Secrets of GrabCut and Kernel K-means

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

*Computer Science Department, 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. *[39, 8, 28, 10], is presented as a generalized "probabilistic" K-means energy [16] for color space clustering. This interpretation reveals some limitations,* e.g. *over-fitting. We propose an alternative approach to color clustering using kernel K-means energy with well-known properties such as non-linear separation and scalability to higher-dimensional feature spaces. Our bound formulation for kernel K-means allows to combine general pair-wise feature clustering methods with image grid regularization using graph cuts, similarly to standard color model fitting techniques for segmentation. Unlike histogram or GMM fitting [39, 28], our approach is closely related to average association and normalized cut. But, in contrast to previous pairwise clustering algorithms, our approach can incorporate any standard geometric regularization in the image domain. We analyze extreme cases for kernel bandwidth* (e.g. *Gini bias) and demonstrate effectiveness of KNN-based adaptive bandwidth strategies. Our kernel K-means approach to segmentation benefits from higher-dimensional features where standard model-fitting fails.*

1. Introduction and Motivation

Many standard segmentation methods combine regularization in the image domain with a likelihood term integrating color appearance models [2, 39, 8, 4, 28]. These appearance models are often treated as variables and estimated jointly with segmentation by minimizing energies like

$$-\sum_{i=1}^{K}\sum_{p\in S^i}\log P^i(I_p)+\lambda||\partial\mathbf{S}|| \qquad (1)$$

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 $||\partial\mathbf{S}||$ is the segmentation boundary length in Euclidean or some contrast sensitive image-weighted metric. This popular approach to unsupervised [39, 8] or supervised [28] segmentation combines smoothness or edge detection in the image domain with the color space clustering by *probabilistic K-means* [16], as explained later. The goal of this paper is to replace standard likelihoods in regularization energies like (1) with a new general term for clustering data points $\{I_p\}$ in the color (or other feature) space based on *kernel K-means*.

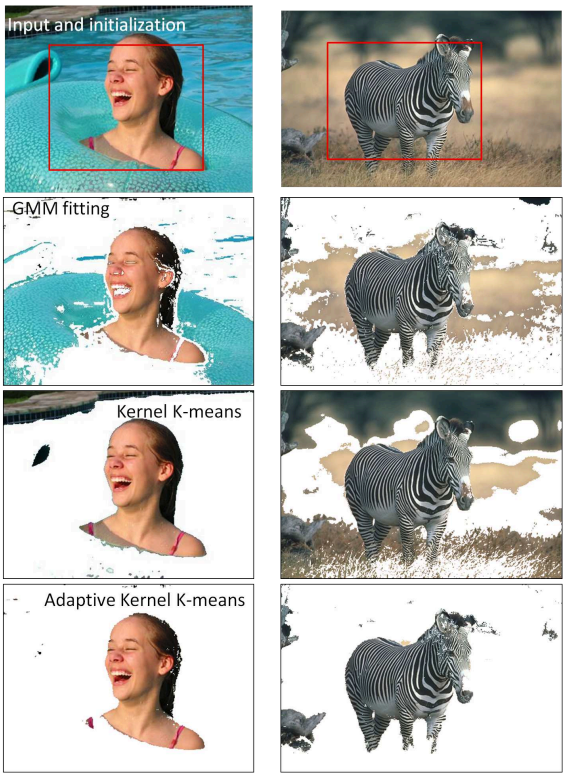


Figure 1: GrabCut vs. kernel K-means for color clustering (no smoothness or hard constraints). In contrast to kernel K-means, descriptive GMMs overfit the data even in $\mathcal{R}^3$.

Our methodology is general and applies to multi-label segmentation problems. For simplicity, our presentation is limited to a binary case $K=2$ where $S_p\in\{0,1\}$. We use $\mathbf{S}=S^1$ and $\bar{\mathbf{S}}=S^0$ to denote two segments.

A. **basic K-means**[1] (*e.g.* Chan-Vese [8])		
	$\sum_{p\in\mathbf{S}} \|I_p-\mu_{\mathbf{S}}\|^2 + \sum_{p\in\bar{\mathbf{S}}} \|I_p-\mu_{\bar{\mathbf{S}}}\|^2$	**Variance criterion**
$=$	$\frac{\sum_{pq\in\mathbf{S}} \|I_p-I_q\|^2}{2\|\mathbf{S}\|} + \frac{\sum_{pq\in\bar{\mathbf{S}}} \|I_p-I_q\|^2}{2\|\bar{\mathbf{S}}\|}$	$= \quad \|\mathbf{S}\|\cdot var(\mathbf{S}) + \|\bar{\mathbf{S}}\|\cdot var(\bar{\mathbf{S}})$
$\stackrel{c}{=}$	$-\sum_{p\in\mathbf{S}} \ln \mathcal{N}(I_p\|\mu_{\mathbf{S}}) - \sum_{p\in\bar{\mathbf{S}}} \ln \mathcal{N}(I_p\|\mu_{\bar{\mathbf{S}}})$	

more complex probability models

more complex data representation

B. **probabilistic K-means** (*e.g.* [39, 31, 29, 28, 10])	C. **kernel K-means** (ours)
(i) *equivalent energy formulations:*	(i) *equivalent energy formulations:*
$\sum_{p\in\mathbf{S}} \|I_p-\theta_{\mathbf{S}}\|_d + \sum_{p\in\bar{\mathbf{S}}} \|I_p-\theta_{\bar{\mathbf{S}}}\|_d$	$\sum_{p\in\mathbf{S}} \|\phi(I_p)-\mu_{\mathbf{S}}\|^2 + \sum_{p\in\bar{\mathbf{S}}} \|\phi(I_p)-\mu_{\bar{\mathbf{S}}}\|^2$
$= \quad -\sum_{p\in\mathbf{S}} \ln \mathcal{P}(I_p\|\theta_{\mathbf{S}}) - \sum_{p\in\bar{\mathbf{S}}} \ln \mathcal{P}(I_p\|\theta_{\bar{\mathbf{S}}})$	$= \quad \frac{\sum_{pq\in\mathbf{S}} \|\|I_p-I_q\|\|_k^2}{2\|\mathbf{S}\|} + \frac{\sum_{pq\in\bar{\mathbf{S}}} \|\|I_p-I_q\|\|_k^2}{2\|\bar{\mathbf{S}}\|}$
	$\stackrel{c}{=} \quad -\frac{\sum_{pq\in\mathbf{S}} k(I_p,I_q)}{\|\mathbf{S}\|} - \frac{\sum_{pq\in\bar{\mathbf{S}}} k(I_p,I_q)}{\|\bar{\mathbf{S}}\|}$
(ii) *example:* descriptive models (histograms or GMM) yield high-order log-likelihood energy	(ii) *example:* normalized kernels (Gaussians) yield high-order Parzen density energy
$-\sum_{p\in\mathbf{S}} \ln \mathcal{P}_h(I_p\|\mathbf{S}) - \sum_{p\in\bar{\mathbf{S}}} \ln \mathcal{P}_h(I_p\|\bar{\mathbf{S}})$	$-\sum_{p\in\mathbf{S}} \mathcal{P}_k(I_p\|\mathbf{S}) - \sum_{p\in\bar{\mathbf{S}}} \mathcal{P}_k(I_p\|\bar{\mathbf{S}})$
$\approx \quad \|\mathbf{S}\|\cdot H(\mathbf{S}) + \|\bar{\mathbf{S}}\|\cdot H(\bar{\mathbf{S}})$ **Entropy criterion** this approximation is valid only for highly descriptive models	$\stackrel{c}{\approx} \quad \|\mathbf{S}\|\cdot G(\mathbf{S}) + \|\bar{\mathbf{S}}\|\cdot G(\bar{\mathbf{S}})$ **Gini criterion** this approximation is valid only for small-width normalized kernels[2]
$\mathcal{P}_h(\mathbf{S}) \equiv \mathcal{P}_h(\cdot\|\mathbf{S})$ - histogram (or GMM) for intensities in $\mathbf{S}$ $H(\mathbf{S})$ - entropy for intensities in $\mathbf{S}$	$\mathcal{P}_k(\mathbf{S}) \equiv \mathcal{P}_k(\cdot\|\mathbf{S})$ - kernel (Parzen) density for intensities in $\mathbf{S}$ $G(\mathbf{S})$ - Gini impurity for intensities in $\mathbf{S}$
(iii) *bound optimization:* auxiliary function at $\mathbf{S}^t$	(iii) *bound optimization:* auxiliary function at $\mathbf{S}^t$
$A_t(\mathbf{S}) = -\sum_{p\in\mathbf{S}} \ln \mathcal{P}_h(I_p\|\mathbf{S}^t) - \sum_{p\in\bar{\mathbf{S}}} \ln \mathcal{P}_h(I_p\|\bar{\mathbf{S}}^t)$	$A_t(\mathbf{S}) \approx -2\sum_{p\in\mathbf{S}} \mathcal{P}_k(I_p\|\mathbf{S}^t) - 2\sum_{p\in\bar{\mathbf{S}}} \mathcal{P}_k(I_p\|\bar{\mathbf{S}}^t)$
$= \quad \|\mathbf{S}\|\cdot H(\mathbf{S}\|\mathbf{S}^t) + \|\bar{\mathbf{S}}\|\cdot H(\bar{\mathbf{S}}\|\bar{\mathbf{S}}^t)$	$+ \quad \|\mathbf{S}\|\cdot\left[G(\bar{\mathbf{S}}^t) - G(\mathbf{S}^t)\right]$

Table 1: *K-means terms for color clustering* combined with (MRF) regularization in segmentation. Basic K-means (A) or Gaussian model fitting minimizes cluster variances. More complex model fitting (elliptic Gaussian, GMM, histograms) corresponds to *probabilistic K-means* (B) [16]. We propose *kernel K-means* (C) using more complex data representation.

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 [8] where probability models P are Gaussian with fixed variances. In this case, the likelihoods in (1) reduce to

$$\sum_{p\in\mathbf{S}} \|I_p-\mu_{\mathbf{S}}\|^2 + \sum_{p\in\bar{\mathbf{S}}} \|I_p-\mu_{\bar{\mathbf{S}}}\|^2 \qquad (2)$$

the sum of squared errors from each cluster mean. This is the standard K-means objective also known as *variance criterion* for clustering, Tab.1A. If both means and covariances for Gaussians are treated as variables, then (1) corresponds to the standard *elliptic K-means* energy [31, 29, 10].

Zhu-Yuille [39] and GrabCut [28] popularized even more complex probability models (GMM or histograms) for segmentation energies like (1). In this case the likelihood term corresponds to a more general *probabilistic K-means* (pKM) energy [16] for color clustering, see Table 1B

$$-\sum_{p\in\mathbf{S}} \log \mathcal{P}(I_p\|\theta_{\mathbf{S}}) - \sum_{p\in\bar{\mathbf{S}}} \log \mathcal{P}(I_p\|\theta_{\bar{\mathbf{S}}}) \qquad (3)$$

where variables θ are ML model parameters for each segment. Assuming $\mathcal{P}(\cdot|\theta)$ is a continuous density of a sufficiently descriptive class (e.g. GMM), information theoretic analysis in [16] shows that probabilistic K-means energy reduces to the standard *entropy criterion* for clustering

$$\approx \quad |\mathbf{S}|\cdot H(\mathbf{S}) + |\bar{\mathbf{S}}|\cdot H(\bar{\mathbf{S}}). \qquad (4)$$

Indeed, for any function $f(x)$ Monte-Carlo estimation gives $\sum_{p\in\mathbf{S}} f(I_p) \approx |\mathbf{S}|\cdot\int d(x)f(x)dx$ where d is a "true" density for points in $\mathbf{S}$. For $f(x) = -\log\mathcal{P}(x|\theta_{\mathbf{S}})$ and $d(x) \approx$

[1] We use $\stackrel{c}{=}$ and $\stackrel{c}{\approx}$ for "up to additive constant" relations.

[2] Optimal bandwidth for accurate Parzen density estimation is near data resolution [37]. Such kernel width is too small for good clustering, Sec.3.1.

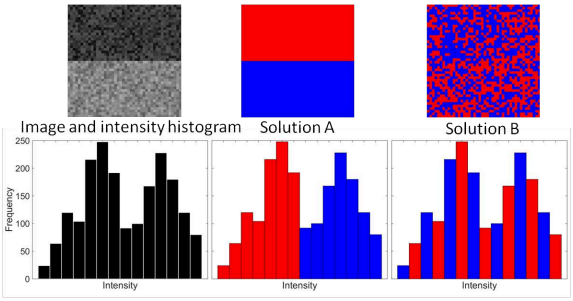


Figure 2: *Histograms in color spaces.* Entropy criterion (4) with histograms can not tell a difference between A and B: bin permutations do not change the histogram's entropy.

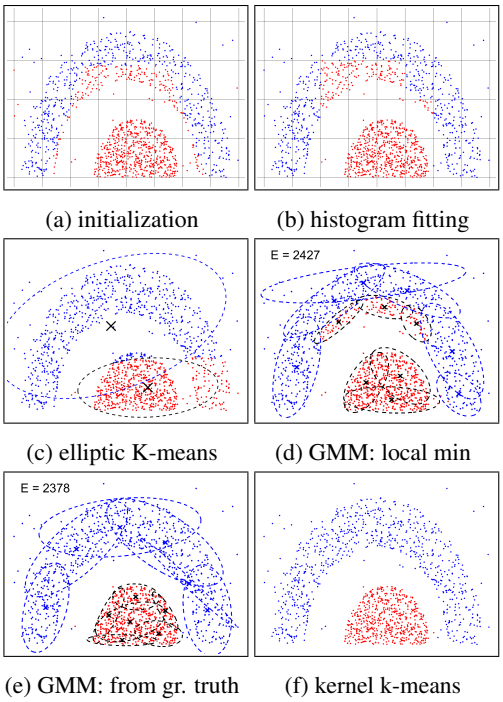
$\mathcal{P}(x|\theta_{\mathbf{S}})$ we get the clustering criterion above for the differential entropy $H(\mathbf{S}) := H(\mathcal{P}(\cdot|\theta_{\mathbf{S}}))$. For histograms $\mathcal{P}_h(\cdot|\mathbf{S})$ the entropy-based interpretation above is exact for discrete entropy $H(\mathbf{S}) := -\sum_x \mathcal{P}_h(x|\mathbf{S}) \cdot \log \mathcal{P}_h(x|\mathbf{S})$.

Intuitively, minimization of the entropy criterion (4) favors clusters with tight or "peaked" distributions. This criterion is widely used in categorical clustering [21] or decision trees [7, 22] where the entropy evaluates histograms over "naturally" discrete features. We show that the entropy criterion with either histograms or GMM densities has 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 (4) can not distinguish the quality of clusterings A and B in Fig.2; some permutation of bins can make B very similar to A.

In case of continuous density models, the problem of entropy criterion (4) is quite different since continuous (GMM or Parzen) densities preserve the notion of continuity in the color space. For example, optimal GMMs for clusterings A and B in Figure 2 will have sufficiently different (differential) entropy values. The main issue for pKM energy (3,4) with GMM densities is optimization. In this case high-order energy (3) requires joint optimization of variables S_p and many additional GMM parameters $\theta_{\mathbf{S}}$ yielding complex objective function with many local minima. Typical block coordinate descent methods [39, 28] iterating optimization of S and θ are very sensitive to initialization and easily overfit the data, see Figs.1 and 3(d). Better solutions exist, see Fig.3(e), but can not be found without good initialization.

These problems of probabilistic K-means with histograms or GMM in color spaces may explain why descriptive model fitting via pKM energy (3) is not a common clustering method in the learning community. Instead of probabilistic K-means they often use a different extension of K-means, that is *kernel K-means* in Table 1C.



(a) initialization (b) histogram fitting

(c) elliptic K-means (d) GMM: local min

(e) GMM: from gr. truth (f) kernel k-means

Figure 3: *Model fitting* (3) *vs kernel K-means* (10)*:* Histogram fitting always converges in one step assigning initially dominant bin label (a) to all points in the bin (b): energy (3) is minimal at any volume balanced solution with one label inside each bin [16]. Basic or elliptic K-means (one mode GMM) under-fit the data (c). Six mode GMMs over-fit (d) similarly to (b), but the problem is local minima since ground-truth initialization (e) yields lower energy (3). Kernel K-means energy (10) gives (f) even from (a).

1.2. Towards Kernel K-means (kKM)

We propose *kernel K-means* energy to replace the standard likelihood term (3) in common regularization functionals for segmentation (1). In machine learning, kernel K-means (kKM) is a well established data clustering technique [34, 25, 13, 11, 9, 15], which can identify complex structures that are non-linearly separable in input space. In contrast to *probabilistic K-means* using complex models, see Tab.1, this approach maps the data into a higher-dimensional Hilbert space using a nonlinear mapping ϕ. Then, the original non-linear problem often can be solved by simple linear separators in the new space.

Given a set of data points $\{I_p | p \in \Omega\}$ kernel K-means corresponds to the basic K-means in the embedding space. In case of two clusters (segments) $\mathbf{S}$ and $\overline{\mathbf{S}}$ this gives energy

$$E_k(\mathbf{S}) := \sum_{p \in \mathbf{S}} \|\phi(I_p) - \mu_{\mathbf{s}}\|^2 + \sum_{p \in \overline{\mathbf{S}}} \|\phi(I_p) - \mu_{\overline{\mathbf{s}}}\|^2. \quad (5)$$

where $\|.\|$ denotes the Euclidean norm, $\mu_{\mathbf{s}}$ is the mean of

segment $\mathbf{S}$ in the new space

$$\mu_{\mathbf{s}} = \frac{\sum_{q\in\mathbf{S}} \phi(I_q)}{|\mathbf{S}|} \tag{6}$$

and $|\mathbf{S}|$ denotes the cardinality of segment $\mathbf{S}$. Plugging $\mu_{\mathbf{s}}$ and $\mu_{\bar{\mathbf{s}}}$ into (5) 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

$$\langle \phi(x), \phi(y) \rangle := k(x, y) \tag{7}$$

and distance

$$\begin{aligned} \|\phi(x) - \phi(y)\|^2 &\equiv k(x, x) + k(y, y) - 2k(x, y) \\ &\equiv \|x - y\|_k^2. \end{aligned} \tag{8}$$

in the embedding space. *Mercer's theorem* [25] 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 (5) with pairwise distances $\|\phi(I_p) - \phi(I_q)\|^2$ in the embedding space implies one of the equivalent kKM formulations in Tab.1C(i)

$$E_k(\mathbf{S}) \equiv \frac{\sum_{pq\in\mathbf{S}} \|I_p - I_q\|_k^2}{2|\mathbf{S}|} + \frac{\sum_{pq\in\bar{\mathbf{S}}} \|I_p - I_q\|_k^2}{2|\bar{\mathbf{S}}|} \tag{9}$$

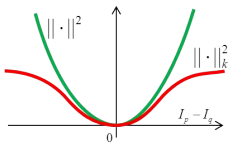
with isometric kernel distance $\|\|_k^2$ as in (8). This *Hilbertian metric*[3] replaces Euclidean metric inside the basic K-means formula in the middle of Tab.1A. Plugging (8) into (9) yields another equivalent (up to a constant) energy formulation for kKM directly using kernel k without any explicit reference to embedding $\phi(x)$

$$E_k(\mathbf{S}) \stackrel{c}{=} -\frac{\sum_{pq\in\mathbf{S}} k(I_p, I_q)}{|\mathbf{S}|} - \frac{\sum_{pq\in\bar{\mathbf{S}}} k(I_p, I_q)}{|\bar{\mathbf{S}}|}. \tag{10}$$

Kernel K-means energy (9) can explain the positive result for the standard Gaussian kernel $k = \exp \frac{-(I_p - I_q)^2}{2\sigma^2}$ in Fig.3(f). Gaussian kernel distance (red plot below)

$$\|I_p - I_q\|_k^2 \propto 1 - k(I_p, I_q) = 1 - \exp \frac{-(I_p - I_q)^2}{2\sigma^2} \tag{11}$$

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



Link to pair-wise clustering: Dhillion et al. [11, 17] first observed the equivalence between kernel k-means and popular spectral clustering criteria. For example, (10) is exactly the negative *average association* cost [11, 30] and (9) is closely related to *average distortion* [27]. Furthermore, [11] showed that weighted versions of energy (5) is equivalent to the popular *normalized cuts* cost [30].

1.3. 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 [8] different from probabilistic K-means [39, 28]. As special cases, our kKM color clustering term includes *normalized cuts* and other pairwise clustering criteria (see detailed discussion in [33]).

Our kKM approach to color clustering has several advantages over standard probabilistic K-means methods [39, 28] based on histograms or GMM. In contrast to histograms, kernels preserve color space continuity without breaking it into unrelated bins, see Fig.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, see Fig.3(d,f).

For high-dimensional data, kernel methods are a prevalent choice in the learning community as EM becomes intractable. Unlike GrabCut, our method extends to higher dimensional feature spaces, see Figure 9.

We analyze the extreme bandwidth cases (Sec.3). It is known that for wide kernels (approaching data range) kKM converges to basic K-means, which has bias to equal size clusters [16, 3]. It has been observed empirically that small-width kernels (approaching data resolution) show bias to compact dense clusters [30]. We provide a theoretical explanation for this bias by connecting kKM energy for small bandwidth with the *Gini criterion* for clustering (19). We analytically prove the bias to compact dense clusters for the continuous case of Gini, see Theorem 2, extending the previous result for histograms by Breiman [7].

We focus on the standard Gaussian and 0-1 kernels and evaluate locally adaptive bandwidth selection strategies avoiding problems reveled by our analysis above. Our tests show that fixed kernels are significantly outperformed by the standard clustering practice [36, 38] choosing local bandwidth from the distance to the *K-th nearest neighbor* (KNN). Efficient parallel implementation for our framework for general (*e.g.* adaptive) kernels is detailed in [33].

[3] Such metrics can be isometrically embedded into a Hilbert space [14].

2. Bound Optimization

In general, bound optimizers are iterative algorithms that optimize *auxiliary functions* (upper bounds) for a given energy $E(\mathbf{S})$ assuming that these auxiliary functions are more tractable than the original difficult optimization problem [18, 32]. $A_t(\mathbf{S})$ is an auxiliary function of $E(\mathbf{S})$ at current solution $\mathbf{S}_t$ (t is the iteration number) if:

$$E(\mathbf{S}) \leq A_t(\mathbf{S}) \quad \forall \mathbf{S} \tag{12a}$$

$$E(\mathbf{S}_t) = A_t(\mathbf{S}_t) \tag{12b}$$

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

$$E(\mathbf{S}_{t+1}) \leq A_t(\mathbf{S}_{t+1}) \leq A_t(\mathbf{S}_t) = E(\mathbf{S}_t).$$

For example, iterative optimization in standard GrabCut algorithm [28] was shown to be an optimizer of a cross-entropy bound [32], see Table 1B(iii).

Theorem 1. *The following is an auxiliary function for the* kKM *energy in* (10)

$$\begin{aligned} E_k(\mathbf{S}) &= -\frac{\sum_{pq\in\mathbf{S}} k_{pq}}{|\mathbf{S}|} - \frac{\sum_{pq\in\bar{\mathbf{S}}} k_{pq}}{|\bar{\mathbf{S}}|} \leq A_t(\mathbf{S}) \quad \textit{where} \\ A_t(\mathbf{S}) &= -2\sum_{p\in\mathbf{S}} \frac{\sum_{q\in\mathbf{S}_t} k_{pq}}{|\mathbf{S}_t|} - 2\sum_{p\in\bar{\mathbf{S}}} \frac{\sum_{q\in\bar{\mathbf{S}}_t} k_{pq}}{|\bar{\mathbf{S}}_t|} \\ &\quad + |\mathbf{S}|\frac{\sum_{pq\in\mathbf{S}_t} k_{pq}}{|\mathbf{S}_t|^2} + |\bar{\mathbf{S}}|\frac{\sum_{pq\in\bar{\mathbf{S}}_t} k_{pq}}{|\bar{\mathbf{S}}_t|^2}. \end{aligned} \tag{13}$$

Proof. See Appendix A in [33]. □

Our technical report [33] shows that the standard iterative kernel K-means algorithm [12] is implicitly a bound optimizer with our auxiliary function (13). However, without explicit use of our bound it is not clear how to combine kKM with MRF image-domain regularization. For example, to combine kKM with the Potts model [17] normalizes the corresponding pairwise constraints by cluster sizes. This alters the Potts model to a form accommodating trace-based formulation. In contrast, our bound-optimization interpretation allows to combine kKM energy and equivalent pairwise clustering energies [33] with any standard (*e.g.* MRF) or geometric regularization in XY domain.

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 $E_k(\mathbf{S})$ in (10):

$$E(\mathbf{S}) = E_k(\mathbf{S}) + \lambda R(\mathbf{S}) \tag{14}$$

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

$$R(\mathbf{S}) = \sum_{\{p,q\}\in\mathcal{N}} w_{pq}[s_p \neq s_q] \;\sim\; ||\partial\mathbf{S}|| \tag{15}$$

where $[\cdot]$ are *Iverson brackets* and $\mathcal{N}$ is the set of neighboring pixels. Pairwise weights w_{pq} are evaluated by the spatial distance and color contrast between pixels p and q as in [4].

Theorem 1 implies that $A_t(\mathbf{S}) + \lambda R(\mathbf{S})$ is an auxiliary function of high-order segmentation functional $E(\mathbf{S})$ in (14). Furthermore, this auxiliary function is a combination of unary (modular) term $A_t(\mathbf{S})$ and a sub-modular term $R(\mathbf{S})$. Therefore, at each iteration of our bound optimization algorithm, the global optimum of the bound can be efficiently obtained by max-flow algorithms [6]. Note that estimation of the unary part (13) of the auxiliary function $A_t(\mathbf{S}) + \lambda R(\mathbf{S})$ has quadratic complexity $O(N^2)$. Efficient implementation of this step is discussed in [33].

3. Parzen Analysis and Bandwidth Selection

This section discusses connections of kKM energy (10) 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 $\mathbf{S}$ can be expressed using normalized Gaussian kernels [1, 13]

$$\mathcal{P}_k(I_p|\mathbf{S}) = \frac{\sum_{q\in\mathbf{S}} k(I_p, I_q)}{|\mathbf{S}|}. \tag{16}$$

It is easy to see that kKM energy (10) is exactly the following high-order Parzen density energy

$$E_k(\mathbf{S}) \overset{c}{=} -\sum_{p\in\mathbf{S}} \mathcal{P}_k(I_p|\mathbf{S}) - \sum_{p\in\bar{\mathbf{S}}} \mathcal{P}_k(I_p|\bar{\mathbf{S}}). \tag{17}$$

3.1. Extreme bandwidth cases

Parzen energy (17) 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 σ approaching the data range. In this case Gaussian kernels k in (16) can be approximated (up to a scalar) by Taylor expansion $1 - \frac{\|I_p - I_q\|^2}{2\sigma^2}$. Then, Parzen density energy (17) becomes (up to a constant)

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

which is proportional to the pairwise formulation for the basic K-means or *variance criteria* in Tab.1A with Euclidean metric $\|\|$. That is, kKM 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 kKM 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 [16, 3].

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 kKM energy (17) as

$$E_k(\mathbf{S}) \overset{c}{\approx} -|\mathbf{S}| \cdot \langle \mathcal{P}_k(\mathbf{S}), d_{\mathbf{S}} \rangle - |\bar{\mathbf{S}}| \cdot \langle \mathcal{P}_k(\bar{\mathbf{S}}), d_{\bar{\mathbf{S}}} \rangle \quad (18)$$

where $\mathcal{P}_k(\mathbf{S})$ is kernel-based density (16) and $d_{\mathbf{S}}$ is a "true" density for intensities in $\mathbf{S}$. Approximation (18) follows directly from the same Monte-Carlo estimation argument given earlier below Eq. (4) with the only difference being $f = -\mathcal{P}_k(\mathbf{S})$ instead of $-\log \mathcal{P}(\theta_{\mathbf{S}})$.

If kernels have small bandwidth optimal for accurate Parzen density estimation[4] we get $\mathcal{P}_k(\mathbf{S}) \approx d_{\mathbf{S}}$ further reducing (18) to approximation

$$\overset{c}{\approx} \quad -|\mathbf{S}| \cdot \langle d_{\mathbf{S}}, d_{\mathbf{S}} \rangle - |\bar{\mathbf{S}}| \cdot \langle d_{\bar{\mathbf{S}}}, d_{\bar{\mathbf{S}}} \rangle$$

that proves the following property.

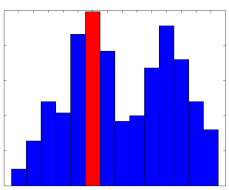
Property 1. *Assume small bandwidth Gaussian kernels optimal for accurate Parzen density estimation. Then kernel K-means energy* (17) *can be approximated by the standard* Gini criterion *for clustering [7]:*

$$E_G(\mathbf{S}) := |\mathbf{S}| \cdot G(\mathbf{S}) + |\bar{\mathbf{S}}| \cdot G(\bar{\mathbf{S}}) \quad (19)$$

where $G(\mathbf{S})$ *is the* Gini impurity *for the data points in* $\mathbf{S}$

$$G(\mathbf{S}) := 1 - \langle d_{\mathbf{S}}, d_{\mathbf{S}} \rangle \equiv 1 - \int_x {d_{\mathbf{S}}}^2(x) dx. \quad (20)$$

Similarly to entropy, *Gini impurity* $G(\mathbf{S})$ can be viewed as a measure of sparsity or "peakedness" for continuous or discrete distributions. Both Gini and entropy clustering criteria are widely used for decision trees [7, 22]. In this discrete context Breiman [7] analyzed theoretical properties of Gini criterion (19) for the case of histograms $\mathcal{P}_h$ where $G(\mathbf{S}) = 1 - \sum_x \mathcal{P}_h(x|\mathbf{S})^2$. 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 Brieman's result to the continuous Gini criterion (19)-(20).



[4] Bandwidth near inter-point distances avoids density oversmoothing.

Theorem 2. (Gini Bias) *Let* d_Ω *be a continuous probability density function over domain* $\Omega \subseteq \mathcal{R}^n$ *defining conditional density* $d_{\mathbf{S}}(x) := d_\Omega(x|x \in \mathbf{S})$ *for any non-empty subset* $\mathbf{S} \subset \Omega$. *Then, continuous version of Gini clustering criterion* (19) *achieves its optimal value at the partitioning of* Ω *into regions* $\mathbf{S}$ *and* $\bar{\mathbf{S}} = \Omega \setminus \mathbf{S}$ *such that*

$$\mathbf{S} = \arg\max_x d_\Omega(x).$$

Proof. See Appendix B in [33]. □

The bias to small dense clusters is practically noticeable for small bandwidth kernels, see Fig.4(d). Similar empirical bias to tight clusters was also observed in the context of average association in [30]. As kernel gets wider the continuous Parzen density (16) no longer approximates the true distribution $d_{\mathbf{S}}$ and Gini criterion (19) is no longer valid as an approximation for kKM energy (17). In pratice, *Gini bias* gradually disappears as bandwidth gets wider. This also agrees with the observations for wider kernel in average association [30]. As discussed earlier, in the opposite extreme case when bandwidth get very large (approaching data range) kKM converges to basic K-means or *variance criterion*, which has very different properties. Thus, kernel K-means properties strongly depend on the bandwidth.

3.2. Adaptive kernels and KNN

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 [16, 3]. In contrast, Gini criteria has bias to small dense clusters (Theorem 2). To avoid these biases kernel K-means should use kernels of width that is neither too small nor too large. Our experiments in Sec.4 compare different strategies with fixed and adaptive-width kernels. Equivalence of kernel-K-means to many standard clustering criteria such as *average distortion*, *average association*, *normalized cuts* (see Sec.1 and [33]) also suggest kernel selection strategies based on practices in prior art.

Section 4.2 in our technical report [33] shows that *Nash embedding* theorem implicitly connects adaptive bandwidth selection strategies with data space transformations changing local density of data points. In particular, we show that Gaussian kernel bandwidth can be selected based on any desired transformation of density $d'(d)$ according to formula

$$\sigma_p \sim \sqrt[n]{d'(d_p)/d_p} \quad (21)$$

where $d_p := d(I_p)$ is an observed local density for data points (in color space) near given point I_p. This formula computes adaptive bandwidth for any desired density transformation $d'(d)$. However, density equalizing transformation $d'(d) = const$ produces adaptive bandwidth

$$\sigma \sim \sqrt[n]{1/d_p} \sim \Delta_{KNN} \quad (22)$$

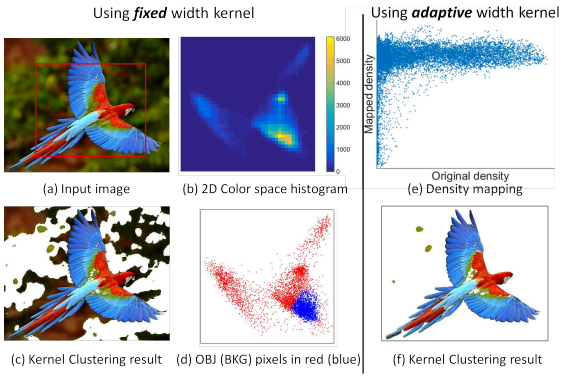


Figure 4: (a)-(d): Gini bias for fixed (small) kernel. (e) Empirical density transform (see [33]) for KNN kernels (22) corresponding to data density equalization. (f) Segmentation for such adaptive KNN kernels.

that worked better than other options we tested. Perhaps, density equalization addresses the Gini bias, see Fig.4(e,f). Interestingly, expression (22) is approximated by distance Δ_{KNN} to the *K-th nearest neighbor*, which is commonly used as adaptive bandwidth [36, 38]. Experiments in Sec.4 use KNN graph for adaptive kernel K-means (aKKM).

4. Experiments

We test our kernel-based segmentation (14) with fixed Gaussian kernel (KKM), its weighted version (wKKM) corresponding to basic *Normalized Cuts*, see [11], and adaptive bandwidth version (aKKM), see Sec.3.2, which closely relates to *Normalized Cuts* with adaptive kernels [36]. We use interactive segmentation as a simple generic application and compare to GrabCut [28] and Boykov-Jolly (BJ) [4] algorithms. We test (i) contrast-sensitive smoothness, (ii) Euclidean smoothness, and (iii) no smoothness to assess relative contributions of image domain regularization and color clustering to segmentation quality. We report the results on GrabCut and Berkeley datasets (50 and 100 images).

Implementation details: All algorithms use LAB color space. For GrabCut we use histograms as probability models [35, 19]. In boundary smoothness (15) we use standard contrast-based penalty $w_{pq} = \frac{1}{d_{pq}} e^{-0.5||I_p - I_q||_2^2/\beta}$ [4, 5] where β is the average of $||I_p - I_q||^2$ over 8-neighbors and d_{pq} is the distance between pixels p and q in the image plane. We use $w_{pq} = \frac{1}{d_{pq}}$ for tests with *Euclidean length* smoothness [5]. For fixed width Gaussian kernel, the bound in (13) is efficiently estimated using fast Bilateral filtering [26] with sampling rate half of the kernel width. Fixed Gaussian and KNN versions of our segmentation algorithm takes a few seconds per image on an average PC, but further speedups are possible with GPU.

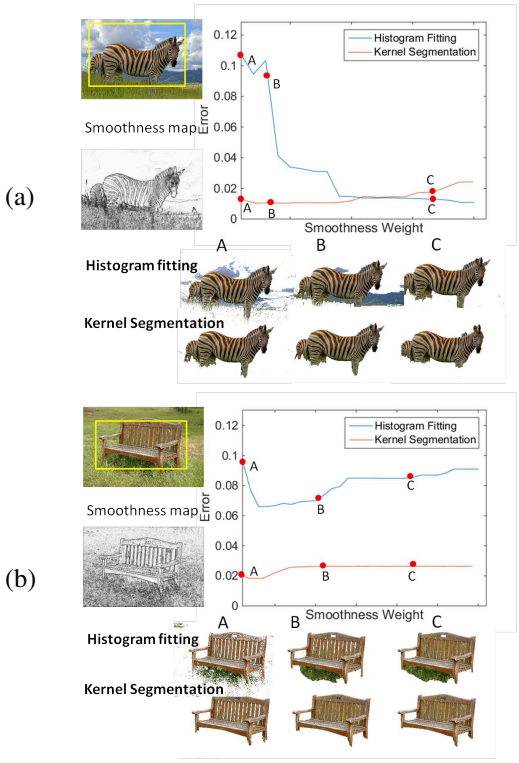


Figure 5: Illustration of robustness to smoothness weight.

4.1. Robustness to regularization weight

We first run all algorithms without smoothness. Then, we experiment with several values of λ for the contrast-sensitive edge term. In the experiments of Fig. 5 (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-based method 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 λ.

4.2. Segmentation on GrabCut & Berkeley datasets.

First, we report results on the GrabCut database (50 images) using the bounding boxes provided in [20]. 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

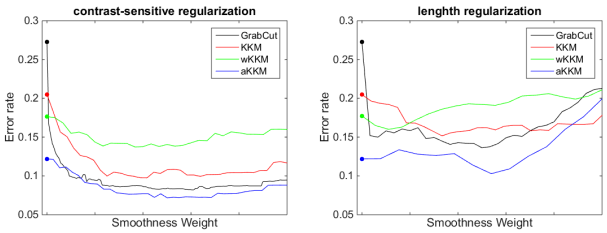


Figure 6: Average error vs. regularization weights for different algorithms on the GrabCut dataset.

boundary smoothness	color clustering term			
	GrabCut	KKM	wKKM	aKKM
none	27.2	20.4	17.6	**12.2**
Euclidean length	13.6	15.1	16.0	**10.2**
contrast-sensitive	8.2	9.7	13.8	**7.1**

Table 2: Box-based interactive segmentation (Fig.7). Error rates (%) are averaged over 50 images in GrabCut dataset.

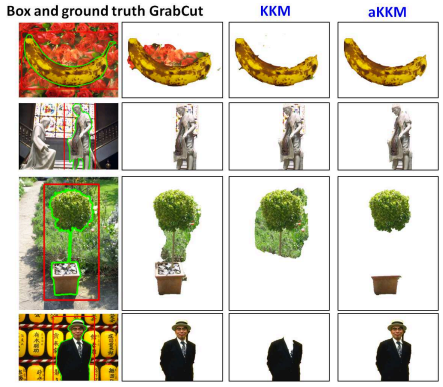


Figure 7: Sample results for GrabCut and our kernel methods with fixed & adaptive widths (KKM, aKKM), see Tab.2.

curves[5] in Fig.6. Table 2 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 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.

Figure 7 shows some results. The top row shows a failure case for GrabCut where the solution aligns with strong edges. The second row show a challenging image where our adaptive kernel method (aKKM) works well. The third and fourth rows shows failure cases for fixed-width kernel (KKM) due to Brieman's bias where image segments of uniform color are separated; see green bush and black suit. Our adaptive approach (aKKM) addresses this bias. We also tested seeds-based segmentation on a different database with ground truth, see Tab.3 and Figs.8.

[5]The smoothness weights for different energies are not directly comparable; Fig. 6 shows all the curves for better visualization.

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

Table 3: Interactive segmentation with seeds, Fig.8. Methods get the same seeds entered by 4 users. Error rates (%) are averaged over 82 images from Berkeley: 100 in [23] minus 18 images with multiple "identical" objects. (GrabCut and aKKM give 3.8 and 3.0 errors on the whole database.)

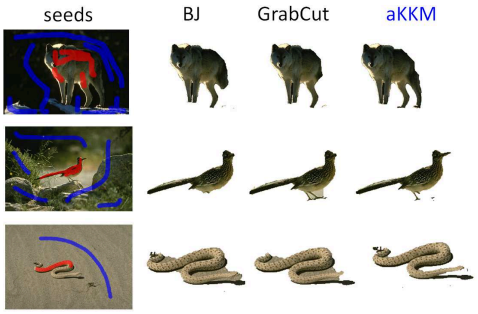


Figure 8: Sample results for BJ [4], GrabCut [28], and our adaptive kernel segmentation (aKKM), see Tab.3.

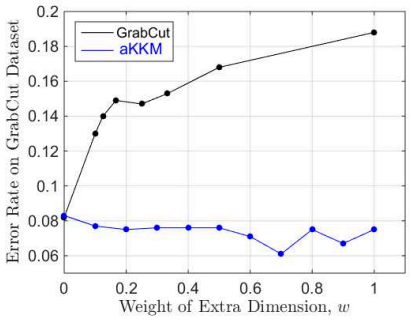


Figure 9: Scalability to high-dimensional features (3x3 patches). We augment color $I_p \in \mathcal{R}^3$ with weighted colors of pixel's 8 neighbors defining feature $[I_p, wI_{\mathcal{N}_p}] \in \mathcal{R}^{27}$. The plot shows errors w.r.t. relative effect of extra dimensions w compared to standard $\mathcal{R}^3$ color space ($w = 0$).

Figure 9 shows that aKKM benefits from extra information (*e.g.* texture) contained in higher-dimensional features (3x3 patches). In contrast, GrabCut fails due to severe overfitting. To handle features in $\mathcal{R}^{27}$ we used sparse histograms for GrabCut and approximate nearest neighbors [24] for aKKM. Thus, performance at $w = 0$ corresponding to standard color space $\mathcal{R}^3$ is slightly different from Tab.2.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, August 2006. 5

[2] A. Blake and A. Zisserman. *Visual Reconstruction*. Cambridge, 1987. 1

[3] Y. Boykov, H. Isack, C. Olsson, and I. B. Ayed. Volumetric Bias in Segmentation and Reconstruction: Secrets and Solutions. In *International Conference on Computer Vision (ICCV)*, December 2015. 4, 6

[4] Y. Boykov and M.-P. Jolly. *Interactive graph cuts* for optimal boundary & region segmentation of objects in N-D images. In *ICCV*, volume I, pages 105–112, July 2001. 1, 5, 7, 8

[5] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *International Conference on Computer Vision*, volume I, pages 26–33, 2003. 7

[6] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004. 5

[7] L. Breiman. Technical note: Some properties of splitting criteria. *Machine Learning*, 24(1):41–47, 1996. 3, 4, 6

[8] T. Chan and L. Vese. Active contours without edges. *IEEE Trans. Image Processing*, 10(2):266–277, 2001. 1, 2, 4

[9] R. Chitta, R. Jin, T. C. Havens, and A. K. Jain. Scalable kernel clustering: Approximate kernel k-means. In *KDD*, pages 895–903, 2011. 3

[10] A. Delong, A. Osokin, H. Isack, and Y. Boykov. Fast Approximate Energy Minization with Label Costs. *Int. J. of Computer Vision (IJCV)*, 96(1):1–27, January 2012. 1, 2

[11] I. Dhillon, Y. Guan, and B. Kulis. Kernel k-means, spectral clustering and normalized cuts. In *KDD*, 2004. 3, 4, 7

[12] I. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors: A multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Learning (PAMI)*, 29(11):1944–1957, November 2007. 5

[13] M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, 2002. 3, 5

[14] 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

[15] 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. 3

[16] 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, 4, 6

[17] B. Kulis, S. Basu, I. Dhillon, and R. Mooney. Semi-supervised graph clustering: a kernel approach. *Machine Learning*, 74(1):1–22, January 2009. 4, 5

[18] 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. 5

[19] V. Lempitsky, A. Blake, and C. Rother. Image segmentation by branch-and-mincut. In *ECCV*, 2008. 7

[20] 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. 7

[21] T. Li, S. Ma, and M. Ogihara. Entropy-based criterion in categorical clustering. In *Int. Conf. on M. Learning*, 2004. 3

[22] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts. Understanding variable importances in forests of randomized trees. In *NIPS*, pages 431–439, 2013. 3, 6

[23] K. McGuinness and N. E. Oconnor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444, 2010. 8

[24] M. Muja and D. G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36, 2014. 8

[25] K. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Trans. on Neural Networks*, 12(2):181–201, 2001. 3, 4

[26] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. In *Computer Vision–ECCV 2006*, pages 568–580. Springer, 2006. 7

[27] V. Roth, J. Laub, M. Kawanabe, and J. Buhmann. Optimal cluster preserving embedding of nonmetric proximity data. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(12):1540—1551, 2003. 4

[28] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cuts. In *ACM trans. on Graphics (SIGGRAPH)*, 2004. 1, 2, 3, 4, 5, 7, 8

[29] M. Rousson and D. R. A variational framework for active and adaptative segmentation of vector valued images. In *Workshop on Motion and Video Computing*, 2002. 2

[30] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactionson Pattern Analysis and Machine Intelligence (PAMI)*, 22:888–905, 2000. 4, 6

[31] K. K. Sung and T. Poggio. Example based learning for view-based human face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 20:39–51, 1995. 2

[32] M. Tang, I. B. Ayed, and Y. Boykov. Pseudo-bound optimization for binary energies. In *European Conference on Computer Vision (ECCV)*, pages 691–707, 2014. 5

[33] M. Tang, I. B. Ayed, D. Marin, and Y. Boykov. Secrets of GrabCut and Kernel K-means. In *arXiv:1506.07439*, June 2015. 4, 5, 6, 7

[34] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998. 3

[35] S. Vicente, V. Kolmogorov, and C. Rother. Joint optimization of segmentation and appearance models. In *International Conference on Computer Vision (ICCV)*, 2009. 7

[36] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007. 4, 7

[37] L. Wasserman. *All of Nonparametric Statistics*. Springer, 2006. 2

[38] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in NIPS*, pages 1601–1608, 2004. 4, 7

[39] 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. 1, 2, 3, 4