Accelerated Convergence Using Dynamic Mean Shift

Kai Zhang[1], Jamesk T. Kwok[1], and Ming Tang[2]

[1] Department of Computer Science,
The Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong
{twinsen, jamesk}@cs.ust.hk
[2] National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences,
Beijing 100080, China
tangm@nlpr.ia.ac.cn

Abstract. Mean shift is an iterative mode-seeking algorithm widely used in pattern recognition and computer vision. However, its convergence is sometimes too slow to be practical. In this paper, we improve the convergence speed of mean shift by dynamically updating the sample set during the iterations, and the resultant procedure is called *dynamic* mean shift (DMS). When the data is locally Gaussian, it can be shown that both the standard and dynamic mean shift algorithms converge to the same optimal solution. However, while standard mean shift only has linear convergence, the dynamic mean shift algorithm has superlinear convergence. Experiments on color image segmentation show that dynamic mean shift produces comparable results as the standard mean shift algorithm, but can significantly reduce the number of iterations for convergence and takes much less time.

1 Introduction

Mean shift is a nonparametric, iterative mode-seeking algorithm widely used in pattern recognition and computer vision. It was originally derived by Fukunaga and Hostetler [1] for nonparametric density gradient estimation, and was later generalized by Cheng [2]. Recent years have witnessed many successful applications of mean shift in areas such as classification [3, 4], image segmentation [5, 6], object tracking [7] and video processing [8].

In a general setting [2], there are two data sets involved in mean shift, namely, the sample (or data) set $\mathcal{S}$, and the "cluster centers" set $\mathcal{T}$. In the standard mean shift algorithm [2], $\mathcal{T}$ evolves iteratively by moving towards the mean, as $\mathcal{T} \leftarrow \mathbf{mean}(\mathcal{T})$. Here, $\mathbf{mean}(\mathcal{T}) = \{\mathbf{mean}(\mathbf{x}) : \mathbf{x} \in \mathcal{T}\}$,

$$\mathbf{mean}(\mathbf{x}) = \frac{\sum_{s \in \mathcal{S}} K(\mathbf{s} - \mathbf{x}) w(\mathbf{s}) \mathbf{s}}{\sum_{s \in \mathcal{S}} K(\mathbf{s} - \mathbf{x}) w(\mathbf{s})},$$

K is the kernel and w is the weight function. The algorithm terminates when a fixed point $\mathbf{mean}(\mathcal{T}) = \mathcal{T}$ is reached.

A. Leonardis, H. Bischof, and A. Pinz (Eds.): ECCV 2006, Part II, LNCS 3952, pp. 257–268, 2006.

However, the mean shift algorithm often converges too slowly to be practical on large-scale applications [9]. Works on improving its convergence are relatively few. Recently, Fashing and Tomasi [10] showed that mean shift is closely related to optimization methods, particularly Newton's method and bound optimization. They conjectured that information on the shape of the kernel K can be used to tighten the bound for faster convergence. However, the difficulty is in finding a bound which is computationally easy to maximize [10]. On a more practical side, Yang *et al.* [9] proposed an improved mean shift algorithm based on quasi-Newton methods. This leads to faster convergence. However, approximating the Hessian matrix and determining the search direction in each iteration become more computationally expensive. Consequently, while the complexity of the standard mean shift algorithm is only linear in the data dimensionality, that of Yang *et al.*'s method rises to cubic.

In this paper, we improve the convergence speed of the mean shift algorithm by dynamically updating the sample set $\mathcal{S}$, depending on its behavior in the iterations. In particular, we focus on the case where $\mathcal{S}$ is updated iteratively based on the set of cluster centers $\mathcal{T}$ computed in the previous step. This modified procedure will be called *dynamic* mean shift (DMS), as opposed to the traditional, *static* mean shift (SMS) algorithm. We will prove that, under certain conditions, this procedure gradually shrinks the data set, and converges asymptotically to the same density maximum as SMS, but with a higher convergence rate (to be more specific, superlinear convergence instead of linear convergence). Besides, the DMS algorithm is also very efficient in that its computational complexity is only linear in the data dimensionality.

The rest of this paper is organized as follows. Section 2 gives a brief review on the traditional mean shift algorithm. Section 3 then describes the dynamic mean shift algorithm. A detailed discussion on its faster convergence properties will be presented in Section 4. Experimental results on color image segmentation are presented in Section 5, and the last section gives some concluding remarks.

2 Standard Mean Shift Algorithm

Let $\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a set of samples in the d-dimensional space $\mathbb{R}^d$. Using kernel k, the kernel density estimator at $\mathbf{x}$ is given by [3]

$$\hat{f}_K(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} |\mathbf{H}_i|^{-\frac{1}{2}} K(\mathbf{x}-\mathbf{x}_i; \mathbf{H}_i),$$

where $\mathbf{H}_i$ is a symmetric, positive definite $d \times d$ bandwidth matrix associated with $\mathbf{x}_i$. Instead of referring to kernel K, it is often convenient to use its *profile* $k : [0,\infty) \to \mathbb{R}$ defined by $K(\mathbf{x};\mathbf{H}) = k(\mathbf{x}'\mathbf{H}^{-1}\mathbf{x})$. To emphasize its dependence on $\mathbf{H}$, we also sometimes write $k(\mathbf{x}'\mathbf{H}^{-1}\mathbf{x})$ as $k_{\mathbf{H}}(\mathbf{x})$.

The *mean shift vector* is defined as [1, 2]

$$\mathbf{m}(\mathbf{x}) \equiv \mathbf{H}_{\mathbf{x}} \cdot \frac{\sum_{i=1}^{n} \mathbf{H}_i^{-1}\mathbf{x}_i |\mathbf{H}_i|^{-\frac{1}{2}} k'_{\mathbf{H}_i}(\mathbf{x}-\mathbf{x}_i)}{\sum_{i=1}^{n} |\mathbf{H}_i|^{-\frac{1}{2}} k'_{\mathbf{H}_i}(\mathbf{x}-\mathbf{x}_i)} - \mathbf{x}, \quad (1)$$

where $\mathbf{H}_{\mathbf{x}}^{-1} \equiv \frac{\sum_{i=1}^{n} \mathbf{H}_i^{-1} |\mathbf{H}_i|^{-\frac{1}{2}} k'_{\mathbf{H}_i}(\mathbf{x}-\mathbf{x}_i)}{\sum_{i=1}^{n} |\mathbf{H}_i|^{-\frac{1}{2}} k'_{\mathbf{H}_i}(\mathbf{x}-\mathbf{x}_i)}$. It can be shown that [3]:

$$\mathbf{m}(\mathbf{x}) = \frac{c}{2} \mathbf{H}_{\mathbf{x}} \frac{\hat{\nabla} f_K(\mathbf{x})}{\hat{f}_G(\mathbf{x})}, \tag{2}$$

where $\hat{f}_G(\mathbf{x})$ is the density estimator using kernel $G(\mathbf{x}; \mathbf{H}_i) = -ck'(\mathbf{x}'\mathbf{H}_i^{-1}\mathbf{x})$, and c is a normalization constant such that G integrates to one. Equation (2) shows that the mean shift vector $\mathbf{m}(\mathbf{x})$ points towards the direction of maximum increase of the density. Therefore, if we initialize the cluster center set $\mathcal{T}^{(t)} = \{\mathbf{y}_1^{(t)}, \mathbf{y}_2^{(t)}, \cdots, \mathbf{y}_n^{(t)}\}$ as the original sample set $\mathcal{S}$, i.e., $\mathcal{T}^{(0)} = \mathcal{S}$, then the iteration

$$\mathbf{y}^{(t+1)} = \mathbf{y}^{(t)} + \mathbf{m}(\mathbf{y}^{(t)}), \quad t = 0, 1, 2, \ldots$$

can be used to locate the local maxima of the estimated density of $\mathcal{S}$.

In this paper, we are particularly interested in the case where the data set follows the normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. This is an assumption commonly used in the theoretical analysis of the mean shift algorithm (e.g., [3]), and is expected to hold at least locally. As will be shown in the sequel, this allows us to obtain the convergence rates explicitly for both the standard and dynamic versions of the mean shift algorithm.

Suppose the use of the Gaussian kernel with fixed bandwidth $\mathbf{H}$ in the mean shift algorithm. Under the normality assumption of the data distribution, the estimated density $\hat{f}_K$ will also be a Gaussian asymptotically, with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma} + \mathbf{H}$ [11]. Plug this into (2), then the mean shift vector $\mathbf{m}(\mathbf{x})$ at $\mathbf{x}$ becomes (note that when K is Gaussian, we have $c = 2$ and $K = G$ in (2) [3])

$$\mathbf{m}(\mathbf{x}) = \mathbf{H} \frac{\hat{\nabla} f_G(\mathbf{x})}{\hat{f}_G(\mathbf{x})} = -\mathbf{H}(\mathbf{H} + \boldsymbol{\Sigma})^{-1}(\mathbf{x} - \boldsymbol{\mu}). \tag{3}$$

3 The Dynamic Mean Shift Algorithm

In the standard mean shift algorithm, the data set $\mathcal{S}$ is fixed and only the cluster center set $\mathcal{T}$ is updated. Each point in $\mathcal{T}$ will keep moving based on the mean shift vector (1) at each step until it reaches a local maximum, and then another point in $\mathcal{T}$ will be processed. In contrast, the dynamic mean shift algorithm updates both $\mathcal{S}$ and $\mathcal{T}$. In each DMS iteration, after moving all the points in $\mathcal{T}$ along their mean shift vectors for one step, we then use the shifted cluster center set $\mathcal{T}'$ to replace the data set $\mathcal{S}$ for the next iteration. More formally, denote the sample set and the cluster center set at the tth iteration by $\mathcal{S}^{(t)} = \{\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \ldots, \mathbf{x}_n^{(t)}\}$ and $\mathcal{T}^{(t)} = \{\mathbf{y}_1^{(t)}, \mathbf{y}_2^{(t)}, \ldots, \mathbf{y}_n^{(t)}\}$ respectively. They are first initialized as $\mathcal{T}^{(0)} = \mathcal{S}^{(0)} = \mathcal{S}$, the original set of samples. At the tth iteration, we have

$$\mathbf{y}_i^{(t+1)} = \frac{\sum_{\mathbf{x}_i^{(t)} \in \mathcal{S}^{(t)}} K\left(\mathbf{x}_i^{(t)} - \mathbf{y}_i^{(t)}\right) \mathbf{x}_i^{(t)}}{\sum_{\mathbf{x}_i^{(t)} \in \mathcal{S}^{(t)}} K\left(\mathbf{x}_i^{(t)} - \mathbf{y}_i^{(t)}\right)}, \quad i = 1, 2, ..., n.$$

The shifted cluster center set $\mathcal{T}^{(t+1)} = \{\mathbf{y}_1^{(t+1)}, \mathbf{y}_2^{(t+1)}, \ldots, \mathbf{y}_n^{(t+1)}\}$ then replaces the sample set at the next iteration,

$$\mathcal{S}^{(t+1)} = \mathcal{T}^{(t+1)},$$

and the whole process is repeated until a fixed state $\mathcal{S}^{(t+1)} = \mathcal{S}^{(t)}$, or equivalently, $\mathcal{T}^{(t+1)} = \mathcal{T}^{(t)}$, is reached.

In the following Sections, we study some properties of this dynamic mean shift algorithm. As mentioned in Section 1, a key advantage of this dynamic version over the standard one is its faster convergence. Hence, we will postpone and dedicate its detailed discussion to Section 4.

3.1 Gradual Shrinking of the Samples

As mentioned in Section 1, we assume that the samples follow a d-dimensional normal distribution, i.e., $\mathcal{S} = \mathcal{S}^{(0)} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$. Recall that this assumption holds at least in the local neighborhood of each sample in $\mathcal{S}$. Moreover, we assume the use of a Gaussian kernel with fixed bandwidth $\mathbf{H}$ (which is positive definite). Besides, the identity matrix will be denoted $\mathbf{I}$, vector/matrix transpose denoted by the superscript $'$, and the determinant of a matrix $\mathbf{A}$ by $|\mathbf{A}|$.

Proposition 1. *Assume that the sample set $\mathcal{S}^{(t)} = \{\mathbf{x}_i^{(t)}\}$ at the tth iteration follows $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}^{(t)})$. After one dynamic mean shift iteration, the updated sample set $\mathcal{S}^{(t+1)} = \{\mathbf{x}_i^{(t+1)}\}$ still follows a normal distribution $\mathcal{N}(\boldsymbol{\mu}, \mathbf{P}^{(t)}\boldsymbol{\Sigma}^{(t)}(\mathbf{P}^{(t)})')$, where*

$$\mathbf{P}^{(t)} = \mathbf{I} - \mathbf{H}(\mathbf{H} + \boldsymbol{\Sigma}^{(t)})^{-1}. \quad (4)$$

Proof. After one iteration, sample $\mathbf{x}_i^{(t)}$ will be moved, according to (3), to

$$\begin{aligned}\mathbf{x}_i^{(t+1)} &= \mathbf{x}_i^{(t)} + \mathbf{m}(\mathbf{x}_i^{(t)}) = \left(\mathbf{I} - \mathbf{H}(\mathbf{H} + \boldsymbol{\Sigma}^{(t)})^{-1}\right)\mathbf{x}_i^{(t)} + \mathbf{H}(\mathbf{H} + \boldsymbol{\Sigma}^{(t)})^{-1}\boldsymbol{\mu} \\ &= \mathbf{P}^{(t)}\mathbf{x}_i^{(t)} + \mathbf{C}^{(t)}, \end{aligned} \quad (5)$$

where $\mathbf{P}^{(t)} = \mathbf{I} - \mathbf{H}(\mathbf{H} + \boldsymbol{\Sigma}^{(t)})^{-1}$, and $\mathbf{C}^{(t)} = \mathbf{H}(\mathbf{H} + \boldsymbol{\Sigma}^{(t)})^{-1}\boldsymbol{\mu}$. Hence, $\mathcal{S}^{(t)}$ and $\mathcal{S}^{(t+1)}$ are related by a linear transform. Since $\mathcal{S}^{(t)}$ follows $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}^{(t)})$, $\mathcal{S}^{(t+1)}$ also follows a normal distribution with mean $\mathbf{P}^{(t)}\boldsymbol{\mu} + \mathbf{C}^{(t)} = (\mathbf{I} - \mathbf{H}(\mathbf{H} + \boldsymbol{\Sigma}^{(t)})^{-1})\boldsymbol{\mu} + \mathbf{H}(\mathbf{H} + \boldsymbol{\Sigma}^{(t)})^{-1}\boldsymbol{\mu} = \boldsymbol{\mu}$ and variance $\mathbf{P}^{(t)}\boldsymbol{\Sigma}^{(t)}(\mathbf{P}^{(t)})'$. □

***Remark*:** In other words, after one dynamic mean shift iteration, the sample mean will remain unchanged while the covariance is updated to

$$\boldsymbol{\Sigma}^{(t+1)} = \mathbf{P}^{(t)}\boldsymbol{\Sigma}^{(t)}(\mathbf{P}^{(t)})'. \quad (6)$$

Moreover, as the original data set $\mathcal{S}$ is assumed to be a Gaussian, all the $\mathcal{S}^{(t)}$'s will also remain as Gaussians.

Before a detailed study on how the covariance $\boldsymbol{\Sigma}^{(t)}$ of the sample set $\mathcal{S}^{(t)}$ evolves in the DMS iterations, we first introduce two useful lemmas.

Lemma 1. *Given two symmetric, positive semi-definite matrices* $\mathbf{A}$ *and* $\mathbf{B}$*, all the eigenvalues of* $\mathbf{C} = \mathbf{AB}$ *are non-negative.*

Proof. Let the eigen-decompositions of $\mathbf{A}$ and $\mathbf{B}$ be $\mathbf{A} = \mathbf{Q}_1\boldsymbol{\Lambda}_1\mathbf{Q}_1'$, $\mathbf{B} = \mathbf{Q}_2\boldsymbol{\Lambda}_2\mathbf{Q}_2'$, where the columns of $\mathbf{Q}_1$ and $\mathbf{Q}_2$ contain the eigenvectors of $\mathbf{A}$ and $\mathbf{B}$, respectively, and the diagonal matrices $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$ contain their corresponding eigenvalues. Let $\mathbf{Q} = \mathbf{Q}_1'\mathbf{Q}_2$ (which is orthonormal) and $\mathbf{M} = \mathbf{Q}_1'\mathbf{C}\mathbf{Q}_1$, then $\mathbf{M} = \mathbf{Q}_1'\mathbf{Q}_1\boldsymbol{\Lambda}_1\mathbf{Q}_1'\mathbf{Q}_2\boldsymbol{\Lambda}_2\mathbf{Q}_2'\mathbf{Q}_1 = \boldsymbol{\Lambda}_1\mathbf{N}$, where $\mathbf{N} = \mathbf{Q}\boldsymbol{\Lambda}_2\mathbf{Q}'$ is positive semi-definite. Let λ be an eigenvalue of $\mathbf{M}$, i.e., $\mathbf{M}v = \lambda v$. Note that $\mathbf{M}$ and $\overline{\mathbf{N}} \equiv \boldsymbol{\Lambda}_1^{1/2}\mathbf{N}\boldsymbol{\Lambda}_1^{1/2}$ share the same eigenvalues as $\mathbf{M}v = \lambda v \;\Rightarrow\; \boldsymbol{\Lambda}_1\mathbf{N}v = \lambda v \;\Rightarrow\; (\boldsymbol{\Lambda}_1^{1/2}\mathbf{N}\boldsymbol{\Lambda}_1^{1/2})(\boldsymbol{\Lambda}_1^{-1/2}v) = \lambda(\boldsymbol{\Lambda}_1^{-1/2}v)$, and $\mathbf{M}$ also has the same eigenvalues with $\mathbf{C}$, therefore $\mathbf{C}$ must have the same eigenvalues with $\overline{\mathbf{N}}$. Moreover, $v'\overline{\mathbf{N}}v = v'\boldsymbol{\Lambda}_1^{1/2}\mathbf{N}\boldsymbol{\Lambda}_1^{1/2}v = (\boldsymbol{\Lambda}_1^{1/2}v)'\mathbf{N}(\boldsymbol{\Lambda}_1^{1/2}v) \geq 0$ for all v's as $\mathbf{N}$ is positive semi-definite. Therefore, $\overline{\mathbf{N}}$ is positive semi-definite, and all its eigenvalues will be non-negative. So all the eigenvalues of $\mathbf{C}$ must be non-negative, too. □

Lemma 2. *For the* $\mathbf{P}^{(t)}$ *defined in (4),* $|\mathbf{P}^{(t)}| < 1$ *for all* t*.*

Proof. From (4),

$$\mathbf{P}^{(t)} = \mathbf{I} - \left((\mathbf{H} + \boldsymbol{\Sigma}^{(t)})\mathbf{H}^{-1}\right)^{-1} = \mathbf{I} - \left(\mathbf{I} + \boldsymbol{\Sigma}^{(t)}\mathbf{H}^{-1}\right)^{-1}. \tag{7}$$

Let the eigen-decomposition of $\boldsymbol{\Sigma}^{(t)}\mathbf{H}^{-1}$ be

$$\boldsymbol{\Sigma}^{(t)}\mathbf{H}^{-1} = \mathbf{U}^{(t)}\boldsymbol{\Lambda}^{(t)}(\mathbf{U}^{(t)})^{-1}, \tag{8}$$

where the columns of $\mathbf{U}^{(t)}$ contain the eigenvectors of $\boldsymbol{\Sigma}^{(t)}\mathbf{H}^{-1}$, and $\boldsymbol{\Lambda}^{(t)} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d)$ contains its eigenvalues. Then $\mathbf{P}^{(t)}$ can be decomposed as (after some simplifications)

$$\mathbf{P}^{(t)} = \mathbf{I} - (\mathbf{I} + \boldsymbol{\Sigma}^{(t)}\mathbf{H}^{-1})^{-1} = \mathbf{U}^{(t)}\left(\mathbf{I} - (\mathbf{I} + \boldsymbol{\Lambda}^{(t)})^{-1}\right)(\mathbf{U}^{(t)})^{-1},$$

and its determinant can be written as

$$|\mathbf{P}^{(t)}| = |\mathbf{I} - (\mathbf{I} + \boldsymbol{\Lambda}^{(t)})^{-1}| = \left|\mathrm{diag}\left(\frac{\lambda_1}{1+\lambda_1}, \frac{\lambda_2}{1+\lambda_2}, \cdots, \frac{\lambda_d}{1+\lambda_d}\right)\right| \tag{9}$$

Note that both $\boldsymbol{\Sigma}^{(t)}$ and $\mathbf{H}$ (and hence $\mathbf{H}^{-1}$) are symmetric, positive semi-definite. Therefore, using Lemma 1, the eigenvalues of $\boldsymbol{\Sigma}^{(t)}\mathbf{H}^{-1}$ must all be non-negative, i.e., $\lambda_i \geq 0$. Hence, except for the meaningless case where all λ_i's are zero, we always have $|\mathbf{P}^{(t)}| < 1$ according to (9). □

Proposition 2. $|\boldsymbol{\Sigma}^{(t)}|$ *decreases with* t*, and* $\lim_{t\to\infty} |\boldsymbol{\Sigma}^{(t)}| = \lim_{t\to\infty} |\mathbf{P}^{(t)}| = 0$.

Proof. From (6), $|\boldsymbol{\Sigma}^{(t)}| = |\boldsymbol{\Sigma}^{(t-1)}| \cdot |\mathbf{P}^{(t-1)}|^2$. Since $|\mathbf{P}^{(t)}| < 1$ by Lemma 2, $|\boldsymbol{\Sigma}^{(t)}|$ will decrease with t. Suppose $|\mathbf{P}^{(\tau)}| = \max\limits_{0\leq\tau\leq t-1} |\mathbf{P}^{(t)}|$. Note that $|\mathbf{P}^{(t)}| < 1$ for all $t \geq 0$, therefore $|\mathbf{P}^{(\tau)}| < 1$. So we have

$$|\boldsymbol{\Sigma}^{(t)}| = |\boldsymbol{\Sigma}^{(0)}| \cdot \prod_{j=0}^{t-1} |\mathbf{P}^{(j)}|^2 < |\boldsymbol{\Sigma}^{(0)}| \cdot \prod_{j=0}^{t-1} |\mathbf{P}^{(\tau)}|^2 = |\boldsymbol{\Sigma}^{(0)}| \cdot |\mathbf{P}^{(\tau)}|^{2t} \to 0,$$

as $t \to \infty$. Using (8), we have $|\boldsymbol{\Lambda}^{(t)}| = |\boldsymbol{\Sigma}^{(t)}\mathbf{H}^{-1}| = |\boldsymbol{\Sigma}^{(t)}|/|\mathbf{H}|$. Therefore,

$$\lim_{t\to\infty} |\boldsymbol{\Lambda}^{(t)}| = \lim_{t\to\infty} |\boldsymbol{\Sigma}^{(t)}\mathbf{H}^{-1}| = 0, \tag{10}$$

and all the eigenvalues (λ_is) of $\boldsymbol{\Sigma}\mathbf{H}^{-1}$ will also approach zero. Substituting this into (9), we then have $\lim_{t\to\infty} |\mathbf{P}^{(t)}| = 0$. □

***Remark*:** Note that $|\boldsymbol{\Sigma}^{(t)}|$ can be used as a measure of the spread of the sample set $\mathcal{S}^{(t)}$ at the tth iteration. Hence, Proposition 2 implies that $\mathcal{S}^{(t)}$ gradually shrinks, and the amount of shrinkage is determined by $|\mathbf{P}^{(t)}|$.

Due to the data shrinkage, a fixed-bandwidth kernel will cover more and more samples in $\mathcal{S}^{(t)}$ as the algorithm proceeds. In other words, using a fixed bandwidth here achieves the same effect as using a variable bandwidth in the standard mean shift algorithm on the original sample set $\mathcal{S}$. Note that the use of variable bandwidth is often superior to the fixed bandwidth case [5].

On the other hand, as the amount of data shrinkage can differ significantly along different directions, this can lead to both very small and very large variance components. This can be problematic if the local covariance matrix of $\mathcal{S}^{(t)}$ is chosen as the bandwidth, as its inverse may be badly scaled. To avoid this numerical problem, one can simply replace the very small eigenvalues of the local covariance matrix by some small number.

3.2 Stopping Rule

The data shrinking behavior discussed in Section 3.1 also allows the design of more efficient stopping rules. As the samples $\mathbf{x}_i^{(t)}$'s move closer and closer towards the density peaks, so once a group of samples have converged inside a small window, they will converge to one point in the following iterations. From the clustering point of view, we will then have enough information to decide their class labels (as these samples must belong to the same class), and so the iterations for these samples can be stopped early. By removing these converged clusters, computations involved in the dynamic mean shift algorithm can be reduced. In comparison, the stopping criterion in standard mean shift is often based on the step length. Since samples usually move very slowly near the density peaks in the standard mean shift algorithm [6], our stopping rule can be much more effective.

3.3 Time Complexity

The complexities of both the standard and dynamic mean shift algorithms are $O(dsN^2)$, where d is the data dimensionality, s is the number of iterations required for convergence, and N is the number of samples. As will be shown in Section 4.2, DMS has superlinear convergence while SMS only has linear convergence. Hence, the number of iterations (s) required by DMS is typically much smaller than that by SMS. Moreover, the stopping rule discussed in Section 3.2 allows samples to be thrown away early near the end of the DMS iteration process. Thus, the number of samples (n) "actively" involved in the remaining computations gradually decreases, which further reduces the time complexity of the DMS algorithm.

One may be concerned that DMS has to move all the samples in order to update the data distribution, and this could be less efficient than the mean shift algorithm that only moves a group of selected samples [12]. Indeed, the dynamic updating of the sample distribution in DMS can be realized as well by only moving a small set of "representative" samples. By decomposing the data set into disjoint, spatially local subsets $Z_1, Z_2, \ldots, Z_m$, one can model the density at each local subset Z_i by a single Gaussian $\frac{n_i}{n}\mathcal{N}(\boldsymbol{\mu}_i, h^2\mathbf{I})$, where n_i and $\boldsymbol{\mu}_i$ are the size and mean of subset Z_i respectively, and h is the bandwidth of the kernel used in the density estimator [13]. The whole density distribution can then be modeled as a combination of these Gaussians $\frac{n_1}{n}\mathcal{N}(\boldsymbol{\mu}_1, h^2\mathbf{I}), \frac{n_2}{n}\mathcal{N}(\boldsymbol{\mu}_2, h^2\mathbf{I}), \ldots, \frac{n_m}{n}\mathcal{N}(\boldsymbol{\mu}_m, h^2\mathbf{I})$. In this variant of the DMS, we only have to shift the representatives $\boldsymbol{\mu}_i$'s, whose movement leads to the update of the corresponding Gaussians $\mathcal{N}_i$s, and hence the whole density function.

4 Convergence Properties of Dynamic Mean Shift

In Section 4.1, we will first show that both the original and dynamic mean shift algorithms converge asymptotically to the same optimal solution, when the data is locally Gaussian. We will then show in Section 4.2 that the dynamic mean shift algorithm has superlinear (and thus faster) convergence while the standard version only has linear convergence.

4.1 Asymptotic Convergence of Dynamic Mean Shift

In the following, we assume, as in Section 3.1, that the samples follow the d-dimensional normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. This holds at least in the local neighborhood of each sample in $\mathcal{S}$. We then have the following property:

Proposition 3. *The dynamic mean shift procedure converges asymptotically to the mean $\boldsymbol{\mu}$.*

Proof. Using (3) and (7), we have

$$\mathbf{m}(\mathbf{x}^{(t)}) = -\mathbf{H}(\mathbf{H} + \boldsymbol{\Sigma}^{(t)})^{-1}(\mathbf{x}^{(t)} - \boldsymbol{\mu}^{(t)}) = -(\mathbf{I} + \boldsymbol{\Sigma}^{(t)}\mathbf{H}^{-1})^{-1}(\mathbf{x}^{(t)} - \boldsymbol{\mu}^{(t)}).$$

Moreover, from (10) in Proposition 2, we have $\lim_{t\to\infty} |\boldsymbol{\Sigma}^{(t)}\mathbf{H}^{-1}| = 0$. Therefore $\lim_{t\to\infty} \mathbf{m}(\mathbf{x}^{(t)}) = -(\mathbf{x}^{(t)} - \boldsymbol{\mu})$. Since $\boldsymbol{\mu}^{(t)} = \boldsymbol{\mu}$ by Proposition 1, one mean shift iteration will ultimately move all $\mathbf{x}^{(t)}$'s to $\mathbf{x}^{(t)} + \mathbf{m}(\mathbf{x}^{(t)}) = \boldsymbol{\mu}$, the mean of the original Gaussian. □

***Remark*:** It is well-known that standard mean shift will find the mode of the underlying density, which is $\boldsymbol{\mu}$ in this case. Thus, both standard and dynamic mean shift converge to the same optimal solution asymptotically.

4.2 Convergence Rates

In this Section, we will show that DMS converges faster than the standard mean shift algorithm. But first, we will provide additional insight on the convergence of standard mean shift by the following 1-D example. Suppose that the data set is the 1-D Gaussian $\mathcal{N}(\mu, \sigma^2)$, the bandwidth of the Gaussian kernel is h^2, and that the iteration starts from $x^{(0)}$. Using (3), $m(x^{(0)}) = -\rho(x^{(0)} - \mu)$ where $\rho = \frac{h^2}{h^2+\sigma^2}$. Then $x^{(0)}$ will be shifted to $x^{(1)} = x^{(0)} + m(x^{(0)}) = x^{(0)} - \rho(x^{(0)} - \mu)$. At the next iteration, the mean shift vector becomes $m(x^{(1)}) = -\rho(x^{(1)} - \mu) = -\rho(1-\rho)(x^{(0)} - \mu)$, and $x^{(1)}$ is shifted to $x^{(2)} = x^{(1)} + m(x^{(1)})$, and so on. It is easy to show by induction that the mean shift vector is of the form $m^{(t)} = m(x^{(t)}) = -\rho(1-\rho)^t(x^{(0)} - \mu)$. Note that $\{|m^{(t)}|\}_{t=1,2,\cdots}$ is a geometric sequence that decreases monotonically, indicating slower and slower convergence. This is illustrated in Figure 1, where we set $\mu = 0$, $\sigma = 1$, $h = 0.1$, and $x^{(0)} = 3$. As can be seen, the step length indeed decreases monotonically. The corresponding step lengths for the dynamic mean shift algorithm are also shown in Figure 1. Note that not only is its step length usually much larger than that for standard mean shift, but it actually increases at the first few iterations.

In the following, we compare the convergence rates of DMS and SMS. In the optimization literature, convergence can be measured by how rapidly the iterates $\mathbf{z}^{(t)}$ converge in a neighborhood of the (local) optimum $\mathbf{z}^*$. If the error $\mathbf{e}^{(t)} = \mathbf{z}^{(t)} - \mathbf{z}^*$ behaves according to $\|\mathbf{e}^{(t+1)}\|_2 / \|\mathbf{e}^{(t)}\|_2^p \to c$, where $c > 0$ and $\|\cdot\|_2$ denotes the (vector) two-norm, then the *order of convergence* is defined to be pth order [14]. In particular, if $p = 1$, we have *first order* or *linear convergence*. Note that linear convergence can be equivalently defined as $\|\mathbf{e}^{(t+1)}\|_2 / \|\mathbf{e}^{(t)}\|_2 \leq c$. Faster convergence can be obtained if the local rate constant c tends to zero, i.e., $\|\mathbf{e}^{(t+1)}\|_2 / \|\mathbf{e}^{(t)}\|_2 \to 0$. This is also known as *superlinear convergence*.

As in previous sections, we will again focus on the case when the samples are normally distributed as $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Recall that Section 4.1 has shown that both DMS and SMS converge to the mean $\boldsymbol{\mu}$, and hence the optimum $\mathbf{z}^* = \boldsymbol{\mu}$ here.

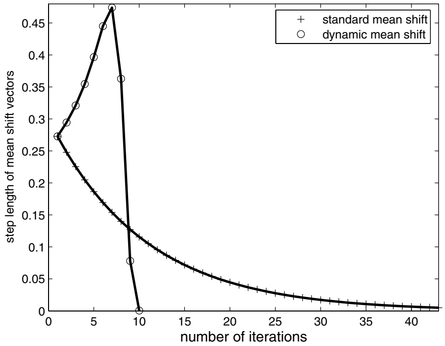


Fig. 1. Step lengths taken by DMS and SMS on a 1-D data set. Note that DMS converges in only 10 iterations.

Theorem 1. *SMS converges linearly, while DMS converges superlinearly.*

Proof. At the tth iteration, both DMS and SMS shift the current $\mathbf{x}^{(t)}$ to $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \mathbf{m}(\mathbf{x}^{(t)})$. Using (3), we have

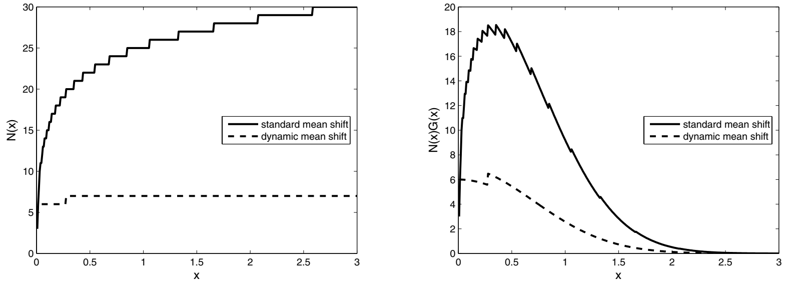
$$\mathbf{x}^{(t+1)} - \mathbf{x}^* = (\mathbf{x}^{(t)} - \boldsymbol{\mu}) - \mathbf{H}(\mathbf{H} + \boldsymbol{\Sigma})^{-1}(\mathbf{x}^{(t)} - \boldsymbol{\mu}) = \mathbf{P}^{(t)}(\mathbf{x}^{(t)} - \boldsymbol{\mu}),$$

with $\mathbf{P}^{(t)}$ defined in (4). Hence,

$$\frac{\|\mathbf{x}^{(t+1)} - \boldsymbol{\mu}\|_2}{\|\mathbf{x}^{(t)} - \boldsymbol{\mu}\|_2} = \frac{\left\|\mathbf{P}^{(t)}(\mathbf{x}^{(t)} - \boldsymbol{\mu})\right\|_2}{\left\|\mathbf{x}^{(t)} - \boldsymbol{\mu}\right\|_2} \leq \|\mathbf{P}^{(t)}\|_2,$$

by definition of the matrix two-norm[1] of $\mathbf{P}^{(t)}$ [15]. In SMS, the sample set $\mathcal{S}$ keeps unchanged. Therefore $\mathbf{P}^{(t)}$'s are all fixed at $\mathbf{P} = \mathbf{I} - \mathbf{H}(\mathbf{H} + \boldsymbol{\Sigma})^{-1}$, implying linear convergence for SMS. For DMS, we have $\lim_{t\to\infty} |\mathbf{P}^{(t)}| \to 0$ by Proposition 2, so all its eigenvalues will approach 0. Since $\|\mathbf{P}^{(t)}\|_2$ is the maximum singular value of $\mathbf{P}^{(t)}$, therefore $\|\mathbf{P}^{(t)}\|_2 \to 0$, i.e., DMS converges superlinearly. □

Here, we give an illustration on the numbers of iterations required for convergence in SMS and DMS. The data set follows the 1-D Gaussian $\mathcal{N}(0, 1)$, and the bandwidth is chosen as $h = 0.5$. Figure 2(a) shows the number of iterations $N(x)$ when starting at different initial positions x's. As can be seen, DMS requires much fewer iterations than the standard mean shift algorithm. Moreover, since we know that the data set follows a normal distribution, we can also compute the average number of iterations by integrating $N(x)$ w.r.t. the (normal) density $G(x)$. Figure 2(b) plots the density-weighted number of iterations $N(x)G(x)$. The average number of iterations required by DMS is calculated to be roughly 70% less than that for SMS.



(a) Unweighted

(b) Weighted by the density

Fig. 2. Number of iterations required for convergence when starting the standard / dynamic mean shift algorithm at different positions

We now investigate the effect of the bandwidth on the number of iterations required for convergence. Again, we use the same 1-D data set that follows

[1] The matrix two-norm of a matrix $\mathbf{A}$ is defined as $\|\mathbf{A}\|_2 = \max_{\mathbf{x}\neq\mathbf{0}}\left\{\frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2}\right\}$. It is also equal to the maximum singular value of $\mathbf{A}$.

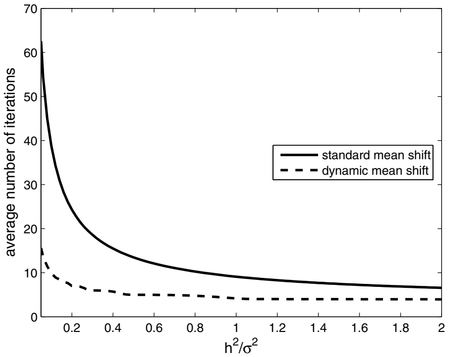


Fig. 3. The average number of iterations for convergence at various values of h^2/σ^2

$\mathcal{N}(0,1)$. As can be seen from Figure 3, DMS needs much fewer iterations than SMS when h^2/σ^2 varies from 0.1 to 2. In practice, h^2/σ^2 should be reasonably small, or else serious misclassifications may occur near the class boundaries.

5 Image Segmentation Experiments

In this Section, we compare the performance of dynamic and standard mean shift algorithms for color image segmentation. The segments are obtained by

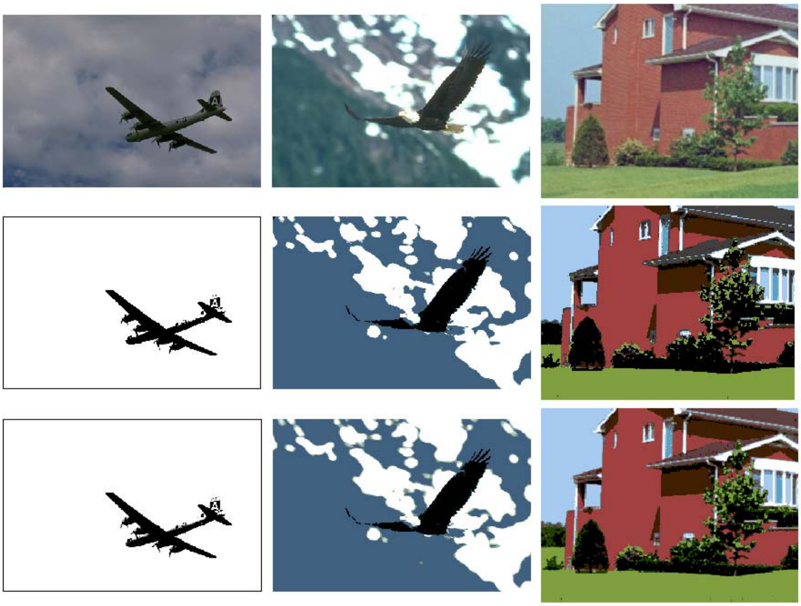


Fig. 4. Segmentation results using SMS and DMS algorithms. Top: Original images; Middle: SMS segmentation results; Bottom: DMS segmentation results.

Table 1. Total wall time (in seconds) and the average number of iterations on the various image segmentation tasks

		SMS		DMS	
image	size	time	# iterations	time	# iterations
plane	321×481	2.62	15.79	1.84	11.86
eagle	321×481	10.03	21.78	4.77	10.79
house	192×255	12.65	20.40	6.43	10.84

clustering in the RGB feature space. The sample size, which is equal to the number of pixels in the image, can be very large (in the order of 100,000). Hence, instead of using/moving all the samples in the mean shift iterations, we only use a set of "representative" samples. As discussed in Section 3.3, the whole data set is first divided into m local subsets, each of them is modeled by a Gaussian $\gamma_i \mathcal{N}(\mathbf{u}_i, h_i^2 \mathbf{I})$. This step can be performed efficiently. Moreover, the number of clusters, m, is much smaller than the sample size. Only these m cluster means, each weighted by the γ_i, are used in the DMS and SMS algorithms. In the experiment, we use the Gaussian kernel with bandwidth $h^2\mathbf{I}$ ($h = 12$). All codes are written in VC++ and run on a 2.26GHz Pentium-III PC.

Figure 4 shows the segmentation results, and Table 1 shows the total wall time (from finding the local cluster representatives to mean shift clustering) and the number of iterations (averaged over all the cluster representatives). One can see that DMS obtains comparable segmentation results as SMS, but converges in much fewer iterations and takes much less time.

6 Conclusions

In this paper, we extend the mean shift algorithm by dynamically updating the set of samples during the iterations. This has the interesting property of gradually shrinking the sample set, and allows a fixed bandwidth procedure to achieve the same effect as variable bandwidth mean shift. More importantly, it allows faster convergence both in theory and practice. When the data is locally Gaussian, it is shown that dynamic mean shift converges to the same optimal solution as the standard version, but while standard mean shift can only converge linearly, the dynamic mean shift algorithm converges superlinearly. Experiments on color image segmentation show that dynamic mean shift produces comparable results as the standard mean shift approach, but the number of iterations and the elapsed time are both reduced by half.

Acknowledgement

This research has been partially supported by the Research Grants Council of the Hong Kong Special Administrative Region. Ming Tang is partially supported by NSFC (Grant No. 60318003 and 60572057).

References

1. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on Information Theory **21** (1975) 32–40
2. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence **17** (1995) 790–799
3. Comaniciu, D.: An algorithm for data-driven bandwidth selection. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 281–288
4. Georgescu, B., Shimshoni, I., Meer, P.: Mean shift based clustering in high dimensions: A texture classification example. In: Proceedings of the International Conference on Computer Vision. (2003) 456–463
5. Comaniciu, D., Meer, P.: The variable bandwidth mean shift and data driven scale selection. In: Proc. ICCV. (2001) 438–445
6. Comaniciu, D., Meer, P.: Mean shift: A robust approach towards feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 603–619
7. Zivkovic, Z., Kröse, B.: An EM-like algorithm for color-histogram-based object tracking. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition. Volume 1. (2004) 798–803
8. DeMenthon, D., Doermann, D.: Video retrieval using spatio-temporal descriptors pages. In: Proceedings of the Eleventh ACM International Conference on Multimedia. (2003) 508 – 517
9. Yang, C., Duraiswami, R., DeMenthon, D., Davis, L.: Mean-shift analysis using quasi-Newton methods. Proceedings of the International Conference on Image Processing **3** (2003) 447 – 450
10. Fashing, M., Tomasi, C.: Mean shift is a bound optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 471–474
11. Stoker, T.: Smoothing bias in density derivative estimation. Journal of the American Statistical Association **88** (1993) 855–863
12. Comaniciu, D., Meer, P.: Distribution free decomposition of multivariate data. Pattern Analysis and Applications **2** (1999) 22–30
13. Zhang, K., Tang, M., Kwok, J.T.: Applying neighborhood consistency for fast clustering and kernel density estimation. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition. (2005) 1001 – 1007
14. Fletcher, R.: Practical Methods of Optimization. Wiley, New York (1987)
15. Noble, B., Daniel, J.: Applied Linear Algebra. 3rd edn. Prentice-Hall, Englewood Cliffs, NJ (1988)