International Journal of Hybrid Information Technology
Vol.8, No.1 (2015), pp.279-294
http://dx.doi.org/10.14257/ijhit.2015.8.1.25

Improvement Cat Swarm Optimization for Efficient Motion Estimation

Prof. Dr. Israa Hadi[1] and Mustafa Sabah[2]
[1]*Professor College of Information Technology University of Babylon,*
[2]*Ph.D. Student, College of Information Technology University of Babylon,*
[1]*Israa_hadi1968@yahoo.com*, [2]*muustafa_bayat@yahoo.com*

Abstract

Cat swarm optimization (CSO) is a novel meta-heuristic for evolutionary optimization algorithms based on swarm intelligence. CSO imitates the behavior of cats through two sub-modes: seeking and tracing. Previous studies have indicated that CSO algorithms outperform other well-known meta-heuristics, such as genetic algorithms and particle swarm optimization, because of complexity, sometimes the pure CSO takes a long time to converge to reach to optimal solution. For improving the convergence of CSO with better accuracy and less computational time, this study presents an improvement structure of cat swarm optimization (ICSO), capable of improving search efficiency within the problem space under the conditions of a small population size and a few iteration numbers.

In this paper, an improved algorithm is presented by mixing two concepts, first concept found in parallel cat swarm optimization (PCSO) method for solving numerical optimization problems. The parallel cat swarm optimization (PCSO) method is an optimization algorithm designed to solve optimization problems Based on cats' cooperation and competition for improving the convergence of Cat Swarm Optimization,, the second concept found in Average-Inertia Weighted CSO (AICSO) by adding a new parameter to the velocity update equation as an inertia weight and used a new form of the position update equation in the tracing mode of algorithm. The performance of ICSO is sensitive to the control parameters selection.

The experimental results show that the proposed algorithm gets higher accuracy than the existing methods and requires less computational time and has much better convergence than pure CSO, and the proposed effective algorithm can provide the optimum block matching in a very short time, finding the best solution in less iteration and suitable for video tracking applications.

Keywords*: Optimization, Block matching algorithms, Cat swarm optimization, Average-Inertia Weighted CSO, parallel cat swarm optimization (PCSO), improvement structure of cat swarm optimization (ICSO)*

1. Introduction

The moving object tracking in video pictures has attracted a great deal of interest in computer vision, The aim of object tracking and detection is to establish a correspondence between objects or object parts in consecutive frames and to extract temporal information about objects such as trajectory, posture, speed and direction. Tracking detected objects frame by frame in video is a significant and difficult task [1].

Motion Estimation (ME) is an important part of video tracking system, since it can achieve significant compression by exploiting the temporal redundancy that commonly exists in a video sequence. Several ME methods have been studied seeking for a complexity reduction at

ISSN: 1738-9968 IJHIT

video coding such as block matching (BM) algorithms, parametric-based models [2], optical flow [3] and percussive techniques [4]. Among such methods, BM seems to be the most popular technique due to its effectiveness and simplicity for both software and hardware implementations [5]. In order to reduce the computational complexity in ME, many BM algorithms have been proposed and employed at implementations for several video compression standards such as MPEG-4 [6] and H.264 [7].

In BM algorithms, the video frames are partitioned into non overlapping blocks of pixels. Each block is predicted from a block of equal size in the previous frame. In particular, for each block at the current frame, the algorithm aims for the best matching block within a search window from the previous frame, while minimizing a certain matching metric such as sum of absolute differences (SAD), Mean Absolute Difference (MAD) and Mean Squared Error (MSE) which is given by equation (1).

$$MSE(i,j) = \frac{1}{MN}\sum_{n=0}^{N-1} \sum_{m=0}^{M-1}[Sc(I+n, J+m) - Sp(I+n+i, J+m+j)]^2 \qquad (1)$$

where *Sc(I+n, J+m,)* and *Sp (I+n+i, J+m+j)* are the pixels value in the current and previous frames, $M \times N$ is the size of block, *(I, J)* represents the coordinates of the upper left corner pixel of the current block and *(i,j)* is the displacement that is relative to current block located at *(I,J)*.The best matching block thus represents the predicted block, whose displacement from the previous block is represented by a transitional motion vector (MV) as seen in Figure 1.

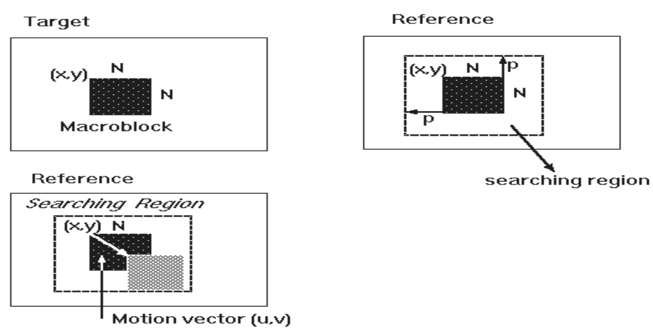

Figure 1. Block Matching Concept

Computational intelligence is a hot research topic and many related algorithms have been proposed in recent years, these algorithms generally were being proposed to solve the optimization problems. Some of these optimization algorithms were developed based on swarm intelligence by simulating the intelligent behavior of animals and The idea of computational intelligence may come from observing the behavior of creatures, like Ant Colony Optimization (ACO) which imitate the behavior of ants, Particle Swarm Optimization (PSO) which imitate the behavior of birds, and the recent finding, Cat Swarm Optimization (CSO) which imitate the behavior of cats. . The number of its successful applications is growing in clustering [8], networks [9-10], solving multi-objective problems [11], image edge enhancement [12], object tracking [13].

Chu, Tsai, and Pan studied the behavior of the cats and modeled their behavior to introduce a novel optimization algorithm [14-15]. Based on their studies they suggested that cats have two modes of behavior: seeking mode and tracing mode. They notice that cat spends most of the time when they are awake on resting. While they are resting, they move their position carefully and slowly. This mode of behavior is called seeking mode. In the tracing mode, a cat moves according to its own velocities for every dimension.

The performance of CSO algorithm was compared to that of different heuristic techniques. It is found that, the convergence speed of CSO is significantly better than that of DE [16], PSO [17], and evolutionary algorithms (EAs) [18-19].it is found that, CSO is the best performing algorithm as it finds the lowest fitness value for the most of the problems considered in that study.

So in this paper, a new modified CSO was proposed in order to order to improve the performance and achieve better convergence in less iteration. This proposed algorithm collected between two concepts that enhancement the core work of CSO. First concept of added a new parameter(inertia weighted) to the position equation as an inertia weight that extracted from Average-Inertia Weighted CSO algorithm (AICSO),therefore a new form of the velocity equation will be obtained .Second, the concept that based on cats' cooperation and competition that clear in a parallel cat swarm optimization PCSO algorithm.

Finally, a new algorithm that combined the two above concepts is proposed in this study in order to take the benefits of two algorithms in solve the optimization problem in block matching algorithm.

The overall paper is organized as follows: Section 2 summarizes studies that are related to the proposed algorithm. In Section 3 holds a brief description about the CSO algorithm. Section 4 provides backgrounds about CSO movement. Section 5 provides brief review to the parallel cat swarm optimization and Section 6 presents the idea of Average-Inertia Weighted CSO while Section 7 exposes the final BM algorithm as a combination of CSO. Section 8 demonstrates experimental results for the proposed approach over tested sequences and some conclusions are drawn in Section 9.

2. Related Work

Enhancement the cat swarm optimization is an important topic in soft computing and it has been studied for several decades. In this section some studies that related to proposed algorithm have been summarized below:

Yan Zhang and Yide Ma [20] present a variation on the standard CSO algorithm called a vibration mutation cat swarm optimization, or VMCSO in order to efficiently increase diversity of the swarm in the global searches. Comparing the new algorithm with CSO and several CSO main variants demonstrates the superiority of the VMCSO for the benchmark functions.

Yuanmei Wen and Yanyu Chen [21] apply support vector machine (SVM) model with modified parallel cat swarm optimization (MPCSO) to forecast next-day cooling load in district cooling system (DCS). By extracting the Eigen value of the input historical load data, principal component analysis (PCA) algorithm is used to reduce the complexity of the data sequence. Thus, the proposed model is effective and applicable to cooling load forecasting.

Maysam Orouskhani, Mohammad Mansouri, and Mohammad Teshnehlab [22] propose a new algorithm of CSO namely, Average-Inertia Weighted CSO (AICSO). For achieving this, they added a new parameter to the position update equation as an inertia weight and used a new form of the velocity update equation in the tracing mode of algorithm. Experimental results using Griewank, Rastrigin and Ackley functions demonstrate that the proposed algorithm has much better convergence than pure CSO.

Carlos E. Klein, Leandro dos S. Coelho, Viviana C. Mariani, and Piergiorgio Alotto [23] propose algorithm to tune the control parameters by Lévy flights (LCSO). Loney's solenoid benchmark problem is used to examine the effectiveness of the conventional CSO and the proposed LCSO algorithms.

Pei-Wei Tsai and Cheng-Wu Chen [24] study the concept of four swarm intelligence methods, including Bat Algorithm (BA), Evolved Bat Algorithm (EBA), Cat Swarm Optimization (CSO), and Parallel Cat Swarm Optimization (PCSO) are given in a comprehensive way. The objective of this review is to provide a brief introduction for new researchers to the swarm intelligence research field.

Pei-wei tsai, Jeng-Shyang Pan, Shyi-Ming Chen and Bin-Yih Liao [25] investigates a parallel structure of cat swarm optimization (CSO) calls it parallel cat swarm optimization (PCSO). In the experiments, compare particle swarm optimization (PSO) with CSO and PCSO can be done. The experimental results indicate that both CSO and PCSO perform well. Moreover, PCSO is an effective scheme to improve the convergent speed of cat swarm optimization in case the population size is small and the whole iteration is less.

Pei-wei tsai, Jeng-Shyang Pan, Shyi-Ming Chen and Bin-Yih Liao [26] present an enhanced parallel cat swarm optimization (EPCSO) method for solving numerical optimization problems. The parallel cat swarm optimization (PCSO) method is an optimization algorithm designed to solve numerical optimization problems under the conditions of a small population size and a few iteration numbers. The Taguchi method is widely used in the industry for optimizing the product and the process conditions. By adopting the Taguchi method into the tracing mode process of the PCSO method, they propose the EPCSO method with better accuracy and less computational time.

3. Cat Swarm Optimization (CSO)

Swarm Intelligence (SI) is a novel artificial intelligence approach inspired by the swarming behaviors of groups of organisms such as ants, termites, bees, birds, fishes in foraging and sharing the information with each other. SI focuses on the collective intelligence of a decentralized system consisting of a group of organisms interacting with each other and their environment. So, by means of their collective intelligence swarms are able to effectively use their environment and resources. SI is also a mechanism that enables individuals to overcome their cognitive limitations and solve problems which are difficult for individuals to resolve alone. Swarm intelligence algorithms are essentially stochastic search and optimization techniques and were developed by simulating the intelligent behavior of these organisms. These algorithms are known to be efficient, adaptive, robust, and produce near optimal solutions and utilize implicit parallelism approaches [14].

One of the more recent optimization algorithm based on swarm intelligence is the Cat Swarm Optimization (CSO) algorithm. The CSO algorithm was developed based on the common behavior of cats. It has been found that cats spend most of their time resting and observing their environment rather that running after things as this leads to excessive use of energy resources. To reflect these two important behavioral characteristics of the cats, the algorithm is divided into two sub-modes and CSO refers to these behavioral characteristics as ―seeking mode‖ and ―tracing mode‖, which represent two different procedures in the algorithm. Tracing mode models the behavior of the cats when running after a target while the seeking mode models the behavior of the cats when resting and observing their environment [15].

Furthermore, previous researches have shown that the CSO algorithm has a better performance in function minimization problems compared to the other similar optimization algorithms like Particle Swarm Optimization (PSO) and weighted-PSO [27].

Cat Swarm Optimization algorithm has two modes in order to solve the problems which are described below:

3.1. Seeking Mode: Resting and Observing

For modeling the behavior of cats in resting time and being-alert, we use the seeking mode. This mode is a time for thinking and deciding about next move. This mode has four main parameters which are mentioned as follow:

Seeking memory pool (**SMP**), seeking range of the selected dimension (**SRD**), counts of dimension to change (**CDC**) and self-position consideration (**SPC**) [26]. The process of seeking mode is describes as follow:

Step1: Make j copies of the present position of catk, where j = SMP. If the value of SPC is true, let j = (SMP-1), then retain the present position as one of the candidates.
Step2: For each copy, according to CDC, randomly plus or minus SRD percent the present values and replace the old ones.
Step3: Calculate the fitness values (FS) of all candidate points.
Step4: If all FS are not exactly equal, calculate the selecting probability of each candidate point by (2); otherwise set all the selecting probability of each candidate point is 1.
Step5: Randomly pick the point to move to from the candidate points, and replace the position of cat_k.

$$Pi = \frac{|\mathrm{SSEi} - \mathrm{SSEmax}|}{\mathrm{SSEmax} - \mathrm{SSEmin}} \qquad (2)$$

If the goal of the fitness function is to find the minimum solution, FSb = FSmax, otherwise FSb = FSmin

3.2. Tracing Mode: Running After a Target

Tracing mode is the second mode of algorithm. In this mode, cats desire to trace targets and foods. The process of tracing mode can be described as follow: [28]

Step1: Update the velocities for every dimension according to (3).

Step2: Check if the velocities are in the range of maximum velocity. In case the new velocity is over-range, it is set equal to the limit.

$$V_{k,d}= V_{k,d} +r_1c_1(X_{best}-X_{k,d}) \qquad (3)$$

Step 3: Update the position of cat k according to (4).

$$X_{k,d} = X_{k,d} + V_{k,d} \qquad (4)$$

$X_{best,d}$ is the position of the cat, who has the best fitness value, $X_{k,d}$ is the position of cat_k, c_1 is an acceleration coefficient for extending the velocity of the cat to move in the solution space and usually is equal to 2.05 and r_1 is a random value uniformly generated in the range of [0,1].

4. CSO Movement = Seeking Mode + Tracing Mode

When applying the CSO algorithm to solve optimization problems, the initial step is to make a decision on the number of individuals or cats to use. Each cat in the population has the following attributes:

a) A position made up of *M* dimensions;
b) Velocities for each dimension in the position;
c) A fitness value of the cat according to the fitness function; and
d) A flag to indicate whether the cat is in seeking mode or tracing mode.

The CSO algorithm keeps the best solution after each cycle and when the termination condition is satisfied, the final solution is the best position of one of the cats in the population. CSO has two sub-modes, namely seeking mode and tracing mode and the mixture ratio *MR* dictates the joining of seeking mode with tracing mode. To ensure that the cats spend most of their time resting and observing their environment, the *MR* is initialized with a small value. The CSO algorithm can be described in 6 steps as presented in [25]:

Step 1: Create *N* cats in the process.
Step 2: Randomly sprinkle the cats into the *M*-dimensional solution space and randomly give values, which are in-range of the maximum velocity, to the velocities of every cat. Then haphazardly pick number of cats and set them into tracing mode according to *MR*, and the others set into seeking mode.
Step 3: Evaluate the fitness value of each cat by applying the positions of cats into the fitness function, which represents the criteria of our goal, and keep the best cat into memory. Note that the position of the best cat *(xbest)* will be remembered because it represents the best solution so far.
Step 4: Move the cats according to their *flags*, if cat_k is in seeking mode, apply the cat to the seeking mode process, otherwise apply it to the tracing mode process.
Step 5: Re-pick number of cats and set them into tracing mode according to *MR*, then set the other cats into seeking mode.
Step 6: Check the termination condition, if satisfied, terminate the program, and otherwise repeat **Step 3** to **Step 5**.

The frame work of CSO is shown in Figure 2 below

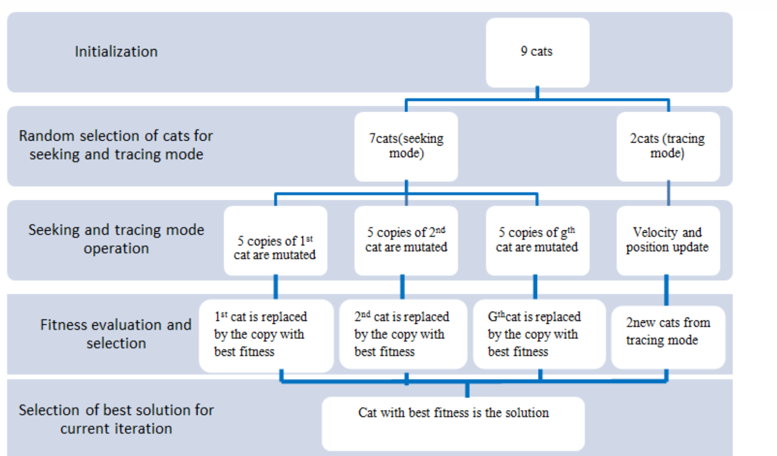

Figure 2. CSO Framework

5. Parallel Cat Swarm Optimization (PCSO)

Tsai *et al.* propose PCSO [25] in 2008. An information exchanging process is add into the original CSO. In addition, the tracing mode process is also modified in this algorithm. Precedents of parallelizing the virtual agents in evolutionary algorithms can be found in many articles. For example, Abramsn *et al.* propose the Parallel Genetic Algorithm (PGA) [29] in 1991, and Chang *et al.* present Parallel Particle Swarm Optimization (PPSO) [30] in 2005. The information exchanging process aims to share the isolated near best solution between different groups of virtual cats.

5.1. Parallel Tracing Mode Process

Since the virtual cats are divided into isolated groups, they can be treated as groups of small-scale CSO clusters. Agents in different clusters should only share their own near best solution. Thus, in the parallel tracing mode process, the velocity update step should follow equation (5) instead of equation (3):

$$V_{k,d}= V_{k,d} + r_1c_1(X_{lbest,d} - X_{k,d}),\ d=1,2,\ldots\ldots,M \qquad (5)$$

Where $X_{lbest,d}$ denotes the coordinates of the near best solution in one cluster.

5.2. Information Exchanging Process

In the information exchanging process, the near best solutions may have change to be copied into different clusters. A parameter called *ECH* is defined to trigger off the information exchanging process. Hence, in PCSO, the information exchanging process is involved every *ECH* iterations. This process can be described in 3 steps:

Step 1: Sort the virtual cats for every cluster by their fitness values.

Step 2: Randomly pick a near best solution from all clusters and replace the virtual cat, which has the worst fitness value in the cluster. But the near best solution and the virtual cat should not come from the same cluster.

Step 3: Repeat step 2 for all clusters

6. Average-Inertia Weighted Cat Swarm Optimization (AICSO)

In the pure CSO, a condition on the velocity equation should be put in order to control the velocities of the cats for every dimension and check whether the velocities are in the range of maximum or not.

For modifying this part, a parameter as an inertia weight to handle this problem will be used. Here the value of inertia weight (w) will be chosen randomly and experimental results indicate that it is better to choose w in the range of [0.4, 0.9].

So selecting the largest value for w in the first iteration (w = 0.9) and then it will be reduced to 0.4 in the next iterations.CSO with inertia weight can converge under certain conditions even without using v_{max}.

For w>1, velocities increase over time, causing cats to diverge eventually beyond the boundaries of the search space. For w<1, velocities decrease over time, eventually reaching 0, resulting convergence behavior. So the new position update equation can be written as

$$V_{k,d}= WV_{k,d} + r_1c_1(X_{best}-X_{k,d}) \qquad (6)$$

Where $c_1$is acceleration coefficient and usually is equal to 2.05 and $r_1$is a random value uniformly generated in the range of [0, 1] and w is inertia weight (ICSO).

Next step, a new form of the position update equation composing two terms will be used. In the first term, the average information of current and previous position and in the second, the average of current and previous velocity information will be used(AICSO). So new position equation is described below: [22]

.

$$X_{i+1} = \frac{X_{i+1}+X_i}{2} + \frac{V_{i+1}+V_i}{2} \qquad (7)$$

7. Proposed Algorithm

The processing of block matching is looking for the best position within the search window, in which a point of the minimum of MSE needs to be found. In order to reaching a better MSE, the more positions within the search window will be matched; however, the more computation times will be spent on searching. A better matching algorithm should spend less computation time on searching and obtain the better position. In this paper, the aim of the application of the ICSO algorithm to ME is to accelerate matching search, obtain higher accuracy, faster computation speed and reach a better ME.

Strong curiosity to moving objects and the outstanding skill of hunting are the two distinctive features of a cat. These two behavioral traits of cats are modeled by CSO: seeking mode and tracing mode, which reflects the cooperation between “cats”. However, in order to further improve the CSO optimization speed and prediction accuracy, ICSO absorbs the advantage of parallel computing to improve the tracing mode such that a parallel tracing mode is adopted. ICSO establishes a plurality of CSO to search the best parameters in the prediction the next block independently and simultaneously by dividing the “cat swarm” into some groups. At the same time, it adds information exchanging mode such that the CSOs can exchange information occasionally, which reflects the cooperation between groups. The information exchanging process aims to share the isolated near best solution between different groups of virtual cats. Hence, ICSO is particularly suitable for optimization problems, because it makes full use of computer resources and obtains the optimal result quickly. When ICSO is running, the “cats” are randomly distributed in the prediction search space. Inevitability, it results in a state such that there more “cats” in some areas and less in others. But sometimes in some cases pure CSO takes a long time to find an acceptable solution. So it affects on performance and convergence of the algorithm. Therefore high speed processor is needed for getting reasonable result.

In this study, proposed a new algorithm (ICSO) in order to improve the performance and achieve better convergence in less iteration. By adding a new parameter to the position equation as inertia weight that will be chosen randomly, then by making a new form of the velocity equation to improve searching ability in the vicinity of the best cats. By using this parameter, a balance between global and local search ability can be made. A large inertia weight facilitates a global search while a small inertia weigh facilitates a local search. First a large value will be used and it will be reduced gradually to the least value. So the maximum inertia weight happens in the first dimension of the each iteration and it will be updated decreasingly in every dimension, the velocity update equation for each cat to a new form can be changed. The block matching algorithm based on ICSO for ME is summarized as follows.

Step1: A population of N cats is generated with random positions within the searching window in the previous frame, the search area called search window which is usually a region centered on the current block position; and then random velocities are assigned to each cat.

Step 2: pick number of cats and set them into tracing mode according to *MR*, and the others set into seeking mode.

Step 3: The fitness of each cat is then evaluated according to the objective function. In the processing of block matching, the MSE as the (matching criterion) will be chosen. In the ICSO algorithm for ME, evaluating the fitness of each cat is calculating the block’s MSE.

Step 4: Evaluate the fitness value of each cat by applying the positions of cats into the fitness function, which represents the criteria of our goal, and keep the best cat into memory.

Step 5: Pick up a group of cats sequentially and sort the cats in this group according to their fitness values.
Step 6: : Move the cats according to their *flags*, if cat_k is in seeking mode, apply the cat to the seeking mode process, otherwise apply it to the tracing mode process.
Step 7: Each time, choose the inertia weight (w) randomly in range of [0.4, 0.9] in order to controlling excessive roaming of cats outside the searching window.
Step 8: Pick the near best solution from the neighbor group and replace the virtual cat, which has the worst fitness value in the group. But the near best solution and the virtual cat should not come from the same group.
Step 9: Repeat step 8 for all groups.
Step 10: Each time in the parallel tracing mode process, the velocity update step as declare in equation (8) instead of equations 3, 5, 6:

$$V_{k,d}= V_{k,d} +r_1c_1(X_{lbest,d} - X_{k,d}),\ d=1,2,\ldots\ldots,M \qquad (8)$$

Where $X_{lbest,d}$ denotes the coordinates of the near best solution in one cluster.

Step 11: use a new form of the position update equation composing two terms. In the first term, the average information of current and previous position and in the second, the average of current and previous velocity information will be used. So new position equation is described below:

$$X_{i+1} = \frac{X_{i+1} + X_i}{2} + \frac{V_{i+1} + V_i}{2}$$

Step 12: Termination criteria. If the number of iteration equals to the maximum (I_{max}), or MSE of the block less than a given small number ε, then iteration terminate; Otherwise go back to step 3.

The flowcharts of proposed algorithm, Information Exchange Mode and Parallel tracing mode are depicted in Figures 3-5 below:

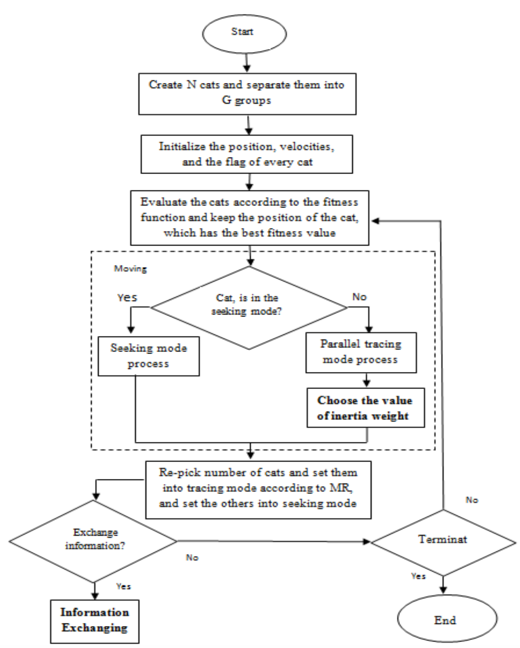

Figure 3. The Flowchart of ICSO Algorithm

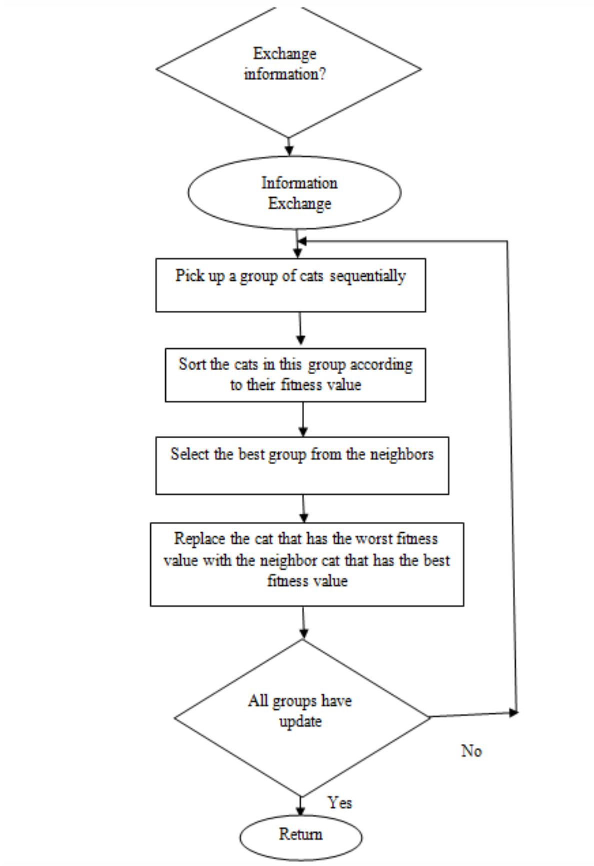

Figure 4. Information Exchange Mode

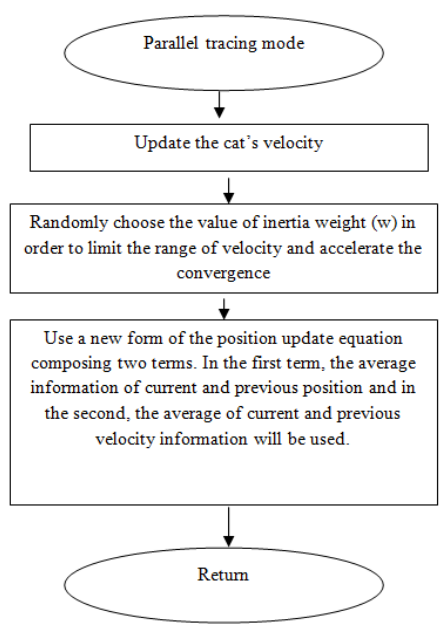

Figure 5. Parallel Tracing Mode

8. Simulation Results

The processing of block matching is looking for the best position within the search window, in which point of the minimum of MSE needs to be found. In order to reaching a better MSE, the more positions within the search window will be matched; however, the more computation times will be spent on searching. A better matching algorithm should spend less computation time on searching and obtain the better position. In this paper, the aim of the application of the ICSO algorithm to ME is to accelerate matching search and reach a better ME.

To illustrate the performance and feasibility of proposed algorithm, an example of video sequence (AVI 25 frame/second 720x576) can be considered (see Fig.6). Fig.7 gives the selected motion object which ME was computed based on ICSO by Matlab software (Block size: 16x16; Search window size: 30). Also the compare computation times per block and MSE per pixel of the block matching algorithm based on ICSO with PSO for the same video sequence have been implemented. The results are proposed in Table1. The table shows that the block matching algorithm based on ICSO is a fast and efficient algorithm.

Figure 6. Tested Sequence

Figure 7. Motion Object Selected

Table 1. Computation Times per Pixels and the MSE per Pixel

Algorithms	PSO	ICSO
Computations	13.94	12.72
MSE	11.92	10.92

The computational complexity of this ME approach depends on the number of fitness function evaluations performed. This is directly related to the population size M and the maximum number of total iteration N_{max} allowed. Theoretically, a maximum of M xN_{max} cost function evaluations will be required for each MB. The value of M is at least equal to 10 and N_{max} should be large enough to guarantee good estimation accuracy. Nevertheless, because this approach exploits the spatial and temporal correlations of the motion vectors, continuously refines the motion search process through cat mutation, and allows an early termination condition for the MBs, it is estimated that the CSO algorithm will converge to the global optimum before N_{max} is reached. Moreover, In this paper ,Cats possess the following characteristics:

(1) Scalability: The cats can change their action by local and distributed agent interactions. This is an important characteristic by which the group is scaled to the desired level, this characteristic is clear when each cat exchange its information with the neighbors.

(2) Fault tolerance: Each cat follows a simple rule. They do not rely on a centralized control mechanism, graceful, scalable degradation.

(3) Adaptation: cats always search for new macro block by roaming around their neighbors. Once they find the goal their members follow the accuracy macro block. While cats follow accuracy macro block, some of the copies of the same cat always search for another accuracy macro block.

(4) Speed: In order to make other cats to know the target, they move faster to their target by apply the modify trace mode and change the value of inertia weight in order to accelerate the optimum macro block while the other cats appear in seek mode in order to find the best cat in each group that has the best fitness value.

(5) Modularity: The cats do not interact directly and act independently to accomplish the task.

(6) Autonomy: No centralized control and hence no supervisor is needed. They work independently and always strive to search the optimum macro block within in the search window.

(7) Parallelism: Cats work independently and the task of searching macro block is carried out by each cat in parallelism. It is parallelism due to which they change their best fitness cat, this effective scheme to improve the convergent speed of cat swarm optimization in case the population size is small and the whole iteration is less.

9. Conclusion

Block-matching algorithm is very popular for video coding and the motion estimation method has a critical impact on the efficiency of block-matching algorithm. Thus, in this paper, a novel adaptive block-matching algorithm based on improvement Cat Swarm Optimization (ICSO) is proposed to reduce the number of search locations in the BM process without the degradation of the image quality. The proposed algorithm can obtain higher accuracy and faster computation speed in block matching .Since the proposed algorithm does not consider any fixed search pattern or any other movement assumption, a high probability for finding the true minimum (accurate motion vector) is expected regardless of the movement complexity contained in the sequence, yet the CSO approach is capable of achieving high accuracy in block matching.

By summarizing proposed improvement algorithm, it is clear that with proper design, the improvement structure can assist the original swarm intelligence algorithm to improve its accuracy. Taking consideration to combine more than two ideas of modified cat swarm intelligence algorithms may be another way to construct new improvement algorithm. However, when the number of the combined algorithms in increased, it may be better that only parts of the actions are taken from different algorithms. Otherwise, the computational complexity would also be increased as well and improves the performance on finding the best global solution and achieves the better accuracy level of convergence in the less iteration.

However, in order to further improve the CSO optimization speed and prediction accuracy, ICSO absorbs the advantage of parallel computing to improve the tracing mode such that a parallel tracing mode is adopted. ICSO establishes a plurality of CSO to search the best parameters in the prediction the next block independently and simultaneously by dividing the “cat swarm” into some groups. At the same time, it adds information exchanging mode such that the CSOs can exchange information occasionally, which reflects the cooperation between groups. The information exchanging process aims to share the isolated near best solution between different groups of virtual cats. Hence, ICSO is particularly suitable for optimization problems, because it makes full use of computer resources and obtains the optimal result quickly. When ICSO is running, the “cats” are randomly distributed in the prediction search space. Inevitability, it results in a state such that there more “cats” in some areas and less in others. But sometimes in some cases pure CSO takes a long time to find an acceptable solution. So it affects on performance and convergence of the algorithm. Therefore high speed processor is needed for getting reasonable result.

In order to limit of velocity range, an adaptive inertia weight to the velocity equation which is updated in each dimension will be added. By using this parameter, a balance between global and local search ability can be made. A large inertia weight facilitates a global search while a small inertia weigh facilitates a local search. First a large value will be used and it will be reduced gradually to the least value. So the maximum inertia weight happens in the first dimension of the each iteration and it will be updated decreasingly in every dimension, the velocity update equation for each cat to a new form can be changed.

Experimental results demonstrate the high performance of the proposed method in terms of computational complexity, finding the global best solution, faster convergence and estimation accuracy.

This work can be further extended by using dynamic search window adjustment in order to reduce the computational complexity and take the advantages of Weight Changes for Learning Mechanisms in Two-Term Back-Propagation Network in order to select the suitable value of weight to reach the optimum value of (w) parameter.

References

[1] W. Qu, F. I. Bashir, D. Graupe, A. Khokhar and D. Schonfeld, “A Motion Trajectory Based Video Retrieval System Using Parallel Adaptive Self Organizing Maps”, IEEE international conference, (**2005),** pp. 1800-1805.

[2] D. Tzovaras, I. Kompatsiaris and M. G. Strintzis, “3D object articulation and motion estimation in model-based stereoscopic videoconference image sequence analysis and coding”, Signal Processing: Image Communication, vol. 14, no. 10, (**1999**), pp. 817–840.

[3] J. L. Barron, D. J. Fleet and S. S. Beauchemin, “Performance of optical flow techniques”, International Journal of Computer Vision, vol. 12, no. 1, (**1994**), pp. 43–77.

[4] J. Skowronski, “Pel recursive motion estimation and compensation in sub bands”, Signal Processing: Image Communication, vol. 14, (**1999**), pp. 389–396.

[5] T. Huang, C. Chen, C. Tsai, C. Shen and L. Chen, "Survey on block matching motion estimation algorithms and architectures with new results", Journal of VLSI Signal Processing, vol. 42, (**2006**), pp. 297–320.
[6] MPEG4, Information Technology Coding of Audio Visual Objects Part 2: Visual, JTC1/SC29/WG11, ISO/IEC14469-2 (MPEG-4 Visual), (**2000**).
[7] H.264, Joint Video Team (JVT) of ITU-T and ISO/IEC JTC1, Geneva, JVT of ISO/IEC MPEG and ITU-T VCEG, JVT-g050r1, Draft ITU-TRec and Final Draft International Standard of Joint Video Specification (ITU-T Rec.H.264-ISO/IEC14496-10AVC), (**2003**).
[8] B. Santosa and M. K. Ningrum, "Cat swarm optimization for clustering," in Proc. the International Conference on Soft Computing and Pattern Recognition, pp. 54-59, (**2009**).
[9] J. Paul and T. Yusiong, "Optimizing Artificial Neural Networks using Cat Swarm Optimization Algorithm," I. J. Intelligent Systems and Applications, vol. 1, (**2013**), pp. 69-80.
[10] S. Chittineni, K. Abhilash, V. Mounica, N. Sharada, and S. Satapathy, "Cat swarm optimization based neural network and particle swarm optimization based neural network in stock rates prediction," in Proc. the 3rd International Conferences on Machine Learning and Computing, (**2011**), pp. 292-296.
[11] P. P. Mohan and P. Ganapati, "Solving multi objective problems using cat swarm optimization," Expert Systems with Applications, vol. 39, no. 3, (**2012**), pp. 2956-2964.
[12] T. R. Benala and S. C. Satapathy, "Cat swarm optimisation for optimizing hybridized smoothening filter in image edge enhancement," in Proc. the International Conferences on Systemic, Cybernetics and Informatics, (**2010**), pp. 247-252.
[13] I. Hadi and M. Sabah, "An Enhanced Video Tracking Technique Based on Nature Inspired Algorithm", International Journal of Digital Content Technology and its Applications (JDCTA), vol. 8, no. 3, (**2014**) June, pp. 32-42.
[14] S.-C. Chu, P.-W. Tsai, and J.-S. Pan, "Cat swarm optimization," in PRICAI 2006: Trends in Artificial Intelligence. Springer, (**2006**), pp. 854–858.
[15] S.-C. Chu and P.-W. Tsai, "Computational intelligence based on the behavior of cats," International Journal of Innovative Computing, Information and Control, vol. 3, no. 1, (**2007**), pp. 163–173.
[16] K. Vaisakh and P. K. Rao, "Differential Evolution based optimal reactive power dispatch for voltage stability enhancement," Journal of Theoretical and Applied Information Technology, (**2008**), pp. 700-709.
[17] S. Granville, "Optimal Reactive Power Dispatch through Interior Point Methods", IEEE Trans. On Power Systems, vol. 9, no. 1, (**1994**), pp. 98-105.
[18] T. Jayabarathi, K. Jayaprakash, D. N. Jeyakumar and T. Raghunathan, "Evolutionary Programming Techniques for Different Kinds of Economic Dispatch Problems", Electric Power Systems Research, vol. 73, (**2005**), pp. 169-176.
[19] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces" TR-95-012, (**1995**) March.
[20] Y. Zhang and Y. Ma, "Cat Swarm Optimization with a Vibrational Mutation Strategy", International Journal of Machine Learning and Computing, vol. 4, no. 6, (**2014**) December, pp. 510-514.
[21] Y. Wen and Y. Chen, "Modified Parallel Cat Swarm Optimization in SVM Modeling for Short-term Cooling Load Forecasting", Journal of Software, vol. 9, no. 8, (**2014**) August, pp. 2093-2104.
[22] M. Orouskhani, M. Mansouri and M. Teshnehlab, "Average-Inertia Weighted Cat Swarm Optimization", ICSI 2011, Part I, LNCS 6728, (**2011**), pp. 321–328.
[23] C. E. Klein, L. dos S. Coelho, V. C. Mariani and P. Alotto", Electromagnetic Optimization by Lévy Cat Swarm Optimization Approach", CEFC2014 - The Sixteenth Biennial IEEE Conference on Electromagnetic Field Computation, (**2014**), May.
[24] P.-W. Tsai and C.-W. Chen, "Review on Swarm Intelligence for Optimization", Computing Science and Technology International Journal, vol. 2, no. 1, (**2014**) April, pp. 13-17.
[25] P.-W. Tsai, J.-S. Pan, S.-M. Chen, B.-Y. Liao and S.-P. Hao, "Parallel Cat Swarm Optimization", In Proceedings of the 7th International Conference on Machine Learning and Cybernetics, (**2008**), pp. 3328-3333.
[26] P.-w. tsai, J.-S. Pan, S.-M. Chen and B.-Y. Liao, "Enhanced parallel cat swarm optimization based on the Taguchi method", vol. 39, no. 7, (**2012**) June 1, pp. 6309–6319.
[27] M. A. Khanesar, M. Teshnehlab, and M. A. Shoorehdeli, "A novel binary particle swarm optimization", in Control & Automation, 2007. MED'07, Mediterranean Conference on, IEEE, (**2007**), pp. 1–6.
[28] J. P. T. Yusiong, "Optimizing Artificial Neural Network Using Cat Swarm Optimization Algorithm", International Journal of Intelligent Systems and Applications, vol. 5, no. 1, (**2012**), pp. 69-80.
[29] D. Abramson, and J. Abela, "A Parallel Genetic Algorithm for Solving the School Timetabling Problem", Technical Report, Division of Information Technology, CSIRO, (**1991**).
[30] J.-F. Chang, S.-C. Chu, J. F. Roddick and J.-S. Pan, "A Parallel Particle Swarm Optimization Algorithm with Communication Strategies", Journal of Information Science and Engineering, vol. 21, no. 4, (**2005**), pp. 809-818.

Authors

Israa Hadi, She received her Ph.D. degree from University of Babylon. She is currently a Professor in Software Department, College of Information Technology, Babylon University, Iraq.. She is primary research interests cover video tracking, data mining and information hiding.

Mustafa Sabah, He is currently a Ph.D. student of College of Information Technology University of Babylon. His primary research interests cover video tracking and swarm optimization.