Fast Nonparametric Clustering with Gaussian Blurring Mean-Shift

Miguel Á. Carreira-Perpiñán MIGUEL@CSEE.OGI.EDU

Dept. of Computer Science & Electrical Engineering, OGI School of Science & Engineering, Oregon Health & Science University, 20000 NW Walker Road, Beaverton, OR 97006, USA

Abstract

We revisit Gaussian blurring mean-shift (GBMS), a procedure that iteratively sharpens a dataset by moving each data point according to the Gaussian mean-shift algorithm (GMS). (1) We give a criterion to stop the procedure as soon as clustering structure has arisen and show that this reliably produces image segmentations as good as those of GMS but much faster. (2) We prove that GBMS has convergence of cubic order with Gaussian clusters (much faster than GMS's, which is of linear order) and that the local principal component converges last, which explains the powerful clustering and denoising properties of GBMS. (3) We show a connection with spectral clustering that suggests GBMS is much faster. (4) We further accelerate GBMS by interleaving connected-components and blurring steps, achieving $2\times$–$4\times$ speedups without introducing an approximation error. In summary, our accelerated GBMS is a simple, fast, nonparametric algorithm that achieves segmentations of state-of-the-art quality.

Consider a dataset $\{\mathbf{x}_n\}_{n=1}^N \subset \mathbb{R}^D$ and define a kernel density estimate

$$p(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} K\left(\left\|\frac{\mathbf{x}-\mathbf{x}_n}{\sigma}\right\|^2\right) \tag{1}$$

with bandwidth $\sigma > 0$ and kernel $K(t)$, e.g. $K(t) = e^{-t/2}$ for the Gaussian or $K(t) = 1 - t$ if $t \in [0, 1)$ and 0 if $t \geq 1$ for the Epanechnikov. One way to find modes of p is to rearrange the stationary-point equation $\nabla p(\mathbf{x}) = \mathbf{0}$ into the iterative scheme $\mathbf{x}^{(\tau+1)} = \mathbf{f}(\mathbf{x}^{(\tau)})$ with

$$\mathbf{f}(\mathbf{x}) = \sum_{n=1}^{N} \frac{K'\left(\left\|\frac{\mathbf{x}-\mathbf{x}_n}{\sigma}\right\|^2\right)}{\sum_{n'=1}^{N} K'\left(\left\|\frac{\mathbf{x}-\mathbf{x}_{n'}}{\sigma}\right\|^2\right)}\mathbf{x}_n \tag{2}$$

where $K' = dK/dt$. The scheme converges to modes of p (fixed points of $\mathbf{f}$) under mild conditions. This algorithm, called *mean-shift algorithm*, originates in an idea of Fukunaga and Hostetler (1975) and has been recently developed by Cheng (1995), Carreira-Perpiñán (2000), Comaniciu and Meer (2002) and others. The mean-shift algorithm can be applied to clustering by declaring each mode of p as representative of one cluster, and assigning data point $\mathbf{x}_n$ (or any point $\mathbf{x} \in \mathbb{R}^D$) to the mode it converges to, $\mathbf{f}^\infty(\mathbf{x}_n)$. Since the algorithm does not depend on parameters such as step sizes, the clustering is deterministic given the bandwidth σ. The algorithm is nonparametric: it assumes neither a model for the clusters nor a number of clusters; the result depends only on σ. Mean-shift has proven particularly successful in image segmentation (Comaniciu & Meer, 2002), where its ability to deal with clusters of arbitrary shape and its user-friendliness are very attractive.

The algorithm proposed by Fukunaga and Hostetler (1975) was in fact different. Following Cheng (1995), we will call it *blurring mean-shift*. In blurring mean-shift, each point $\mathbf{x}_m$ of the dataset actually moves to the point $\mathbf{f}(\mathbf{x}_m)$ given by eq. (2). That is, given the dataset $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, for each $\mathbf{x}_m \in \mathbf{X}$ we obtain a new point $\tilde{\mathbf{x}}_m$ by applying one step of the mean-shift algorithm: $\tilde{\mathbf{x}}_m = \mathbf{f}(\mathbf{x}_m)$. Thus, one iteration of blurring mean-shift results in a new dataset $\tilde{\mathbf{X}}$ which is a blurred (shrunk) version of $\mathbf{X}$. By iterating this process we obtain a sequence of datasets $\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \dots$ (and a sequence of kernel density estimates $p^{(0)}(\mathbf{x}), p^{(1)}(\mathbf{x}), \dots$) where $\mathbf{X}^{(0)}$ is the original dataset and $\mathbf{X}^{(\tau)}$ is obtained by blurring $\mathbf{X}^{(\tau-1)}$ with one mean-shift step (see fig. 2). Focussing on the Epanechnikov kernel for computational efficiency, Fukunaga and Hostetler (1975) observed that this algorithm could be used for clustering and dimensionality reduction (or denoising), since points converge locally to cluster centroids or medial lines for appropriate values of σ. Cheng (1995) proved convergence of blurring mean-shift, as follows. (1) For kernels broad enough to cover the dataset $\mathbf{X}$ (e.g. infinite-support kernels such as the Gaussian) convergence is to a dataset $\mathbf{X}^{(\infty)}$ with all points coincident ($\mathbf{x}_1^{(\infty)} = \cdots = \mathbf{x}_N^{(\infty)}$), independent of σ. (2) For finite-support kernels and small enough σ, convergence is to several clusters with all points coinci-

Appearing in *Proceedings of the 23^{rd} International Conference on Machine Learning*, Pittsburgh, PA, 2006.

dent in each of them; the clusters depend on the value of σ. To our knowledge, no other work (theoretical or practical) exists on blurring mean-shift apart from these two papers.

Different kernels give rise to different versions of the mean-shift and blurring mean-shift algorithms. Much previous work (including Fukunaga and Hostetler 1975 and Cheng 1995) mostly uses the Epanechnikov kernel for computational efficiency, since the kernel evaluations involve only pairs of neighbouring points (at distance $< \sigma$) rather than all pairs of points (though the neighbours must still be found at each iteration), and convergence occurs in a finite number of iterations. However, in practice with mean-shift the Gaussian kernel produces better segmentations than the Epanechnikov kernel (Comaniciu & Meer, 2002). In this paper we focus on *Gaussian blurring mean-shift* (GBMS), where the kernel density estimate is a Gaussian mixture and the mean-shift iteration can be written in the following, elegant form (Carreira-Perpiñán, 2000):

$$p(n|\mathbf{x}^{(\tau)}) = \frac{\exp\left(-\frac{1}{2}\left\|(\mathbf{x}^{(\tau)} - \mathbf{x}_n)/\sigma\right\|^2\right)}{\sum_{n'=1}^{N} \exp\left(-\frac{1}{2}\left\|(\mathbf{x}^{(\tau)} - \mathbf{x}_{n'})/\sigma\right\|^2\right)} \quad (3a)$$

$$\mathbf{x}^{(\tau+1)} = \sum_{n=1}^{N} p(n|\mathbf{x}^{(\tau)})\mathbf{x}_n \quad (3b)$$

i.e., the new iterate is the data average under the posterior probabilities given the current iterate. Figures 1**A** and 1**B** give the algorithms for GMS and GBMS, respectively. Both algorithms require a final connected-components step after stopping the iterative scheme. This step merges all modes (GMS) or data points (GBMS) that lie within a small distance $\texttt{min\_diff} > 0$ from each other.

At first sight, GBMS might seem uninteresting since it eventually leads to a dataset with all points coincident for any starting dataset and bandwidth (Cheng, 1995). However, our practical experience shows that, first, the dataset very quickly collapses into clusters which depend on σ (see fig. 2), and then these clusters move towards each other relatively slowly. If we could stop the GBMS iteration at this point, the algorithm would have potential application to clustering. However, Fukunaga and Hostetler (1975) did not provide a stopping criterion, and simply stopping when the parameter update is small does not work reliably. Thus, the first question that we address is how to stop the iteration in a robust way so that we obtain meaningful clusters (section 1). We then show that, for Gaussian clusters, GBMS converges cubically (section 2), which explains its speed, and we show a connection with spectral clustering (section 3). We introduce an accelerated version of GBMS which produces the same clustering as GBMS with $2\times$–$4\times$ speedups (section 4). Finally (section 5), in experiments in the task of image segmentation, we demonstrate that our stopping criterion and accelerated GBMS reliably result in excellent clustering results, of quality comparable to mean-shift clustering but obtained much faster.

A. Gaussian mean-shift (GMS) algorithm

for $n \in \{1, \dots, N\}$ — For each data point
  $\mathbf{x} \leftarrow \mathbf{x}_n$ — Starting point
  $\forall n$: $p(n|\mathbf{x}) \leftarrow \frac{e^{-\frac{1}{2}\left\|\frac{\mathbf{x}-\mathbf{x}_n}{\sigma}\right\|^2}}{\sum_{n'=1}^{N} e^{-\frac{1}{2}\left\|\frac{\mathbf{x}-\mathbf{x}_{n'}}{\sigma}\right\|^2}}$ — Eq. (3a)
  repeat — Iteration loop
    $\mathbf{x} \leftarrow \sum_{n=1}^{N} p(n|\mathbf{x})\mathbf{x}_n$ — Update $\mathbf{x}$, eq. (3b)
    $\forall n$: $p(n|\mathbf{x}) \leftarrow \frac{e^{-\frac{1}{2}\left\|\frac{\mathbf{x}-\mathbf{x}_n}{\sigma}\right\|^2}}{\sum_{n'=1}^{N} e^{-\frac{1}{2}\left\|\frac{\mathbf{x}-\mathbf{x}_{n'}}{\sigma}\right\|^2}}$ — Eq. (3a)
  until $\mathbf{x}$'s update $< \texttt{tol}$
  $\mathbf{z}_n \leftarrow \mathbf{x}$ — Mode
end
connected-components($\{\mathbf{z}_n\}_{n=1}^N$,min_diff) — Clusters

B. Gaussian blurring mean-shift (GBMS) algorithm

repeat — Iteration loop
  for $m \in \{1, \dots, N\}$ — For each data point
    $\forall n$: $p(n|\mathbf{x}_m) \leftarrow \frac{e^{-\frac{1}{2}\left\|\frac{\mathbf{x}_m-\mathbf{x}_n}{\sigma}\right\|^2}}{\sum_{n'=1}^{N} e^{-\frac{1}{2}\left\|\frac{\mathbf{x}_m-\mathbf{x}_{n'}}{\sigma}\right\|^2}}$ — Eq. (3a)
    $\mathbf{y}_m \leftarrow \sum_{n=1}^{N} p(n|\mathbf{x}_m)\mathbf{x}_n$ — One GMS step, eq. (3b)
  end
  $\forall m$: $\mathbf{x}_m \leftarrow \mathbf{y}_m$ — Update whole dataset
until stop — See section 1
connected-components($\{\mathbf{x}_n\}_{n=1}^N$,min_diff) — Clusters

C. GBMS algorithm in matrix form

repeat — Iteration loop
  $\mathbf{W} = \left(\exp\left(-\frac{1}{2}\left\|\frac{\mathbf{x}_m-\mathbf{x}_n}{\sigma}\right\|^2\right)\right)_{nm}$ — Gaussian affinity matrix
  $\mathbf{D} = \operatorname{diag}\left(\sum_{n=1}^{N} w_{nm}\right)$ — Degree (normalising) matrix
  $\mathbf{X} = \mathbf{X}\mathbf{W}\mathbf{D}^{-1}$ — Update whole dataset
until stop — See section 1
connected-components($\{\mathbf{x}_n\}_{n=1}^N$,min_diff) — Clusters

D. Accelerated GBMS algorithm

$\mathbf{\Pi} = \frac{1}{N}\mathbf{I}$ — Equal weights
repeat — Iteration loop
  reduce $\mathbf{X}$, $\mathbf{\Pi}$ — See section 4
  $\mathbf{W} = \left(\exp\left(-\frac{1}{2}\left\|\frac{\mathbf{x}_m-\mathbf{x}_n}{\sigma}\right\|^2\right)\right)_{nm}$ — Gaussian affinity matrix
  $\mathbf{D} = \operatorname{diag}\left(\sum_{n=1}^{N} (\mathbf{\Pi}\mathbf{W})_{nm}\right)$ — Degree (normalising) matrix
  $\mathbf{X} = \mathbf{X}\mathbf{\Pi}\mathbf{W}\mathbf{D}^{-1}$ — Update whole dataset
until stop — See section 1
connected-components($\{\mathbf{x}_n\}_{n=1}^N$,min_diff) — Clusters

Figure 1. Pseudocode for all algorithms. **C** is **B** but rewritten to emphasise the relationship with spectral clustering, where $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ is a $D \times N$ matrix, and $\mathbf{W}$ and $\mathbf{D}$ are $N \times N$ matrices. For clarity, in **D** we omit the housekeeping necessary for relating original data points to the reduced set in $\mathbf{X}$. The stopping criterion in **B**–**D** is explained in section 1.

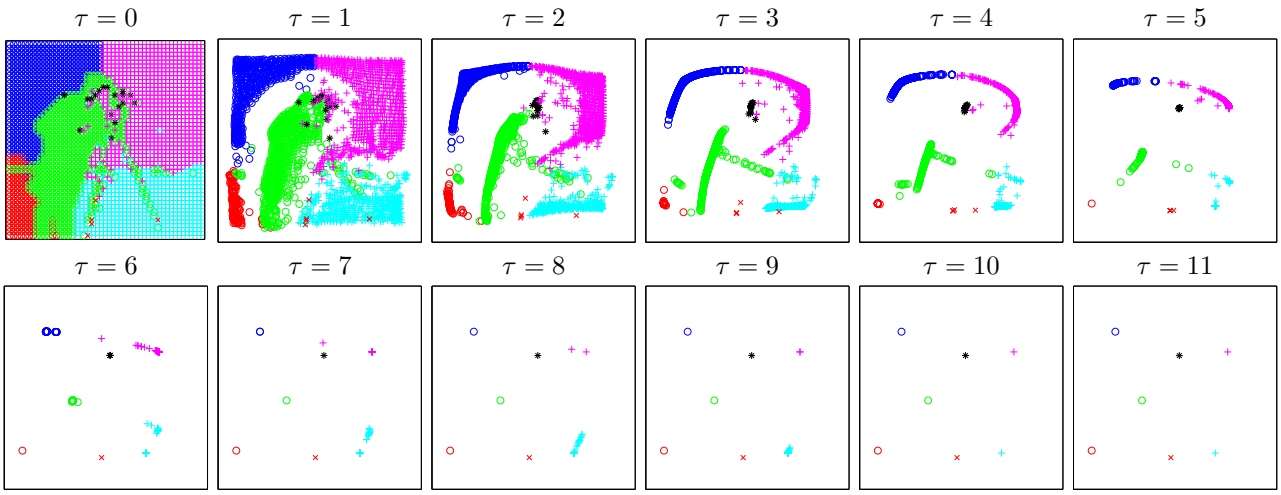

Figure 2. Sequence of datasets $\mathbf{X}^{(0)}, \ldots, \mathbf{X}^{(11)}$ obtained by GBMS for `cameraman` 50×50 (fig. 7) with $\sigma = 6$. The dataset contains $N = 2\,500$ points in 3D; the 2D projection on the spatial domain is shown, coloured according to the final 7 clusters. Note: (1) points very quickly move towards a centroid and collapse into it (clustering property); (2) for each cluster-to-be, the local direction of maximum variance collapses much more slowly than the lower-variance directions, producing linearly shaped clusters (denoising property) that straighten and shorten. Phase 2 (see section 1) starts at $\mathbf{X}^{(10)}$, where the dataset consists of 7 clusters of coincident points, and the stopping criterion of section 1 stopped GBMS at iteration $\tau = 11$. These clusters keep moving, so simply stopping when the average update (4) is less than `tol` usually results in stopping either too early or too late, depending on the `tol` value.

1. Stopping Criterion for GBMS

The typical behaviour of GBMS is shown in fig. 2 and consists of two phases. In *phase 1*, which lasts for a few iterations, points merge into extremely compact clusters (coincident points, to machine precision). This clustering structure depends on the dataset and on σ and corresponds to meaningful clusters for appropriate values of σ. In *phase 2*, which lasts from a few to hundreds of iterations, these clusters do not change but simply approach each other till they eventually merge at a single point (convergence of GBMS). Our objective is to design a stopping criterion (as opposed to a convergence criterion) that will stop the algorithm just after phase 1—neither too early that clusters have not compacted, nor too late that either clusters merge or we waste iterations where the structure does not change.

It might seem that stopping when the average update drops below a small tolerance `tol` (say 10^{-3}) would work:

$$\frac{1}{N}\sum_{n=1}^{N} e_n^{(\tau)} < \texttt{tol} \text{ where } e_n^{(\tau)} = \|\mathbf{x}_n^{(\tau)} - \mathbf{x}_n^{(\tau-1)}\| \quad (4)$$

but it does not in general. The reason is that clusters keep moving in phase 2, so that (1) the average update is not necessarily small compared with the average update in phase 1 iterations (so it cannot discriminate) and (2) its value varies widely depending on the dataset and σ. We also found statistics of the dataset such as its variance or entropy to be very unreliable no matter how small or large `tol` is.

The key to solve this is to realise that all points in one cluster behave equally in phase 2. Thus, at iteration τ the update $e_n^{(\tau)}$ for $n = 1 \ldots, N$ takes at most K different values (for K clusters), and a histogram of $\{e_n^{(\tau)}\}_{n=1}^N$ has K or fewer nonempty bins. We do not know K so we cannot use the value of the entropy (or the number of nonempty bins) to detect this situation reliably. Consider however the histogram of $\{e_n^{(\tau+1)}\}_{n=1}^N$: since points in each cluster have moved equally, the histogram at $\tau + 1$ will have the same number of nonempty bins with the same frequencies as for τ (not necessarily at the same bin locations). Thus the entropy of the two histograms (whatever its value is) will be the same, since the entropy is invariant to bin reorderings.

In summary, our stopping criterion is:

$$\begin{gathered}\left(|H(\mathbf{e}^{(\tau+1)}) - H(\mathbf{e}^{(\tau)})| < 10^{-8}\right) \text{ OR} \\ \left(\tfrac{1}{N}\textstyle\sum_{n=1}^{N} e_n^{(\tau+1)} < \texttt{tol}\right)\end{gathered} \quad (5)$$

where $H(\mathbf{e}) = -\sum_{i=1}^{B} f_i \log f_i$ is the entropy, f_i is the relative frequency of bin i (so $\sum_{i=1}^{B} f_i = 1$), and the bins span the interval $[0, \max(\mathbf{e})]$. The number of bins B should be larger than the number of clusters but smaller than N; we use $B = 0.9N$. As shown in section 5, our criterion makes the clustering result dependent only on the bandwidth σ and not on the specifics of the convergence criterion. Checking the criterion takes $\mathcal{O}(N)$, thus negligible overall.

2. Rate of Convergence of GBMS

We show that GBMS compresses a Gaussian cluster towards its mean with cubic convergence rate. Write the

Gaussian density as $\mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$. To eliminate the dependence on a particular dataset, we consider an infinite sample, the whole $\mathbb{R}^D$ distributed with density $q(\mathbf{x})$. Sums over data points such as (3b) become then integrals over $\mathbb{R}^D$. The kernel density estimate (1) is the convolution of the data distribution q with the kernel $K(\mathbf{x})=\mathcal{N}(\mathbf{x};\mathbf{0},\sigma^2\mathbf{I})$:

$$p(\mathbf{x})=\int_{\mathbb{R}^D} q(\mathbf{y})K(\mathbf{x}-\mathbf{y})\,d\mathbf{y}. \tag{6}$$

The new dataset after one blurring mean-shift iteration is

$$\tilde{\mathbf{x}}=\int_{\mathbb{R}^D} p(\mathbf{y}|\mathbf{x})\mathbf{y}\,d\mathbf{y}=\mathrm{E}\left\{\mathbf{y}|\mathbf{x}\right\} \qquad \forall \mathbf{x}\in\mathbb{R}^D \tag{7}$$

with the posterior probability of datapoint $\mathbf{y}$ given $\mathbf{x}$ given by Bayes' theorem: $p(\mathbf{y}|\mathbf{x})=K(\mathbf{x}-\mathbf{y})q(\mathbf{y})/p(\mathbf{x})$. Since translation and rotation do not affect the result, we take the dataset as zero-mean and diagonal covariance w.l.o.g. Then, $q(\mathbf{x})$, $K(\mathbf{x})$ and $p(\mathbf{x})$ factorise, so we consider each dimension separately. Letting $q(x)=\mathcal{N}(x;0,s^2)$ and $r=\frac{1}{1+(\sigma/s)^2}\in(0,1)$, we get the kernel density estimate

$$p(x)=\int_{-\infty}^{\infty} q(y)K(x-y)\,dy=\mathcal{N}(x;0,s^2+\sigma^2)$$

(the original data density but with an inflated variance) and the posterior distribution $p(y|x)=\mathcal{N}(y;rx,r\sigma^2)$. The new dataset is linearly contracted towards its mean ($\tilde{x}=rx$ $\forall x\in\mathbb{R}$) with distribution $p(\tilde{x})=\mathcal{N}(\tilde{x};0,(rs)^2)$, i.e., with the same mean and standard deviation $\tilde{s}=rs$. The sequence of standard deviations satisfies:

$$s^{(\tau+1)}=\frac{1}{1+(\sigma/s^{(\tau)})^2}s^{(\tau)} \text{ for } s^{(0)}>0 \tag{8}$$

which converges to zero. The order p of convergence is the largest p for which

$$r_s=\lim_{\tau\to\infty}\frac{\left|s^{(\tau+1)}-s^{(\infty)}\right|}{\left|s^{(\tau)}-s^{(\infty)}\right|^p}=\lim_{\tau\to\infty}\frac{(s^{(\tau)})^{3-p}}{(s^{(\tau)})^2+\sigma^2}<\infty.$$

This happens for $p=3$ (cubic convergence) with asymptotic rate $r_s=\sigma^{-2}$; we have verified this empirically. The reason for this fast convergence is that, since σ is kept constant but the dataset shrinks, effectively σ increases. Thus, at each iteration both $s^{(\tau)}$ and $\frac{1}{1+(\sigma/s^{(\tau)})^2}$ decrease. Note that the smaller $s^{(0)}$ is, the faster the convergence and so the direction of largest variance (principal component) collapses much more slowly than all other directions.

This explains the practical behaviour shown by GBMS (see fig. 2): (1) clusters collapse extremely fast (in a handful of iterations, for a suitable bandwidth); (2) after a few iterations only the local principal component survives, resulting in temporary linearly-shaped clusters (that quickly straighten). These two behaviours make GBMS useful for clustering and denoising, respectively. However, a cubic convergence does not necessarily imply a small number of iterations; this depends on the bandwidth as well. Figure 3 shows the number of iterations required to reduce $s^{(\tau)}$ below a given tolerance, as a function of σ (relative to $s^{(0)}$). While for σ of the order of $\frac{1}{4}s^{(0)}$ or larger convergence occurs in a few iterations, a small σ takes hundreds of iterations. This property is responsible for the clustering ability of GBMS: by choosing σ around $\frac{1}{4}$ of a cluster's size, clusters quickly collapse to their centroids (phase 1) while the centroids barely move towards each other (phase 2).

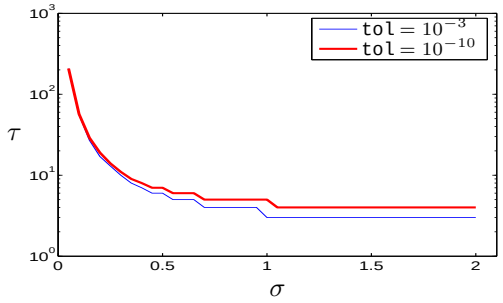

Figure 3. Number of GBMS iterations τ necessary to achieve $s^{(\tau)}<\texttt{tol}$ as a function of the bandwidth σ, for a 1D Gaussian data distribution of standard deviation $s^{(0)}=1$.

For GMS (which is an EM algorithm; Carreira-Perpiñán and Williams 2003; 2005) the dataset remains constant and the convergence is only of linear order ($p=1$), thus requiring many iterations to converge to a mode.

3. Connection with Spectral Clustering

In fig. 1**C** we rewrite the GBMS algorithm of fig. 1**B** in a matrix form that suggests a relation with spectral clustering. Each GBMS iteration is now a product $\tilde{\mathbf{X}}=\mathbf{XP}$ where $\mathbf{X}=(\mathbf{x}_1\cdots\mathbf{x}_N)$ is a $D\times N$ matrix of data points; $\mathbf{W}$ is the $N\times N$ matrix of Gaussian affinities $w_{nm}=\exp\left(-\left\|\mathbf{x}_n-\mathbf{x}_m\right\|^2/2\sigma^2\right)$; $\mathbf{D}=\operatorname{diag}\left(\sum_{n=1}^N w_{nm}\right)$ is the degree matrix; and $\mathbf{P}=\mathbf{WD}^{-1}$ is an $N\times N$ stochastic matrix: $p_{nm}=p(n|\mathbf{x}_m)\in(0,1)$ and $\sum_{n=1}^N p_{nm}=1$. $\mathbf{P}$ (or rather its transpose) is the stochastic matrix of the random walk in a graph (Chung, 1997), which in GBMS represents the posterior probabilities of each point under the kernel density estimate (1). $\mathbf{P}$ is closely related to the matrix $\mathbf{N}=\mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$ (equivalent to the normalised graph Laplacian) commonly used in spectral clustering, e.g. in the normalised cut (Shi & Malik, 2000). The eigenvalue/eigenvector pairs $(\mu_n,\mathbf{u}_n)$ and $(\lambda_n,\mathbf{v}_n)$ of $\mathbf{P}$ and $\mathbf{N}$ satisfy $\mu_n=\lambda_n$ and $\mathbf{u}_n=\mathbf{D}^{-\frac{1}{2}}\mathbf{v}_n$. In spectral clustering, given σ and $\mathbf{X}$ one computes the eigenvectors associated with the top K eigenvalues of $\mathbf{N}$ (if K clusters are desired); in this spectral space the clustering structure of the data is considerably enhanced and so a simple algorithm such as

k-means can often find the clusters.

In GBMS, we iterate the product $\mathbf{X} \leftarrow \mathbf{XP}$. If $\mathbf{P}$ were kept constant, this would be the power method (Golub & van Loan, 1996) and $\mathbf{X}$ would converge to the leading left eigenvector of $\mathbf{P}$ (the vector of ones, i.e., a single cluster), with a rate of convergence given by the second eigenvalue $\mu_2 < 1$ (the Fiedler eigenvalue in spectral clustering). However, the dynamics of GBMS is more complex because $\mathbf{P}$ also changes after each iteration. In practice $\mathbf{P}$ and $\mathbf{X}$ quickly reach a quasistable state (phase 2) where points have collapsed in clusters which slowly approach each other, and $\mathbf{P}$ does remain almost constant in phase 2. Thus, GBMS can be seen as refining the original affinities into an almost perfectly blocky matrix (phase 1) and then (trivially) extracting piecewise-constant eigenvectors for each cluster with the power method (phase 2).

Computing eigenvectors in spectral clustering is $\mathcal{O}(N^3)$ if $\mathbf{W}$ is not sparse and roughly $\mathcal{O}(KN^2)$ if $\mathbf{W}$ is sparse (for K clusters), e.g. if imposing a graph structure on the data (Shi & Malik, 2000; Carreira-Perpiñán & Zemel, 2005). Computing $\mathbf{W}$ and the graph is an extra cost. For GBMS, the total cost for non-sparse $\mathbf{W}$ is $\mathcal{O}(kN^2)$ where the number of iterations k is very small, independently of the number of clusters ($k \approx 4$–6 for accelerated GBMS; see section 5). Thus, our accelerated GBMS produces good segmentations at no more cost than spectral clustering. Using a finite-support kernel (e.g. Epanechnikov or truncated Gaussian) or a neighbourhood graph makes $\mathbf{W}$ sparse and further reduces the computational cost. GBMS has the additional advantage that the clustering result is deterministic given σ. In contrast, in spectral clustering, given σ one has to select the number of clusters (possibly with help of the eigenvalue curve) and to use a secondary clustering method in the spectral domain. Both issues introduce uncertainty in the result and are problematic in practice.

The view of GBMS as iterated products with a stochastic matrix gives a direct proof of the result of Cheng (1995) that for broad kernels the only fixed point of blurring mean-shift is the dataset where all points coincide. For broad kernels, by definition $p(\mathbf{x}_m|n) > 0 \; \forall m, \; n$ so $\mathbf{P}$ is a positive stochastic matrix at every iteration. By the Perron-Frobenius theorem (Horn & Johnson, 1986, ch. 8), $\mathbf{P}$ has a simple left eigenvalue $\mu_1 = 1$ associated with the eigenvector of ones $\mathbf{u}_1 = \mathbf{1} = (1, \dots, 1)$ and all other eigenvalues have magnitude less than 1. Since a fixed point verifies $\mathbf{X} = \mathbf{XP}$ then $\mathbf{X} = \mathbf{x}\mathbf{1}^T$ for some $\mathbf{x} \in \mathbb{R}^D$, i.e., all points coincide. For non-broad kernels, $p(\mathbf{x}_m|n) = 0$ for some $m, \; n$ and so $\mathbf{P}$ is a nonnegative matrix. By the Perron-Frobenius theorem, the eigenvalue $\mu_1 = 1$ can have multiplicity $K > 1$, where K is the number of clusters, and is associated with piecewise-constant eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_K$, where $\mathbf{u}_{kn} = 1$ if $\mathbf{x}_n$ is in cluster k and $\mathbf{u}_{kn} = 0$ otherwise.

4. Accelerated GBMS Algorithm

The fact that GBMS collapses clusters to a single point suggests that as soon as one cluster collapses we could replace it with a single point with a weight porportional to the cluster's number of points. This will be particularly effective if clusters collapse at different speeds, which happens if they have different sizes, as predicted in section 2; e.g. see fig. 2. The total number of iterations remains the same as for the original GBMS but each iteration uses a dataset with fewer points and is thus faster. Specifically, the Gaussian kernel density estimate is now $p(\mathbf{x}) = \sum_{n=1}^{N} \pi_n p(\mathbf{x}|n)$ where $p(\mathbf{x}|n) = \mathcal{N}(\mathbf{x}; \mathbf{x}_n, \sigma^2\mathbf{I})$ and the posterior probability is $p(n|\mathbf{x}_m) = p(\mathbf{x}_m|n)\pi_n / p(\mathbf{x}_m)$ as in Carreira-Perpiñán (2000). At the beginning $\pi_n = \frac{1}{N} \; \forall n$ and when clusters m and n merge then the combined weight is $\pi_m + \pi_n$. Using the matrix notation of section 3 we have $w_{nm} \propto p(\mathbf{x}_m|n)$, $\mathbf{\Pi} = \operatorname{diag}(\pi_n)$, $d_m = \sum_{n=1}^{N} w_{nm}\pi_n = p(\mathbf{x}_m)$ and $(\mathbf{\Pi W D}^{-1})_{nm} = p(n|\mathbf{x}_m)$ (we omit a proof that this is equivalent to the original GBMS).

The reduction step where coincident points are replaced with a single point can be approximated by a connected-components step where points closer than `min_diff` are considered coincident, where `min_diff` takes the same value as in the final connected-components step of GBMS (fig. 1**B**). Thus, `min_diff` is the resolution of the method (below which points are indistinguishable), and while GBMS applies it only after having stopped iterating, our accelerated version applies it at each iteration. The accelerated GBMS algorithm, given in fig. 1**D**, is then a sequence of alternating connected-components reduction and blurring mean-shift steps.

The approximate computational cost in multiplications (summarised in table 1 for all three algorithms) is as follows. One iteration of GMS (fig. 1**A**) over the whole dataset costs $2N^2D$ multiplications ($2ND$ per pixel). One iteration of GBMS (fig. 1**C**) costs $\frac{3}{2}N^2D$ multiplications. Iteration τ of accelerated GBMS (fig. 1**D**) costs[1] $\frac{3}{2}(N^{(\tau-1)})^2D$ multiplications, where $N^{(\tau)}$ is the number of points at iteration τ. The algorithm in fig. 1**D** is $\mathcal{O}(N^2)$ in memory, but for large datasets it can be implemented exactly in $\mathcal{O}(N)$ at a small speed loss.

5. Experiments with Image Segmentation

We consider the problem of segmenting greyscale and colour images, for which GMS produces very good results (Comaniciu & Meer, 2002; DeMenthon, 2002). Each

[1] The reduction step has a negligible cost: computing the distances is for free, since they were necessary to compute $\mathbf{W}$ in the previous iteration, and using depth-first search (Cormen et al., 1990) the connected-components step costs $\mathcal{O}(N^{(\tau)} + E)$ where the number of edges in the graph is $E \ll (N^{(\tau)})^2$.

GMS	GBMS	Accelerated GBMS
$2N^2Dk_1$	$\frac{3}{2}N^2Dk_2$	$\frac{3}{2}D\sum_{\tau=1}^{k_2}(N^{(\tau-1)})^2$

Table 1. Computational cost in multiplications for the algorithms. k_1 is the average number of iterations per point for GMS and k_2 is the number of iterations for GBMS (equal to that of accelerated GBMS). The speedup of accelerated GBMS over GBMS is $\sum_{\tau=0}^{k_2-1}(N^{(\tau)})^2/k_2N^2$. In the experiments, all iterations are normalised to GBMS iterations so figures can be compared directly.

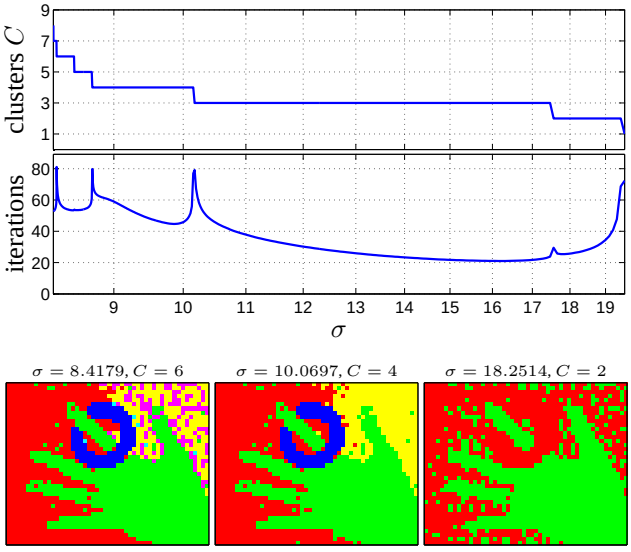

Figure 4. Segmentation results with GMS for `hand` 50×40 over a range of σ. *Above*: number of clusters C and number of iterations as a function of σ. The iterations are normalised so that one iteration of GBMS equals 1. *Below*: sample segmentations.

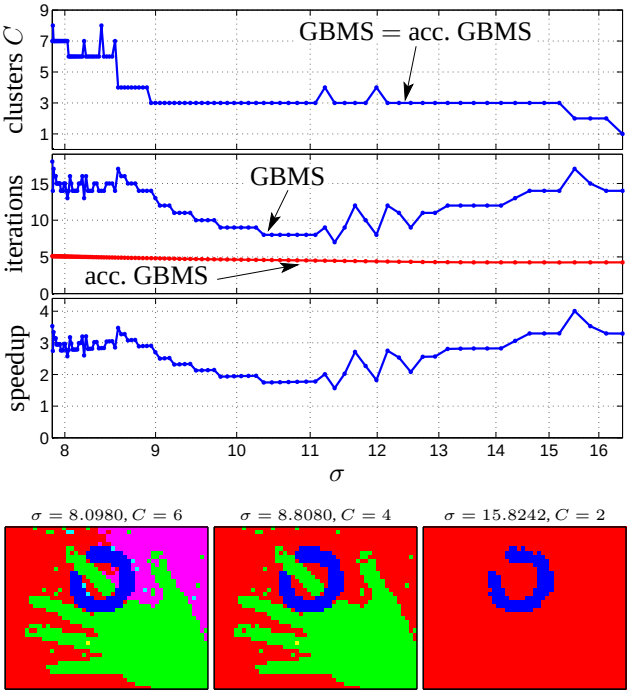

Figure 5. Segmentation results with GBMS and accelerated GBMS for `hand` 50×40 ($N = 2\,000$ points in $D = 5$ dimensions) over a range of bandwidths σ (100 different σ values). *Above*: number of clusters C, number of iterations and speedup of accelerated GBMS over GBMS, as a function of σ. *Below*: sample segmentations. Figs. 4–5 may need to be viewed in colour.

data point $\mathbf{x}_n$, i.e., each pixel, is represented by *spatial* and *range* features, e.g. $(i, j, I) \in \mathbb{R}^3$ or $(i, j, L^*, u^*, v^*) \in \mathbb{R}^5$ where (i, j) is the pixel location in the image and I and (L^*, u^*, v^*) the pixel value in a greyscale or colour image, respectively. We prescale the range features and use an isotropic kernel; thus, each component of the feature vector $\mathbf{x}$ has now pixel units, as does σ. For each image we try a range of values of σ (to obtain different numbers of clusters). The convergence criterion for GMS is that $\left\|\mathbf{x}^{(\tau+1)} - \mathbf{x}^{(\tau)}\right\| < \texttt{tol}$ while the stopping criterion for GBMS and accelerated GBMS is given by eq. (5). For all three methods we take $\texttt{tol} = 10^{-3}$ and $\texttt{min\_diff} = 10^{-2}$. Unlike in other segmentation methods, the segmentations shown use no pre- or post-processing (such as removing small clusters). We tested different images of different sizes; we give results mainly for the images of fig. 7: `hand` (colour) and `cameraman` (greyscale). We report the number of iterations relative to GBMS as given in table 1.

Figure 4 shows the results for GMS for a colour image. When using mean-shift with the Gaussian kernel, best segmentations are obtained for σ of the order of $\frac{1}{5}$ of the image size (unlike for the Epanechnikov kernel, which requires a much smaller σ and results in many small clusters that must then be grouped; Comaniciu and Meer 2002, DeMenthon 2002). The number of clusters decreases monotonically with σ; this is not necessary (Carreira-Perpiñán & Williams, 2003) but almost always holds in practice. The number of iterations generally decreases with σ and is between 20 and 80 for `hand`, though it can be more than 100 for other images, e.g. `cameraman`. The spikes in the iterations curve are caused when two modes of the density p merge (note how they match decreases in the number of clusters). The reason is that the density around the resulting mode is extremely flat and the rate of linear convergence of GMS becomes almost 1 (sublinear convergence; Carreira-Perpiñán 2005), requiring a very large number of iterations for those pixels converging to that mode. The spike is large if the two clusters that merge are large.

Figure 5 shows the results for GBMS and accelerated GBMS. The number of clusters decreases mostly monotonically with σ with occasional increases. These are caused by GBMS stopping slightly too early, so that some clusters are left unmerged. These occurrences are rare and just

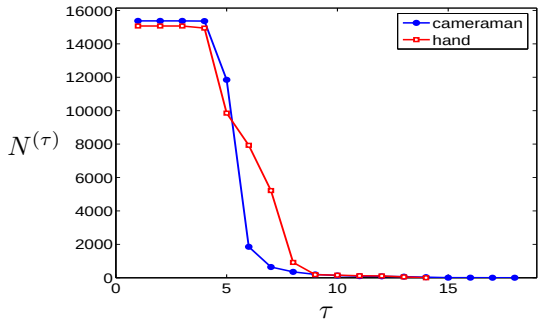

Figure 6. Number of points $N^{(\tau)}$ in the dataset as a function of the iteration number τ with accelerated GBMS for the segmentations in fig. 7. Phase 2 of GBMS (see section 1) starts at $\tau = 17$ for `cameraman` 124×124 (where there are 3 clusters of coincident points) and at $\tau = 13$ for `hand` 137×110 (where there are 3 clusters of coincident points).

result in a slightly oversegmented image. The number of iterations for GBMS remains between 10 and 25 over the whole range of σ, much smaller than that of GMS for similar segmentations. Overall, our stopping criterion works reliably, limiting the number of iterations while achieving good clusterings. If using the average update (4) instead, the segmentations obtained are very sensitive to the value of `tol` and can take several hundred iterations. The segmentations of GBMS are similar to those of GMS but not identical. GBMS seems to deal better than GMS with textured regions in the `hand` image; its segmentation of the background is much smoother than GMS's. All results are confirmed with larger images in fig. 7. The segmentations are similar to those of Comaniciu and Meer (2002).

Given the segmentation obtained by GMS for a certain σ value, the corresponding σ value for which GBMS approximately achieves the same segmentation is slightly smaller. For example, for `hand`, the 3-cluster segmentation occurs for $\sigma \in [10.1, 17.4]$ with GMS and $\sigma \in [8.9, 15.3]$ with GBMS. The reason for this is that, while for GMS both σ and the dataset remain constant, for GBMS the dataset shrinks after each iteration, so effectively σ increases.

Accelerated GBMS obtains the same segmentation as GBMS. Remarkably, the speedup over GBMS is such that the number of iterations for accelerated GBMS remains almost constant at 4–6 iterations over the whole range of σ (for all images we tested). This means a $2\times$–$4\times$ speedup over GBMS and $5\times$–$60\times$ over GMS. The explanation is given by fig. 6, which is typical for all images and σ values considered. It shows how the number of points $N^{(\tau)}$ in the dataset (which decreases after each connected-components reduction) sharply drops at $\tau \approx 5$, with an even sharper drop in the computational cost (which is proportional to $(N^{(\tau)})^2$). Thus the computational cost is dominated by the first 5 iterations.

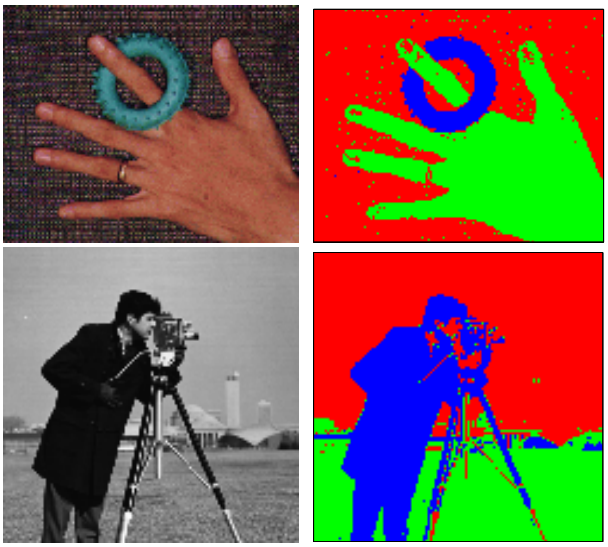

Image	Number of iterations GMS	GBMS	Accel. GBMS
`cameraman` 124×124	71.5	18	4.6
`hand` 137×110	36.4	14	4.8

Figure 7. Segmentation results for larger images with GBMS (`cameraman`: $\sigma = 20.3$, 3 clusters; `hand`: $\sigma = 24$, 3 clusters). GMS obtained very similar segmentations (not shown).

6. Discussion

As mentioned in the introduction, the literature on (Gaussian) blurring mean-shift appears to be very scarce. Fukunaga and Hostetler (1975) seem to have been the first to propose the idea of blurring the data with mean-shift and Cheng (1995) proved its convergence. These authors limited their application of blurring mean-shift to toy or very small problems and did not discuss the slippery issue of stopping the iteration in the Gaussian-kernel case.

Due to their nonparametric nature, both GMS and GBMS are particularly well suited for image segmentation, where clusters can take complicated shapes. The user can easily control the number of clusters through a single parameter, the bandwidth σ, which has pixel units. No post-processing of the resulting clusters is necessary, unlike with other methods. Values of σ that produce good segmentations can easily be obtained by using a slightly smaller σ than would be appropriate for GMS (see section 5); the latter are of the order of $\frac{1}{5}$ of the image side in pixels (DeMenthon, 2002). Another useful way is to try values of σ in a scaled-down version of the image and then scale up σ.

Although our accelerated GBMS is much faster than other nonparametric methods based on pairwise distances, such as GMS and spectral clustering, clustering large datasets (large N, as in image segmentation) is still computationally costly because GBMS is $\mathcal{O}(kN^2)$ in time. Since the cost

is almost entirely due to the first 4–6 iterations, strategies for further acceleration should probably attempt to reduce the cost per iteration rather than the number of iterations. A possible approach is to approximate the mean-shift iteration of eq. (3) with the fast Gauss transform (Greengard & Strain, 1991; Elgammal et al., 2003), which reduces the cost from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ but is only efficient for low dimensions (3 or less). Other strategies for pairwise-distance methods that have a high setup cost, such as kd–trees (Moore, 1999), may not be helpful here because the dataset (thus the kd-tree) changes at each iteration. Another approach is to make the affinity matrix $\mathbf{W}$ sparse, which also reduces the cost from $\mathcal{O}(N^2)$ to roughly $\mathcal{O}(N)$. This can be done by using a truncated kernel, or more generally by combining the affinity matrix with a neighbourhood graph (having few edges), as has been done for spectral clustering (Shi & Malik, 2000; Carreira-Perpiñán & Zemel, 2005). Also, strategies do exist to accelerate GMS (Carreira-Perpiñán, 2006); $5\times$—$100\times$ speedups are possible in image segmentation at a small approximation error.

7. Conclusion

The idea proposed by Fukunaga and Hostetler (1975) of iteratively altering a dataset using Gaussian mean-shift to enhance its clustering structure has received very little attention. We have turned this idea into a practical algorithm, accelerated GBMS, by introducing a reliable stopping criterion and an acceleration strategy, and applied it to image segmentation for the first time. Compared with Gaussian mean-shift, a state-of-the-art segmentation algorithm, our accelerated GBMS is also nonparametric, requires no post-processing and obtains results of similar quality with granularity controlled by the user through a single bandwidth parameter; but GBMS is 5 to 60 times faster. GBMS is also very simple to implement (a few lines in Matlab). We hope that its excellent performance and our theoretical insights into it will spur its use in other applications where mean-shift has been successful, such as tracking, or in clustering problems other than segmentation.

The spectral clustering view of GBMS suggests its extension to arbitrary affinities (e.g. graph-defined), not necessarily derived from a kernel density estimate. Other extensions are the use of adaptive bandwidths (different for each data point) and the use of an online version of GBMS (as opposed to batch) that updates data points immediately.

Acknowledgements This work was partially supported by NSF CAREER award IIS–0546857.

References

Carreira-Perpiñán, M. Á. (2000). Mode-finding for mixtures of Gaussian distributions. *IEEE Trans. PAMI, 22*, 1318–1323.

Carreira-Perpiñán, M. Á. (2005). Gaussian mean shift is an EM algorithm. Submitted.

Carreira-Perpiñán, M. Á. (2006). Acceleration strategies for Gaussian mean-shift image segmentation. *CVPR'06*, to appear.

Carreira-Perpiñán, M. Á., & Williams, C. K. I. (2003). On the number of modes of a Gaussian mixture. In L. Griffin and M. Lillholm (Eds.), *Scale space methods in computer vision, LNCS* vol. 2695, 625–640. Springer-Verlag.

Carreira-Perpiñán, M. Á., & Zemel, R. S. (2005). Proximity graphs for clustering and manifold learning. *NIPS 2004* (pp. 225–232).

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Trans. PAMI, 17*, 790–799.

Chung, F. R. K. (1997). *Spectral graph theory*. Providence, RI: American Mathematical Society.

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Trans. PAMI, 24*, 603–619.

Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (1990). *Introduction to algorithms*. Cambridge, MA: MIT Press.

DeMenthon, D. (2002). Spatio-temporal segmentation of video by hierarchical mean shift analysis. *Statistical Methods in Video Processing Workshop (SMVP 2002)*. Copenhagen, Denmark.

Elgammal, A., Duraiswami, R., & Davis, L. S. (2003). Efficient kernel density estimation using the fast Gauss transform with applications to color modeling and tracking. *IEEE Trans. PAMI, 25*, 1499–1504.

Fukunaga, K., & Hostetler, L. D. (1975). The estimation of the gradient of a density function, with application in pattern recognition. *IEEE Trans. Inf. Theory, 21*, 32–40.

Golub, G. H., & van Loan, C. F. (1996). *Matrix computations*. Baltimore: Johns Hopkins U. Press. Third edition.

Greengard, L., & Strain, J. (1991). The fast Gauss transform. *SIAM J. Sci. Stat. Comput., 12*, 79–94.

Horn, R. A., & Johnson, C. R. (1986). *Matrix analysis*. Cambridge, U.K.: Cambridge University Press.

Moore, A. W. (1999). Very fast EM-based mixture model clustering using multiresolution kd-trees. *NIPS 1998* (pp. 543–549).

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Trans. PAMI, 22*, 888–905.