HIERARCHICALLY DISTRIBUTED DYNAMIC MEAN SHIFT

Kohei Inoue and Kiichi Urahama

Department of Visual Communication Design, Kyushu University, Japan
e-mail: {k-inoue, urahama}@design.kyushu-u.ac.jp

ABSTRACT

A fast and memory-efficient method is presented for dynamic mean shift (DMS) algorithm, which is an iterative mode-seeking algorithm. The DMS algorithm requires a large amount of memory to run because it dynamically updates all samples during the iterations. Therefore, it is difficult to use the DMS for clustering a large set of samples. The difficulty of the DMS is solved by partitioning a set of samples into subsets hierarchically, and the resultant procedure is called the hierarchically distributed DMS (HDDMS). Experimental results on image segmentation show that the HDDMS requires less memory than that of the DMS.

Index Terms— mean shift, dynamic mean shift, clustering, image segmentation, stochastic matrix

1. INTRODUCTION

Mean shift (MS) algorithm [1] is an iterative mode-seeking algorithm, which is widely used in image processing, pattern recognition and computer vision fields. The MS algorithm originated with Fukunaga et al. [2]. Cheng [3] applied MS to clustering and Hough transform. Comaniciu and Meer [1] applied MS to clustering, image segmentation and image smoothing. Fashing and Tomasi [4] proved that MS procedure is a quadratic bound optimization. Carreira-Perpiñán [5] showed that Gaussian MS is an EM algorithm. Li et al.[6] proposed anisotropic MS and proved its convergence. In computer vision community, MS is applied to object tracking [7, 8, 9].

Recentry, Zhang et al. [10] proposed dynamic MS (DMS), which dynamically updates sample set during the iterations, and improved the convergence speed of MS. They also proved that both MS and DMS converge to the same solution when the data is locally Gaussian. Carreira-Perpiñán [11] proposed Gaussian blurring MS (GBMS), which is similar to Zhang's DMS [10]. Carreira-Perpiñán [11] pointed out that the typical behavior of DMS or GBMS consists of two phases. That is, in the first phase, points merge into extremely compact clusters, and in the second phase, these clusters approach to each other till they merge at a single point. He proposed a stopping criterion that stops the algorithm just after the first phase. The stopping criterion is also used in this paper.

Although DMS improves the convergence speed of MS, it needs a large amount of memory to run because DMS moves all samples during the iterations. Therefore, it is still difficult for DMS to handle large size data which exceed the memory capacity of machine. In this paper, we propose a fast and memory-efficient method for DMS. The proposed method is derived from the relationship between DMS and a stochastic matrix. In the derivation, DMS iteration is understood as a fixed point iteration. The proposed method distributes samples into mutually disjoint subsets hierarchically, and is called hierarchically distributed DMS (HDDMS).

Boutsinas and Gnardellis [12] presented a methodology for distributing clustering process, which is called the prosegisis methodology. However, their methodology is not hierarchical, and so the scalability is limited.

The rest of this paper is organized as follows: we summarize DMS and show the relationship between DMS and a stochastic matrix in section 2. In section 3, we propose hierarchically distributed DMS (HDDMS) algorithm. Experimental results on image segmentation are shown in section 4. Finally, we give concluding remarks and a discussion of future work in section 5.

2. DYNAMIC MEAN SHIFT

Let $\{x_1, ..., x_n\}$ be a set of samples in the d-dimensional space $\mathbb{R}^d$, and let $\{\pi_1, ..., \pi_n\}$ be the corresponding weights. Then the $(\tau+1)$-th iteration of dynamic mean shift (DMS) [10, 11] algorithm can be written as follows:

$$x_i^{(\tau+1)} = \frac{1}{z_i^{(\tau)}} \sum_{j=1}^{n} x_j^{(\tau)} \pi_j^{(\tau)} w_{ij}^{(\tau)}, \quad i = 1, ..., n, \tag{1}$$

where

$$w_{ij}^{(\tau)} = e^{-\alpha \| x_i^{(\tau)} - x_j^{(\tau)} \|^2} \tag{2}$$

with a scale parameter $\alpha > 0$ and $z_i^{(\tau)} = \sum_{j=1}^{n} \pi_j^{(\tau)} w_{ij}^{(\tau)}$. $x_i^{(\tau)}$ and $\pi_i^{(\tau)}$ are initialized as $x_i^{(0)} = x_i, \pi_i^{(0)} = 1$. Equation (1) can be rewritten in matrix form as

$$X^{(\tau+1)} = X^{(\tau)} \Pi^{(\tau)} W^{(\tau)} (D^{(\tau)})^{-1}, \tag{3}$$

where $X^{(\tau)} = [x_1^{(\tau)}, ..., x_n^{(\tau)}]$, $\Pi^{(\tau)} = \mathrm{diag}(\pi_1^{(\tau)}, ..., \pi_n^{(\tau)})$, $W^{(\tau)} = [w_{ij}^{(\tau)}]$, and $D^{(\tau)} = \mathrm{diag}(z_1^{(\tau)}, ..., z_n^{(\tau)})$. During the

1-4244-1437-7/07/$20.00 ©2007 IEEE

iteration, collapsed samples are reduced to a single sample. Specifically, if $\|x_i^{(\tau)} - x_j^{(\tau)}\| < \epsilon$ for a constant $\epsilon > 0$ and $i < j$, then we renew $\pi_i^{(\tau)}$ to $\pi_i^{(\tau)} + \pi_j^{(\tau)}$ and delete $x_j^{(\tau)}$ and $\pi_j^{(\tau)}$. The stopping criterion is $\frac{1}{n}\sum_{i=1}^{n} e_i^{(\tau)} < \epsilon$ or $|E^{(\tau+1)} - E^{(\tau)} < \delta$ for a constant $\delta > 0$, where $e_i^{(\tau)} = \|x_i^{(\tau)} - x_i^{(\tau-1)}\|$ for $\tau \geq 1$ and $E^{(\tau)}$ is the entropy of the normalized histogram of $e_1^{(\tau)}, ..., e_n^{(\tau)}$. Details are found in Ref. [11].

Property 1 Let $P^{(\tau)} = [p_{ij}^{(\tau)}] = \Pi^{(\tau)} W^{(\tau)} (D^{(\tau)})^{-1}$. Then $P^{(\tau)}$ is a stochastic matrix, i.e., $p_{ij}^{(\tau)} \geq 0$ and $\sum_{i=1}^{n} p_{ij}^{(\tau)} = 1$.

Proof. The j-th column of $P^{(\tau)}$ may be written as

$$[p_{1j}^{(\tau)}, ..., p_{nj}^{(\tau)}]^T = [\pi_1^{(\tau)} w_{1j}^{(\tau)}, ..., \pi_n^{(\tau)} w_{nj}^{(\tau)}]^T / z_j^{(\tau)}. \qquad (4)$$

Since $\pi_i^{(\tau)} \geq 0,\ w_{ij}^{(\tau)} \geq 0$, it follows that $p_{ij}^{(\tau)} \geq 0$. From Eq. (2), we have $w_{ij}^{(\tau)} = w_{ji}^{(\tau)}$ and thus $\sum_{i=1}^{n} p_{ij}^{(\tau)} = \sum_{i=1}^{n} \pi_i^{(\tau)} w_{ij}^{(\tau)} / z_j^{(\tau)} = \sum_{i=1}^{n} \pi_i^{(\tau)} w_{ji}^{(\tau)} / z_j^{(\tau)} = z_j^{(\tau)} / z_j^{(\tau)} = 1$.

Property 2 Let $u^{(\tau)} = [u_1^{(\tau)}, ..., u_n^{(\tau)}]^T$ be the stationary probability vector of $P^{(\tau)}$; that is, $u_i^{(\tau)} \geq 0,\ \sum_{i=1}^{n} u_i^{(\tau)} = 1$ and $P^{(\tau)} u^{(\tau)} = u^{(\tau)}$ hold. Then $X^{(\tau+1)} u^{(\tau)} = X^{(\tau)} u^{(\tau)}$. That is, the weighted mean of samples with weights $u^{(\tau)}$ is a fixed point of DMS iteration.

Proof. From Eq. (3) and property 1, we have

$$\begin{aligned} X^{(\tau+1)} u^{(\tau)} &= X^{(\tau)} \Pi^{(\tau)} W^{(\tau)} (D^{(\tau)})^{-1} u^{(\tau)} \\ &= X^{(\tau)} P^{(\tau)} u^{(\tau)} \\ &= X^{(\tau)} u^{(\tau)}. \end{aligned} \qquad (5)$$

Consider a Markov chain defined by the stochastic matrix $P^{(\tau)}$. If the set of samples forms several clusters, then the corresponding Markov chain is also decomposable into several essential sets approximately. Thus, the above properties hold for each cluster separately. Property 2 implies that the samples in a cluster move toward a fixed point (or a weighted mean) and collapse into it. That is, DMS algorithm estimates the local modes of a density function as the means of samples weighted by the stationary probabilities of stochastic matrices.

3. HIERARCHICALLY DISTRIBUTED DYNAMIC MEAN SHIFT

Let r be a weighted mean of $X = [x_1, ..., x_n]$ with nonnegative weights $u = [u_1, ..., u_n]^T$ normalized as $\sum_{i=1}^{n} u_i = 1$:

$$r = Xu = \sum_{i=1}^{n} u_i x_i. \qquad (6)$$

Let $C_1, ..., C_m$ be the mutually disjoint subsets of $\{1, ..., n\}$. Then we may write Eq. (6) as

$$\begin{aligned} r &= \sum_{k=1}^{m} \left(\sum_{i \in C_k} u_i \right) \frac{\sum_{i \in C_k} u_i x_i}{\sum_{i \in C_k} u_i} \\ &= \sum_{k=1}^{m} \tilde{u}_k \tilde{r}_k, \end{aligned} \qquad (7)$$

where $\tilde{u}_k = \sum_{i \in C_k} u_i$ and $\tilde{r}_k = \sum_{i \in C_k} u_i x_i / \tilde{u}_k$. Equation (7) indicates that r can be computed as the weighted mean of $\{\tilde{r}_1, ..., \tilde{r}_m\}$ with weights $\{\tilde{u}_1, ..., \tilde{u}_m\}$. Further, we expand each $\tilde{r}_k$ as r in Eq. (7). Repeating this expansion hierarchically, we have

$$\begin{aligned} r &= \sum_{k_1=1}^{m_1} \tilde{u}_{k_1} \tilde{r}_{k_1} \\ &= \sum_{k_1=1}^{m_1} \tilde{u}_{k_1} \left(\sum_{k_2=1}^{m_2} \tilde{u}_{k_2}^{(k_1)} \tilde{r}_{k_2}^{(k_1)} \right) \\ &\vdots \\ &= \sum_{k_1=1}^{m_1} \tilde{u}_{k_1} \left(\sum_{k_2=1}^{m_2} \tilde{u}_{k_2}^{(k_1)} \cdots \right. \\ &\quad \left. \left(\sum_{k_N=1}^{m_N} \tilde{u}_{k_N}^{(k_1, ..., k_{N-1})} \tilde{r}_{k_N}^{(k_1, ..., k_{N-1})} \right) \cdots \right), \end{aligned} \qquad (8)$$

where N is the number of levels in the tree structure, m_l is the number of mutually disjoint subsets or nodes in l-th level, $\tilde{r}_{k_l}^{(k1, ..., k_{l-1})}$ is the k_l-th weighted mean in the k_l-th subset on the l-th level in the k_{l-1}-th subset on the $(l-1)$-th level $\cdots$ in the k_1-th subset on the first level, and $\tilde{u}_{k_l}^{(k_1, ..., k_{l-1})}$ is its weight.

The algorithmic procedure of hierarchically distributed DMS (HDDMS) is summarized as follows:

[HDDMS algorithm]

Step 1: Distribute a set of samples into mutually disjoint subsets hierarchically and construct the tree structure of them.

Step 2: Execute the DMS algorithm at each node from leaf nodes to root node.

Step 3: Output local modes obtained at root node.

The amount of memory required in the HDDMS algorithm is $O((n/\prod_{l=1}^{N} m_l)^2)$, and the computational cost in multiplications is $O(d(n/\prod_{l=1}^{N} m_l)^2 \tau_{\max})$, where $\tau_{\max}$ is the maximal number of iterations. On the other hand, the required memory and the computational cost in DMS are $O(n^2)$ and $O(dn^2 \tau_{\max})$, respectively.

4. EXPERIMENTS

In this section, we compare the performance of DMS, HDDMS and DDMS which is a special case of HDDMS for

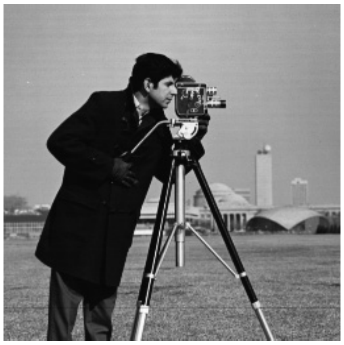


Fig. 1. Cameraman image (256×256 pixels).

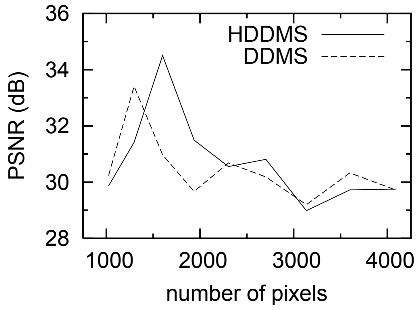


Fig. 2. PSNR.

$N = 1$. DDMS is a proseggisis methodological [12] implementation of DMS. An original image shown in Fig. 1 is rescaled from 32×32 pixels to 64×64 pixels and segmented using the three algorithms. Segments are obtained by clustering pixels in the joint spatial-range domain [1]. Peak signal-to-noise ratio defined by

$$\mathrm{PSNR} = 10 \log_{10} \frac{255^2}{\sigma^2} \tag{9}$$

is shown in Fig. 2, where $\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(f_i^{\mathrm{(H)DDMS}} - f_i^{\mathrm{DMS}})^2$ and f_i^{DMS}, $f_i^{\mathrm{(H)DDMS}}$ indicate the outputs of DMS, (H)DDMS algorithms respectively. Figure 2 shows that the output images of DDMS and HDDMS are similar to that of DMS. The values of PSNR are about 30 dB. It should be noted that DMS fails to segment images of more than 64×64 pixels because of the out of memory error. Zhang et al. [10] showed the results of image segmentation with 321×481 pixels. However, they only used a set of representative samples instead of using all the samples or pixels. Segmentation results are shown in Fig. 3. Output images by DMS, DDMS and HDDMS shown in Fig. 3(b), (c) and (d) respectively are similar to each other. CPU time is shown in Fig. 4. DDMS and HDDMS are much faster than DMS, especially for larger images. All codes are written in MATLAB and run on a PC with Pentium 4 CPU 3.40GHz, 2.00GB RAM. We set $\alpha = 0.005$, $\epsilon = 10^{-3}$, $\delta = 10^{-8}$. In DDMS, we set $m = \lceil n/h \rceil$ for $h = 300$. In HDDMS, we construct a binary tree and restrict the maximum number of elements in each disjoint subset to $h = 300$.

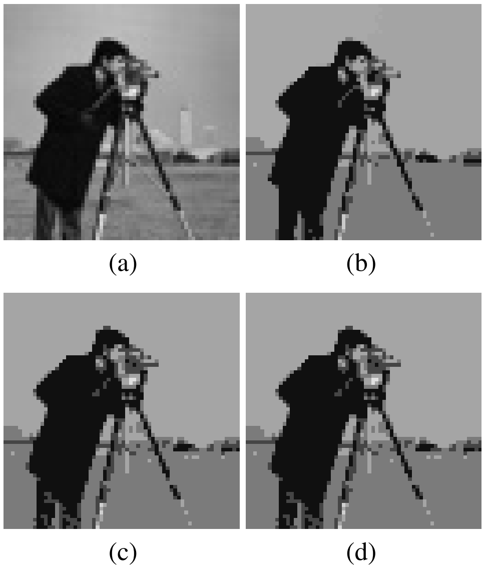

(a) (b)
(c) (d)

Fig. 3. Segmentation results: (a) input image (64×64 pixels), (b) DMS, (c) DDMS and (d) HDDMS.

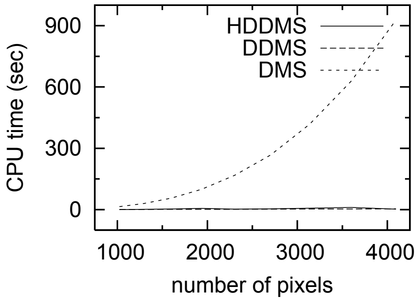


Fig. 4. CPU time.

Another example image shown in Fig. 5 is rescaled from 128×128 to 1024×1024 and segmented using DDMS and HDDMS. CPU time is shown in Fig. 6. Although DDMS fails to segment images of more than 256×256 pixels because of the out of memory error, HDDMS can segment all images. Segmented images by HDDMS are shown in Fig. 7.

5. CONCLUSION

In this paper, a fast and memory-efficient algorithm has been proposed for dynamic mean shift (DMS) algorithm. The proposed algorithm distributes a dataset into mutually disjoint subsets hierarchically and constructs a tree structure of the

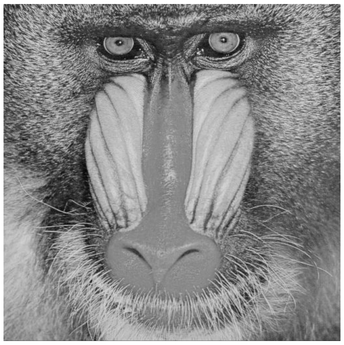


Fig. 5. Mandrill image (512 × 512 pixels).

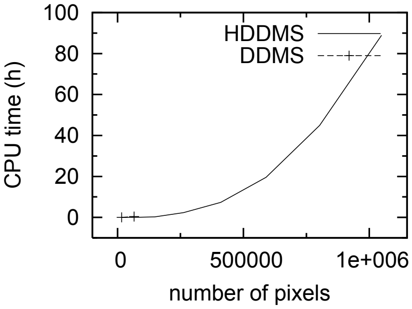
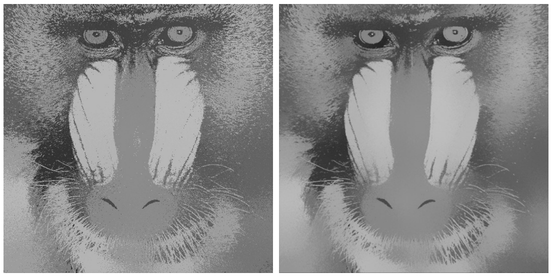


Fig. 6. CPU time.



(a) (b)

Fig. 7. Segmentation results: (a) 512 × 512 pixels and (b) 1024 × 1024 pixels.

dataset. A bottom-up approach on the tree structure enables us to segment large-sized images. Moreover, a relationship between the DMS algorithm and a stochastic matrix is also shown. This relation suggests the connection between DMS and spectral clustering [13, 14, 15], in which left eigenvectors of a stochastic matrix are used. Application of the proposed algorithm to anisotropic DMS is future work.

6. REFERENCES

[1] D. Comaniciu and P. Meer, “Mean Shift: A Robust Approach toward Feature Space Analysis,” *IEEE Trans. PAMI*, vol. 24, no. 5, pp. 603-619, May 2002.

[2] K. Fukunaga and L. D. Hostetler, “The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition,” *IEEE Trans. IT*, vol. IT-21, no. 1, pp. 32-40, 1975.

[3] Y. Cheng, “Mean Shift, Mode Seeking, and Clustering,” *IEEE Trans. PAMI*, vol. 17, no. 8, pp. 790-799, Aug. 1995.

[4] M. Fashing and C. Tomasi, “Mean Shift Is a Bound Optimization,” *IEEE Trans. PAMI*, vol. 27, no. 3, pp. 471-474, Mar. 2005.

[5] M. A. Carreira-Perpiñán, “Gaussian Mean Shift Is an EM Algorithm,” *IEEE Trans. PAMI*, vol. 29, no. 5, pp. 767-776, 2007.

[6] X. Li, Z. Hu and F. Wu, “A Note on the Convergence of the Mean Shift,” *Pattern Recognition*, vol. 40, no. 6, pp. 1756-1762, 2007.

[7] D. Comaniciu, V. Ramesh and P. Meer, “Real-Time Tracking of Non-Rigid Objects Using Mean Shift,” *IEEE Proc. CVPR 2000*, **2**, pp. 142-149, June 2000.

[8] R. Collins, “Mean-Shift Blob Tracking through Scale Space,” *IEEE Proc. CVPR 2003*, pp. 234-240, June 2003.

[9] Z. Zivkovic and B. Köse, “An EM-like Algorithm for Color-Histogram-Based Object Tracking,” *IEEE Proc. CVPR 2004*, vol. 1, pp. 798-803, 2004.

[10] K. Zhang, J. T. Kwok and M. Tang, “Accelerated Convergence Using Dynamic Mean Shift,” *Proc. ECCV 2006*, Part II, LNCS 3952, pp. 257-268, 2006.

[11] M. A. Carreira-Perpiñán, “Fast Nonparametric Clustering with Gaussian Blurring Mean-Shift,” *Proc. ICML 2006*, pp. 153-160, 2006.

[12] B. Boutsinas and T. Gnardellis, “On Distributing the Clustering Process,” *Pattern Recognition Letters*, vol. 23, no. 8, pp. 999-1008, June 2002.

[13] A. Y. Ng, M. I. Jordan and Y. Weiss, “On Spectral Clustering: Analysis and an Algorithm,” *Proc. NIPS 2001*, vol. 14, 2002.

[14] J. Shi and J. Malik, “Normalized Cut and Image Segmentation,” *IEEE Trans. PAMI*, vol. 22, no. 8, pp. 888-905, Aug. 2000.

[15] M. Maila and J. Shi, “A Random Walks View of Spectral Segmentation,” *Proc. AISTATS 2001*, Jan. 2001.