An approximation to mean-shift via swarm intelligence

M. Thomas and C. Kambhamettu
Video/Image Modeling and Synthesis Lab
University of Delaware, Newark, DE
{manivt, chandra}@cis.udel.edu

Abstract

Mean shift based feature space analysis has been shown to be an elegant, accurate and robust technique. The elegance in this non-parametric algorithm is mainly due to its simplicity in performing gradient ascent to estimate the modes in a multidimensional data. One characteristic aspect of mean shift is that the mode estimation is performed at each data point. Since it is important to describe the data in as succinct manner as possible, it is important to focus on modal points in the data instead of every data point. In this paper, we attempt to tackle the mean shift problem through a "mode centric" approach using swarm intelligence. Here, the mode estimation is cast as a problem of goal seeking for the swarm as it moves through the multidimensional data space. Local maxima/minima and plateaus are avoided through information exchange between each member of the swarm, thereby converging at the mode values efficiently.

1. Introduction

In today's world, information is extremely ubiquitous. Unfortunately, this pervasiveness brings with it the need to sift through the vast amounts of data which may or may not be of consequence. Data analysis has thus become an integral part of every single field of research that we have advanced into. Despite the ever-increasing computational prowess, analysis of a real life problem is an extremely daunting task due to the sheer influx of information content. An important task for any knowledge system would thus be to sieve through the vast amounts of data that is present and to retain a representative sample that provides the most concise and precise description of the information that they represent.

An obstacle that most research is constantly plagued with is known as the "Curse of Dimensionality". Bellman [4] defined this problem as the exponential growth of the hyper-volume of the data space with increasing dimensions. This increase in the volume causes data points to be extremely spread out and far apart (Sain [27], Georgescu et al. [16]). This decrease in the density of data in very high dimensions makes the estimation of any pattern in data extremely complex. This invariably has an important implication in developing a statistical technique, since most real world data tend to occupy a very high dimensional space. Thus any type of statistical analysis would invariably have to contend with this problem.

Mean shift algorithm is one such statistical tool that had been originally proposed by Fukunaga and Hostetler [15] and later researched upon by Cheng [6]. But it had remained largely hidden to the vision community until 2002, when Comaniciu and Meer [9] discussed the convergence characteristics of mean shift as an application to image segmentation and as a continuity preserving filter.

Typically, the mean shift approach has an $\mathcal{O}(N^2)$ complexity in low dimensional spaces with the constant typically in the range of $\sim$10. But with increasing dimensions of the data set, the computational complexity tends to become exponential with respect to d, the number of dimensions. One of the most expensive aspects of performing the mean shift operation arises in the computation of the nearest neighbors for a data point. As described by Arya et al. [1], most algorithms that perform "exact nearest neighbor" computation in very high dimensional space has an order of complexity that is very close to that of a brute force search.

The problem of improving the computational efficiency of the mean shift has been approached in multiple ways in the past. Georgescu et al. [16] describe an approximation algorithm to tackle this problem of multi-dimensional data by using the Locality Sensitive Hashing (LSH) technique developed by Indyk et al. [18]. Another possible improvement of the computation burden was proposed by Yang et al. [30], by approximating the Hessian matrix computation via a quasi-Newton method. Carreira-Perpiñán [5] describes various acceleration techniques to tackle the Gaussian Mean-Shift problem efficiently. The work indicates at least an order of magnitude speedup in computing Gaussian Mean-Shift procedure.

In this paper we try to improve the efficiency of this problem via a stochastic goal seeking mechanism using swarm intelligence. This scheme is driven by the estimation of the modal features present in the data, thereby making it "mode centric". The information exchange principle built into the swarm, helps it to efficiently avoid local minima/maxima when searching for the modes present in the data.

2. Background Work

As discussed above, it is essential to compress the vast amounts of data into a representative set of data samples that provide the best description of the entire data set. This problem of data clustering has been a much researched topic [12][20]. One of the most commonly applied clustering techniques is the $\mathcal{K}$-means clustering [22], which has gained significant popularity due to its low computational complexity of $\mathcal{O}(n\mathcal{K}Nd)$, when compared to many other techniques [16]. But $\mathcal{K}$-means imposes a strong constraint on the output clustering: the number of clusters, $\mathcal{K}$, must be known prior to performing the clustering. The second problem with $\mathcal{K}$-means is that the clusters are computed so as to minimize the intra cluster variance thereby creating hyper-ellipsoidal cluster edges. Nevertheless, under the assumption that the data can be represented as a mixture of Gaussians, the $\mathcal{K}$-means algorithm provides the best estimate for the cluster.

The use of stochastic algorithms for data clustering has been attacked by many researchers. In [24], Omran et al. describe a PSO based clustering scheme that is developed over the $\mathcal{K}$-means algorithm. The authors show that the coherency of the PSO clusters were better than the ones obtained by the standard $\mathcal{K}$-means algorithm with respect to the quantization error, inter and intra cluster distances. Despite the improved results that is indicated in the work, since the algorithm is developed over a $\mathcal{K}$-means framework, the algorithm inherently requires that the number of clusters be known prior to the clustering.

The Mean shift based approach, on the other hand, computes the clusters as a function of the modes present in the data, rather than as a function of the means of the data tessellation. Unlike the $\mathcal{K}$-means, it does not require any knowledge about the number of groups that are present in the data prior to clustering. The caveat is in the variable h that defines the bandwidth of the kernel being used to compute the local estimate, thereby placing a similar limiting condition as $\mathcal{K}$-means. But this problem of optimal bandwidth selection has been tackled using a two pass mechanism to estimate a local kernel bandwidth [7]. This mechanism, though robust and accurate, can prove to be computationally very expensive, especially with multidimensional data.

Since the work by Comaniciu and Meer [9], many variants and practical applications of the mean shift approach have appeared in research literature. Boomgaard and Weijer [28] described the connection between mean shift analysis and robust estimators. Fashing and Tomasi [13] showed that the mean shift procedure can be interpreted as a bounded quadratic optimization problem. Wang et al. [29] proposed a variant by modulating the mean shift vector with an anisotropic kernel instead of the radially symmetric kernel, thereby producing better segmentation results especially across the edges of coherent regions. DeMenthon [11] described a hierarchical mean shift mechanism to perform spatio-temporal video segmentation by analyzing the 7D feature vector composed of color and motion information.

The central principle involved in the mean-shift operation is given below for completeness. The readers are directed to [6] and [9] for more details. Given N data points $\mathbf{x}_i, i = 1 \cdots N$ in a d-dimensional space $\mathbb{R}^d$, the multivariate kernel density estimate obtained with the kernel $K(\mathbf{x})$ having a bandwidth parameter of h can be described as

$$f(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{N} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \tag{1}$$

The mean shift vector, $\mathbf{m}_h(\mathbf{x})$, can be subsequently computed from the gradient of the density estimator (Eq. 1).

$$\mathbf{m}_h(\mathbf{x}) = \frac{\sum_{i=1}^{N} \mathbf{x}_i g\left(\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\|^2\right)}{\sum_{i=1}^{N} g\left(\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\|^2\right)} - \mathbf{x} \tag{2}$$

where $g(s) = -k'(s)$ and k is called the shadow of the kernel K. The algorithm proceeds by iteratively shifting the data point $\mathbf{x}_i$ by the mean shift vector $\mathbf{m}_h(\mathbf{x})$ until the mean stabilizes to an estimate of the local mode of the data. This algorithm thus performs a gradient ascent to compute the mode of the distribution and has proven to have guaranteed convergence properties for specific kernels [9].

As can be observed, the biggest impediment in performing the mean shift operation is the computational cost that is incurred. Typically, the mean shift is applied to every data point and the computational complexity could be quite steep depending on the number of data points. This condition is further exacerbated when performing the analysis in high dimensional spaces, since the order of complexity would be $\mathcal{O}(N^2d)$ where d is the dimension of the space. One of the main factor that drives the computational cost is the number of data points that is involved in the process. It would be advantageous if we could perform this operation only on a small subset of the data points, thereby improving computational efficiency. The main principle of this paper is to develop a mechanism of selecting points from the feature space and using those points to estimate the modes intelligently. This intelligent selection of data points is accomplished via a stochastic scheme called Particle Swarm Optimization (PSO) that was developed in 1995 by Kennedy and Eberhart [21]. Unlike other evolutionary optimization schemes such as Genetic Algorithms, the PSO is a pseudo-optimization method inspired by the collective intelligence of swarms of biological populations. The most important characteristic of this method is that it is a zero-order, non-calculus based method, which can be used to solve for continuous variables [17].

2.1. Particle Swarm Optimization

In the original formulation by Kennedy and Eberhart [21], each particle in the population (also called the swarm) adjusted its trajectory towards its own best position and towards the best position attained by the whole group, mimicking the social behavior among flocks of birds [26]. The system dynamics are governed by the following equations.

$$\begin{aligned}\mathbf{v}_i^{(t+1)} &= \zeta[\omega\mathbf{v}_i^{(t)} + c_1\chi_1(\mathbf{p}_i^{(t)} - \mathbf{x}_i^{(t)}) \\ &+ c_2\chi_2(\mathbf{g}^{(t)} - \mathbf{x}_i^{(t)})] \quad (3)\\ \mathbf{x}_i^{(t+1)} &= \mathbf{x}_i^{(t)} + \mathbf{v}_i^{(t+1)} \quad (4)\end{aligned}$$

where $\chi_1, \chi_2 \sim U[0,1]$ are two $N_s \times N_s$ diagonal matrices of uniform random numbers with N_s being the total number of particles in the swarm. ω is the "inertia weight" that regulates the trade-off between the global (wide-ranging) and the local (nearby) exploratory capabilities of the swarm [26]. $\mathbf{x}_i^{(t-1)}$ is the i^{th} particle in the swarm at the $(t-1)^{th}$ iteration and $\mathbf{v}_i^{(t-1)}$ is its corresponding "velocity" component. ζ is called the "constriction term" which controls the influence of the velocity term to the positional term. $\mathbf{p}_i^{(t-1)}$ corresponds to the position of the best fitness value for the i^{th} particle while $\mathbf{g}^{(t-1)}$ corresponds to the best fitness value for the entire swarm.

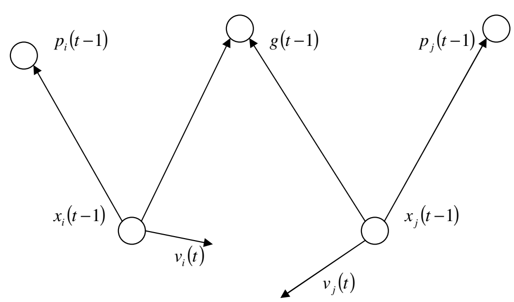


Figure 1. Particle system in the Particle Swarm Optimization model for a two particle ($\mathbf{x}_i^{(t-1)}$ and $\mathbf{x}_j^{(t-1)}$) system.

Among the three components of this dynamical equation, $\omega\mathbf{v}_i^{(t-1)}$ is the "inertial component", which constrains the velocity state estimate along the direction of $\mathbf{v}_i^{(t-1)}$. The second component, the "cognitive term" for each particle, $c_1\chi_1(\mathbf{p}_i^{(t-1)} - \mathbf{x}_i^{(t-1)})$ constrains the particle motion in the direction of its previous best value while the third component, the "social component", $c_2\chi_2(\mathbf{g}^{(t-1)} - \mathbf{x}_i^{(t-1)})$, directs the particles towards the best value among all the elements in the swarm. The random variables χ_1 and χ_2 provide for the stochastic parameters for the search with c_1 and c_2 as two positive weights that control each component (Eq. 1). It is interesting to note that functional optimizations in the PSO framework can be completely accomplished via additions and multiplications alone and is thus computationally very efficient.

2.2. Problem Statement

The PSO system dynamics is very similar to the "food" searching pattern that is seen in biological swarms. If the modes in the mean shift procedure can be represented as positions in the feature space that represents the "food", then the algorithm rests on the ability of the swarm in estimating these positions.

As with most stochastic optimization techniques, the primary necessity is in developing a fitness functional that could be used to evaluate the "goodness" of a particle in the swarm, so that the fitness would be high when the particle is "close to a goal" and low when it is "away from the goal". Another essential aspect of this fitness measure is that the fitness should be low at the vicinity of already estimated modes and high elsewhere, thereby forcing the swarm to move to other locations in search of possibilities. This "consume and move" paradigm is very similar to the "deflection" and "stretching" mechanism developed by Parsopoulos and Vrahatis [25].

Intuitively, this technique tries to capture the modes in the feature space of the data by searching at locations and passing on the information about the best locations as encountered by individual members of the swarm to the others. This mutual information sharing provides the swarm with a greater chance of escaping the local minima. Typically a swarm is composed of particles that are way smaller in number when compared to the input data, leading to efficient convergence patterns. In fact, this algorithm could be easily used as an initialization mechanism in order to be able to apply the exact mean shift at the vicinity of the estimate.

3. Algorithm Details

We begin this algorithm by defining a swarm of $N_s(N_s \ll N)$ particles. Let $\mathbf{x}_i^{(t)}, i = 1 \ldots N_s$ be the particles in the swarm. Each particle, $\mathbf{x}_i^{(t)}$ represents a point in the d-dimensional space $\mathbb{R}^d$. t denotes the current iteration number. At $t = 0$, $\mathbf{x}_i^{(t)}$ are mapped to N_s data points drawn randomly from the input data. The velocity, $\mathbf{v}_i^{(0)}$, for each particle is initialized to a d-dimensional random vector drawn from a uniform probability density function, $\mathcal{U}\left[-\mathbf{r}/2, \mathbf{r}/2\right]$, where $\mathbf{r}$ is the radius estimated for the mean shift operation.

Having thus initialized the swarm of particles to positions in the feature space, the subsequent stage of processing requires the evaluation of the fitness of each particle. Due to the unavailability of a parametric form of the fitness functional to perform the mode search, we computed an error metric that explained the local data statistics and also satisfied the constraints that were stated in section 2.2.

In this problem of estimating the modes, we resorted to an error metric instead of a fitness functional, with the mode estimation defined as a minimization of the error rather than as a maximization of the fitness. The error metric was composed by three components and can be defined as

$$\varepsilon_i^{(t)} = \rho_i^{(t)} * (\mathcal{D}_i^{(t)} + \mathcal{G}_i^{(t)}) \tag{5}$$

and the particle having a smaller $\varepsilon_i^{(t)}$ is considered to be a particle that is closer to the mode.

The first component, $\rho_i^{(t)}$, evaluates the average "density" of all particles within the spheroid around $\mathbf{x}_i^{(t)}$ within the radius $\mathbf{r}$.

$$\rho_i^{(t)} = \tfrac{1}{\eta} \textstyle\sum_{j=1}^{\eta} \|\mathbf{x}_j^{(t)} - \mathbf{x}_i^{(t)}\|, \quad \|\mathbf{x_j}^{(t)}\| \leq \mathbf{r} \tag{6}$$

where $\mathbf{x}_j$ are composed of all the points within the spheroid of radius $\mathbf{r}$. The density thus defines the average local spread of the data.

The second term in the fitness metric, $\mathcal{D}_i^{(t)}$, is composed of the mean shift error term that determines how close $\mathbf{x}_i^{(t)}$ is to the mean of the data points within $\mathbf{r}$. This term would have a low value if the magnitude of the error is small and large when the error is large. We used a "flipped" Gaussian to handle this characteristic.

$$\mathcal{D}_i^{(t)} = 1 - \exp\left[-\frac{\left(\frac{1}{\eta}\sum_{i=1}^{\eta}\mathbf{x}_j^{(t)}\right) - \mathbf{x}_i^{(t)}}{\mathbf{r}}\right] \tag{7}$$

where $\mathbf{x}_j^{(t)}$ corresponds to all the data points that are within the radius $\mathbf{r}$ at the t^{th} iteration.

The third term comes into play if a mode has been estimated in the previous mode seeking iteration. This term, $\mathcal{G}_i^{(t)}$, modifies the error metric for a data point as an inverse function of the distance to an estimated mode. Thus a point that is "closer" to an estimated mode would have a higher error than a point that is "further" away. Parsopoulos and Vrahatis [25] apply a "$\tanh$" function to apply "deflection" and "stretching" which has a similar profile with the Gaussian that has been applied here.

$$\mathcal{G}_i^{(t)} = \sum_{k=1}^{\phi} \exp\left[-\frac{\mathcal{M}_k - \mathbf{x}_i^{(t)}}{\mathbf{r}}\right] \tag{8}$$

where $\mathcal{M}_k$ is the k^{th} mode that has been estimated and ϕ is the total number of modes that have been currently estimated. The "flipped" Gaussian profile of $\mathcal{D}_i^{(t)}$ provides the normalization required to combine it with the $\mathcal{G}_i^{(t)}$ without imposing a bias towards either of the two terms.

During the iteration, the social term, $\mathbf{p}_i^{(t)}$, is updated to a new value from $\mathbf{x}_i^{(t)}$ if and only if the error has decreased between iteration t and $t-1$. The cognitive term $\mathbf{g}^{(t)}$ is always initialized to the position of the particle having the smallest error metric through it's entire history of iterations. The iterative dynamical system (Eq. 3) is applied to the particles until convergence. The convergence is indicated by the constancy of the cognitive term, $\mathbf{g}^{(t)}$, over a period of iterations that we define as "HISTORY". The iteration is repeated until the change in the cognitive term is significantly smaller than the convergence criterion ($\delta = 0.001$).

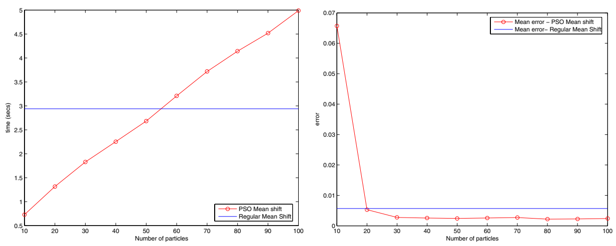
We also implemented the repulsion mechanism [25] to constrain the swarm to move away from an already estimated point. The repulsion coefficient was provided so as to push the particles from a previously estimated mode along the vector connecting the particles and the mode estimated in the given iteration. But this appeared to do more harm than good with an increase in the estimated error and increase in time complexity. It was observed that the particles hovered at a particular position, unable to converge, due to the repulsion effects occurring due to the presence of an already estimated mode in the vicinity.

4. Results and Analysis

The algorithm was tested on synthetic and real data. The synthetic data was rather a simple test to understand the convergence characteristics of the swarm approach. Two specific applications were tested using the proposed swarm technique. The first application involved image segmentation and the second application involved the problem of object tracking under non rigid dynamics under unknown initial positional estimate.

4.1. Synthetic test-case

This data set denotes a simple bimodal Gaussian distribution with the two modes of the distribution occurring at $[5.016, 4.962, 5.029]$ and $[-0.0145, 0.0157, 0.0190]$. The data is composed of 2000 data points and the estimated means of the data correspond to the mode of the data since they are individually Gaussian. Figure 2 shows the error variation with the number of particles used in the approximation and the time required against the number of particles required.



(a) (b)

Figure 2. Variation of the (a) computational time and (b) error against the number of particles used in the iterations.

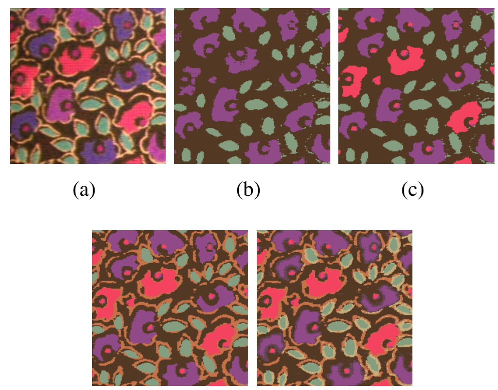
Due to the randomness involved in the particle trajectory selection, the algorithm was repeated 20 times for each particle number. The time and error was averaged across all the 20 runs per particle number. The error was computed against the actual mode as obtained from the bimodal Gaussian data. The blue line indicates the time taken by the regular mean shift. It can be seen that as the number of particles increase, the time required to compute the modes, in this simple case, also increases. $20 \sim 30$ particles were sufficient for a minimum error estimate. The constants that were used in the dynamical system (Eq. 1) were $\zeta = 0.5, w_1 = 0.9, w_2 = 0.4, c_1 = 2, c_2 = 2$. The inertia weight was computed as $w = w_1 + (w_2 - w_1)\frac{t}{MAXITER}$ where $MAXITER = 100$. This provided a means to reduce the exploratory characteristics of the swarm with increasing time, similar to the temperature variable in simulated annealing [3]. The convergence criterion, δ, was measured across the "HISTORY" of 5 iterations of $\mathbf{g}^{(t)}$.

4.2. Image segmentation

Image segmentation is one of the applications where the mean shift provides a very intuitive solution. In the mean shift approach the data points are clustered according to their modes instead of their locality specific means.

In the examples shown, the RGB components of the image were first projected into the LUV space prior to the mean shift operation [8]. The input image has a dimension of 200×200 leading to a feature space of 40000×3. The number of particles (N_s) that were used in this procedure were $\sim 0.01 \times N$, where N is the total number of data points. Thus the mode computation was performed at a fraction of the total input data points, thereby being computationally efficient.

It can be noted that the mode seeking characteristics of the swarm drives the mode estimation, so that the modes are obtained in approximately descending order of importance. With increasing selection of the modes, smaller structures within the images can be obtained as an output from the PSO based mean shift operation. This is evident in figure 3 where the PSO based mean shift was used to segment the image containing flowers ("fabric.png" in MATLAB). The first 3 estimated modes, segment the image into three important zones - background, leaves and flowers. Addition of the next mode into the classification provides a means to distinguish the two sets of flowers. The next mode adds the brown structure into the classification mechanism that is obtained from the background and so on and so forth.



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

Figure 3. (a) Original image. Segmentation using (b) the first 3 estimated modes (c) the first 4 estimated modes (d) the first 5 estimated modes and (e) the first 6 estimated modes

4.3. Algorithm Complexity

It is worth observing the computational time and accuracy of the mode estimation achieved by the PSO optimization scheme for the image segmentation. The algorithm was applied to the "peppers" image for varying sizes of the image. The regular mean shift and the particle swarm optimization based mean shift were applied to the images, and the segments were thus computed. Each algorithm was run twice and the time was averaged across the two runs to reduce any bias errors. As can be seen in figure 4, the regular mean shift has a quadratic complexity $\mathcal{O}(N^2)$ with respect to the input data points. The curves for increasing the size of the swarm shows an increase in the slope, but the significant loss in the accuracy is negligible as can be seen from the output clusters obtained from the PSO based scheme.

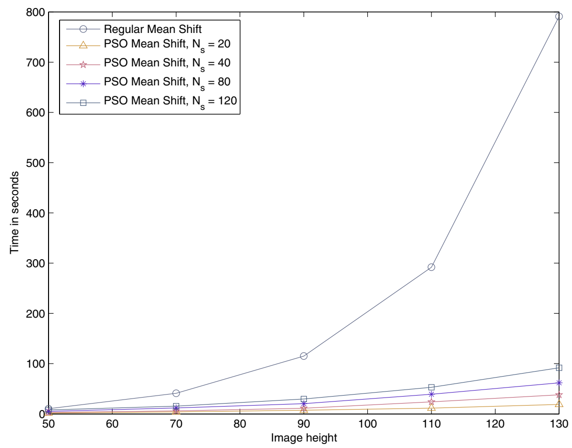


Figure 4. The time required to estimate the modes using the PSO based mean shift and the regular mean shift for the "peppers" image. The x-axis gives the height of the square images

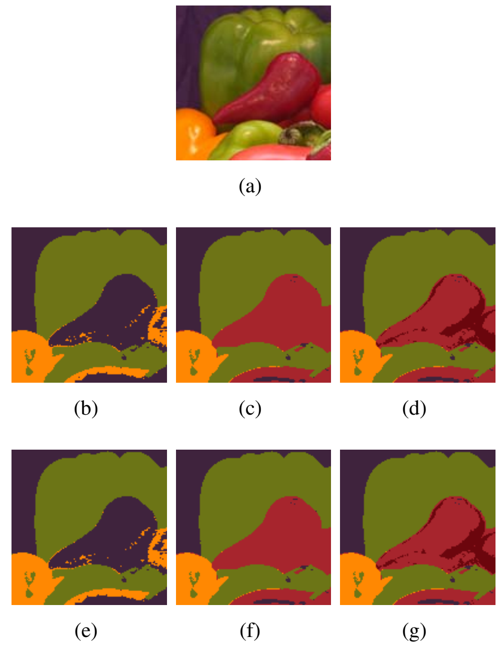


Figure 5. (a) Original image. Segmentation using (b)(e) the first 3 estimated modes (c)(f) the first 4 estimated modes and (d)(g) the first 5 estimated modes. The two rows indicate the segmentation using $N_s = 20$ and $N_s = 80$.

Shown in figure 5 is the segmentation of the peppers image using two (N_s = 20 and N_s = 80) of the four different particle numbers. As can be observed, the segmentation difference between the two columns is extremely close to each other, with minor variations at around the cluster edges.

5. Tracking as a mode estimation problem

It is important to validate the scalability of a technique across different problem domains. The PSO based dominant mode estimation can be shown, in this respect, to be easily adaptable to the problem of tracking. The field of surveillance and tracking has a plethora of research works, and it would be beyond the scope of this paper to give a detailed description of the techniques currently available. The readers are directed to the work by Bashir and Porikli [2] where several tracking characteristics are discussed and compared across many algorithms. Other interesting papers also include, but are not limited to, the work by Jacquot et al. [19] and [23], in which they describe a particle filter based system to perform color image tracking. When handling non rigid dynamics, the tracking problem is significantly more complex especially under scale/illumiantion variation. In [10], Comaniciu et al. proposes an algorithm for tracking non-rigid objects using the mean-shift based kernel, but as described by the authors (page 567 in [10]) the approximation of the Bhattacharyya coefficient [14] is satisfactory if the target does not change drastically from the initial model. This condition is a valid assumption when observing the change between consecutive frames but would be less accurate under large temporal difference between two frames. In this paper, we attempt to provide an accurate estimate of the location of the object under large inter frame temporal distance. This solution could be further improved upon using more complex local mechanisms.

Thus, a typical description of the tracking problem consists of estimating a maxima of some similarity criterion so as to locate the object spatially. As is evident, correlation based similarity metric would fail due to the presence of the non rigid motion and occlusion/disocclusion of pixels in the target image. To tackle this situation, we transformed the problem of maximization into that of estimating the position of the dominant mode. We used the Bhattacharyya coefficient (Eq. 9) between the model histogram and candidates from the target frame as the similarity metric.

$$\mathcal{B}(y) = \sum_{i=1}^{m}(\sqrt{p_i(y)q_i}) \tag{9}$$

where $p_i(y)$ are the candidate histograms, q_i is the model histogram and i is the histogram bin under consideration. We can observe that the Bhattacharyya coefficient computes the cosine similarity between two histograms. Since p and q are probability distributions (under proper normalization), $\mathcal{B}$ varies from 0 to 1 which indicates no overlap to maximum

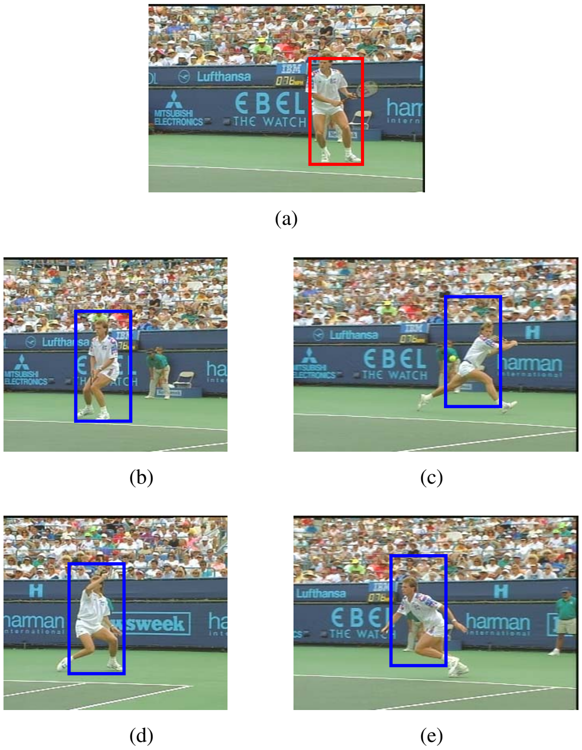


Figure 6. (a) Image 1 in the sequence - Red box indicates the model object to be tracked. Tracked object in the (b) 60^{th} (c) 170^{th} (d) 190^{th} and (e) 220^{th} frame. The input in all three cases is the model from the first frame alone.

overlap between the distributions respectively.

In order to map $\mathcal{B}$ to a set of data points, we create data point distributions that vary with the magnitude of $\mathcal{B}$. To achieve this, we use the Cauchy cumulative distribution function (Eq. 10) such that the number of data points varies from 0 to κ as $\mathcal{B}$ varies from 0 to 1.

$$F(x; x_0, \lambda, \kappa) = \kappa \left(\frac{1}{2} + \frac{1}{\pi} \arctan\left[\lambda(x - x_0)\right] \right) \quad (10)$$

with $\lambda = 100, x_0 = 1$ and $\kappa = 1000$. The parameter κ thus provides a scaling factor to control the number of samples that are selected. Thus regions in the image that have a higher Bhattacharyya coefficient would correspond to regions where the number of data points generated would be higher. This can be easily likened to the roulette wheel based selection where the probability of the selection of a slot is directly proportional to the similarity metric. It is typically observed that the concentration of the data points would be higher in the vicinity of the model under consideration. Thus finding the position of the highest concentration entails the position of the model in the collection of target candidates. Furthermore, it is essential to note that the application of a simple threshold does not provide a solution due to two reasons. They are the absence of a unimodal peak at the target position and the observation that the threshold would be data specific, which might cause the tracking to fail under incorrect threshold initialization.

To provide a brief description of the algorithm that was applied, we tessellate each image into coarsely overlapping regions and compute $\mathcal{B}$ between the model and the target tessellation. The data points are generated by sampling the Cauchy cdf (Eq. 10) for each tessellation. Given the coarse overlap between the tessellations, the positions where the concentration of data points were higher would thus correspond to a higher probability of the existence of the model. Applying the PSO based mean shift algorithm to find the position of the mode (highest concentration) of the data points, the target under consideration could be easily tracked.

In figure 5, we used a linearized color histogram obtained from the Y, U and V components of the image. The original YUV (4:2:2) image stream, in SIF (240×352) format, was obtained from the University of Missouri video repository. It can be easily seen that the model (red box in figure 5(a)) undergoes significant distortion and the estimated targets (blue box in figure 5(b)(c)(d)(e)) are quite accurate, despite the fact that the input is the model from the first frame and the tracking is on frames that are considerably advanced in time. One deficiency in this current method is that we have not added any mechanism to handle scale variation which is being currently researched upon. A simple solution to this would be to use the estimate obtained from our mechanism as an initialization to other complex algorithms that tackle the issues related to scale.

6. Comparative analysis

In both the applications above, it is observable that the mode estimation can be accomplished in a fraction of the original cost due to simplistic approach of using lesser number of candidates. In checking the efficiency of the algorithm, it is necessary to stress upon the fact that the algorithm was tested against a vanilla implementation of mean shift. This, we believed, provided a better approach to compare time complexities since optimizations can be performed on any mechanism, making it faster. First and foremost, due to fact that all estimates are performed at a position in feature space independently, the entire procedure can be modified to work in parallel.

Figure 4 indicates the approximate linear order when using particle swarm optimization which would also translate to speed in the problem of tracking. Despite the MATLAB implementation, the tracking algorithm was able to identify the object in $\sim$3 seconds despite significant non rigid dynamics. Thus an implementation using C would provide at least an order of magnitude improvement. Further comparison and results are available at vims.cis.udel.edu/∼mani/PSO

7. Conclusion

This paper describes a swarm intelligence approach to perform the mean shift operation. The mechanism uses Particle Swarm Optimization to constrain a random selection of data points to move through the d-dimensional space in search of modes of the given data. Information exchanged between the particles of the swarm through the social and cognitive components help the algorithm to converge to the goal efficiently. This information passing also helps the technique to move out of plateaus and local minima that occur when estimating the modes. Since the algorithm is "mode centric" rather than "data centric", it converges very fast to the most significant modes in the data based on their importance. Thus it provides an important tool to obtain a hierarchical distribution of the modes in the data. The two applications, in image segmentation and tracking, show the easy adaptability of the algorithm for tackling different problem types.

References

[1] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. *Journal of the ACM*, 45(6):891–923, 1998.

[2] F. Bashir and F. Porikli. Performance evaluation of object detection and tracking systems. In *9th IEEE Int. Workshop on Perf. Eval. of Tracking and Surveillance*, 2006.

[3] R. Battiti, M. Brunato, and S. Pasupuleti. Do not be afraid of local minima: Affine shaker and particle swarm. Technical report DIT-05-049, university of Trento, Italy, 2005.

[4] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[5] M. A. Carreira-Perpinan. Acceleration strategies for gaussian mean-shift image segmentation. In *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 2006.

[6] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8):790–799, 1995.

[7] D. Comaniciu. An algorithm for data-driven bandwidth selection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(2):281–288, 2003.

[8] D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. In *Proc. of the IEEE Conf. on CVPR*, pages 750–755, 1997.

[9] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.

[10] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):564–575, 2003.

[11] D. DeMenthon. Spatio-Temporal Segmentation of Video by Hierarchical Mean Shift Analysis. In *Statistical Methods in Video Processing Workshop*, June 2002.

[12] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2 edition, 2000.

[13] M. Fashing and C. Tomasi. Mean shift is a bound optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(3), 2005.

[14] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 2nd edition, 1990.

[15] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. in Inf. Theory*, pages 32–40, 1975.

[16] B. Georgescu, I. Shimshoni, and P. Meer. Mean shift based clustering in high dimensions: A texture classification example. In *9th International Conference on Computer Vision*, October, 2003.

[17] R. Hassan. Particle swarm optimization: Method and applications. Presentation at http://ocw.mit.edu, 2004.

[18] P. Indyk, R. Motwani, P. Raghavan, and S. Vempala. Locality-preserving hashing in multidimensional spaces. In *Proc. of the 29th annual ACM symposium on Theory of computing*, pages 618–625, 1997.

[19] A. Jacquot, P. Sturm, and O. Ruch. Adaptative tracking of non rigid objects based on color histograms and automatic parameter selection. In *Proceedings of IEEE Workshop on Motion and Video Computing*, pages 103–109, 2005.

[20] A. K. Jain, M. N. Murthy, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[21] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.

[22] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[23] K. Nummiaro, E. Koller-Meier, and L. V. Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 21:99–110, 2002.

[24] M. Omran, A. P. Engelbrecht, and A. Salman. Particle swarm optimization method for image clustering. *IJPRAI*, 19(3):297–321, 2005.

[25] K. E. Parsopoulos and M. N.Vrahatis. On the computation of all global minimizers through particle swarm optimization. *IEEE Trans. on Evol. Comp.*, 8(3):211–224, 2004.

[26] K. E. Parsopoulos and M. N. Vrahatis. Particle swarm optimization method for constrained optimization problems. In *Proc. of the Euro-Intl. Sym. on Comp. Int.*, 2002.

[27] S. Sain. *Adaptive Kernel Density Estimation*. PhD thesis, Rice University, 1994.

[28] R. van den Boomgaard and J. van de Weijer. On the equivalence of local-mode finding, robust estimation and mean-shift analysis as used in early vision tasks. In *Proc. ICPR*, 2002.

[29] J. Wang, B. Thiesson, Y. Xu, and M. Cohen. Image and video segmentation by anisotropic kernel mean shift. In *Proc. of ECCV*, 2004.

[30] C. Yang, R. Duraiswami, D. DeMenthon, and L. Davis. Mean-shift analysis using quasi-newwton methods. In *Proc. of ICIP*, pages 447–450, 2003.