

## Research Article

# Video Superresolution via Parameter-Optimized Particle Swarm Optimization

Yunyi Yan, Yujie He, Yingying Hu, and Baolong Guo

*School of Aerospace Science and Technology, Xidian University, Xi'an 710071, China*

Correspondence should be addressed to Yunyi Yan; [yyyan@xidian.edu.cn](mailto:yyyan@xidian.edu.cn)

Received 19 April 2014; Revised 15 July 2014; Accepted 7 August 2014; Published 28 August 2014

Academic Editor: Antonio Ruiz-Cortes

Copyright © 2014 Yunyi Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Video superresolution (VSR) aims to reconstruct a high-resolution video sequence from a low-resolution sequence. We propose a novel particle swarm optimization algorithm named as parameter-optimized multiple swarms PSO (POMS-PSO). We assessed the optimization performance of POMS-PSO by four standard benchmark functions. To reconstruct high-resolution video, we build an imaging degradation model. In view of optimization, VSR is converted to an optimization computation problem. And we take POMS-PSO as an optimization method to solve the VSR problem, which overcomes the poor effect, low accuracy, and large calculation cost in other VSR algorithms. The proposed VSR method does not require exact movement estimation and does not need the computation of movement vectors. In terms of peak signal-to-noise ratio (PSNR), sharpness, and entropy, the proposed VSR method based POMS-PSO showed better objective performance. Besides objective standard, experimental results also proved the proposed method could reconstruct high-resolution video sequence with better subjective quality.

## 1. Introduction

Extended and expanded from the definition of still image superresolution reconstruction proposed by Harris [1] and Goodman [2] in 1960s, video superresolution (VSR) was investigated quite intensively in recent years [3–7]. The two common types resolution algorithms to VSR are based on multiframe complementary information [8] and based on motion estimation [9, 10]. The former utilizes the redundancy information of different frames to reconstruct a high-resolution video sequence. So it is noneffective when object moves fast or no redundancy information is available. On the other hand, the latter describes movement rule of moving objects by motion vector, whose performance depends on the accuracy in motion estimation. Unfortunately, the accuracy is not very high and even very low in some cases, and yet the process is highly calculation-costed [11].

Particle swarm optimization (PSO) is an established method for optimization with the advantage of simplicity of implementation [12, 13] and it has been applied in many fields including scientific research and engineering problems [12, 14–18]. Compared to other heuristic methods, like genetic

algorithm (GA) and ant colony optimization (ACO), PSO is proven to have higher computational accuracy [19] and lower time cost [20] in many complicated applications.

However, the parameter selection will affect the performance deeply [21], but there are no strategies proven to work well for all problems [21–24]. In many cases, we have to adjust the parameters in PSO several times to obtain satisfactory precision. The accession of subswarms with different evolutionary strategies ensured the quality both intensively and extensively [25–27]. In this paper, we introduced one swarm to determine those configuration parameters of PSO and proposed the parameter-optimized multiple swarms PSO (POMS-PSO) which could avoid the limitation of fixed parameters and achieve dynamical adjustment of parameters throughout the whole optimized process.

The goal of video superresolution reconstruction (VSR) is to obtain a video with higher resolution that meets certain required specifications, for example, sharpness, succession, details, and so forth. Hence, in view of optimization, VSR can be thought as an optimal procedure, which looks for the optimal result satisfying objective and subjective quality

requirements. If imaging degeneration model built, optimization technique can be used to search for the global minimum points according to proper fitness function.

In this paper, we propose a novel video superresolution reconstruction (VSR) scheme which is based on image degeneration model solution via POMS-PSO. The paper is organized as follows. In Section 2, the 3D video degeneration model and its solution are introduced. In Section 3, the parameter-optimized multiple swarms PSO and its performance are presented. In Section 4, we implement the video reconstruction using POMS-PSO algorithm and evaluate the result by several criteria, and finally the conclusion of this paper is given in Section 5.

## 2. Video Superresolution Model

**2.1. 3D Video Degeneration Model.** Space-time dynamic scene can be presented by coordinate system like  $(x, y, z, t)$ . However in most real imaging cases, space-time dynamic scene can be presented by three-coordinate system  $(x, y, t)$  under some suitable conditions, such that (1) the scene is at one identical plane and the movement takes place within this plane; (2) the distance between various cameras is much smaller than the distance between the camera and the scene, which is to say, there is almost no parallax between two cameras. In this situation, arrange all the elements into a vector  $\vec{L}$  according to the lexicographical order  $(x-y-t)$ . In order to simplify calculations, only gray value is used to represent the characterization of pixels in images.

The value on  $p^l$  in a low-resolution sequence  $S^l$  can be deemed as a mapping of corresponding value on  $p^h$  in a high-resolution sequence  $S$  as follows:

$$S^l(p^l) = \iiint_{p=(x,y,t) \in \text{sup}(B^h)} S(p) B^h(p - p^h) dp + n, \quad (1)$$

where  $S^l$  is a low-resolution sequence,  $p^l = (x^l, y^l, t^l)$  is a point in the low-resolution sequence,  $p^h = (x^h, y^h, t^h)$  is the mapping of corresponding value on  $p^l$  in the high-resolution sequence, and  $B^h$  is the space-time position-dependent blur kernel function in high-resolution coordinates. It consists of spatial domain point spread effect, which can be simulated by kind of Gaussian function model, and time domain fuzzy caused by integration effect by the reason of exposure time during imaging;  $n$  is white noise which is subjected to Gaussian distribution.

Obviously, the range of support set of the space-time blur kernel function determines the space-time resolution relationship between the high-resolution sequence and low-resolution sequence. It determines whether the reconstruction occurs in time domain and/or spatial domain(s) and amplification factor used during reconstruction.

Equation (1) is defined in continuous space which presents the map relationship between high-resolution scenes and the low-resolution ones in real world. But in digital processing system, like computers or digital signal processors (DSPs), all data need to be stored and processed in digital

format. This means we have to transfer the continuous space to discrete space.

The unknown continuous scene  $S$  can be thought as the unknown high-resolution sequence  $S^h$ , and the relationship between low-resolution and high-resolution sequence can be presented by matrix computing. Hence, we obtain the discrete form of (1) as follows, one giant system composed of linear equations carrying the relation between high- and low-resolution sequence,

$$\vec{L} = M\vec{H} + \vec{N}, \quad (2)$$

where  $\vec{L}$  is a vector composed of all the elements in the low-resolution sequence arranging into lexicographical order,  $\vec{H}$  is a vector composed of all the elements in the high-resolution sequence arranging into lexicographical order,  $\vec{N}$  is a noise vector, and  $M$  is a sparse Toeplitz matrix corresponding to the space-time blur kernel function.

**2.2. Model Solution.** Like other ill-posed problems, in super-resolution reconstruction, the known elements in low-resolution sequence are much less than the unknown elements in high-resolution sequence. To solve this kind of problems, extra proper criteria or constraints should be set reasonably. In this paper, a visual criteria were taken as the object optimization function, and the solution of the VSR problem can be shown as follows:

$$\vec{H}^* = \arg \min f(\vec{H}), \quad (3)$$

where  $\vec{H}^*$  is a vector composed of all the elements in rebuilt high-resolution sequence arranging into lexicographical order and  $f(\vec{H})$  is the object optimization function known as the fitness function in  $r/KPSO$  with visual criteria taken into consideration, and it is shown as below:

$$f(\vec{H}) = \|M\vec{H} - \vec{L}\|^2 + \text{Sm}(\vec{H})_{x,y,t} + \text{Dt}(\vec{H})_{x,y}. \quad (4)$$

$\|M\vec{H} - \vec{L}\|^2$  reflects fidelity, and  $\text{Sm}(\vec{H})_{x,y,t}$  reflects smoothness in directions of  $x-y-t$ , and more specifically, it is defined by the following equation:

$$\begin{aligned} \text{Sm}(\vec{H})_{x,y,t} &= \lambda_x \|W_x G_x(\vec{H})\|^2 + \lambda_y \|W_y G_y(\vec{H})\|^2 \\ &+ \lambda_t \|W_t G_t(\vec{H})\|^2. \end{aligned} \quad (5)$$

$\lambda_i (i = x, y, t)$  is the image smooth coefficients in  $i$  direction,  $W_i (i = x, y, t)$  is the weight matrix in  $i$  direction, and  $G_i (i = x, y, t)$  is the second-order differential operator in  $i$  direction, and  $\text{Dt}(\vec{H})_{x,y}$  reflects details of the retention, which can be computed as follows:

$$\text{Dt}(\vec{H})_{x,y} = \left( \|D_x(\vec{H})\|^2 + \|D_y(\vec{H})\|^2 + \varepsilon \right)^{-1}. \quad (6)$$

$D_i(\vec{H}) (i = x, y)$  is the gradient operator in  $i$  direction;  $\varepsilon$  is a small positive constant.

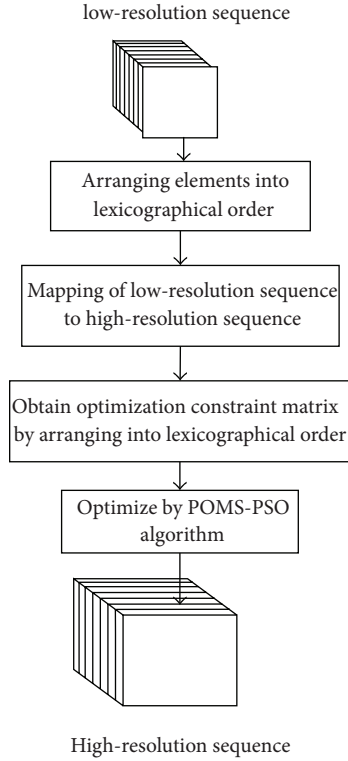


FIGURE 1: Schematic diagram of a reconstruction procedure.

The schematic diagram of the reconstruction procedure in this paper is given in Figure 1. The input low definition video is transformed into a sequence, and then all elements are arranged in lexicographical order to compose a matrix. Based on the video degeneration model, there is a map between the low-resolution sequence and the high-resolution sequence.

In Figure 1, POMS-PSO is used to search the best reconstructed results which is represented as  $\vec{H}^*$  in (3). In fact, we code the reconstructed high definition video as a particle of subswarm in the inner loop of Figure 2. And the  $f(\vec{H})$  in (4) is used as the fitness function during the optimization process. The best solution to (4) will be thought as reconstructed video. The utilization of POMS-PSO algorithm changes the reconstruction procedure into a problem that search a solution satisfying the video degeneration process in best.

### 3. Parameter-Optimized Multiple Swarms PSO (POMS-PSO)

Particle swarm optimization (PSO) has surprising ability of handle optimization problems with multiple local optima and its simplicity of implementation. We combined the concepts of multiswarm and parameter autooptimizing, and then we proposed the improvement of multiswarm PSO named as parameter-optimized PSO.

In fact, many improved PSO algorithms [15, 28–30] have been given with fixed parameters [22, 31]. The parameter selection can affect the performance of PSO [21], but there

are no strategies proven to work well for all problems [21–24]. In many cases, when facing real engineering problems, we have to adjust the parameters in PSO several times to obtain satisfactory precision. The concept of parameter-optimizing is to determine those free parameters of the subswarms via PSO program. In this paper, the concept of multi-subswarm is based on an ecological approach [25, 26], in which all the particles are divided into two different subswarms depending on their fitness. By adjusting parameters dynamically throughout the whole optimized process, the method can avoid the limitation of fixed parameters. At the same time, those parameters can automatically change when problem to be solved is changed. By introducing different evolutionary strategies using on different subswarms [27], the new framework ensures quality both intensively and extensively. All the advantages above over current methods result in better optimization performance than parameter-fixed PSO methods [32, 33].

**3.1. Algorithm Introduction.** Initialization of the particles which are used to search for the global optimum is spanned as follows:

$$\begin{pmatrix} x_{11} & \cdots & x_{1n} & v_{11} & \cdots & v_{1n} \\ x_{21} & \cdots & x_{2n} & v_{21} & \cdots & v_{2n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{Nn} & v_{N1} & \cdots & v_{Nn} \end{pmatrix}, \quad (7)$$

where  $N$  and  $n$  are the size and the dimension of particle swarm, respectively. And  $\{x_{ij}, i = 1, 2, \dots, N; j = 1, 2, \dots, n\}$  refers to the  $i$ th particle with position  $j$ , and its corresponding speed is  $v_{ij}$ . General speaking, one particle's iteration and update of PSO algorithm can be presented by the following two equations:

$$\begin{aligned} v_{id}^{n+1} &= \omega v_{id}^n + c_1 r_1 (p_{id}^n - x_{id}^n) + c_2 r_2 (p_g^n - x_{id}^n), \\ x_{id}^{n+1} &= x_{id}^n + v_{id}^{n+1}, \end{aligned} \quad (8)$$

where  $d$  is dimensions index of particle's position vector;  $i$  is particle index;  $n$  notes for iteration number;  $c_1$  and  $c_2$  are positive constants called *cognitive* and *social* factor, respectively;  $\omega$  is called *inertia weight*;  $r_1$  and  $r_2$  are random numbers;  $p_i$  is the best position visited by this particle; and  $p_g$  is the global best position over the whole swarm. More details on PSO model and its implementation can be found in [28].

We proposed a novel optimization method with a feature that the parameters in PSO can be determined and optimized by its self, which was named as POMS-PSO. To accomplish this goal, we introduced two layer loops into the iteration process. In outer loop, we placed one swarm to optimize the parameter setting for the two subswarms in inner loop. And the inner loop is used to find the optimal resolution to given problem. The proposed method is named as POMS-PSO, which comes from parameter-optimized multiple swarms PSO. The schematic diagram of PMOS-PSO is shown in Figure 2.

In the view of particle swarm optimization, a particle means a potential solution to problem. The PSO algorithm

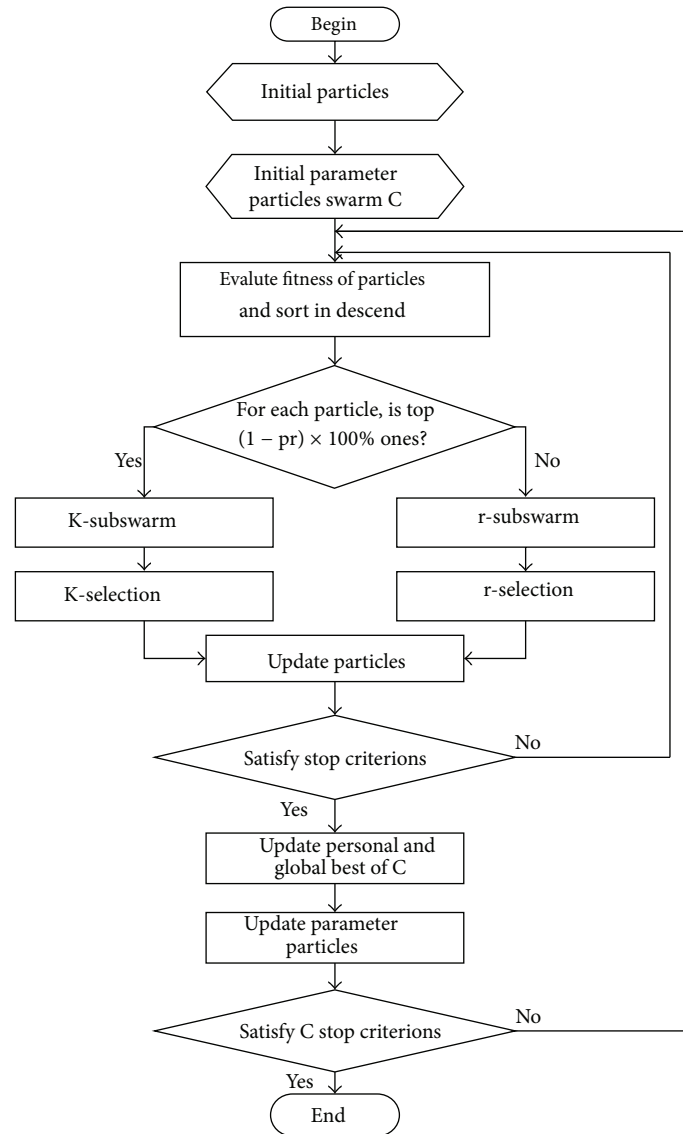


FIGURE 2: Schematic diagram of POMS-PSO.

is inspired by the bird or fish swarms' seeking food procedure in nature. In PSO algorithm, a bird is modelled as a particle, and a bird position is presented by a vector. In Figure 2, POMS-PSO employs double layer loops and three swarms. The swarm in outer loop is named as C-swarm and the particles in C-swam is called C-particle which is based on the fact that the C-particle's valued is used to configure the two subswarms' parameter selection (in inner loop) and the inner loop's computing is performed with a certain C-particle's configuration to find the best solution to given problem. In the proposed POMS-PSO, a C-particle means a parameter setting for the inner loop and a particle in inner loop means a solution to given problem.

As shown in Figure 2, the first step of optimization initializes the particles with random numbers normally distributed in the search space. The C-swarm's search space is different from the r- and K-subswarm. For example, the

search space for the four benchmark functions in Section 3.2 is  $[-100, 100]^D$ ,  $D = 30$ , but the C-swarm's is recommended to set in  $[0, 10]^D$ ,  $D = 11$  due to empirical information. In the step of fitness evaluation, all particles in inner loop are evaluated by given fitness function and are sorted in descend. The top  $(1 - P_r) \times 100\%$  particles are allocated to K-subswarm and the other to r-subswarm. For example, we totally have 20 particles in inner loop, and  $P_r$  is set to 0.1, then 18 out of 20 particles belong to K-subswarm, and 2 out of 20 belong to r-subswarm. The r- and K-subswarms will perform r- and K-selection strategies, respectively, and update their positions.

In most of all optimization based on iteration, the stop criterion can be defined as the iteration could satisfy one of the following two conditions. (1) The maximum generation is reached and (2) the error or the fitness value is good enough to satisfy our requirement. To our experimental experience, when POMS-PSO is used to solve complicated problems

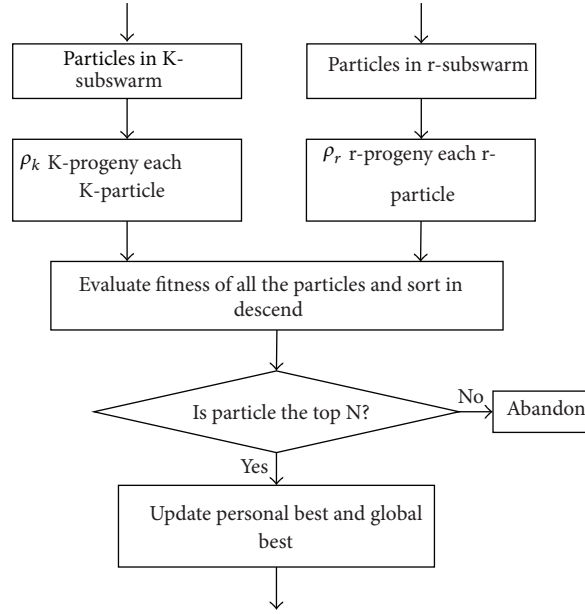


FIGURE 3: Schematic diagram of K-selection and r-selection.

(e.g., VSR), the error threshold is difficult to estimate and we often just set it to zero. In this case, the inner loop is often stopped by the maximum iteration generation reached unless our luck is good enough to get a solution with zero error.

For every C-particle, all of its various positions mean various configuration of inner loop. The inner loop deploys two subswarms, the r-subswarm and the K-subswarm. In ecology, r-selection is termed for those species that breed many offspring and live in unstable environments and K-selection for those species that produce few offspring and live in stable environments. The r-selection can be characterized as quantitative, little parent care, large growth rate, and rapid development and K-selection can be characterized as qualitative, much parent care, small growth rate, and slow development. K-selection is performed for those particles with high fitness, and so K-subswarm only can produce relatively fewer progenies but the progenies are nurtured delicately with much parent care. On the other hand, r-selection is performed for those particles with relatively lower fitness. Due to little parent care, r-subswarm can produce a large number of progenies; the progenies have to compete with other members in r-subswarm for survival and only the best ones can survive. The main task of r-subswarm is to explore the search space as possible as they can, and K-subswarm should try to keep the current optimum solutions and exploit the space as they can.

In fact, (8) are the core procedure in most of all PSO algorithms. The r-selection and K-selection are defined in ecology to describe various species' idiosyncrasy; however, in our POMS-PSO algorithm, the two strategies are defined by different parameter setting.

Judging from (8), we can find there are three parameters that could affect optimization performance, the *inertia weight*  $\omega$ , the *cognitive* factor  $c_1$ , and the *social* factor  $c_2$ . Due to their importance to PSO, these three parameters of r-subswarm

together with those of K-subswarm are selected to be part of the C-particle. And POMS-PSO will optimize these six parameters.

For the inner loop, we divide the whole swarm into r-subswarm and K-subswarm as shown in Figure 2. That is why we call them not swarm but subswarm. Now we have to determine how many particles divided into r-subswarm. In other words, the proportion of r-subswarm  $P_r$  should be in the range of  $[0, 1]$ , but it needs to be determined. And on the other hand, the proportion of K-subswarm will be  $(1 - P_r)$ .

As the r-selection is likely to produce more progenies than K-selection, the fertility rate of r-subswarm  $\rho_r$  is larger than that of K-subswarm  $\rho_K$ . So we have

$$\rho_r > \rho_K \geq 1. \quad (9)$$

Three steps in Figure 2, the r- and K-selection and update particles, are presented in details in Figure 3. Suppose the total particle number of r- and K-subswarms is  $N$  as shown in Figure 3; after one evolutionary generation, the total number of progenies will be

$$N_p = \rho_K \times N \times P_r + \rho_r \times N \times (1 - P_r). \quad (10)$$

Considering (9), we can draw a conclusion that  $N_p \geq N$ . To keep stability of r- and K-subswarms size, we only select the best  $N$  out of  $N_p$  to survive to next iteration in inner loop and the others are abandoned. And we have to notice that like the classic PSO, C-particles only have one progeny.

Another important difference between r- and K-subswarm is that the velocity in r-subswarm may be much higher than the K-subswarm, because r-subswarm is designed to explore the search space to the greatest extent. The maximum velocity of r-subswarm,  $T_r$ , should be larger than that of K-subswarm  $T_K$ . These two parameters,  $T_r$  and  $T_K$ , should be optimized by C-swarm in POMS-PSO. Totally

TABLE 1: Eleven parameters optimized by outer loop in POMS-PSO.

Number	Symbols	Meanings
1	$\omega_K$	<i>Inertia weight</i> of K-subswarm
2	$c_{K1}$	<i>Cognitive factor</i> of K-subswarm
3	$c_{K2}$	<i>Social factor</i> of K-subswarm
4	$\omega_r$	<i>Inertia weight</i> of r-subswarm
5	$c_{r1}$	<i>Cognitive factor</i> of r-subswarm
6	$c_{r2}$	<i>Social factor</i> of r-subswarm
7	$\rho_K$	Fertility rate of K-subswarm
8	$\rho_r$	Fertility rate of r-subswarm
9	$P_r$	Proportion of r-subswarm
10	$T_r$	Maximum velocity of r-subswarm
11	$T_K$	Maximum velocity of K-subswarm

we have eleven parameters for r-subswarm and K-subswarm to be determined. Table 1 lists these parameters and their meanings. All of these parameters will be optimized in outer loop of POMS-PSO. The optimization procedure will be running with these eleven parameters' configuration. Although the parameter setting of r- and K-subswarm needs to be determined by the C-swarm in POMS-PSO, we have to notice that C-swarm perform the iteration procedure using the classic setting as given in [28].

As the inner loop is run under the configuration of a certain C-particle, the best fitness of inner loop got can be used to evaluate the corresponding C-particle. As shown in Figure 2, we take the best particle inner loop as the validation of C-particle. By evolution and update in the outer loop, POMS-PSO can find the best C-particle. And for inner loop, r- and K-subswarms can evolve in best parameter settings.

**3.2. Performance Evaluation of POMS-PSO.** We assessed the performance of parameter-optimized PSO method by employing a suite of benchmark functions and compared the result to that of the standard PSO (SPSO) and the constriction type PSO (CPSO) methods. The benchmark suite consists of two unimodal functions (Tablet and Quadric) and two multimodal functions (Rastrigin and Schaffer) [28]. All functions except for the 2D Schaffer function are optimized in 30 dimensional spaces.

The four functions are presented as follows:

Schaffer:

$$f(x) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{1 + 10^{-3}(x_1^2 + x_2^2)^2}, \quad x \in [-100, 100]^2. \quad (11)$$

Tablet:

$$f(x) = 10^6 x_1^2 + \sum_{i=2}^D x_i^2, \quad x \in [-100, 100]^D, \quad D = 30. \quad (12)$$

Rastrigin:

$$f(x) = 10D + \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i)), \quad (13)$$

$$x \in [-100, 100]^D, \quad D = 30.$$

TABLE 2: Optimized swarm parameters for four benchmark functions.

	Sch	Tab	Ras	Qua
$\omega_{K1}$	2.97	2.64	1.09	1.13
$c_{K1}$	3.35	2.75	3.65	2.78
$c_{K2}$	0.14	2.75	2.74	1.65
$\omega_{r1}$	1.05	0.97	1.81	2.74
$c_{r1}$	3.86	2.54	2.87	2.70
$c_{r2}$	1.94	1.92	3.38	2.26
$\rho_K$	2.83	3.31	1.24	1.36
$\rho_r$	7.76	3.79	8.69	9.21
$P_r$	0.74	0.01	0.01	0.01
$T_r$	0.56	0.77	1.00	1.00
$T_K$	0.07	0.36	0.01	0.01

Quadric:

$$f(x) = \sum_{i=1}^D \left( \sum_{j=1}^i x_j \right)^2, \quad x \in [-100, 100]^D, \quad D = 30. \quad (14)$$

The four functions are often used in optimization performance evaluation [28]. To test parameter-optimized PSO's optimal ability, we used the search space  $[-100, 100]^D$  which was larger than those in [28]. The Tablet and Quadric functions are evolved from spherical function, which have only one minimum point in the search space. And the Schaffer and the Rastrigin are multimodal, which means they have multiple local minimum points in the space. All of the benchmark functions have the global minimum  $f(x) = 0$  at  $x_i = 0$ . Unimodal and multimodal should be both evaluated for complete comparison performance assessment. General speaking, multimodal functions are much more difficult to find the optimal point.

As mentioned in Section 3.1, POMS-PSO employs 3 swarms in two loops, that is, the r-subswarm, the K-subswarm (in inner loop), and the C-swarm (in outer loop). One task of parameter-optimized PSO is to find the optimal swarm parameters setting for r- and K-subswarms, which is performed by the C-swarm. In our experiments, one population of 20 particles was chosen for r- and K-subswarms and 10 for C-swarm. The maximum number of iterations of outer loop is set to 40 and inner loop is set to 20. Table 2 lists the optimal results of C-particles of the four benchmark functions.

The parameter-optimized PSO can perform optimization for every benchmark function's parameter selection to guarantee various functions can get the optimal or better solution under various configurations instead of the common used settings. The parameter-optimized PSO is a problem-oriented PSO solution. In fact, we noticed that even we run the outer loop only once, we could obtain the results comparable to SPSO and CPSO in our experiments. In Table 2, all of the eleven parameters are shown for Sch (Schaffer), Tab (Tablet), Ras (Rastrigin), and Qua (Quadric), respectively. It is obvious that various functions have various configuration.

TABLE 3: Performance comparison of SPSO, CPSO, and POMS-PSO implementation. Maximum evolution generation of SPSO and CPSO: 800. Maximum evolution generation of POMS-PSO outer loop: 40, inner loop: 20.

Functions	Schaffer	Tablet	Rastrigin	Quadric
Standard PSO				
Mean	$8.61E - 04$	$1.57E + 03$	$1.21E + 03$	$3.29E + 02$
Standard deviation	$8.56E - 04$	$9.10E + 02$	$3.84E + 02$	$3.08E + 02$
Minimum	$4.88E - 15$	$1.84E + 02$	$4.54E + 02$	$3.51E + 01$
Median	$9.85E - 04$	$1.39E + 03$	$1.18E + 03$	$1.17E + 02$
Maximum	$8.76E - 03$	$6.11E + 03$	$2.92E + 03$	$2.13E + 03$
CPSO				
Mean	$8.43E - 04$	$2.52E + 03$	$1.93E + 03$	$9.69E + 02$
Standard deviation	$7.60E - 04$	$1.19E + 03$	$6.66E + 02$	$6.50E + 02$
Minimum	$0.00E + 00$	$4.01E + 02$	$7.99E + 02$	$1.06E + 02$
Median	$9.85E - 04$	$2.35E + 03$	$1.84E + 03$	$7.95E + 02$
Maximum	$8.76E - 03$	$7.38E + 03$	$5.24E + 03$	$4.71E + 03$
POMS-PSO				
Mean	$2.14E - 05$	$8.61E + 02$	$4.40E + 02$	$1.25E + 02$
Standard deviation	$2.80E - 05$	$3.30E + 03$	$2.39E + 02$	$2.20E + 02$
Minimum	$9.01E - 12$	$5.59E - 04$	$4.86E + 01$	$3.20E - 03$
Median	$1.08E - 05$	$6.72E + 01$	$4.09E + 02$	$1.59E + 01$
Maximum	$1.96E - 04$	$3.80E + 04$	$9.93E + 02$	$9.45E + 02$

The performance comparison between parameter-optimized PSO, SPSO, and CPSO methods is given in Table 3 for four benchmark functions. The optimization performance comparison is based on 200 trials. We compared the minimum, maximum, mean, and standard deviation of those three methods. The statistical results are listed when the SPSO, CPSO, and parameter-optimized PSO run 200 times. The size of all the populations is 20. The maximum evolution generation of SPSO and CPSO is 800, while that of POMS-PSO is also  $40 \times 20 = 800$  (outer loop: 40, inner loop: 20).

From the values listed in Table 3, we can see that at most time, the POMS-PSO can outperform the other two, no matter the minimum, maximum, mean, or standard deviation values, although POMS-PSO has much smaller iteration times. The performance of the POMS-PSO method decreases the errors by one order of magnitude compared with the other two methods.

As shown in Figure 4, multiruns experiments show POMS-PSO results vary in much smaller range intervals which means the proposed methods can give more stable optimum. Due to POMS-PSO's precision improvement under multi-subswarms strategy framework, it is a reasonable choice to adapt it into video superresolution reconstruction by solving the video degeneration model showed in (2), (3), and (4).

## 4. Video Superresolution by POMS-PSO

### 4.1. Reconstruction Quality Evaluation Criteria

**4.1.1. Similarity-Based Evaluation Criteria.** The PSNR is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise

that affects the fidelity of its representation. The PSNR is most commonly used as a measure of quality of reconstruction in image compression. It is defined via the mean squared error (MSE) as follows:

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i, j) - K(i, j)\|^2. \quad (15)$$

$I$  and  $K$  are two  $m \times n$  monochrome images where one of the images is considered a noisy approximation of the other.

The PSNR is defined as follows:

$$\text{PSNR} = 10 \log \left( \frac{255^2}{\text{MSE}} \right) (\text{dB}). \quad (16)$$

**4.1.2. Detail-Based Evaluation Criteria.** Sharpness is a photograph term that reflects image's detail as an integral part of its appeal. Sharpness is defined by the boundaries between zones of different tones or colors. It is illustrated by the bar pattern of increasing spatial frequency.

As shown in Figure 5, the top portion is sharp and its boundaries are crisp steps, not gradual. The bottom portion illustrates how the degraded pattern is blurred.

The sharpness is most commonly defined as the following equation for one  $m \times n$  monochrome image  $I$ :

$$\text{sharpness} = \frac{\sqrt{2}}{2(m-1)(n-1)} \times \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \sqrt{(I_{i,j} - I_{i+1,j})^2 + (I_{i,j} - I_{i,j+1})^2}. \quad (17)$$

The sharpness lines indicate the average gradient values of all frames in the video, which express whether the variation of scene is large or not in adjacent pixels.

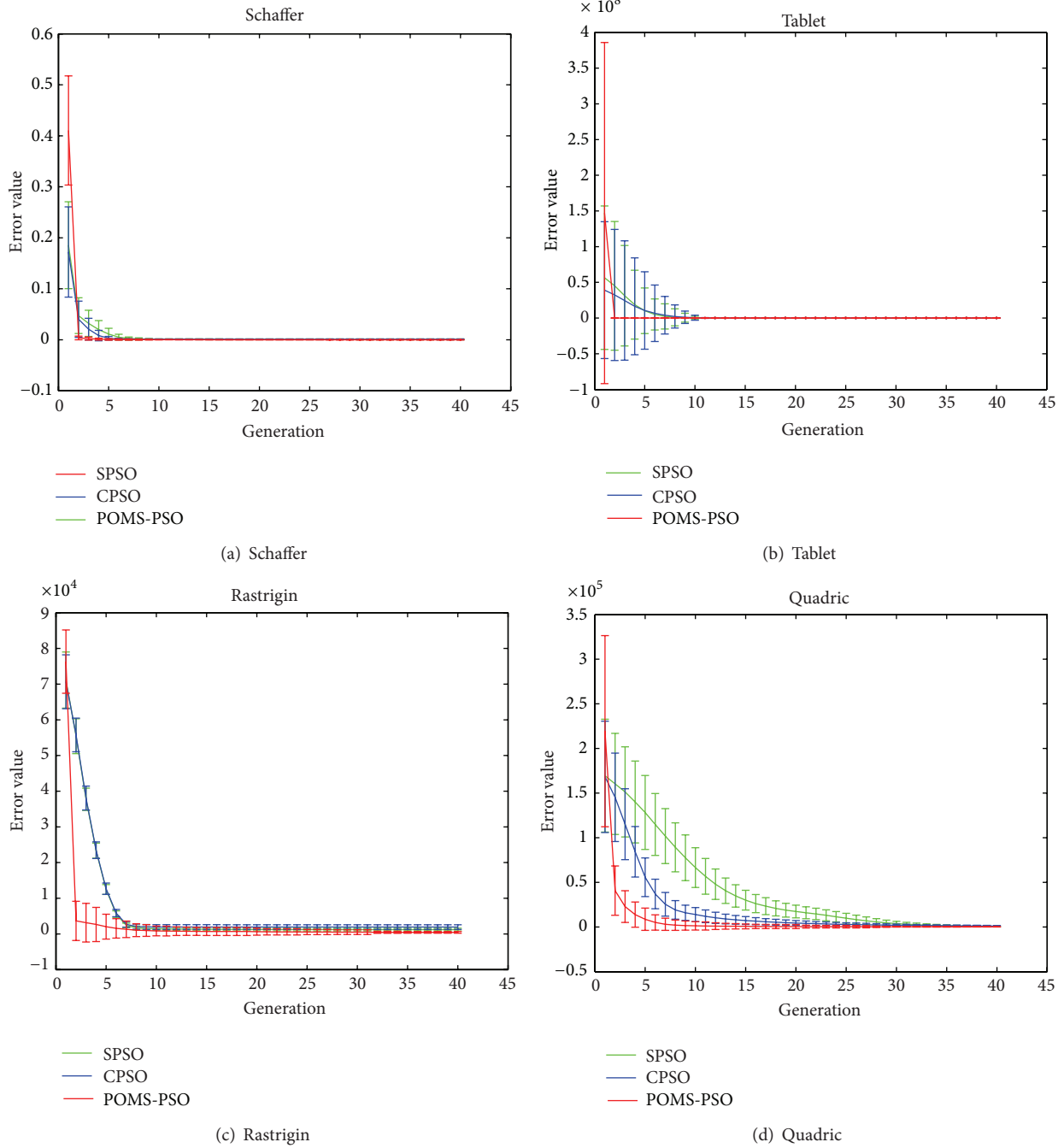


FIGURE 4: The convergence procedure of four functions between POMS-PSO, SPSO, and CPSO methods (The corresponding evolution generation is outer loop generation for parameter-optimized PSO and  $Generation \times 20$  for SPSO and CPSO method.).



FIGURE 5: Bar pattern: original (top); with degradation (bottom).

4.1.3. *Information-Based Evaluation Criteria.* Image entropy reflects the average amount of information for a given image and the entropy of grayscale image indicates the amount

of information contained in aggregation feature of gray distribution for the given image and it can be computed as follows:

$$Entropy = - \sum_{i=0}^{M-1} P_i \log_2 (P_i), \quad (18)$$

where  $M$  is the number of gray levels and  $P_i$  is the probability associated with gray level  $i$ .



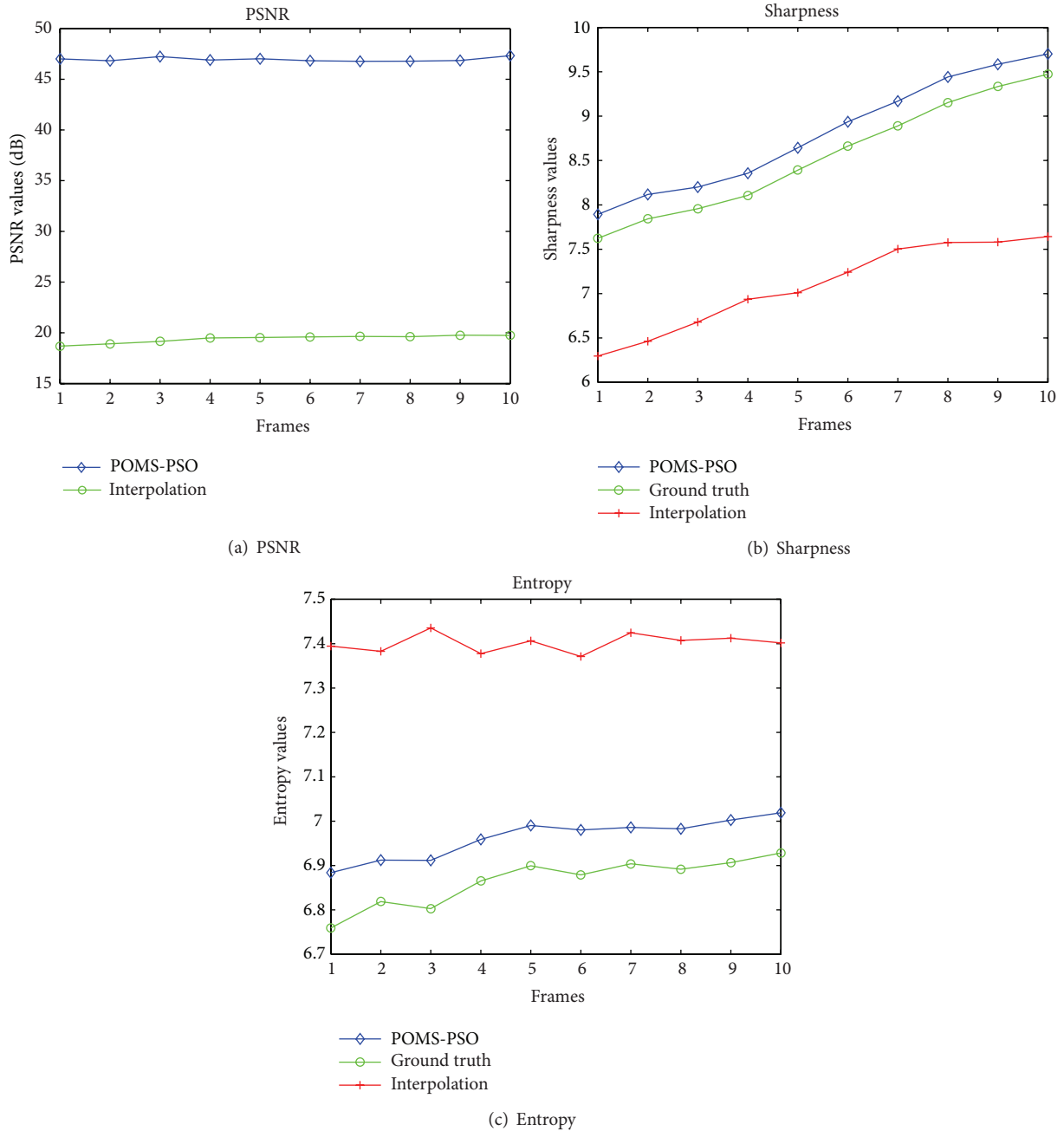


FIGURE 6: Criteria comparison on VSR performance for 10 frames video.

4.2. *Reconstruction Simulation and Experimental Results.* We assessed the performance of the video superresolution process using POMS-PSO algorithm on several low-resolution video sequences and compared the result to that of the bilinear interpolation method. The resolution of the reconstructed video is twice as large as incoming video, in both height and width. To compare the objective criteria, the input video used to perform superresolution procedure was obtained by 2-downsampling the ground truth video. Due to minimize the errors and noise effect on the reconstructed sequence, we introduced an Gaussian filter with size  $3 \times 3$ .

All of the reconstructed video should be compared with the ground truth video sequence. Taking 10 frames as experimental cases, we achieved the PSNR, sharpness, and entropy results as shown in Figure 6. The maximum iteration of POMS-PSO is set to 15. The size of whole populations is set to 10 particles.

From PSNR lines in Figure 6, we can see that the result using POMS-PSO algorithm achieved PSNR value approximating about 47 dB and bilinear interpolation only got 18 dB. That is to say, the proposed method has a much better objective image quality than that of bilinear interpolation



(a) The input



(b) VSR results of bilinear interpolation



(c) VSR results of POMS-PSO



(d) Ground truth

FIGURE 7: Results of the 9th (left) and 10th (right) frames of the video.

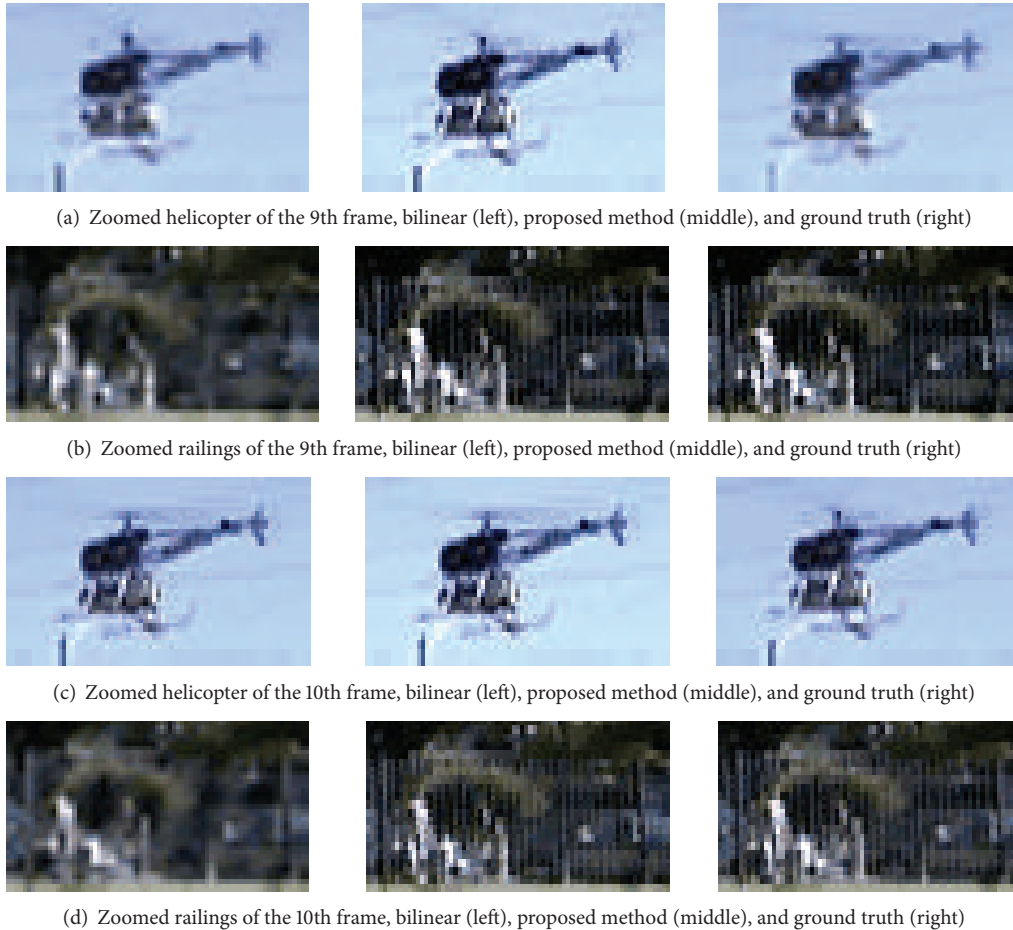


FIGURE 8: Zoomed parts of the 9th and 10th frames.

method. The higher sharpness value means the more details the frame shows. We can see that the sharpness values of POMS-PSO algorithm is much higher than those of bilinear interpolation method by 1.5 in average and even better than those of ground truth by 0.5 in average. The entropy of proposed method is lower than the bilinear interpolation, but it is very close to the ground truth. The close entropy means the reconstructed sequence has similar pixel value distribution to ground truth, while the higher entropy got by interpolation could be seen as artificial effect.

The results of the reconstruction using bilinear interpolation and POMS-PSO are given as Figure 7. The images given are the 9th and the 10th frames of the corresponding reconstructed video. The images in Figure 7 show that video reconstruction using POMS-PSO could obtain high-quality results with a small number of iterations. The video is continuous smoothly.

For more clear details, the helicopter and the railings from the 9th and 10th frames are zoomed and shown in Figure 8. Obviously, the helicopter is of great interest in the whole image, and the railings are the most difficult to recognize fine details part. For the bilinear interpolation, the helicopter and the railings are blurred seriously. Judging from Figure 8, we can draw a conclusion that the objective

vision quality of POMS-PSO result is much better than that of bilinear interpolation, especially in the regions of interest. Considering other parts of the images, say the sky and the trees, VSR by POMS-PSO can also achieve better performance.

## 5. Conclusions and Future Works

In this paper, we proposed POMS-PSO which could help to find out better swarm configuration for given problem. The POMS-PSO employed two layer loops and three subswarms. Performance comparison on four standard benchmark functions showed POMS-PSO could achieve higher accuracy in unimodal and multimodal functions.

The model of imaging degeneration is very important during reconstruction. To some extent, proper model determination is the basis of finding the solution. Based on a imaging degeneration model and POMS-PSO, we proposed a novel VSR method in view of optimization computation. In view of computation intelligence, the video sequence with higher resolution is thought as the optimal resolution for swarm optimization. Experimental results showed that the proposed novel method could obtain high objective and subjective quality results.

In the future, more efforts should be focused on the perfect degeneration model building and the fast implementation of multiple swarms based adaptive optimization algorithms.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### Acknowledgments

This work was supported by the National Natural Science Foundation of China with no. 61305041 and the Fundamental Research Funds for the Central Universities of China with no. K5051304024.

### References

- [1] J. L. Harris, "Diffraction and resolving power," *Journal of the Optical Society of America*, vol. 54, no. 7, pp. 931–936, 1964.
- [2] J. W. Goodman, *Introduction to Fourier Optics*, McGraw-Hill, New York, NY, USA, 1968.
- [3] B. R. Hunt, "Super-resolution of images: algorithms, principles, performance," *International Journal of Imaging Systems and Technology*, vol. 6, no. 4, pp. 297–304, 1995.
- [4] M. Ben-Ezra, A. Zomet, and S. K. Nayar, "Video super-resolution using controlled subpixel detector shifts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 977–987, 2005.
- [5] C. Liu and D. Sun, "On Bayesian adaptive video super resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 346–360, 2014.
- [6] J. Buss, C. Coltharp, and J. Xiao, "Super-resolution imaging of the bacterial division machinery," *Journal of Visualized Experiments*, no. 71, 2013.
- [7] H. Su, Y. Wu, and J. Zhou, "Super-resolution without dense flow," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1782–1795, 2012.
- [8] A. J. Patti, M. I. Sezan, and A. M. Tekalp, "Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time," *IEEE Transactions on Image Processing*, vol. 6, no. 8, pp. 1064–1076, 1997.
- [9] R. R. Schultz and R. L. Stevenson, "Extraction of high-resolution frames from video sequences," *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 996–1011, 1996.
- [10] B. Bascle, A. Blake, and A. Zisserman, "Motion deblurring and super-resolution from an image sequence," in *Proceeding of European Conference on Computer Vision (ECCV '96)*, pp. 312–320, Springer, 1996.
- [11] M. Irani and S. Peleg, "Motion analysis for image enhancement: resolution, occlusion, and transparency," *Journal of Visual Communication and Image Representation*, vol. 4, no. 4, pp. 324–335, 1993.
- [12] M. Belkacem, T. Bouktir, and K. Srairi, "Strategy based PSO for dynamic control of UPFC to enhance power system security," *Journal of Electrical Engineering and Technology*, vol. 4, no. 3, pp. 315–322, 2009.
- [13] H.-T. Yau, C.-J. Lin, and Q.-C. Liang, "PSO based PI controller design for a solar charger system," *The Scientific World Journal*, vol. 2013, Article ID 815280, 13 pages, 2013.
- [14] A. Chander, A. Chatterjee, and P. Siarry, "A new social and momentum component adaptive PSO algorithm for image segmentation," *Expert Systems with Applications*, vol. 38, no. 5, pp. 4998–5004, 2011.
- [15] Z. Geng and Q. Zhu, "A multi-swarm PSO and its application in operational optimization of ethylene cracking furnace," in *Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA '08)*, pp. 103–106, Chongqing, China, June 2008.
- [16] S. Peters and A. Koenig, "A hybrid texture analysis system based on non-linear & oriented kernels, particle swarm optimization, and kNN vs. support vector machines," *Neural Network World*, vol. 17, no. 6, pp. 507–527, 2007.
- [17] M. Meissner, M. Schmucker, and G. Schneider, "Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training," *BMC Bioinformatics*, vol. 7, article 125, 2006.
- [18] A. Subasi, "Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders," *Computers in Biology and Medicine*, vol. 43, no. 5, pp. 576–586, 2013.
- [19] S. Nema, J. Goulermas, G. Sparrow, and P. Cook, "A hybrid particle swarm branch-and-bound (HPB) optimizer for mixed discrete nonlinear programming," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 38, no. 6, pp. 1411–1424, 2008.
- [20] B. Sharma, R. K. Thulasiram, and P. Thulasiraman, "Portfolio management using particle swarm optimization on GPU," in *Proceedings of the 10th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA '12)*, pp. 103–110, July 2012.
- [21] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '00)*, pp. 84–88, July 2000.
- [22] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Proceedings of the 7th Annual Conference on Evolutionary Programming*, pp. 591–600, New York, NY, USA, 1998.
- [23] A. Carlisle and G. Dozier, "An off-the-shelf PSO," in *Proceedings of the Workshop on Particle Swarm Optimization*, pp. 1–6, Indianapolis, Ind, USA, 2001.
- [24] A. Lari, A. Khosravi, and F. Rajabi, "Controller design based on mu analysis and PSO algorithm," *ISA Transactions*, vol. 53, no. 2, pp. 517–523, 2014.
- [25] T. Blackwell and J. Branke, "Multi-swarm optimization in dynamic environments," in *Applications of Evolutionary Computing*, vol. 3005 of *Lecture Notes in Computer Science*, pp. 489–500, 2004.
- [26] Y. Y. Yan and B. L. Guo, "Particle swarm optimization inspired by r- and k-selection in ecology," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, pp. 1117–1123, Hong Kong, China, June 2008.
- [27] Y. Yan and B. Guo, "Convergence analysis of PSO inspired by r-and K-selection," in *Proceedings of the 8th International Conference on Intelligent Systems Design and Applications (ISDA '08)*, vol. 2, pp. 247–252, Kaohsiung, Taiwan, November 2008.
- [28] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

- [29] K. E. Parsopoulos and M. N. Vrahatis, "Recent approaches to global optimization problems through particle swarm optimization," *Natural Computing*, vol. 1, no. 2-3, pp. 235–306, 2002.
- [30] N. Zeng, Z. Wang, Y. Li, M. Du, and X. Liu, "A hybrid EKF and switching PSO algorithm for joint state and parameter estimation of lateral flow immunoassay models," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 2, pp. 321–329, 2012.
- [31] Y. Yan and B. Guo, "Two image denoising approaches based on wavelet neural network and particle swarm optimization," *Chinese Optics Letters*, vol. 5, no. 2, pp. 82–85, 2007.
- [32] Y. Yan, B. Guo, Z. Yang, and X. Fu, "Image noise removal via wavelet transform and r/K-PSO," in *Proceedings of the 4th International Conference on Natural Computation (ICNC '08)*, vol. 5, pp. 544–548, Jinan, China, October 2008.
- [33] Y. Y. Yan and B. L. Guo, "r/K-PSO and its convergence speed analysis," in *Proceedings of the 8th International Conference on Intelligent Systems Design and Applications (ISDA '08)*, vol. 2, pp. 247–252, Kaohsiung, Taiwan, November 2008.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

