matrices \( R \) generate a class of isometric embeddings \( \phi_R(I_p) := R \phi(I_p) \).

It is easy to build lower dimensional embeddings \( \phi(I_p) \) by weakening the exact isometry requirements (8,9) following the standard multi-dimensional scaling (MDS) methodology [31], as detailed below. Consider given rank \( m < |\Omega| \) approximation \( \tilde{\mathcal{K}} \) for kernel matrix \( \mathcal{K} \) minimizing Frobenius norm errors [31]

\[
||\mathcal{K} - \tilde{\mathcal{K}}||_F := \sum_{p,q \in \Omega} (K_{pq} - \tilde{K}_{pq})^2. \tag{D-2}
\]

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

\[
\tilde{\mathcal{K}} = (V^m)^T \Lambda^m V^m
\]

where \( V^m \) is a submatrix of \( V \) including \( m \) rows corresponding to the largest \( m \) eigen values of \( \mathcal{K} \) and \( \Lambda^m \) is the diagonal matrix of these eigen values, see Fig.29(b). The corresponding minimum Frobenius error is given by the norm of zeroed out eigen values

\[
||\mathcal{K} - \tilde{\mathcal{K}}||_F = \sqrt{\lambda_{m+1}^2 + \cdots + \lambda_{|\Omega|}^2} \tag{D-3}
\]

It is easy to check that lower dimensional embedding

\[
\tilde{\phi}(I_p) := \sqrt{\Lambda^m} V^m_p \in \mathbb{R}^m \tag{D-4}
\]

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

\[
\langle \tilde{\phi}(I_p), \tilde{\phi}(I_q) \rangle = \tilde{K}_{pq}.
\]

Examples of low-dimensional approximate isometry embeddings (D-4) for a Gaussian kernel are shown in Fig.30. Note that \( \tilde{\phi}(I_p) \in \mathbb{R}^m \) in (D-4) can be directly obtained from \( \phi(I_p) \in \mathbb{R}^{|\Omega|} \) in (D-1) by selecting coordinates corresponding to dimensions of the largest \( m \) eigen values.

According to (D-3) lower dimensional embedding \( \tilde{\phi}(I_p) \) in (D-4) is nearly-isometric to kernel matrix \( \mathcal{K} \) if ignored dimensions have sufficiently small eigen values. Then (D-4) may allow efficient approximation of kernel K-means. For example, if sufficiently many eigen values are close to zero then a small rank \( m \) approximation \( \tilde{\mathcal{K}} \) will be sufficiently accurate. In this case, we can use basic iterative K-means procedure directly in \( \mathbb{R}^m \) with \( O(|\Omega| m) \) complexity of each iteration. In contrast, each iteration of the standard kernel K-means (11) is \( O(|\Omega|^2) \) in general [14].

There is a different way to justify approximate low-dimensional embedding \( \tilde{\phi}(I_p) \) ignoring small eigen value dimensions in \( \phi(I_p) \). The exact \( k \)KM for \( \mathcal{K} \) corresponds to a basic K-means for points \( \phi(I_p) \) in (D-1). This is equivalent to basic K-means in the space of columns \( V_p \) in orthonormal matrix \( V \) using weighted distance measure

\[
||V_p - \mu||_A^2 := \sum_{i=1}^{|\Omega|} \lambda_i ((V_p)_i - \mu[i])^2 = ||\phi(I_p) - \sqrt{\Lambda} \mu||^2
\]

where index \( [i] \) specifies coordinates of the column vectors. An approximation of \( k \)KM can ignore coordinates for small enough eigen values with low weight in the distance measure above. This is equivalent to K-means for points (D-4).

**Relation to spectral clustering methods:** Our approximation of \( k \)KM via basic K-means over explicit low-dimensional embedding (D-4) is closely related to popular spec-
Table 6: Spectral relaxation (a) and kKM formulations (b) for common pairwise clustering criteria: average association (AA), ratio cut (RC), and normalized cut (NC) for given 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 on the left in (a), spectral methods use (as in \cite{84}) an embedding, see $\tilde{\phi}(I_p)$ in (a). For easier comparison, (a) also shows equivalent representations of this embedding based on the same eigen decompositions $V^T A V$ as those used for isometry eigenmaps in (b). In contrast to (a), K-means embeddings in (b) are derived from justified approximations of the original non-relaxed AA, RC, or NC clustering problems. Note that NC corresponds to a weighted case of K-means with data point weights $w_p = d_p$. See \cite{4, 36} and Section 3.

### Spectral relaxation [98]

| Method | Formula | Description |
|--------|---------|-------------|
| AA $A\mathbf{u} = \lambda \mathbf{u}$ | $\phi(I_p) := U_p^T V_p$ | Unit eigenvectors $\mathbf{u}$ of $A$; $V$ is diagonal. |
| RC $(D-A)\mathbf{u} = \lambda \mathbf{u}$ | $\phi(I_p) := U_p^T V_p$ | Laplacian eigenvectors $\mathbf{u}$; $V$ is diagonal. |
| NC $(D-A)\mathbf{u} = \lambda D\mathbf{u}$ | $\phi(I_p) := U_p^T (V^T D^{-\frac{1}{2}})^m$ | Normalized cut eigenvectors $\mathbf{u}$; $V$ is diagonal. |

(a) Spectral methods: relaxation and discretization

### kKM approach: exact kernels and isometric Euclidean embeddings

| Method | Formula | Description |
|--------|---------|-------------|
| AA $K = A$ | $\phi(I_p) := \sqrt{\lambda} V_p^m$ | As in \cite{84}. |
| RC $K = \sigma I + A - D$ | $\phi(I_p) := \sqrt{\sigma I - \lambda} V_p^m$ | |
| NC $K = D^{-1} A D^{-1}$, $w_p = d_p$ | $\tilde{\phi}(I_p) := [\sqrt{\lambda} V m D^{-\frac{1}{2}}]^m \equiv \sqrt{\frac{1}{d_p} \lambda} V_p$ | |

(b) kKM approach: exact kernels and isometric Euclidean embeddings
arbitrary affinities via [84] as in (15). Note that the eigenvectors, i.e. the rows of matrix $V$ illustrated in Fig.29, also solve the spectral relaxation for $AA$, see Tab.6(a). However, ad hoc discretization by K-means over points $V^K_p$ may substantially differ from the result for embedding (D-4).

**Ratio cut (RC):** As discussed in [37], ratio cut\(^{15}\) clustering for affinity $A$ is equivalent to minimizing $\frac{1}{2} \phi^T \Lambda \phi$ objective with kernel $\bar{K} = \sigma I + A - D$ where $D$ is a diagonal matrix of node degrees $d_p := \sum_q A_{pq}$. Diagonal shift $\sigma I$ is needed to guarantee positive definiteness of the kernel. Eigen decomposition for $D - A = V^T \Lambda V$ implies $\bar{K} = V^T (\sigma I - \Lambda) V$. Therefore, the corresponding rank-$m$ approximate isometry embedding $\sqrt{\sigma} \tilde{\text{I}}^m - \bar{\Lambda}^m V^m = \tilde{\text{K}}_p$ in Tab.6(b) uses eigenvectors, i.e. the rows of matrix $V^m$, that also solve the standard spectral relaxation of RC criteria in Tab.6(a). But, discretization via K-means over $V^K_p$ may differ from the results for the justified embedding.

**Normalized cut (NC):** According to [4, 36] and a simple derivation in Sec.3, normalized cut clustering criterion for affinity $A$ is equivalent to weighted $k$KM with kernel $\bar{K} = D^{-\frac{1}{2}}A D^{-\frac{1}{2}}$ and node weights $w_p = d_p$ based on their degree. Weighted $k$KM can be interpreted as weighted K-means in the embedding space where $w_p$ weighs each data point $\phi(I_p)$ within its cluster as in (27,28). Most algorithms for K-means can be trivially adapted to such point weights. The main issue is computing $m$-dimensional embeddings approximately isometric to $\bar{K}$. Note that previously discussed solutions $\bar{\phi}$ like (D-4) use eigen decomposition of matrix $\bar{K}$ to minimize the sum of quadratic errors between $\tilde{K}_{pq}$ and approximating kernel $\tilde{K}_{pq} = \langle \bar{\phi}(I_p), \bar{\phi}(I_q) \rangle$. While this solution could be still acceptable, 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 $\tilde{\text{K}}$ minimizing the sum of weighted squared errors w.r.t. $\bar{K}$:

$$\sum_{p,q \in \Omega} w_p w_q (\tilde{K}_{pq} - \tilde{\bar{K}}_{pq})^2 = \|D^{\frac{1}{2}} (\bar{K} - \bar{K}) D^{\frac{1}{2}}\|_F.$$  

Indeed, using $\bar{K} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ inside the expression on the right hand side, we get an equivalent Frobenius objective

$$\|D^{\frac{1}{2}} A D^{-\frac{1}{2}} - D^{\frac{1}{2}} \tilde{K} D^{\frac{1}{2}}\|_F.$$  

Considering rank-$m$ matrix $\tilde{M} := D^{\frac{1}{2}} \tilde{K} D^{\frac{1}{2}}$ as a new minimization variable, its optimal value $(V^m)^T \tilde{\Lambda}^m V^m$ follows from an eigen decomposition of matrix

$$D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \equiv V^T \Lambda V.$$  

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

$$\tilde{K} = D^{-\frac{1}{2}} (V^m)^T \tilde{\Lambda}^m V^m D^{-\frac{1}{2}}.$$ 

\(^{15}\)Ratio cut is also known as average cut.

It is easy to check that $m$-dimensional embedding

$$\tilde{\phi}(I_p) = \sqrt{\frac{1}{d_p}} \Lambda^m V^m$$  

is isometric to kernel $\tilde{K}$, that is $\langle \tilde{\phi}(I_p), \tilde{\phi}(I_q) \rangle = \tilde{K}_{pq}$. Therefore, Euclidean dot product affinities between points $\tilde{\phi}(I_p) \in \mathbb{R}^m$ minimize the weighted sum of errors wrt kernel $\tilde{K}$. Note that our method still needs weighted K-means for approximate embedding $\tilde{\phi}(I_p)$ with weights $w_p = d_p$.

**Embeddings in spectral relaxation methods for NC:** Despite similarity, there are noticeable differences between our low-dimensional embedding (D-6) provably approximating the kernel $\bar{K} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ in the exact $k$KM formulation of NC [4, 36] and common ad-hoc embeddings used for K-means discretization step in the spectral relaxation methods. For example, one such discretization heuristic [89, 96] presented on the right side of Tab.6(a) uses embedding $\tilde{\phi}(I_p)$ defined by the columns of matrix $U^K$ whose rows are the $K$ top (unit) eigenvectors of the standard eigen system on the left side. It is easy to verify that the rows of matrix $V D^{-\frac{1}{2}}$ are non-unit eigenvectors for the generalized eigen system for NC. The following relationship

$$\tilde{\phi}(I_p) = U^K \equiv [V^K D^{-\frac{1}{2}}]^m,$$

where operator $[\cdot]^m$ normalizes matrix rows, demonstrates certain differences between ad hoc embeddings used by many spectral relaxation methods in their heuristic K-means discretization step and our justified approximation embedding (D-6) shown in Tab.6(b). Note that our formulation scales each embedding dimension, i.e. rows in matrix $V^K D^{-\frac{1}{2}}$, according to eigen values 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, [7, 68, 2] use

$$\tilde{\phi}(I_p) := \lfloor \Lambda^{-\frac{1}{2}} U \rfloor_p^K$$

discretization of the relaxed NC solution. The motivation comes from the physics-based mass-spring system interpretation [7] of the generalized eigenvalue system.

Some spectral relaxation methods motivate their discretization procedure differently. For example, [101, 4] 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 (D-5) that we used above. Yu and Shi [101] represent the subspace via matrix

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

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

\[ \phi(I_p) := \sqrt{\frac{1}{d_p} V^m_p} \]  \hspace{1cm} (D-7)

and weights \( w_p = d_p \). This embedding is different from (D-6) only by eigen values scaling.

Interestingly, a footnote in [4] states that NC objective is equivalent to (weighted) K-means objective (27) for high-dimensional embedding

\[ \phi(I_p) = \frac{1}{d_p} G_p \in \mathbb{R}^{[2]} \]  \hspace{1cm} (D-8)

based on any decomposition \( A \equiv G^T G \). For example, our exact isometry map (D-6) for \( m = |\Omega| \) and \( G = \sqrt{A V D^k} \) is a special case. While [4] were first to reduce NC to K-means\(^\dagger\), their low-dimensional embedding \( \phi \) in (D-7) is derived to approximate the subspace of relaxed NC solutions. In contrast, our low-dimensional embeddings (D-6) directly approximate our exact isometry map \( \phi \) ignoring any relaxed solutions. It is not obvious if decomposition \( A \equiv G^T G \) for the exact embedding (D-8) can be used to find any approximate lower-dimensional embeddings like (D-6).

Low-dimensional optimization ideas: Our approximate isometry lower-dimensional embedding (D-6) suggests new principled versions of NC algorithm using weighted K-means directly in \( \mathbb{R}^m \) where \( m \) could be varied in order to obtain the lowest value of the original energy. Unlike the spectral methods, our choice of dimension \( m \) is not restricted to the number of needed clusters \( K \). However, our preliminary synthetic experiments in Fig. 31 show that the quality of optimization decreases as \( m \) gets much larger probably due to higher sensitivity of K-means to local minima in higher dimensions. This sensitivity is not compensated by the better quality of approximation of the true kernel/affinities for larger \( m \). Our observations are consistent with the results of Dhillon et al. [36, 37]. Their kernel K-means procedure is equivalent to K-means for the exact embedding \( \phi(I_p) \in \mathbb{R}^{[2]} \) and it is often gives weaker results than the lower-dimensional K-means discretizing the relaxed spectral solutions. We offer a new principled low-dimensional approach. Moreover, unlike previous methods our bound and pseudo-bound optimization allows to combine common pairwise clustering with MRF regularization.

Acknowledgements

We would like to thank Carl Olsson (Lund University) for highly stimulating discussions and useful feedback at different stages of the project. Anders Eriksson (Lund University) helped with related work. We also thank Jianbo Shi (UPenn) for his feedback and for his spectral-relaxation optimization code for normalized cuts.

References

[1] A. Adams, J. Baek, and M. A. Davis. Fast high-dimensional filtering using the permutohedral lattice. Computer Graphics Forum, 29(2):753–762, 2010. 16
[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 33(5):898–916, 2011. 20, 29, 30
[3] I. B. Ayed, A. Mitiche, and Z. Belhadj. Multiregion level set partitioning of synthetic aperture radar images. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 27(5):793—800, 2005. 2, 3
[4] F. Bach and M. Jordan. Learning spectral clustering. Advances in Neural Information Processing Systems (NIPS), 16:305–312, 2003. 27, 29, 30, 31
[5] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In International Conference on Knowledge Discovery and Data Mining (KDD), pages 1601–1608, August 2004. 3

\(^\dagger\)Objective (27) for points \( \phi(I_p) \) in \( \mathbb{R}^{[2]} \) is not manageable by a basic (weighted) K-means procedure. Dhillon et al. [36] later observed that the corresponding pairwise formulation (26) for \( A_{pw} = \langle \phi(I_p), \phi(I_q) \rangle \) can be addressed by the standard quadratic kernel K-means procedure.
[41] K. Fukunaga and L. D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21:32–40, 1975. 26
[42] M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, 2002. 4, 5, 13
[43] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012. 28
[44] J.-M. Gottfried, J. Fehr, and C. S. Garbe. Computing range flow from multi-modal kinect data. In *Advances in Visual Computing*, pages 758–767. Springer, 2011. 19
[45] V. Gulshan, V. Lempitsky, and A. Zisserman. Humanising grabbed: Learning to segment humans using the kinect. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1127–1133. IEEE, 2011. 19
[46] M. Hein, T. N. Lal, and O. Bousquet. Hilbertian metrics on probability measures and their application in svms. *Pattern Recognition*, LNCS 3175:270—277, 2004. 4, 7
[47] D. S. Hochbaum. 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. 6
[48] T. Hofmann and J. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(1):1–14, January 1997. 6, 9
[49] Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304, 1998. 9, 27
[50] H. Ishikawa. Exact optimization for Markov Random Fields with convex priors. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, 2003. 15
[51] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi. Kernel methods on riemannian manifolds with gaussian rbf kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, In press, 2015. 4
[52] I. Jermyn and H. Ishikawa. Globally optimal regions and boundaries as minimum ratio weight cycles. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(10):1075–1088, 2001. 6
[53] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1943–1950. IEEE, 2010. 17
[54] M. Kearns, Y. Mansour, and A. Ng. An Information-Theoretic Analysis of Hard and Soft Assignment Methods for Clustering. In *Conf. on Uncertainty in Artificial Intelligence (UAI)*, August 1997. 1, 2, 3, 9, 13, 14
[55] Y. Kee, M. Souiai, D. Cremers, and J. Kim. Sequential convex relaxation for mutual information-based unsupervised figure-ground segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4082–4089, 2014. 9
[56] J. Kim, J. W. F. III, A. J. Yezzi, M. Çetin, and A. S. Willsky. A nonparametric statistical method for image segmentation using information theory and curve evolution. *IEEE Transactions on Image Processing*, 14(10):1486–1502, 2005. 9
[57] P. Kohli, P. H. Torr, et al. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82(3):302–324, 2009. 21
[58] V. Kolmogorov. Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, October 2006. 15
[59] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. In *IEEE International Conference on Computer Vision (ICCV)*, 2007. 6, 12
[60] B. Kulis, S. Basu, I. Dhillon, and R. Mooney. Semi-supervised graph clustering: a kernel approach. *Machine Learning*, 74(1):1–22, January 2009. 7, 10, 21
[61] K. Lange, D. R. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1):1–20, 2000. 9
[62] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006. 21
[63] V. Lempitsky, A. Blake, and C. Rother. Image segmentation by branch-and-mincut. In *ECCV*, 2008. 16
[64] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *Int. Conference on Computer Vision (ICCV)*, pages 277–284, 2009. 16
[65] T. Li, S. Ma, and M. Oghara. Entropy-based criterion in categorical clustering. In *Int. Conf. on M. Learning*, 2004. 4
[66] G. Louppe, L. Wainenkel, A. Sutera, and P. Geurts. Understanding variable importances in forests of randomized trees. In *NIPS*, pages 431–439, 2013. 4, 13
[67] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 21
[68] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International journal of computer vision*, 43(1):7–27, 2001. 20, 30
[69] D. Martin, C. Fowlkes, D. Tal, and J. Malik. 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*, volume 2, pages 416–423. IEEE, 2001. 18
[70] K. McGuinness and N. E. O Connor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444, 2010. 17
[71] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 20, 25
[72] A. Mitiche and I. B. Ayed. *Variational and Level Set Methods in Image Segmentation*. Springer, 2010. 3
[83] X. Ren, L. Bo, and D. Fox. RGB-(D) scene labeling: A variational framework for active foreground extraction using iterated graph cuts. In Proceedings of the IEEE International Conference on Computer Vision, pages 1368–1376, 2009.

[88] B. Schölkopf. The kernel trick for distances. In Advances in Neural Information Processing Systems (NIPS), pages 301–307, 2001.

[89] J. Shi and J. Malik. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 22:888–905, 2000.

[90] K. K. Sung and T. Poggio. Example based learning for viewbased human face detection. IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI), 20:39–51, 1995.

[91] M. Tang, I. Ben Ayed, and Y. Boykov. Pseudo-bound optimization for binary energies. In European Conference on Computer Vision (ECCV), pages 691–707, 2014.

[92] V. Vapnik. Statistical Learning Theory. Wiley, 1998.

[93] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. International Journal of Computer Vision, 62(1-2):61–81, 2005.

[94] S. Vicente, V. Kolmogorov, and C. Rother. Joint optimization of segmentation and appearance models. In International Conf. on Computer Vision (ICCV), 2009.

[95] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In Computer Vision and Pattern Recognition (CVPR), pages 2217–2224, 2011.

[96] U. Von Luxburg. A tutorial on spectral clustering. Statistics and computing, 17(4):395–416, 2007.

[97] S. Wang and J. M. Siskind. Image segmentation with ratio cut. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 25(6):675–690, 2003.

[98] L. Wasserman. All of Nonparametric Statistics. Springer, 2006.

[99] T. Werner. A linear programming approach to max-sum problem: A review. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 29(7):1165–1179, 2007.

[100] L. Xu, W. Li, and D. Schuurmans. Fast normalized cut with linear constraints. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 2866–2873, 2009.

[101] S. Yu and J. Shi. Multiclass spectral clustering. In International Conference on Computer Vision (ICCV), 2003.

[102] S. X. Yu and J. Shi. Segmentation given partial grouping constraints. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 26(2):173–183, 2004.

[103] Y. Yu, C. Fang, and Z. Liao. Piecewise flat embedding for image segmentation. In Proceedings of the IEEE International Conference on Computer Vision, pages 1368–1376, 2015.

[104] S. C. Zhu and A. Yuille. 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, Sept. 1996.