FAST NONPARAMETRIC CLUSTERING WITH GAUSSIAN BLURRING MEAN-SHIFT.

Miguel Á. Carreira-Perpiñán.
Dept. of CSEE, OGI, Oregon Health & Science University.

1 Abstract

We revisit Gaussian blurring mean-shift (GBMS), a procedure that iteratively blurs 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.

2 Gaussian mean-shift (GMS)

Given dataset $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N \subset \mathbb{R}^D$, define a Gaussian kernel density estimate with bandwidth σ:

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

GMS is an iterative algorithm to find a mode of p:

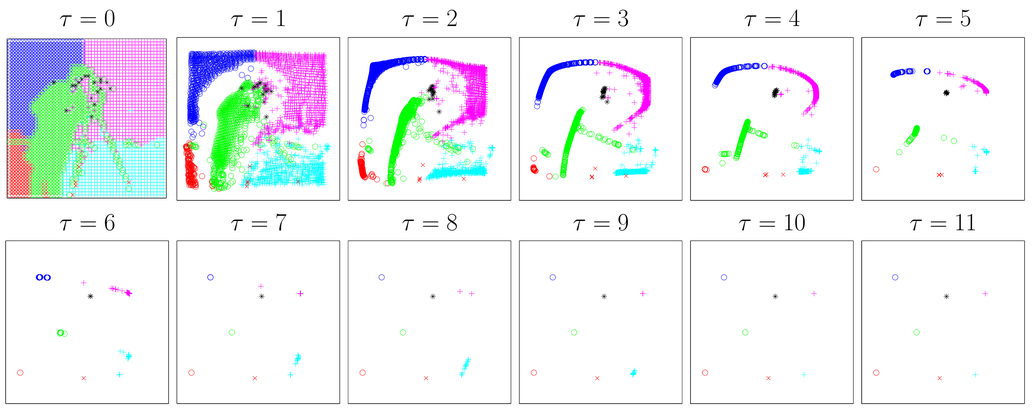
- it has no step size and is an EM algorithm with global, linear convergence (Carreira-Perpiñán & Williams 03)
- clustering: $\mathbf{x}_n$, $\mathbf{x}_m$ in same cluster if they converge to same mode
- nonparametric clustering; σ determines the number of clusters
- popular in computer vision (segmentation, tracking; Comaniciu & Meer)
- based on Fukunaga & Hostetler '75 (also Cheng '95, Carreira-Perpiñán '00, Comaniciu & Meer '02, etc.)

for $n \in \{1,\dots,N\}$ *For each data point*
  $\mathbf{x} \leftarrow \mathbf{x}_n$ *Starting point*
  repeat *Iteration loop*
    $\forall n$: $p(n|\mathbf{x}) \leftarrow \frac{\exp\left(-\frac{1}{2}\|(\mathbf{x}-\mathbf{x}_n)/\sigma\|^2\right)}{\sum_{n'=1}^N \exp\left(-\frac{1}{2}\|(\mathbf{x}-\mathbf{x}_{n'})/\sigma\|^2\right)}$ *Post. prob. (E step)*
    $\mathbf{x} \leftarrow \sum_{n=1}^N p(n|\mathbf{x})\mathbf{x}_n$ *Update* $\mathbf{x}$ *(M step)*
  until $\mathbf{x}$'s update $<$ `tol`
  $\mathbf{z}_n \leftarrow \mathbf{x}$ *Mode*
end
connected-components($\{\mathbf{z}_n\}_{n=1}^N$,`min_diff`) *Clusters*

3 Gaussian blurring mean-shift (GBMS)

This is the algorithm that Fukunaga & Hostetler really proposed (gone largely unnoticed). Same iterative scheme, but now the data points move at each iteration. Result: sequence of progressively shrunk datasets $\mathbf{X}^{(0)}, \mathbf{X}^{(1)}, \dots$ converging to a single, all-points-coincident cluster.

repeat *Iteration loop*
  for $m \in \{1,\dots,N\}$ *For each data point*
    $\forall n$: $p(n|\mathbf{x}_m) \leftarrow \frac{\exp\left(-\frac{1}{2}\|(\mathbf{x}_m-\mathbf{x}_n)/\sigma\|^2\right)}{\sum_{n'=1}^N \exp\left(-\frac{1}{2}\|(\mathbf{x}_m-\mathbf{x}_{n'})/\sigma\|^2\right)}$
    $\mathbf{y}_m \leftarrow \sum_{n=1}^N p(n|\mathbf{x}_m)\mathbf{x}_n$ *One GMS step*
  end
  $\forall m$: $\mathbf{x}_m \leftarrow \mathbf{y}_m$ *Update whole dataset*
until stop
connected-components($\{\mathbf{x}_n\}_{n=1}^N$,`min_diff`) *Clusters*



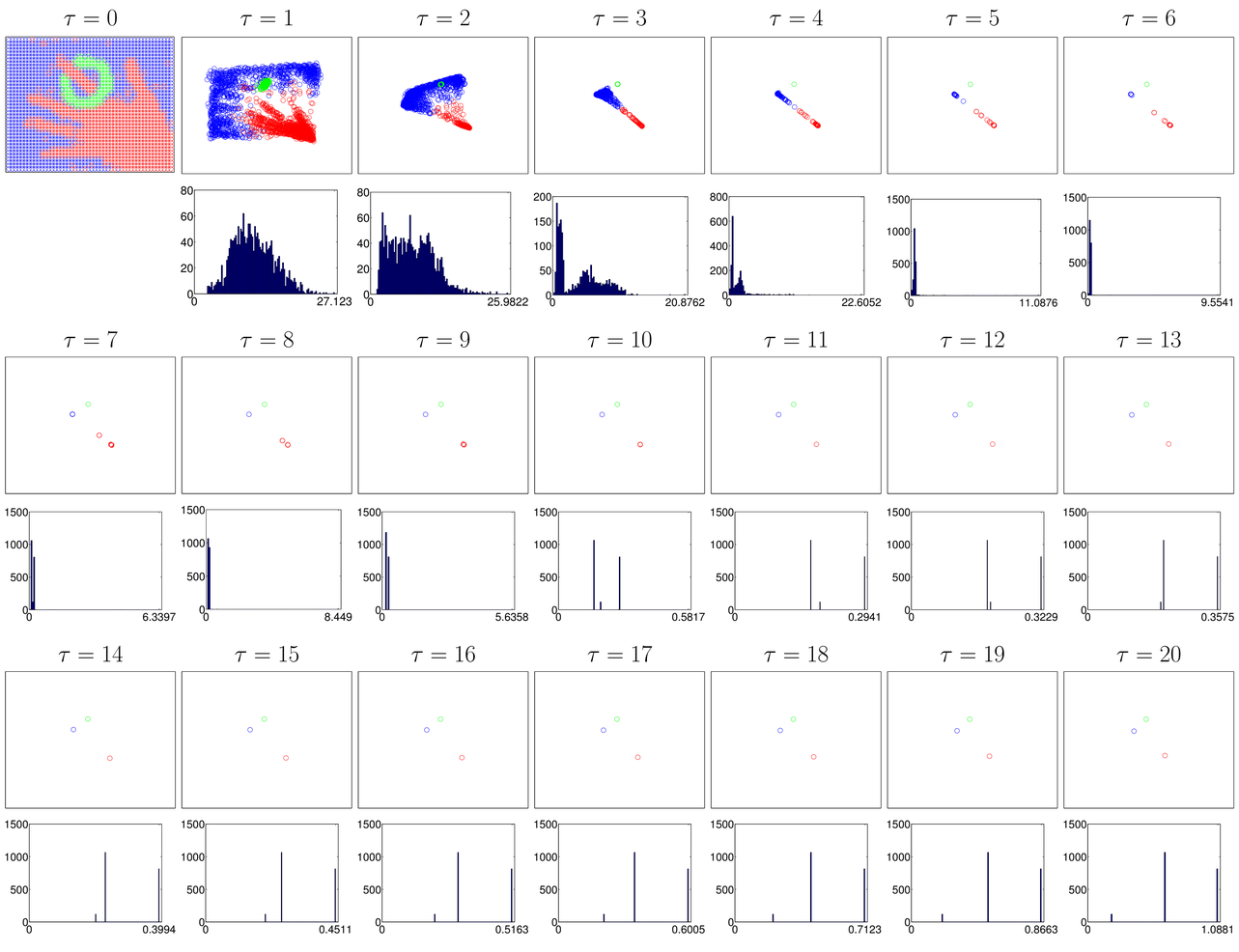
4 Stopping criterion

Behaviour of GBMS:

Phase 1: points merge into clusters of coincident points (a few iterations); we want to stop here.

Phase 2: clusters keep approaching and merging (a few to hundreds of iterations); slowly erases clustering structure.

Simply checking $\|\mathbf{X}^{(\tau)} - \mathbf{X}^{(\tau-1)}\| <$ `tol` does not work because the points are always moving. Instead, consider the histogram of updates $\{\|\mathbf{x}_n^{(\tau)} - \mathbf{x}_n^{(\tau-1)}\|\}_{n=1}^N$. Though the histograms change as points move, in phase 2 the entropy does not change (the histogram bins change their position but not their values) ⇒ stop then.

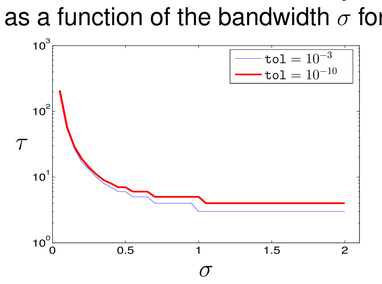


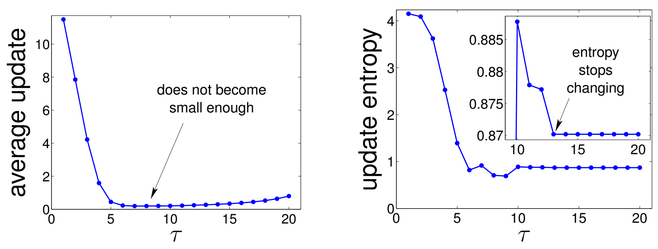
5 Convergence rate

GBMS shrinks a Gaussian cluster towards its mean with cubic convergence rate (see proof in paper), much faster than GMS's linear rate. The stdev s along each direction decreases as $s^{(\tau+1)} = \frac{1}{1+(\sigma/s^{(\tau)})^2} s^{(\tau)}$. Reason: since the dataset shrinks, effectively the bandwidth σ increases. The local principal component of the data collapses far slower than other directions. This explains the practical behaviour shown by GBMS:

1. Clusters collapse extremely fast (clustering).
2. After a few iterations only the local principal component survives, resulting in temporary linearly-shaped clusters (denoising).

Number of GBMS iterations τ necessary to achieve $s^{(\tau)} <$ `tol` as a function of the bandwidth σ for $s^{(0)} = 1$:





6 GBMS & spectral clust.

GBMS can be written as repeated products $\mathbf{X} \leftarrow \mathbf{XP}$ with the random-walk matrix $\mathbf{P} = \mathbf{WD}^{-1}$ equivalent to the graph Laplacian ($\mathbf{W}$: Gaussian affinities, $\mathbf{D}$: degree matrix, $\mathbf{P}$: posterior prob. $p(n|\mathbf{x}_m)$). In phase 1 $\mathbf{P}$ is quickly changing as points cluster. In phase 2 $\mathbf{P}$ is almost constant (and perfectly blocky) so GBMS implicitly extracts the leading eigenvectors (power method) like spectral clustering. Thus GBMS is much faster than computing eigenvectors (about 5 matrix-vector products are enough).

Since $\mathbf{P}$ is a positive matrix it really has a single leading eigenvector (Perron-Frobenius th.) with constant components, so eventually all points collapse.

7 Accelerated GBMS algorithm

At each iteration, replace clusters already formed with a single point of mass = no. points in cluster. The algorithm is now an alternation between GMS blurring steps and connected-component reduction steps, and is equivalent to the original GBMS but faster. The effective number of points $N^{(\tau)}$ (thus the computation) decreases very quickly.

Computational cost: 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). In the experiments, all iterations are normalised to GBMS iterations ($= 2N^2D$) so the figures can be compared directly.

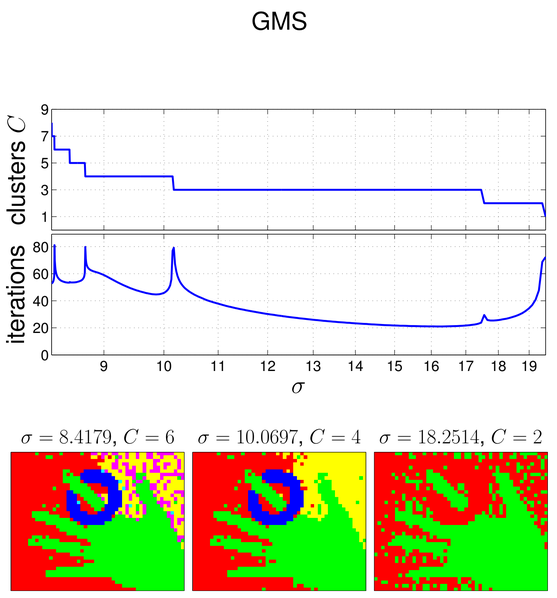
GMS	GBMS	Accelerated GBMS
$2N^2Dk_1$	$\frac{3}{2}N^2Dk_2$	$\frac{3}{2}D\sum_{\tau=1}^{k_2}(N^{(\tau-1)})^2$

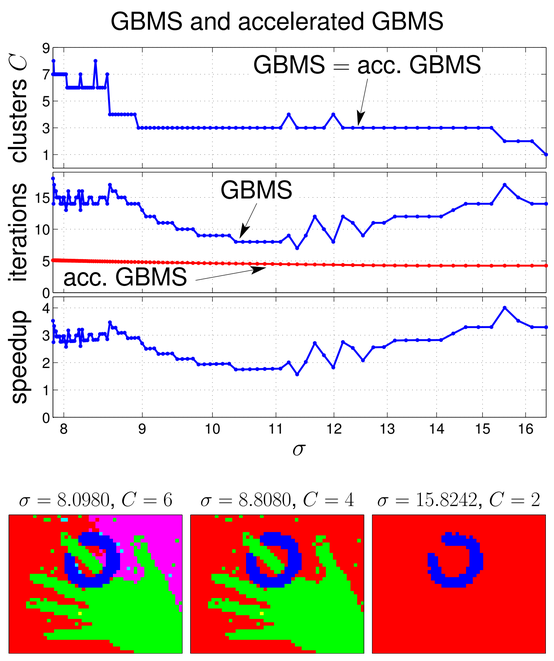
Further acceleration techniques applicable: fast Gauss transform, kd-trees, subsampling, making $\mathbf{W}$ sparse (by truncating the Gaussian or by using a proximity graph).

8 Experiments with image segmentation

Dataset: $\mathbf{x}_n = (i_n, j_n, I_n)$ (greyscale) or $\mathbf{x}_n = (i_n, j_n, L_n^*, u_n^*, v_n^*)$ (colour) where (i,j) is the pixel's spatial position and I or (L^*, u^*, v^*) are prescaled to pixel units. No pre- or postprocessing of clusters (e.g. removal of small clusters). Best segmentations for GMS appear for $\sigma \approx \frac{1}{5}\times$(image side); GBMS needs a somewhat smaller σ.

Both GMS and GBMS produce excellent segmentations (though these can differ). Accelerated GBMS takes effectively only 4–5 iterations (each $\mathcal{O}(N^2D)$) and is $5\times$–$60\times$ faster than GMS (which takes 20–hundreds of iterations).





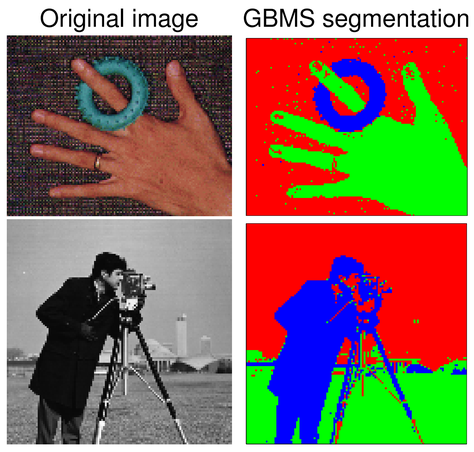
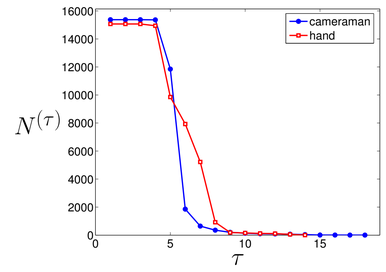


Image	Number of iterations GMS	GBMS	Acc. GBMS
`cameraman` 124×124	71.5	18	4.6
`hand` 137×110	36.4	14	4.8

Effective number of points in acc. GBMS



9 Conclusion

We have developed an old but neglected algorithm by providing a reliable stopping criterion and an acceleration, and applied it to image segmentation. GBMS produces results comparable to state-of-the-art methods such as GMS and spectral clustering at a fraction of their computation and is very simple to implement.

Partially supported by NSF CAREER award IIS–0546857.