THE PATH ASSIGNED MEAN SHIFT ALGORITHM: A NEW FAST MEAN SHIFT IMPLEMENTATION FOR COLOUR IMAGE SEGMENTATION

Akash Pooransingh [*], *Cathy-Ann Radix*

Department of Electrical
and Computer Engineering,
University of the West Indies

Anil Kokaram

Department of Electronic
and Electrical Engineering,
Trinity College, Dublin

ABSTRACT

This paper presents a novel method for colour image segmentation derived from the mean shift theorem. When applied to colour image segmentation tasks, the path assigned mean shift algorithm performed 1.5 to 5 times faster than existing fast mean shift methods such as the Hierarchical 'Neighbourhood Consistency' FMS Method proposed by Zhang [1] with comparable results. The complexity of the new PAMS algorithm can be represented as $O(\phi^2)$ where ϕ represents the total number of unassigned points per iteration of the algorithm.

***Index Terms*—** Mean Shift, Fast Mean Shift, Colour Image Segmentation, Path Assigned Mean Shift

1. INTRODUCTION

Colour is a property that provides a strong clue to characterize objects. Image segmentation is a commonly used low level image processing technique for classifying objects. Clustering techniques may be used to classify homogeneous regions based on colour feature space. The mean shift method is a well established and powerful non-parametric clustering technique [2]. Mean shift is a popular choice for image segmentation due to its non parametric nature and the minimal user input.

The general mean shift algorithm was first proposed by Fukunaga and Hostetler in 1975 and reintroduced by Cheng [3] in 1995. According to Cheng, mean shift is a simple iterative procedure that shifts each data point to the average of the data points in its neighbourhood. The shift is determined by considering the gradient of the kernel density estimate.

Comaniciu [4] explored the meaning of the different components of the gradient, $\nabla\hat{f}(x)$ of the kernel density estimate (k.d.e.). Assuming that x_i corresponds to any point in the neighbourhood of size, n with a kernel bandwidth, h and assuming a Gaussian kernel, $g()$, the gradient of the kernel density estimate can be expressed as the product of two terms:

$$\nabla\hat{f}(x) = \underbrace{\left[\frac{1}{n}\sum_{i=1}^{n} g\left\|\frac{x-x_i}{h}\right\|^2\right]}_{K.D.E.} \underbrace{\left[\overbrace{\frac{\sum_{i=1}^{n} x_i.g\left\|\frac{x-x_i}{h}\right\|^2}{\sum_{i=1}^{n} g\left\|\frac{x-x_i}{h}\right\|^2}}^{C} - x\right]}_{m(x)} \tag{1}$$

In equation 1, the first term, is the K.D.E. at the point x. The second, displacement term is called the mean shift vector, $m(x)$. The gradient of the K.D.E., $\nabla\hat{f}(x)$ is proportional to the K.D.E. by a factor given by the mean shift vector, $m(x)$. The first part of $m(x)$ corresponds to the calculated neighbourhood centre of mass, C.

This shows that the mean shift vector always points toward the maximum increase of density i.e. the centre of mass. A proof of the convergence of the mean shift procedure is given in [2].

The mean shift method is effective in high density regions but for multidimensional data sets proves to be computationally expensive. Recently, [5], [6], [7], [1], work has been done to achieve fast mean shift methods capable of processing multidimensional data sets easily. This paper outlines one such method in the Path Assigned Mean Shift (PAMS).

In order to understand how the new algorithm addresses computational issues in the mean shift assignment, the next section presents a succinct statement of the general mean shift procedure (GMS). That is followed by a summary of fast mean shift algorithms including the PAMS.

2. THE GENERAL MEAN SHIFT METHOD

The mean shift algorithm may be applied directly to colour space feature sets in order to segment images. For the purposes of this paper, the YUV colour space will be considered. The algorithm below describes the general mean shift process applied to the UV colour domain:

Algorithm 1: General Mean Shift in the Colour Domain

1. Select a point site (p, q) at random in the image.
2. Extract the colour values vector of the pixel at that point $I_{U,V}(p, q)$.
3. Find the j neighbourhood vectors, $I_{(U,V)_j}(p, q)$, within the colour bandwidth, h_c.

[*] Thanks to the University of the West Indies for funding.

4. Compute the Center of Mass, C_c in the colour domain.
5. Translate by the mean shift vector, $m_c(U, V)$.
6. Repeat 3 and 5 till convergence to stationary mode vector, $I_{(U_m,V_m)}$. Assign the initial point, (p, q) the value of the final mode vector, $I_{(U_m,V_m)}$.

The main computational load for the general mean shift lies in the calculation of the mean shift vector, $m_c(U, V)$ (See Equation 1). The computational cost of the mean shift algorithm is $O(n^2)$ where n is the size of the data set.

3. FAST MEAN SHIFT METHODS

Comaniciu [8] improved the performance of the mean shift algorithm by reducing the dimensionality of the colour information. A single metric was used to represent each data point, thus the mean shift process was accelerated.

For the purpose of colour image segmentation, Comaniciu in [2] and [4] also showed that the mean shift procedure can be applied to both the spatial and colour domains. The major idea was to reduce the burden of feature space analysis by exploiting spatial (image) domain information. At any point site, only the vectors within a spatial bandwidth, h_s and a colour bandwidth, h_c are considered.

Comaniciu's methods discussed thus far improved the performance but the complexity still remained at $O(n^2)$.

De Menthon [5] introduced a hierarchical mean shift method. Here a hierarchical clustering method was used by repeatedly applying the mean shift over increasingly large bandwidth, with each step using the results of the previous to initialize. The complexity is stated as $O(p^2)$ where p is the number of vectors within a specified search range.

Zhang [1] introduced a unique fast mean shift concept and applied it to colour segmentation. This algorithm was arranged in two steps. 1. Partition: For fast mean shift the original data set is decomposed into a number of local subsets of similar size and centre calculated. 2. Clustering: The mean shift is calculated for each sample rather than the whole data set to find a single class for each sample - this is the notion of neighbourhood consistency. At the end similar subsets are fused together to create a single class. Here the complexity is stated as $O(Nlog(m)) + O(m^2)$ where m is the number of partitioned data subsets and N is the total number of points.

Major drawbacks of the these methods [1], [2], [4], [5] are: i) all feature space vectors are not considered and ii) each gives an approximate result.

4. THE PATH ASSIGNED MEAN SHIFT (P.A.M.S.): A NEW FAST MEAN SHIFT METHOD

The notion of 'neighbourhood consistency' was stated by Zhang in [1]. In 'neighbourhood consistency'the class of any random point is assumed by the class of its nearest neighbours. Comaniciu [8] showed that for any random start point, the mean shift vector always points to the mode point i.e. the stationary point. Thus far in fast mean shift segmentation, [2], [4], [1], [5] feature vectors are assigned to the final mode value.

In the PAMS assignment, all points along the path toward the mode point are assigned to that final mode value.

By so doing points already assigned modes are eliminated from the mean shift process and are not traversed in the future. Since large swathes of feature space vectors are now assigned in one iteration step, the complete mean shift process converges much faster. For the purposes of this paper, the YUV colour space will be considered. The algorithm below describes the path assigned mean shift process applied to the UV colour domain::

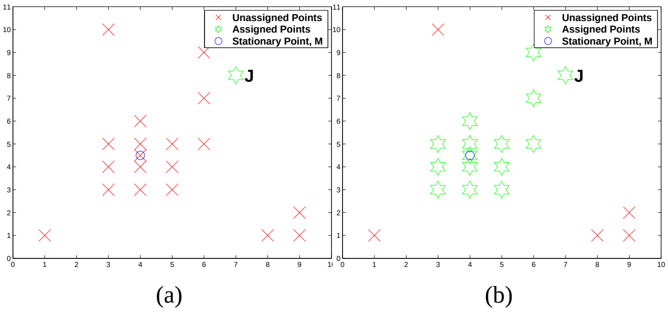
Algorithm 2: Path Assigned Mean Shift Algorithm in the Colour Domain

1. Select a point site (p, q) at random in the image.
2. Extract the colour values vector of the pixel at that point $I_{U,V}(p, q)$.
3. Find the j neighbourhood vector, $I_{(U,V)_j}(t)$, within the colour bandwidth, h_c.
4. Compute the Center of Mass, CoM_c in the colour domain.
5. Translate by the mean shift vector, $m_c(U, V)$.
6. Repeat 3 and 5 till convergence to stationary mode vector, $I_{(U_m,V_m)}$. Assign the final mode vector, $I_{(U_m,V_m)}$, to the entire mean shift path, $\bigcup_{t=0}^{t=i} I_{(U,V)_j}(t)$, where i is the number of iterations to convergence.

This reduces the complexity to $O(\phi^2)$ where ϕ represents the total number of unassigned points per iteration of the algorithm. Also computation time is reduced significantly with each iteration.

5. EXAMPLE: GENERAL MEAN SHIFT ASSIGNMENT VS PATH ASSIGNED MEAN SHIFT ASSIGNMENT

To illustrate the PAMS algorithm, take the following case. Consider a two dimensional normally distributed data set with a random starting point, J. Applying the mean shift algorithm would converge or 'climb the hill' to the mode point, M. Using the general mean shift assignment, the point J is assigned the mode value, M as shown in Figure 1(a). Using the PAMS method of assignment, all points along the path to the convergence point are assigned to the mode point M as shown in Figure 1(b). Here all points traversed towards the final stationary point are assigned (green) while the remaining points are left unassigned (red).



(a) (b)

Fig. 1. Example illustrating the comparison between (a) GMS Assignment and (b) PAMS Assignment in one iteration. Assigned points (Green,∗). Unassigned points (Red,X). Stationary Point (Blue,O)

6. RESULTS

The PAMS method is compared with: i) General Mean Shift Method [3] and ii) Hierarchical 'Neighbourhood Consistency' FMS Method proposed by Zhang [1]. All three methods were applied to colour image segmentation tasks in the YUV and UV colour spaces with a uniformly distributed kernel and a constant bandwidth h_c.

PAMS was compared against the existing methods both in terms of quality of segmentation and computational performance.

6.1. Image Segmentation

Figure 2 shows a comparison of segmentation between the general mean shift (GMS), the Hierarchical 'Neighbourhood Consistency' FMS Method proposed by Zhang (ZFMS) and the PAMS algorithm for the U-V dimension. Each algorithm performed with bandwidth, ($h_c = 15$) while the segmentation results are represented by the final mode value of each feature point.

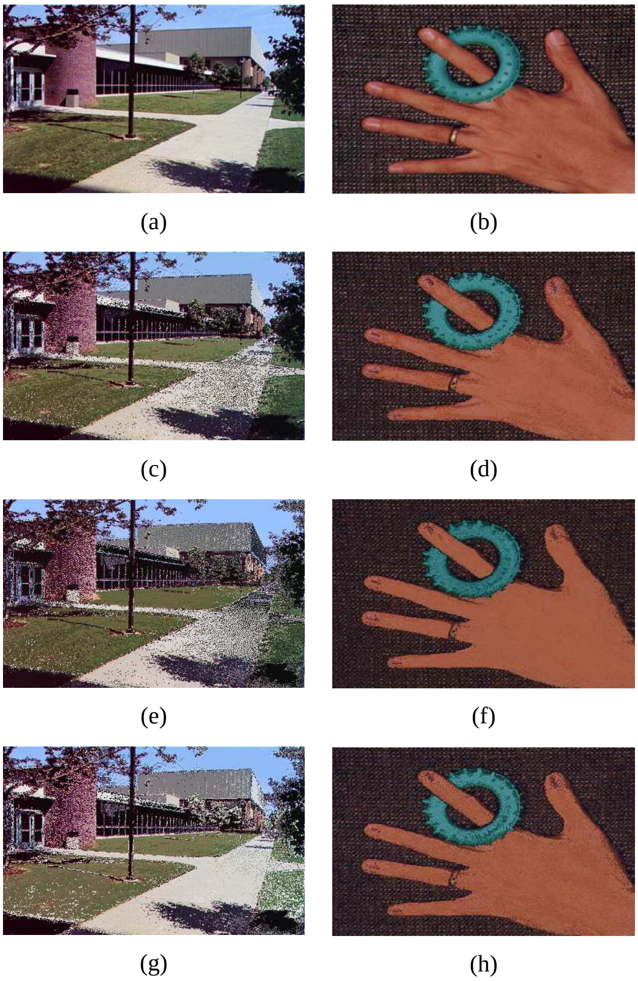


Fig. 2. Comparison Between the GMS, ZFMS and PAMS Segmentation in the U-V dimensions. (a)Original 1, (b)Original 2, (c)GMS 1, k = 105, (d)GMS 2, k = 86, (e)ZFMS 1, k = 23, (f)ZFMS 2, k = 25, (g)PAMS 1, k = 64 (h)PAMS 2, k = 52 where k is the number of classes.

6.2. Mislabeling Rate

Visual comparison is insufficient for determining the performance of the segmentation procedure. Hence a measure of similarity to the general mean shift method can be established to validate the results of segmentation. The Average Mislabeling Rate can be defined as:

$$\textit{Average Mislabeling Rate,}\hat{r} = \frac{\sum r_c}{N} \tag{2}$$

where N is the number of segmentation classes and r_c is defined as:

$$\textit{Mislabeling Rate per Mode Class, } r_c = \frac{G_c - F_c}{G_c} \tag{3}$$

where r_c is the mislabeling rate per mode class, G_c is the number of pixels assigned to a mode class in the general mean shift case and F_c is the number of pixels assigned to a mode class in the fast mean shift case. To illustrate this, the image shown in Figure 3, is segmented using the general mean shift algorithm to reveal a low number of base classes such that $k = 5$. Segmentation is performed by the PAMS and ZFMS algorithms to reveal similar base classes and the mislabeling rate is calculated as shown in Table 1.

AMR, ZFMS $\hat{r}_{zfms}\%$	19.27
Maxima, $\hat{r}_{zfms}\%$	40.51
Minima, $\hat{r}_{zfms}\%$	8.97
AMR, PAMS $\hat{r}_{pams}\%$	28.75
Maxima, $\hat{r}_{pams}\%$	56.02
Minima, $\hat{r}_{pams}\%$	2.71

Table 1. Mislabeling Rates: AMR, ZFMS - Average Mislabeling Rate, ZFMS; AMR, PAMS - Average Mislabeling Rate, PAMS

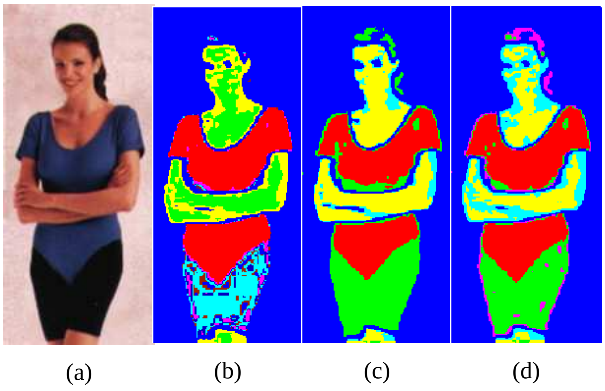


Fig. 3. Plots to show the dominant segmentation classes for an image for the general mean shift, the ZFMS and the PAMS algorithms. (a)Original Image, (b)GMS, hc = 115, (c)PAMS, hc = 40, (d)ZFMS, hc= 40.

6.3. Computational Performance

The algorithms mentioned above were implemented using MATLAB. The speed of each implementation can be compared by using MATLAB's internal stop watch timer. Randomly selected two dimensional, normally distributed test cases of sample size $N = 1000$, with increasing modes were chosen. In each case the mean shift procedure was performed until convergence, for 100 iterations and the average time was taken for each mode. Figure 4 shows a comparison of average running time versus number of modes for each algorithm.

(Note here that in this procedure, test data was generated deliberately to ensure that all algorithms compute the exact classes as in these cases only the computational performance of the algorithms were tested.) The PAMS algorithm performed ten (10) times faster than the GMS algorithm in the under segmented case (> 100 modes) and over one hundred (100) times faster in the over segmented case (< 10 modes)(results not shown). Figure 4 shows the PAMS algorithm performed (1.5) times faster than the ZFMS algorithm in the under segmented case (> 100 modes) and over five (5) times faster in the over segmented case (< 10 modes).

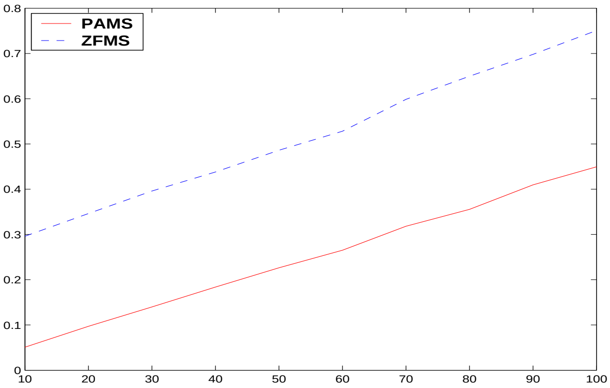


Fig. 4. Plot showing the comparison of computational performance (Average running time) for 1000 points over 100 iterations for ZFMS and PAMS algorithms for 10 - 100 Modes.

7. FINAL COMMENTS

This paper describes a new method for the fast mean shift. This method was applied to colour image segmentation in the YUV color domain for two dimensional (UV) clustering. This method was compared to the general mean shift and a new fast mean shift method proposed by Zhang, ZFMS. These methods were compared using both qualitative and quantitative measures.

Figure 4 showed that the PAMS algorithm operated at least five (5) times faster than the fast mean shift method proposed by Zhang for low number of modes (< 10) and 1.5 times faster for high mode cases(< 100). The new method also gave reasonable qualitative results compared to other methods. Table 1 shows that the PAMS segmentation and the ZFMS algorithm are comparable when compared to the GMS method.

Unlike most fast mean shift methods, the new method is a *true* mean shift as no preprocessing phase is needed such as Zhang's method and the ones proposed in [5] and [7]. For acceptable segmentation, no post processing is needed as in the case of Comaniciu [2] and [4]. In the new method only one input parameter is needed, that is the choice of the bandwidth, h_c. This is an improvement from the other methods that require a number of other inputs.

The complexity of the new PAMS process can be represented as $O(\phi^2)$ where ϕ represents the total number of unassigned points per iteration of the algorithm. In this process all points of the data set are considered unlike other methods that use a sample of the data set to reduce the complexity of the algorithm such as Zhang's method, and the one proposed by Yang [6].

One possible method of refinement for the PAMS method is the introduction of supervised classification. For example the PAMS may be considered as a first pass process to a Bayesian classification framework.

The PAMS can also be combined with the other fast mean shift concepts such as the Hierarchical 'Neighbourhood Consistency' FMS Method proposed by Zhang [1] to improve the performance.

Comaniciu [4] Collins [9] and others show that using a normally distributed, or Epanechnikov kernel, or an adaptive, varying kernel may yield more desirable results than using a uniform kernel. Comaniciu [10] and others show that the bandwidth can be automatically selected based on the calculation of the normalized density gradient that minimizes the *mean integrated squared error* (M.I.S.E.). Future work will focus on the proper selection of bandwidth and kernel to minimize the mislabeling rate.

8. REFERENCES

[1] Kai Zhang, Ming Tang, and J.T. Kwok, "Applying neighborhood consistency for fast clustering and kernel density estimation," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on Volume 2, 20-25 June*, Washington, DC, USA, 2005, pp. 1001 – 1007, IEEE Computer Society.

[2] Dorin Comaniciu and Peter Meer, "Mean shift analysis and applications," in *ICCV (2)*, 1999, pp. 1197–1203.

[3] Yizong Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, 1995.

[4] D. Comaniciu and P. Meer., "Mean shift: A robust approach toward feature space analysis.," in *PAMI, 24(5) May*, Washington, DC, USA, 2002, pp. 603–619, IEEE Computer Society.

[5] De Menthon D., "Spatio-temporal segmentation of video by hierarchical mean shift analysis," in *In Proc. Statistical Methods in Video Processing Workshop*, Washington, DC, USA, 2002, IEEE Computer Society.

[6] Changjiang Yang, Ramani Duraiswami, Nail A. Gumerov, and Larry Davis, "Improved fast gauss transform and efficient kernel density estimation," in *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, Washington, DC, USA, 2003, p. 464, IEEE Computer Society.

[7] Bogdan Georgescu, Ilan Shimshoni, and Peter Meer, "Mean shift based clustering in high dimensions: A texture classification example," in *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, Washington, DC, USA, 2003, p. 456, IEEE Computer Society.

[8] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on Volume 2, 13-15 June*, Washington, DC, USA, 2000, pp. 142 – 149, IEEE Computer Society.

[9] R.T. Collins, "Mean-shift blob tracking through scale space," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, Volume 2, 18-20 June*, Washington, DC, USA, 2003, pp. II –234–40, IEEE Computer Society.

[10] D. Comaniciu, V. Ramesh, and P. Meer, "The variable bandwidth mean shift and data-driven scale selection," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on Volume 1, 7-14 July*, Washington, DC, USA, 2001, pp. 438 – 445, IEEE Computer Society.