Research Article

Perceptual Dominant Color Extraction by Multidimensional Particle Swarm Optimization

Serkan Kiranyaz,[1] Stefan Uhlmann (EURASIP Member),[1] Turker Ince,[2] and Moncef Gabbouj (EURASIP Member)[1]

[1] Signal Processing Department, Tampere University of Technology, P.O. Box 553, 33101 Tampere, Finland
[2] Faculty of Computer Science, Izmir University of Economics, 35330 Izmir, Turkey

Correspondence should be addressed to Serkan Kiranyaz, serkan.kiranyaz@tut.fi

Color is the major source of information widely used in image analysis and content-based retrieval. Extracting dominant colors that are prominent in a visual scenery is of utmost importance since the human visual system primarily uses them for perception and similarity judgment. In this paper, we address dominant color extraction as a dynamic clustering problem and use techniques based on Particle Swarm Optimization (PSO) for finding optimal (number of) dominant colors in a given color space, distance metric and a proper validity index function. The first technique, so-called Multidimensional (MD) PSO can seek both positional and dimensional optima. Nevertheless, MD PSO is still susceptible to premature convergence due to lack of divergence. To address this problem we then apply Fractional Global Best Formation (FGBF) technique. In order to extract perceptually important colors and to further improve the discrimination factor for a better clustering performance, an efficient color distance metric, which uses a fuzzy model for computing color (dis-) similarities over HSV (or HSL) color space is proposed. The comparative evaluations against MPEG-7 dominant color descriptor show the superiority of the proposed technique.

1. Introduction

DOMINANT Color (DC) extraction is basically a dynamic color quantization process, which seeks for such prominent color centers that minimize the quantization error. To this end, studying human color perception and similarity measurement in the color domain becomes crucial and there is a wealth of research performed in this field. For example in [1], van den Broek et al. focused on the utilization of color categorization (called as focal colors) for content-based image retrieval (CBIR) purposes and introduced a new color matching method, which takes human cognitive capabilities into account. They exploited the fact that humans tend to think and perceive colors *only* in 11 basic categories. In [2], Mojsilović et al. performed a series of psychophysical experiments analyzing how humans perceive and measure similarity in the domain of color patterns. One observation worth mentioning here is that the human eye cannot perceive a large number of colors at the same time, nor it is able to distinguish similar (close) colors well. Based on this, they showed that at the coarsest level of judgment, the human visual system (HVS) primarily uses dominant colors (i.e., few prominent colors in the scenery) to judge similarity.

The usual approach for DC extraction is to perform clustering in a color domain. The most popular clustering method, which is also used for MPEG-7 DC descriptor (DCD) [3], is K-means, [4]. However, clustering is a multimodal problem especially in high dimensions, which contains many suboptimum solutions resulting in over- and underclustering. Therefore, well-known deterministic methods such as K-means, Max-Min [4], FCM [4], and SOM [4] are susceptible to get trapped to the closest local minimum since they are nothing but greedy descent methods, which start from a random point in the solution space and perform a localized search. This fact eventually turns the focus on stochastic Evolutionary Algorithms (EAs) [5] such as Genetic Algorithms (GAs) [6], Genetic Programming (GP) [7], Evolution Strategies (ESs), [8] and Evolutionary

Programming (EP) [9], all of which are motivated by the natural evolution process and thus make use of evolutionary operators. The common point of all is that EAs are in population-based nature and can perform a global search. So they may avoid becoming trapped in a local optimum and find the optimum solution; however, this is never guaranteed.

Conceptually speaking, Particle Swarm Optimization (PSO) [10–12], which has obvious ties with the EA family, lies somewhere in between GA and EP. Yet unlike GA, PSO has no complicated evolutionary operators such as crossover, selection, and mutation. In a PSO process, a swarm of particles (or agents), each of which represents a potential solution to an optimization problem, navigate through the search space. Particles are initially distributed randomly over the search space and the goal is to converge to the global optimum of a function or a system. Several researchers have shown that PSO exhibits a better clustering performance than the aforementioned techniques [12–15]; however, when the problem is multimodal, PSO may also become trapped in local optima [16] due to the premature convergence problem especially when the search space is of high dimensions [12]. Furthermore, PSO has so far been applied to simple clustering problems [12–15], where the data space is limited and usually in low dimensions and the number of clusters (hence the solution space dimension) is kept reasonably low (e.g., <10). Moreover, all clustering methods mentioned earlier are static in nature; that is, the number of clusters has to be specified a priori. This is also true for PSO since in its basic form it can only be applied to a search space with a fixed dimension. Particularly for dominant color extraction, the optimal (true) number of DCs in an image is unknown and should thus be determined within the (PSO) process.

In this paper, we shall address data clustering as an optimization problem and present techniques, which extend PSO in a proper way to find optimal (number of) clusters in a multidimensional space. To alleviate the premature convergence problem, the so-called Fractional Global Best Formation (FGBF) collects all promising components from each particle and fractionally creates an artificial Global Best (GB) particle, the *aGB*, which may guide the swarm better than the swarm's native *gbest* particle [11] in such a way that the swarm can converge to the global optimum (or near-optimum) solution even in high dimensions and usually in earlier stages. In order to achieve a *dynamic* clustering where the optimum number of clusters is also determined within the process, we shall then present the so-called MultiDimensional Particle Swarm Optimization (MD PSO) method, which extends the native structure of PSO particles in such a way that they can make inter-dimensional passes with a dedicated dimensional PSO process [17]. Therefore, in a multidimensional search space where the optimum dimension is unknown, swarm particles can seek for both positional and dimensional optima. In recent works, both techniques have been successfully applied over multi-dimensional nonlinear function minimization and 2D data clustering [17], optimization over dynamic environments [18], and automatic design of artificial neural networks [19, 20], respectively. In this paper we adapt both techniques

to extract the optimal (number of) DCs in an image with respect to a cluster validity index and color domain.

Cluster validity analysis is the assessment of the clustering method's output using a specific criterion for optimality, that is, the so-called clustering validity index function [21]. Hence the *optimality* of any clustering method can *only* be assessed with respect to the validity index, which can be defined over a particular data representation with a proper distance (similarity) metric. Many existing DC extraction techniques, particularly the ones widely used in CBIR systems such as MPEG-7 DCD, have severe drawbacks and thus show a limited performance. The main reason for this is because most of them are designed based on some heuristics or naïve rules that are *not* formed with respect to what humans or more specifically the human visual system (HVS) finds "relevant" in color similarity. Therefore, it is of decisive importance that human color perception is considered whilst modeling and describing any color composition of an image. In other words, when a particular color descriptor is designed entirely based on HVS and color perception rules, further discrimination power and hence certain improvements in retrieval performance can be achieved. For this reason we shall propose a *fuzzy* model to achieve a perceptual distance metric over HSV (or HSL) color space, which provides means of modeling color in a way HVS does. In this way the discrimination between distinct colors is further enhanced, which in turns improves the clustering (and DC extraction) performance.

The rest of the paper is organized as follows. Section 2 surveys related work on DC extraction whilst presenting a brief overview on data clustering. The applications of MD PSO and FGBF for optimal dynamic clustering and the proposed DC extraction technique are presented in detail in Section 3. Section 4 provides the experiments conducted over a real image database and discusses the results. Finally, Section 5 concludes the paper.

2. Related Work

There is a wealth of research done and still going on in developing efficient DC extraction methods, which can be used in many applications, such as lossy compression techniques, mobile and hand-held devices, low-cost color displays, color look-up tables, and CBIR. In this article, we shall restrict the focus on the CBIR domain, which employs *color* as the descriptor for image retrieval. We shall then briefly introduce major data clustering methods.

2.1. DC Descriptors. In order to solve the problems of static quantization in color histograms, various DC descriptors, for example, [3, 22–25], have been developed using dynamic quantization with respect to image color content. DCs, if extracted properly according to the aforementioned color perception rules, can indeed represent the prominent colors in any image. They have a global representation, which is compact and accurate, and they are also computationally efficient. MPEG-7 DC descriptor (DCD) is adopted as in [23] where the method is designed with respect to HVS

color perceptual rules. For instance, HVS is more sensitive to changes in smooth regions than in detailed regions. Thus colors are quantized more coarsely in the detailed regions while smooth regions have more importance. To exploit this fact, a smoothness weight ($w(p)$) is assigned to each pixel (p) based on the variance in a local window. Afterwards, the *General Lloyd Algorithm* (*GLA*, also referred to as *Linde-Buzo-Gray* and it is equivalent to the well-known *K-means* clustering method [4]) is used for color quantization. For a color cluster C_i, its centroid c_i is calculated by

$$c_i = \frac{\sum w(p)x(p)}{\sum w(p)}, \quad x(p) \in C_i \qquad (1)$$

and the initial clusters for *GLA* are determined by using a weighted distortion measure, defined as,

$$D_i = \sum w(p)\|x(p) - c_i\|^2, \quad x(p) \in C_i. \qquad (2)$$

This is used to determine which clusters to split until either a maximum number of clusters (DCs), N_{DC}^{max}, is achieved or a maximum allowed distortion criteria, ε_D, is met. Hence, pixels with smaller weights (detailed sections) are assigned fewer clusters so that the number of color clusters in the detailed regions where the likelihood of outliers' presence is high is therefore suppressed. As the final step, an agglomerative clustering (AC) is performed on the cluster centroids to further merge similar color clusters so that there is only one cluster (DC) hosting all similar color components in the image. A similarity threshold T_S is assigned to the maximum color distance possible between two similar colors in a certain color domain (CIE-Luv, CIE-Lab, etc.). Another merging criterion is the color area; that is, any cluster should have a minimum amount of coverage area, T_A, so as to be assigned as a DC; otherwise, it will be merged with the closest color cluster since it is just an outlier. Another important issue is the choice of the color space since a proper color clustering scheme for DC extraction tightly relies on the metric. Therefore, a perceptually uniform color space should be used and the most common ones are CIE-Luv and CIE-Lab, which are designed such that color distances perceived by HVS are also equal in L_2 (Euclidean) distance in these spaces. For CIE-Luv, a typical value for T_S is between 10 and 25, T_A is between 1% and 5%, and $\varepsilon_D < 0.05$ [3].

2.2. Data Clustering. As the process of identifying natural groupings in a multidimensional data space based on some distance metric (e.g., *Euclidean*), data clustering can be divided into two main categories: hierarchical and partitional [4]. Each category then has a wealth of subcategories and different algorithmic approaches for finding the clusters. Clustering can also be performed in two different modes: hard (or crisp) and fuzzy. *K-means* [4] is a well known and widely used hard clustering method, which first assigns each data point to one of the K cluster *centroids* and then updates them to the *mean* of their associated points. Yet as a hard clustering method, *K-means* suffers from the several drawbacks, for example, (1) The number of clusters, K, needs to be set in advance, (2) Its performance directly depends

on the initial (random) centroid positions as the method converges to the closest local optima and (3) *K-means* is also dependent on the data distribution. There are many other clustering variants that are skipped where an extensive survey can be found in [4, 26].

A hard clustering technique based on the basic PSO (*bPSO*) was first introduced by Omran et al. in [13] and this work showed that the *bPSO* can outperform *K-means*, FCM, KHM, and some other state-of-the-art clustering methods in any (evaluation) criteria. This is indeed an expected outcome due to the PSO's aforementioned ability to cope with the local optima by maintaining a guided random search operation through the swarm particles. In clustering, similar to other PSO applications, each particle represents a potential solution at a particular time t, that is, the particle a in the swarm, $\xi = \{x_1, \ldots, x_a, \ldots, x_S\}$, is formed as $x_a(t) = \{c_{a,1}, \ldots, c_{a,j}, \ldots, c_{a,K}\} \Rightarrow x_{a,j}(t) = c_{a,j}$ where $c_{a,j}$ is the jth (potential) cluster centroid in N dimensional data space and K is the number of clusters fixed in advance. Note that the data space dimension, N, is now different than the solution space dimension, K. Furthermore, the fitness (validity index) function, f that is to be optimized, is formed with respect to two widely used criteria in clustering:

(i) *Compactness*: Data items in one cluster should be similar or close to each other in N dimensional space and different or far away from the others when belonging to different clusters.

(ii) *Separation*: Clusters and their respective centroids should be distinct and well-separated from each other.

The fitness functions for clustering are then formed as a regularization function fusing both *Compactness* and *Separation* criteria and in this problem domain they are known as clustering validity indices. The minimization of a validity index will simultaneously try to minimize the intra-cluster distances (for better *Compactness*) and maximize the inter-cluster distance (for better *Separation*). In such a regularization approach, different priorities (weights) can be assigned to both subobjectives via proper setting of weight coefficients; however, this makes the approach strictly parameter dependent. Another traditional and well-known validity index is Dunn's index [27], which suffers from two drawbacks. It is computationally expensive and sensitive to noise [21]. Several variants of Dunn's index were proposed in [26] where robustness against noise is improved. There are many other validity indices, that is, proposed by Turi [28], Davies and Bouldin [29], Halkidi et al. [21], and so forth. A throughout survey can be found in [21]. Most of them presented promising results; however, none of them can guarantee the "optimum" number of clusters in every clustering scheme.

Although PSO-based clustering outperforms many well-known clustering methods, it still suffers from two major drawbacks. The number of clusters K (being the solution space dimension as well) must still be specified in advance and similar to other *bPSO* applications, the method tends to trap in local optima particularly when the complexity of the

clustering scheme increases. This also involves the dimension of the solution space, that is, convergence to "optimum" number of "true" clusters can only be guaranteed for low dimensions. This is also true for dynamic clustering schemes, DCPSO [15] and MEPSO [22], both of which eventually present results only in low dimensions and for simple data distributions. All these drawbacks and limitations have successfully been addressed in [17].

3. The Proposed DC Extraction Technique

Humans tend to think and describe color the way they perceive it. Therefore, in order to achieve a color (dis) similarity metric taking HVS into account, HSV (or HSL), which is a perceptual color space and provides means of modeling color in a way HVS does, is used in the proposed technique for extracting dominant colors. Note that in a typical image with 24-bit RGB representation, there can be several thousands of distinct colors, most of which cannot be perceived by HVS. Therefore, to reduce the computational complexity of RGB to HSV color transformation and particularly to speed up the dynamic clustering process via MD PSO and FGBF, a pre-processing step, which creates a limited color palette in RGB color domain, is first performed. In this way such a massive, yet unperceivable amount of colors in RGB domain can be reduced to a reasonable number, for example, $256 < n < 512$. To this end, we used the Median Cut method [30] because it is fast (i.e., $O(n)$) and for such a value of n, it yields an image which can hardly be (colorwise) distinguished from the original. Only the RGB color components in the color palette are then transformed into HSV (or HSL) color space over which the proposed dynamic clustering technique is applied to extract the dominant colors, as explained next.

3.1. Dynamic Clustering by MD PSO with FGBF. Based on the earlier discussion it is obvious that the clustering problem requires the determination of the solution space dimension (i.e., number of clusters, K) and an effective mechanism to avoid local optima (i.e., traps, both dimensionally and spatially) particularly in complex clustering schemes in high dimensions (e.g., $K > 10$). The former requirement justifies the use of the proposed MD PSO technique while the latter calls for FGBF. A brief description of both techniques is given in the appendix and see [17] for details.

At time t, a particle a in the swarm, $\xi = \{x_1, \ldots, x_a, \ldots, x_S\}$, has the positional component formed as, $xx_a^{xd_a(t)}(t) = \{c_{a,1}, \ldots, c_{a,j}, \ldots, c_{a,xd_a(t)}\} \Rightarrow xx_{a,j}^{xd_a(t)}(t) = c_{a,j}$ meaning that it represents a potential solution (i.e., a cluster centroid) for the $xd_a(t)$ number of clusters whilst jth component being the jth cluster centroid. Apart from the regular limits such as (spatial) velocity, $V_{\max}$, dimensional velocity, $VD_{\max}$, and dimension range $D_{\min} \le xd_a(t) \le D_{\max}$, the (variable) N dimensional data space is also limited with some practical spatial range, that is, $X_{\min} < xx_a^{xd_a(t)}(t) < X_{\max}$. In case this range is exceeded even for a single dimension j, $xx_{a,j}^{xd_a(t)}(t)$, then all positional components of the particle for the respective dimension $xd_a(t)$ are initialized randomly within the range and this further contributes to

the overall diversity. Let Z be the set of points, $Z = \{z_p\}$, in the N-dimensional data space. The following validity index is used to obtain computational simplicity with minimal or no parameter dependency,

$$f\left(xx_a^{xd_a(t)}, Z\right) = Q_e\left(xx_a^{xd_a(t)}\right)(xd_a(t))^\alpha, \quad \text{where}$$

$$Q_e\left(xx_a^{xd_a(t)}\right) = \frac{1}{xd_a(t)} \sum_{j=1}^{xd_a(t)} \frac{\sum_{\forall z_p \in xx_{a,j}^{xd_a(t)}} \left\| xx_{a,j}^{xd_a(t)} - z_p \right\|}{\left\| xx_a^{xd_a(t)} \right\|},$$

(3)

where Q_e is the quantization error (or the average intra-cluster distance) as the *Compactness* term and $(xd_a(t))^\alpha$ is the *Separation* term, by simply penalizing higher cluster numbers with an exponential, $\alpha > 0$. Using $\alpha = 1$, the validity index yields the simplest form and becomes entirely parameter-free.

On the other hand, (hard) clustering has some constraints. Let $C_j = \{xx_{a,j}^{xd_a(t)}(t)\} = \{c_{a,j}\}$, for all $j \in [1, xd_a(t)]$, be the set of data points assigned to a (potential) cluster centroid $xx_{a,j}^{xd_a(t)}(t)$ for a particle a at time t. The partitions C_j, for all $j \in [1, xd_a(t)]$ should maintain the following constraints:

(1) Each data point should be assigned to one cluster set: $\bigcup_{j=1}^{xd_a(t)} C_j = Z$.

(2) Each cluster should contain at least one data point: $C_j \ne \{\phi\}$, for all $j \in [1, xd_a(t)]$.

(3) Two clusters should have no common data points: $C_i \cap C_j = \{\phi\}$, $i \ne j$ and for all $i, j \in [1, xd_a(t)]$.

In order to satisfy the 1st and 3rd (hard) clustering constraints, before computing the clustering fitness score via the validity index function in (3), all data points are first assigned to the *closest* centroid. Yet there is no guarantee for the fulfillment of the 2nd constraint since $xx_a^{xd_a(t)}(t)$ is set (updated) by the internal dynamics of the MD PSO process and hence any dimensional component (i.e., a potential cluster candidate), $xx_{a,j}^{xd_a(t)}(t)$, can be in an abundant position (i.e., no closest data point exists). To avoid this, a high penalty is set for the fitness score of the particle, that is, $f(xx_a^{xd_a(t)}, Z) \approx \infty$, if $\{xx_{a,j}^{xd_a(t)}\} = \{\phi\}$ for any j.

The major outlines so far given are sufficient for applying the MD PSO technique to dynamic clustering and for this purpose the details for the application of FGBF over MD PSO can be found in [17].

3.2. Fuzzy Model over HSV-HSL Color Domains. Let $c_1 = \{h_1, s_1, v_1\}$ and $c_2 = \{h_2, s_2, v_2\}$ be two colors in HSV domain. Assume for the sake of simplicity that the hue is between 0 to 360 degrees and both s and v are *unit* normalized. The normalized *Euclidean* distance between s and v can be defined as follows:

$$\|c_1 - c_2\|^2 = (v_1 - v_2)^2 + (s_1 \cos(h_1) - s_2 \cos(h_2))^2$$
$$+ (s_1 \sin(h_1) - s_2 \sin(h_2))^2.$$

(4)

During the dynamic clustering process by MD PSO and FGBF, the problem of using this equation for computing a color distance between a candidate color centroid, $xx_{a,j}^d(t)$, for all $j \in [1,d]$ and a color in the palette, $z_p \in xx_a^d(t)$, as in (3) is that it has a limited discrimination power between distinct colors, as it basically yields arbitrary fractional numbers despite the fact that HVS finds "no similarity" in between. Therefore, instead of using this typical distance metric for all color pairs, we adopt a perceptual approach in order to improve discrimination between different colors. Recall from the earlier discussion that humans can recognize and distinguish 8 to 12 colors. Recall that in [1], the authors exploited the fact that humans tend to think and perceive colors only in 11 basic categories. Hence above a certain hue difference between two colors, it is obvious that they become entirely different for HVS; for example, yellow and green are as different as yellow and blue or cyan or black or purple, and so forth. So if the hue difference is above a certain limit, a maximum difference should be used (i.e., 1.0). We have selected an upper limit by considering distinct colors number as only 8, therefore, the perceptual threshold is $\Delta T_H = 360/8 = 45$ degrees. In practice, however, even a lower hue threshold can also be used, because two colors, for instance, with 40 degrees of hue difference can hardly have any similarity—yet 45 degrees present a safe margin leaving any subjectivity out.

We then use a *fuzzy* color model for further discrimination. As shown in Figure 1, for a fixed hue, for example, red for HSV and green for HSL, a typical saturation (S) versus Value (V) or Lightness (L) plot can be partitioned into 5 regions: White (W), Black (B), Gray (G), Color (C), and Fuzzy (F), which is a transition area among others. W, B and G are the areas where there is absolutely no color (hue) component whereas in F there is a hint of a color presence with a known hue but perhaps not fully saturated. In C, the color described by its hue is fully perceivable with a varying saturation and value. It is a fact that the borders among color regions are highly subjective and this is the sole reason to use a large Fuzzy region, so as to address this subjectivity in color perception and thus to contain the error. This is the reason why there is no need for drawing precise boundaries of F (even if possible) or the boundaries between $W \leftrightarrow G$ and $B \leftrightarrow G$ because between two colors, *say* one in C and one in F, or both in C or both in F, the same distance metric shall anyway be applied (as in (4)) provided that they have hue differences less than ΔT_H thus presenting some degree of color similarity. This is not a condition in other cases where at least one color is from either of the "no color" areas. For instance, between $W \leftrightarrow G$ and $B \leftrightarrow G$, the distance should only be computed over V (or L) components because they have no *perceivable* color components. The boundaries are only important to distinguish areas such as C, W and B (and between $C \leftrightarrow G$) where there is no similarity among them. Therefore, as shown in the HSL map on the right with blue arrows, if two colors, despite the fact that they have similar hues (i.e., $\Delta H < \Delta T_H$), happen to be in such regions, maximum distance (1.0) shall be applied rather than computing (4).
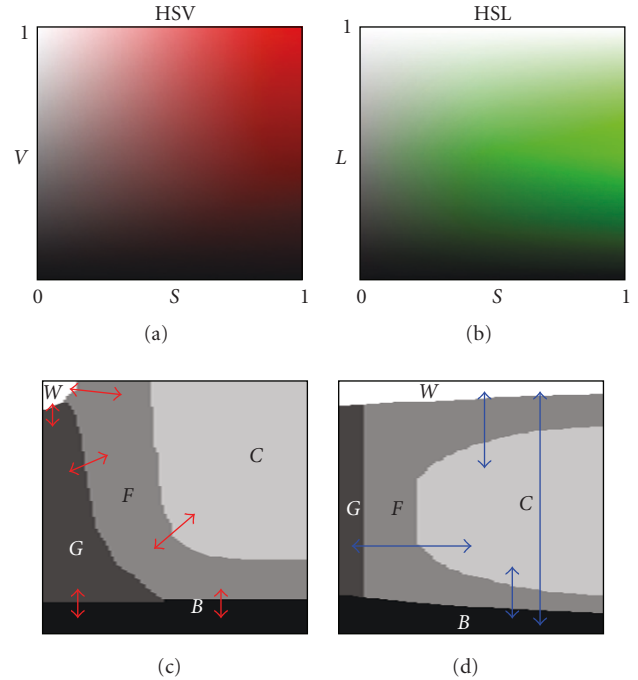

FIGURE 1: Fuzzy model for distance computation in HSV and HSL color domains (best viewed in color).

4. Experimental Results

We have made comparative evaluations against MPEG-7 DCD over a sample database with 110 images, which are selected from *Corel* database in such a way that their prominent colors (DCs) are easy to be recognized by ground-truth. We used the typical internal PSO parameters ($c_1, c_2,$ and w) as in [31]. Unless otherwise stated, in all experiments in this section, the two critical PSO parameters, swarm size (S), and number of iterations (*iterNo*) are set as 100 and 500, respectively. Their effects over the DC extraction are then examined. The dimension (search) range for DC extraction is set as $D_{\min} = 2$, $D_{\max} = 25$. This setting is in harmony with the maximum number of DCs set by the MPEG-7 DCD, that is, $N_{DC}^{\max} = 25$. Finally the size of the initial color palette created by the Median Cut method is set as 256.

4.1. Comparative Evaluations against MPEG-7 DCD. In order to demonstrate the strict parameter dependency of MPEG-7 DCD, we have varied only two parameters, T_A and T_S whilst keeping the others fixed, that is, $N_{DC}^{\max} = 25$, and $\varepsilon_D = 0.01$. Experiments are performed with three sets of parameters: P1: $T_A = 1\%$, $T_S = 15$, P2: $T_A = 1\%$, $T_S = 25$ and P3: $T_A = 5\%$, $T_S = 25$. The number of DCs (per image) plots obtained from the 110 images in the sample database using each parameter set can be seen in Figure 2. It is evident from the figure that the number of DCs is strictly dependent on the parameters used and can vary significantly, for example, between 2 to 25.
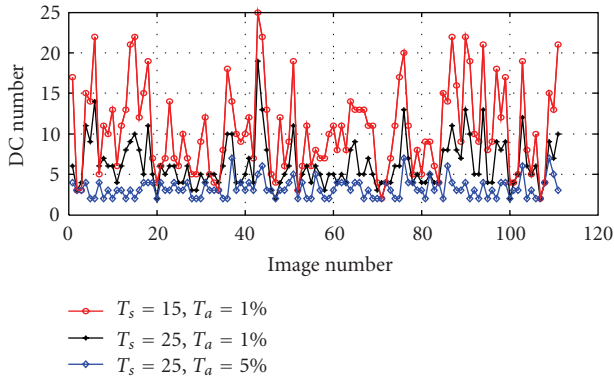
FIGURE 2: Number of DC plot from three MPEG-7 DCDs with different parameter set over the sample database.

Figures 3 and 4 show some visual examples from the sample database. In both figures, the first column shows the output of the Median-Cut algorithm with 256 (maximum) colors, which is almost identical to the original image. The second and the rest of the three columns show the back-projected images using the DCs extracted from the proposed technique and MPEG-7 DCD with those three parameter sets, respectively. Note that the parts, where DC centroids cannot be accurately localized or missed completely by MPEG-7 DCD, are pointed with (yellow) arrows. There is an ambiguity for deciding which parameter set yields the best visual performance although it would have naturally been expected from the first set, $P1$: $T_A = 1\%$, $T_S = 15$, where the highest number of DCs is extracted (see the *red* plot in Figure 2), but it is evident that $P2$ and $P3$ can also yield "comparable or better" results; however it is a highly subjective matter.

According to the results, one straightforward conclusion is that not only does the number of DCs significantly vary but DC centroids, as well, change drastically depending on the parameter values used. On the other hand, it is obvious that the best DC extraction performance is achieved by the proposed technique, where none of the prominent colors are missed or mislocated whilst the "true" number of DCs is extracted. However, we do not and, in any way, cannot claim that the proposed technique achieves the minimum quantization error (or the mean square error, MSE), due to two reasons. First, the optimization technique is applied over a regularization (fitness) function where the quantization error minimization (i.e., minimum Compactness) is only one part of it. The other part, implying maximum Separation, presents a constraint so that minimum MSE has to be achieved using the least number of clusters (DCs). The second and the main reason is that computation of MSE is typically performed in RGB color space, using the Euclidean metric. Recall that the proposed DC extraction is performed over HSV (or HSL) color domain, which is discontinuous and requires nonlinear transformations, and using a fuzzy distance metric with respect to the HVS perceptual rules for enhancing the discrimination power. Therefore, the optimization in this domain using such a fuzzy

metric obviously cannot ensure a minimum MSE in RGB domain. Besides that, several studies show that MSE is not an appropriate metric for visual (or perceptual) quality (e.g., [32]) and thus we hereby avoid using it as a performance measure.

4.2. Robustness and Parameter Invariance of the Proposed Method. Due to its stochastic nature, there is a concern about robustness (or repeatability) of the results. In this section, we perform several experiments to examine whether or not the results are consistent in regard to accuracy of the DC centroids and their numbers. Repeatability would be a critical problem for deterministic methods such as K-means, and Min-Max. if the initial color (cluster) centroids are randomly chosen, as the original algorithm suggests. Eventually such methods would create different clustering scheme each time they are performed since they are bound to get trapped to *the nearest* local optimum from the initial position. The solution to this problem induced by MPEG-7 DCD method is to change the random initialization part to a fixed (deterministic) initial assignment to the existing data points so that the outcome, DC centroids and the number of DCs extracted, will be the same each time the algorithm is performed over a particular image with the same parameters. This would also be a practical option for the proposed technique, that is, fixing the initialization stage and using a constant *seed* for the random number generator that MD PSO uses. However, as a global optimization method, we shall demonstrate that MD PSO with FGBF can most of the time converge to (near-) optimal solutions, meaning that, the number of DCs and their centroids extracted from the proposed dynamic clustering technique shall be consistent and perceptually intact. Furthermore, in order to show that significant variations for two major parameters, *iterNo* and *S,* do not cause drastic changes on the DC extraction, we will use three parameter sets: $P1$: $S = 50$, *iterNo* = 500, $P2$: $S = 50$, *iterNo* = 1000, and $P3$: $S = 100$, *iterNo* = 1000. With each parameter set, we run the proposed technique (with random initialization and random seeds) 100 times over two images. The DC number histograms per image and per parameter set are as shown in Figure 5. In the first image (left), it is certain that the number of DCs is either 2 or 3, as one might argue that the yellowish color of the dot texture over the object can be counted as a DC or not. For the image on the right, it is rather difficult to decide the exact number of DCs, since apart from blue, the remaining 5 colors, red, pink, yellow, green, and brown have certain *shades*. It is, first of all, obvious that the proposed technique is parameter invariant since in both cases, the significant parameter variations, (particularly from $P1$ to $P3$ where both *iterNo* and S are doubled) only make a *slight* difference over the histograms. A high degree of robustness (repeatability) is also achieved since *all* runs in the first image yielded either 2 or 3 DCs, as desired, and >95% of the runs in the second image; the number of DCs is in the range 16 ± 1. Among the back-projected images, it is evident that quite similar/almost identical DCs are anyway extracted even though they have different number of DCs

Median-cut (original) | Proposed | MPEG-7 DCD ($P1$) | MPEG-7 DCD ($P2$) | MPEG-7 DCD ($P3$)
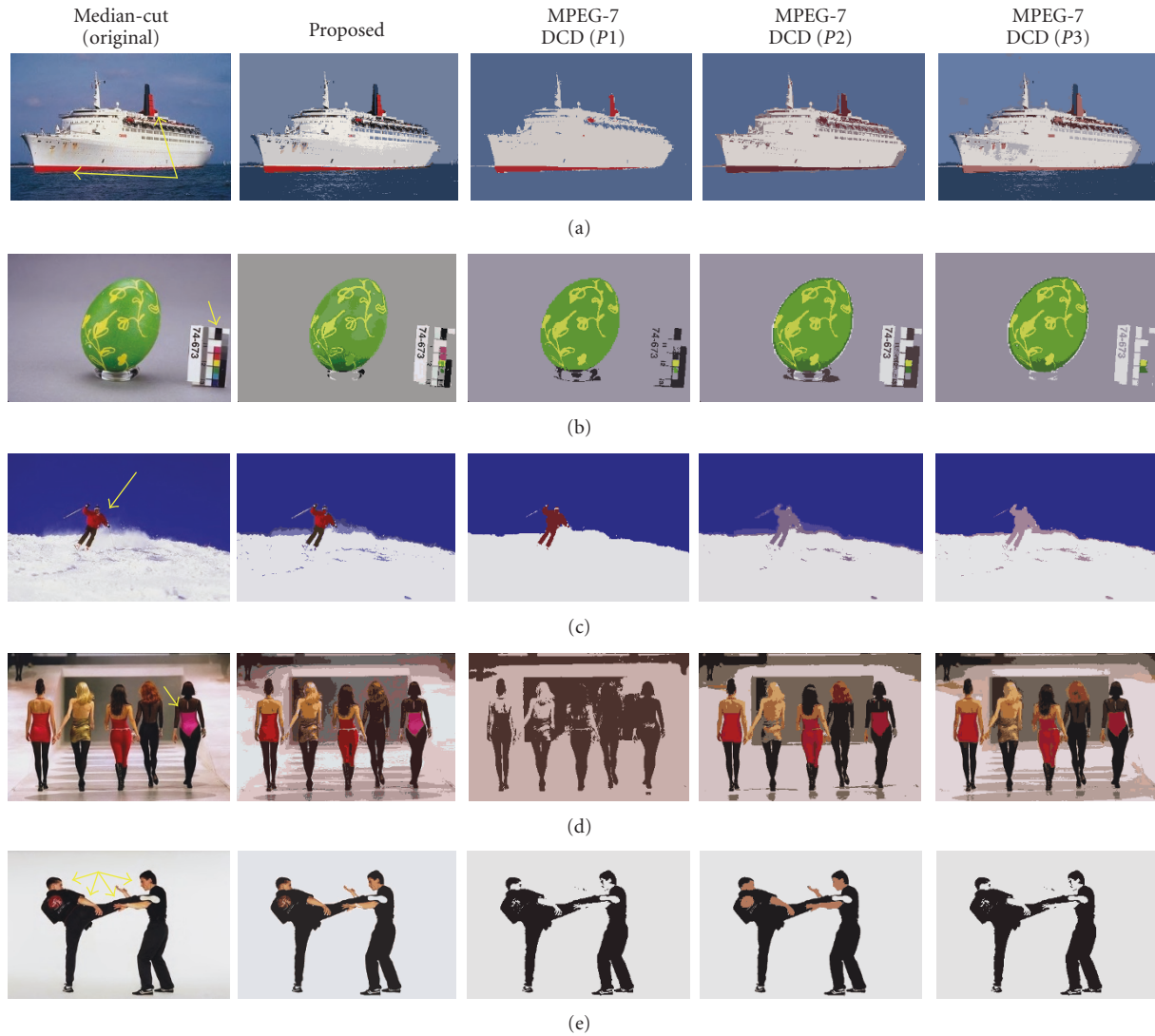
(a)

(b)

(c)

(d)

(e)

FIGURE 3: The DC extraction results over 5 images from the sample database (best viewed in color).

(e.g., see the two with 14 and 18 DCs). As a result of the perceptual model used, the number of DCs can slightly vary, somewhat reflecting the *subjectivity* in HVS color perception, but similar DCs are extracted by the proposed technique regardless of the parameter set used.

4.3. Computational Complexity Analysis. The computational complexity of the proposed method depends on two distinct processes. First is the preprocessing stage which creates a limited color palette in RGB color domain using Median Cut method and the following RGB to HSV color transformation. Recall that Median Cut method is a fast method (i.e., $O(n)$), which has the same computational complexity as K-means. The following color transformation has an insignificant processing time since it is only applied to a reduced number of colors. The computational complexity analysis for the dynamic clustering technique based on MD-PSO with FGBF is performed in [17]. Moreover, the complexity of the validity index used has a direct impact over the total computational

cost since for each particle (and at each iteration) it is used to compute the fitness of that particle. This is the main reason of using such a simple (and parameter independent) validity index as in (3). In that, the proposed fuzzy color model makes the computational cost primarily dependant on the color structure of the image because the normalized *Euclidean* distance that is given in (4) and is used within the validity index function is obviously quite costly; however, recall that it may not be used at all for such color pairs that do not show any perceptual color similarity. This further contributes the infeasibility of performing an accurate computational complexity analysis for the proposed technique. For instance, using a PC with P-IV 3 GHz CPU and 1 GB RAM, the proposed DC extraction technique with parameter set $P1$ took 126 and 911 milliseconds, respectively, for the two sample images shown in Figure 5. In short, as any other evolutionary algorithm the DC extraction based on MD PSO with FGBF is slow in nature and may require indefinite amount of iterations to converge to the global solution.

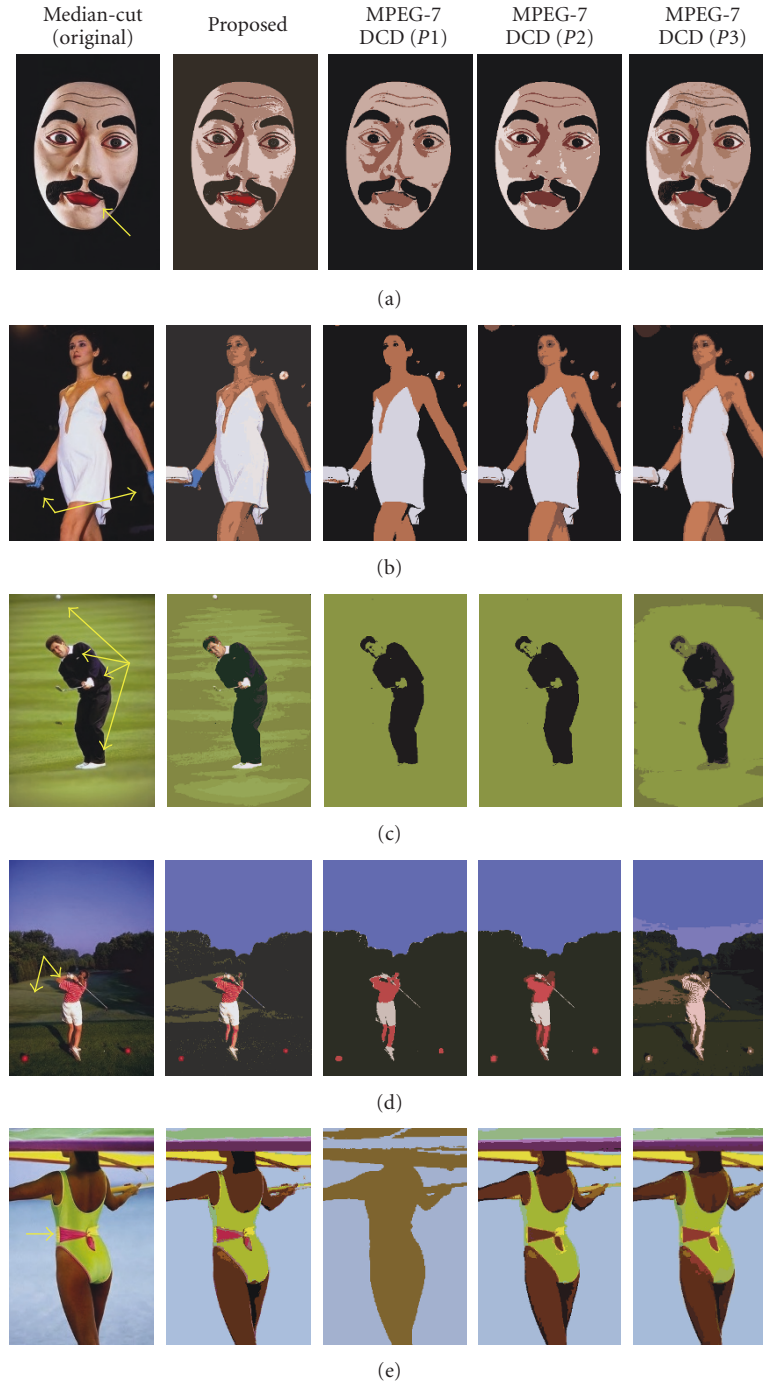| Median-cut (original) | Proposed | MPEG-7 DCD (*P*1) | MPEG-7 DCD (*P*2) | MPEG-7 DCD (*P*3) |

(a)

(b)

(c)

(d)

(e)

FIGURE 4: The DC extraction results over 5 images from the sample database (best viewed in color).

5. Conclusions

In this paper, we first presented two efficient techniques, MD PSO and FGBF, as a solution to common drawbacks of the family of PSO methods such as *a priori* knowledge of the search space dimension and premature convergence to local optima. A novel dynamic clustering technique based on MD PSO with FGBF is then proposed and applied for extracting "*true*" number of dominant colors in an image. In order to

improve the discrimination among different colors, a *fuzzy* model over HSV (or HSL) color space is then proposed so as to achieve such a distance metric that reflects HVS perception of color (dis) similarity.

The DC extraction experiments using MPEG-7 DCD have shown that the method, although a part of the MPEG-7 standard, is highly dependent on the parameters. Moreover, since it is entirely based on K-means clustering method, it can create artificial colors and/or misses some important
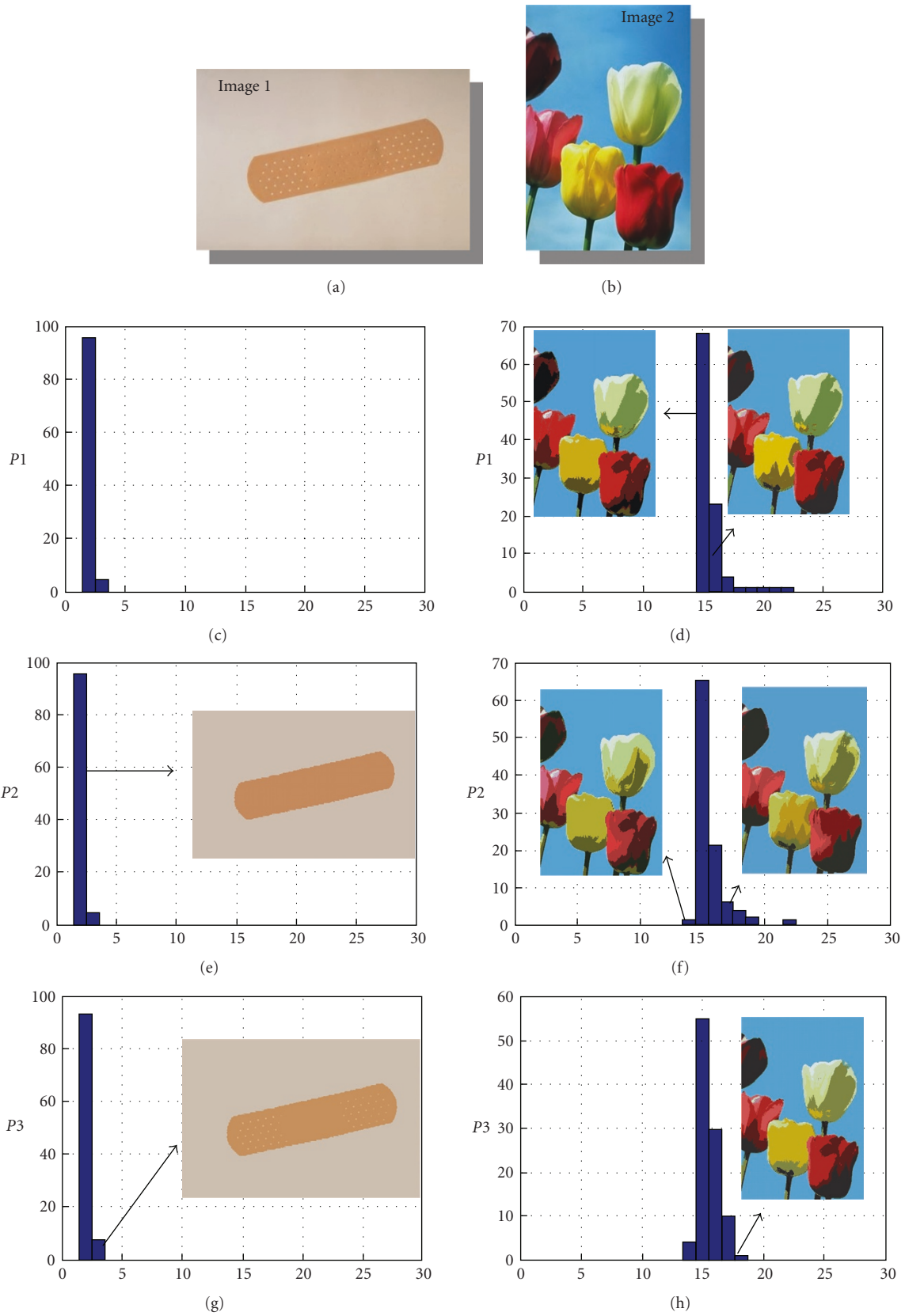
FIGURE 5: DC number histograms of 2 sample images using 3 parameter sets. Some typical back-projected images with their DC number pointed are shown within the histogram plots (best viewed in color).

DCs due to its convergence to local optima, thus yielding critical over- and under-clustering. Consequently, a mixture of different colors, and hence artificial DCs or DCs with shifted centroids, may eventually occur. This may also cause severe degradations over color textures since the regular textural pattern cannot be preserved if the true DC centroids are missed or shifted. Using a simple clustering validity index, we have successfully addressed these problems and a superior DC extraction is achieved with ground-truth DCs. The optimum number of DCs can slightly vary on some images, but the number of DCs on such images is hardly definitive, rather subjective and thus on such occasions the dynamic clustering based on a stochastic optimization technique can converge to some near-optimal solutions. The proposed technique shows a high level of robustness for parameter invariance and hence the main idea is that instead of struggling to fine tune several parameters to improve performance, which is not straightforward, if possible at all, the focus can now be drawn to designing better validity index functions or improving the ones for the purpose of higher DC extraction performance in terms of perceptual quality, which shall be the subject to our future work.

Appendices

A. MD PSO Algorithm

Instead of operating at a fixed dimension N, the MD PSO algorithm is designed to seek both positional and dimensional optima within a dimension range ($D_{\min} \leq N \leq D_{\max}$). In order to accomplish this, each particle has two sets of components, each of which has been subjected to two independent and consecutive processes. The first one is a regular positional PSO, that is, the traditional velocity updates and following positional moves in N-dimensional search (solution) space. The second one is a dimensional PSO, which allows the particle to navigate through dimensions. Accordingly, each particle keeps track of its last position, velocity and personal best position (*pbest*) in a particular dimension so that when it revisits the same dimension at a later time, it can perform its regular "positional" fly using this information. The dimensional PSO process of each particle may then move the particle to another dimension where it will remember its positional status and keep "flying" within the positional PSO process in this dimension, and so on. The swarm, on the other hand, keeps track of the *gbest* particles in all dimensions, each of which, respectively, indicates that the best (global) position so far achieved and can thus be used in the regular velocity update equation for that dimension. Similarly the dimensional PSO process of each particle uses its personal best dimension in which the personal best fitness score has so far been achieved. Finally, the swarm keeps track of the global best dimension, *dbest*, among all the personal best dimensions. The *gbest* particle in *dbest* dimension represents the optimum solution (and the optimum dimension).

For the description of the positional and dimensional components, we shall use the following *two-letter* notation rule for parameterization: the first letter signifies whether it is a position (x) or velocity (v) component and the second letter corresponds to the type of the component: positional (x) or dimensional (d). As in PSO notation, we also used y for personal best *(pbest)* position for the positional component. Therefore, the following enlists all the components of MD PSO:

(i) $xx_{a,j}^{xd_a(t)}(t)$: jth component (dimension) of the position of particle a, in dimension $xd_a(t)$,

(ii) $vx_{a,j}^{xd_a(t)}(t)$: jth component (dimension) of the velocity of particle a, in dimension $xd_a(t)$,

(iii) $xy_{a,j}^{xd_a(t)}(t)$: jth component (dimension) of the personal best (*pbest*) position of particle a, in dimension $xd_a(t)$,

(iv) $x\hat{y}_j^d(t)$: jth component (dimension) of the global best position of swarm, in dimension d,

(v) $xd_a(t)$: dimension component of particle a,

(vi) $vd_a(t)$: velocity component of dimension of particle a,

(vii) $x\widetilde{d}_a(t)$: personal best dimension component of particle a,

(viii) $gbest(d)$: global best particle index in dimension d.

Figure 6 shows sample MD PSO and *bPSO* particles with index a. The *bPSO* particle that is at a (fixed) dimension, $N = 5$, contains only positional components whereas MD PSO particle contains both positional and dimensional components, respectively. In the figure the dimension range for the MD PSO is given between 2 and 9; therefore the particle contains 8 sets of positional components (one for each dimension). In this example, the current dimension where the particle a resides is 2 ($xd_a(t) = 2$) whereas its personal best dimension is 3 ($x\widetilde{d}_a(t) = 3$). Therefore, at time t, a positional PSO update is first performed over the positional elements, $xx_a^2(t)$, and then the particle may move to another dimension by the dimensional PSO.

Let f denote the dimensional fitness function that is to be optimized within a certain dimension range ($D_{\min} \leq N \leq D_{\max}$). Without loss of generality assume that the objective is to find the minimum (position) of f at the optimum dimension within a multidimensional search space. Assume that the particle a visits (back) the same dimension after T iterations (i.e., $xd_a(t) = xd_a(t + T)$), then the personal best position can be updated in iteration $t + T$ as follows,

$$xy_{a,j}^{xd_a(t+T)}(t + T)$$
$$= \begin{cases} xy_{a,j}^{xd_a(t)}(t) & \text{if } f\left(xx_a^{xd_a(t+T)}(t+T)\right) \\ & > f\left(xy_a^{xd_a(t)}(t)\right) \\ xx_{a,j}^{xd_a(t+T)}(t+T) & \text{else} \end{cases} \quad (A.1)$$
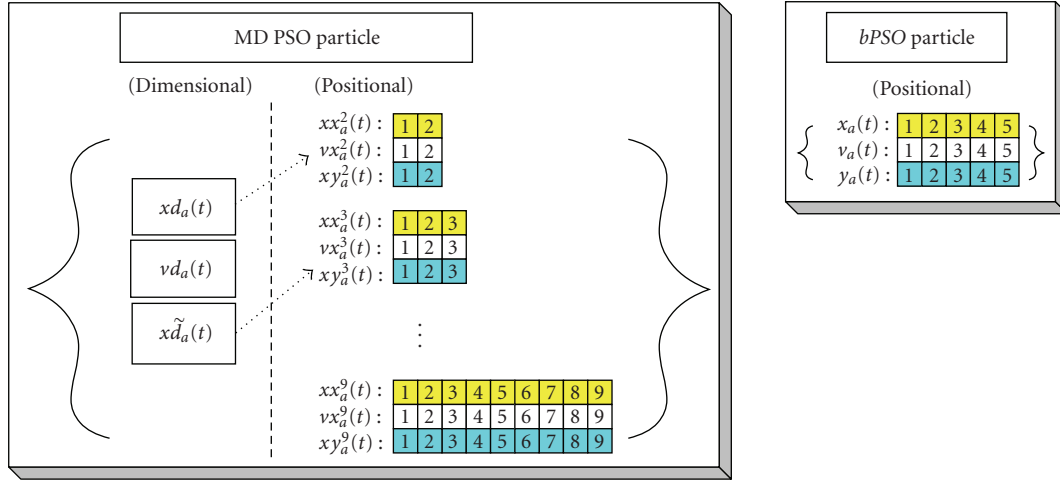
$$\forall j \in [1, xd_a(t)].$$

FIGURE 6: Sample MD PSO (right) versus *bPSO* (left) particle structures. For MD PSO $[D_{\min} = 2, D_{\max} = 9\,]$ and at the current time t, $xd_a(t) = 2$ and $x\tilde{d}_a(t) = 3$. For *bPSO* $N = 5$.

Furthermore, the personal best dimension of particle a can be updated in iteration $t + 1$ as follows:

$$x\tilde{d}_a(t+1) = \begin{cases} x\tilde{d}_a(t) & \text{if } f\left(xx_a^{xd_a(t+1)}(t+1)\right) \\ & > f\left(xy_a^{x\tilde{d}_a(t)}(t)\right) \\ xd_a(t+1) & \text{else} \end{cases} \quad \text{(A.2)}$$

Recall that $gbest(d)$ is the index of the global best particle at dimension d then $x\hat{y}^{dbest}(t) = xy_{gbest(dbest)}^{dbest}(t) = \arg\min_{\forall i\in[1,S]}(f(xy_i^{dbest}(t))$. For a particular iteration t, and for a particle $a \in [1,S]$, first the positional components are updated in the current dimension, $xd_a(t)$, and then the dimensional update is performed to determine the next $(t + 1\text{st})$ dimension, $xd_a(t + 1)$. The positional update is performed for each dimension component, $j \in [1, xd_a(t)]$ as follows:

$$vx_{a,j}^{xd_a(t)}(t+1) = w(t)\, vx_{a,j}^{xd_a(t)}(t) + c_1 r_{1,j}(t)$$
$$\times \left(xy_{a,j}^{xd_a(t)}(t) - xx_{a,j}^{xd_a(t)}(t)\right)$$
$$+ c_2 r_{2,j}(t)\left(x\hat{y}_j^{xd_a(t)}(t) - xx_{a,j}^{xd_a(t)}(t)\right),$$
$$xx_{a,j}^{xd_a(t)}(t+1) = xx_{a,j}^{xd_a(t)}(t) + vx_{a,j}^{xd_a(t)}(t+1).$$
$$\text{(A.3)}$$

Note that the particle's new position, $xx_a^{xd_a(t)}(t + 1)$, will still be in the same dimension, $xd_a(t)$; however, the particle may fly to another dimension afterwards with the following dimensional update equations:

$$vd_a(t+1) = \left\lfloor vd_a(t) + c_1 r_1(t)\left(x\tilde{d}_a(t) - xd_a(t)\right) \right.$$
$$\left. + c_2 r_2(t)(dbest - xd_a(t)) \right\rfloor \quad \text{(A.4)}$$
$$xd_a(t+1) = xd_a(t) + vd_a(t+1),$$

where $\lfloor \cdot \rfloor$ is the *floor* operator. The inertia weight, w, is not used for *dimensional* velocity update, since no benefit was obtained experimentally for dimensional PSO. To avoid exploding, along with the positional velocity limit $V_{\max}$, two more clamping operations are applied for dimensional PSO components, such as $|vd_{a,j}(t+1)| < VD_{\max}$ and the initial dimension range set by the user, $D_{\min} \le xd_a(t) \le D_{\max}$. Once the MD PSO process terminates, the optimum solution will be $x\hat{y}^{dbest}$ at the optimum dimension, $dbest$, achieved by the particle $gbest(dbest)$ and finally the best (fitness) score achieved will naturally be $f(x\hat{y}^{dbest})$.

B. FGBF Algorithm

Fractional GB formation (FGBF) is designed to avoid premature convergence by providing a significant diversity obtained from a proper *fusion* of the swarm's best components (the individual dimension(s) of the current position of each particle in the swarm). At each iteration in a *bPSO* process, an artificial GB particle (*aGB*) is (fractionally) formed by selecting the most promising (or simply the best) particle (dimensional) components from the entire swarm. Therefore, especially during the initial steps, the FGBF can most of the time be a better alternative than the native *gbest* particle since it has the advantage of assessing each dimension of every particle in the swarm individually, and forming the *aGB* particle fractionally by using the most promising (or simply the best) components among them. This process naturally uses the available diversity among individual dimensional components and thus it can prevent the swarm from trapping in local optima. Suppose for a swarm ξ, that FGBF is performed in a PSO process at a (fixed) dimension N. In a particular iteration, t, each PSO particle, a, has the following components: position $(x_{a,j}(t))$, velocity $(v_{a,j}(t))$ and the personal best position $(y_{a,j}(t)), j \in [1, N])$. The *aGB* particle, first of all, does not use a velocity term, since instead of velocity updates, the *aGB* particle
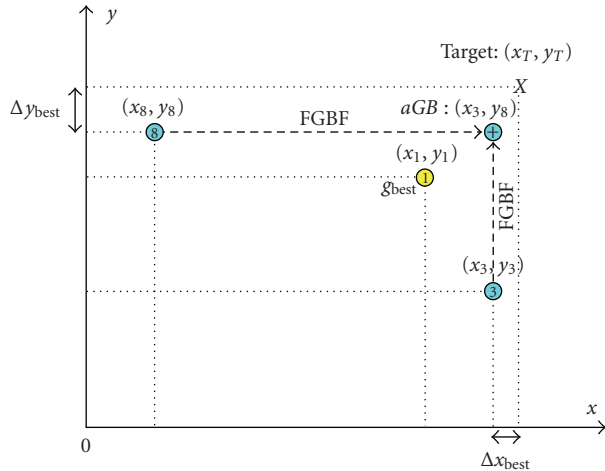
FIGURE 7: A sample FGBF in 2D space.

is fractionally (re) created from the dimensions of some swarm particles. Consequently, $y_{aGB}(t)$ is set to the best of $x_{aGB}(t)$ and $y_{aGB}(t-1)$. As a result, the FGBF process creates one aGB particle providing a (potential) GB solution ($y_{aGB}(t)$). Let $f(a, j)$ be the dimensional fitness score of the jth component of particle a and the computation of $f(a, j)$ depends entirely on the optimization problem. It keeps track of partial fitness contributions from each individual dimension from each particle's position (the potential solution). For those problems without any constraints (e.g., nonlinear function minimization), the best dimensional components can simply be selected whereas in others (e.g., clustering), some promising components, which satisfy the constraints, are first selected, grouped and the most suitable one in each group is then used for FGBF. Here, the internal nature of the problem will determine the "suitability" of the selection. Take for instance the function minimization problem as illustrated in Figure 7 where 2D space is used for illustration purposes. In the figure, three particles in a swarm are ranked as the 1st (or the *gbest*), the 3rd and the 8th with respect to their proximity to the target position (or the global solution) of some function. Although *gbest* particle (i.e., 1st ranked particle) is the closest in the overall sense, the particles ranked 3rd and 8th provide the best x and y dimensions (closest to the target's respective dimensions) in the entire swarm and hence the aGB particle via FGBF yields a better (closer) particle than the swarm's *gbest*.

Recall that in an MD PSO process there is one *gbest* particle per (potential) dimension of the solution space. Recall further from the earlier discussion that in a particular iteration, t, each MD PSO particle, a, has the following components: position ($xx_{a,j}^{xd_a(t)}(t)$), velocity ($vx_{a,j}^{xd_a(t)}(t)$) and the personal best position ($xy_{a,j}^{xd_a(t)}(t)$) for each potential dimensions in solution space (i.e., $xd_a(t) \in [D_{min}, D_{max}]$ and $j \in [1, xd_a(t)]$) and their respective counterparts in the dimensional PSO process (i.e., $xd_a(t)$, $vd_a(t)$ and $x\tilde{d}_a(t)$). aGB particle does not need dimensional components where a single positional component with the maximum dimension

D_{max} is created to cover all dimensions in the range, $\forall d \in [D_{min}, D_{max}]$, and as explained earlier, there is no need for the velocity term either. Furthermore, the aforementioned competitive selection ensures that $xy_{aGB}^d(t)$, $\forall d \in [D_{min}, D_{max}]$ is set to the best of the $xx_{aGB}^d(t)$ and $xy_{aGB}^d(t-1)$. As a result, the FGBF process creates one aGB particle providing (potential) GB solutions ($xy_{aGB}^d(t)$) for all dimensions in the given range (i.e., for all $d \in [D_{min}, D_{max}]$). Further details and pseudocode for FGBF and its application along with the MD PSO can be obtained from [17].

Acknowledgment

References

[1] E. L. van den Broek, P. M. F. Kisters, and L. G. Vuurpijl, "The utilization of human color categorization for content-based image retrieval," in *Human Vision and Electronic Imaging IX*, vol. 5292 of *Proceedings of SPIE*, pp. 351–362, San Jose, Calif, USA, 2004.

[2] A. Mojsilović, J. Kovačević, J. Hu, R. J. Safranek, and K. Ganapathy, "Matching and retrieval based on the vocabulary and grammar of color patterns," *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 38–54, 2000.

[3] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 703–715, 2001.

[4] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 316–323, 1999.

[5] T. Back and H. P. Schwefel, "An overview of evolutionary algorithm for parameter optimization," *Evolutionary Computation*, vol. 1, no. 1, pp. 1–23, 1993.

[6] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.

[7] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, Mass, USA, 1992.

[8] T. Back and F. Kursawe, "Evolutionary algorithms for fuzzy logic: a brief overview," in *Fuzzy Logic and Soft Computing*, pp. 3–10, World Scientific, Singapore, 1995.

[9] U. M. Fayyad, G. P. Shapire, P. Smyth, and R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining*, MIT Press, Cambridge, Mass, USA, 1996.

[10] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons, New York, NY, USA, 2005.

[11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, 1995.

[12] F. van den Bergh, *An analysis of particle swarm optimizers*, Ph.D. thesis, Department of Computer Science, University of Pretoria, Pretoria, South Africa, 2002.

[13] M. Omran, A. Salman, and A. P. Engelbrecht, "Image classification using particle swarm optimization," in *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL '02)*, vol. 1, pp. 370–374, Singapore, 2002.

[14] M. G. Omran, A. Salman, and A. P. Engelbrecht, *Particle Swarm Optimization for Pattern Recognition and Image Processing*, Springer, Berlin, Germany, 2006.

[15] M. G. H. Omran, A. Salman, and A. P. Engelbrecht, "Dynamic clustering using particle swarm optimization with application in image segmentation," *Pattern Analysis and Applications*, vol. 8, no. 4, pp. 332–344, 2006.

[16] J. Riget and J. S. Vesterstrom, "A diversity-guided particle swarm optimizer—the ARPSO," Tech. Rep., Department of Computer Science, University of Aarhus, 2002.

[17] S. Kiranyaz, T. Ince, A. Yildirim, and M. Gabbouj, "Fractional particle swarm optimization in multi-dimensional search space," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*. In press.

[18] S. Kiranyaz, J. Pulkkinen, and M. Gabbouj, "Multi-dimensional particle swarm optimization for dynamic environments," in *Proceedings of the 5th International Conference on Innovations in Information Technology (IIT '08)*, pp. 34–38, Al Ain, UAE, December 2008.

[19] T. Ince, S. Kiranyaz, and M. Gabbouj, "A generic and robust system for automated patient-specific classification of ECG signals," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 5, pp. 1415–1426, 2009.

[20] S. Kiranyaz, T. Ince, A. Yildirim, and M. Gabbouj, "Unsupervised design of artificial neural networks via multi-dimensional particle swarm optimization," in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR '08)*, Tampa, Fla, USA, December 2008.

[21] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, no. 2-3, pp. 107–145, 2001.

[22] A. Abraham, S. Das, and S. Roy, "Swarm intelligence algorithms for data clustering," in *Soft Computing for Knowledge Discovery and Data Mining Book, Part IV*, pp. 279–313, 2007.

[23] Y. Deng, C. Kenney, M. S. Moore, and B. S. Manjunath, "Peer group filtering and perceptual color image quantization," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '99)*, vol. 4, pp. 21–24, Orlando, Fla, USA, May 1999.

[24] J. Fauqueur and N. Boujemaa, "Region-based image retrieval: fast coarse segmentation and fine color description," in *Proceedings of the International Conference on Image Processing (ICIP '02)*, Rochester, NY, USA, September 2002.

[25] A. Mojsilović, J. Hu, and E. Soljanin, "Extraction of perceptually important colors and similarity measurement for image matching, retrieval, and analysis," *IEEE Transactions on Image Processing*, vol. 11, no. 11, pp. 1238–1248, 2002.

[26] N. R. Pal and J. Biswas, "Cluster validation using graph theoretic concepts," *Pattern Recognition*, vol. 30, no. 6, pp. 847–857, 1997.

[27] J. C. Dunn, "Well separated clusters and optimal fuzzy partitions," *Journal of Cybernetics*, vol. 4, pp. 95–104, 1974.

[28] R. H. Turi, *Clustering-based colour image segmentation*, Ph.D. thesis, Monash University, Clayton, Australia, 2001.

[29] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 224–227, 1979.

[30] A. Kruger, "Median-cut color quantization," *Dr. Dobb's Journal*, vol. 19, no. 10, pp. 46–92, 1994.

[31] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, Anchorage, Alaska, USA, May 1998.

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.