THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 16, Number 1/2015, pp. 95–102

PSO AND GA BASED NEIGHBOR EMEDDING SUPER RESOLUTION

Saeid FAZLI [1], Maryam TAHMASEBI [2]

[1] Zanjan University, Research Institute of Modern Biological Techniques, Zanjan, Iran
[2] Zanjan University, Eng. Faculty, Electrical Dept., Zanjan, Iran
E-mail: m.tahmasebi@znu.ac.ir

In this paper a novel technique for Neighbor embedding single image super resolution (SR) is proposed. Given a low-resolution image, its high-resolution image is reconstructed from a set of training images, which are composed of one or more low-resolution and corresponding high-resolution image pairs. In this paper we propose a new approach to a single image super-resolution through neighbor embedding using Genetic Algorithms (GA) and Particle Swarm Optimization (PSO). GAs and PSO are used for patch size, overlap and K nearest neighbor parameters tuning of neighbor embedding super resolution by maximizing the PSNR as a fitness value. Experiments show that the use of GA and PSO for finding the parameters of neighbor embedding method is more accurate than setting the parameters as random. Also, it can be seen from the results that the proposed method increases the average of PSNR 2.2db in comparing with Bicubic interpolation, but the PSNR differences between PSO and GAs are not significant.

Key words: neighbor embedding, genetic algorithms (GA), particle swarm optimization (PSO).

1. INTRODUCTION

The super-resolution (SR) methods can be divided into two classes: one class is multiple-frame SR [1, 2, etc.], which generates a high-resolution (HR) image from multiple low-resolution (LR) images of the same scene. The other one is single-frame SR [3, 4, 5, 6, etc.], which generates a HR image from a single LR image, with the help of training set images. In this paper, we focus on the single-image SR problem.

Neighbor embedding algorithm has been widely used in example-based super-resolution reconstruction from a single frame, which assumes that neighbor patches embedded are contained in a single manifold. Chang et al. (2004) first propose the neighbor embedding super-resolution method, which assumes the patches of high- and low-resolution images, can form manifolds with similar local geometry in the two different feature spaces. First, they compute the reconstruction weights of each low-resolution patch's neighbors in low-resolution training image set by minimizing the reconstruction error. Second, they estimated the high-resolution embedding from the training image pairs by preserving local geometry. Finally, they enforce local compatibility between adjacent high-resolution patches.

According to the study-based super resolution algorithm, the training set is often a subset of all patches of one or several images. With a good patch selecting strategy, the generality and reliability of the super-resolution algorithm will be largely improved. If the size of the patch is too small, the training set is enlarged and more patches of the input image would be calculated; if the patch is too large in size, the matching error is magnified and the acquired high-resolution image is low in quality. Additionally, the local patch information is not enough to predict the detailed information of the high-resolution, but the effect of the spatial neighborhood should be taken into account. So, during the process of breaking the low-frequency image by raster scan order, every patch should be partially overlapped by its neighbor to keep the accordance of the space neighborhood [7].

In this paper, we apply both the PSO and GA approaches for optimizing the value of patch size, overlap and K nearest neighbor in Neighbor Embedding methods in order to obtain higher PSNR than Bicubic interpolation. Instead of using the super resolution problem directly with highly computationally complex algorithms, genetic algorithms and particle swarm optimization can be applied to find the optimal parameters of neighbor embedding method and promote the output results. Experimental results show that this approach can effectively obtain high resolution image and make the super-resolution algorithm of the image more practical.

This paper is organized as follows. In section 2 we give a review of neighbor embedding algorithm for super-resolution. In section 3 we outline the methodology of this genetic algorithm and particle swarm optimization to solve the neighbor embedding super resolution parameters, followed by the experiments and analysis in section 4. The conclusion is explained in section 5.

2. REVIEW OF NEIGHBOR EMBEDDING FOR SUPER-RESOLUTION RECONSTRUCTION

The idea of neighbor embedding for super-resolution reconstruction was first proposed by Chang et al. [8]. As follows, we will give a brief formulation of neighbor embedding for image super-resolution.

The target high-resolution image Y_t of a low-resolution image X_t is estimated using a training set of one or more low resolution images X_s and the corresponding high-resolution images Y_s . Each low- or high-resolution image represents as a set of small overlapping image patches. X_t and Y_t have the same number of patches, and each low-resolution image in X_s and the corresponding high-resolution image in Y_s also have the same number of patches. The sets of image patches denote corresponding to x_s , y_s , x_t and y_t as $\{x_s^p\}_{p=1}^{N_s}$, $\{y_s^p\}_{p=1}^{N_s}$, $\{x_t^q\}_{q=1}^{N_t}$ and $\{y_t^q\}_{q=1}^{N_t}$. $\{y_t^q\}_{q=1}^{N_t}$. Obviously, N_s and N_t depend on the patch size and the degree of overlap between adjacent patches.

Neighbor embedding method for SR reconstruction can be summarized in five steps [8].

(a) For each patch x_t^q in image X^t do.

(b) Find the set N_q of K nearest neighbors in X_s .

(c) Calculate the reconstruction weights of the neighbors for minimizing the error of reconstructing x_t^q

$$\varepsilon^q = \left\| x_t^q - \sum_{x_s^p N^q} w_{qp} x_s^p \right\|^2 . \tag{1}$$

In the equation (1), w_{qp} is the weight for x_s^p , subject to the following constraints in the equation (2)

$$\sum_{x_s^p \in N_q} w_{qp} = 1 \text{ and } w_{qp} = 0 \text{ for any } x_s^p \notin N_q . \tag{2}$$

(d) Compute the high-resolution embedding y_t^q using the appropriate high-resolution features of the K nearest neighbors and the reconstruction weights.

$$y_t^q = \sum_{x_s^p \in N_q} w_{qp} y_s^p . \tag{3}$$

(e) Construct the target high-resolution image Y^t .

3. PROPOSED ALGORITHM

In this section, we will describe the idea of the proposed method. Our neighbor embedding method has only three parameters that we have explored to optimize them. The first parameter is the number of K nearest neighbors for neighbor embedding. The second and third parameters are the patch size and the degree of overlap between adjacent patches. Our aim is to find the best set of values for these three parameters which can produce the optimal result (better PSNR). In this paper, PSO and GA are used to optimize the patch size, overlap and K nearest neighbor parameters of the neighbor embedding method. We propose the use of particle swarm optimization and genetic algorithm techniques that speeds up the convergence and reduces the computation time.

3.1. Determining neighbor embedding parameters using genetic algorithms

Although GA started much earlier than 1975, Holland (1975) is the key literature that introduced GA to broader audiences. In GA, the solutions are represented as chromosomes. The chromosomes (a string of genes that represents a solution) are evaluated for fitness values and they are ranked from best to worst based on fitness value. The process is accomplished by repeating applications of three genetic operators: selection, crossover, and mutation. First, the better offspring are selected to become parents to produce new chromosomes. To actuate the remaining of the fittest, the chromosomes with better fitness value are selected with higher probabilities than the chromosomes with weaker fitness. The selection probabilities are usually defined using the relative ranking of the fitness values. As soon as the parent chromosomes are selected, the crossover operator incorporates the chromo some of the parents to produce new offspring (perturbation of old solutions). Since stronger (fitter) individuals are being selected more often, there is a trend that the new solutions may become very similar after several generations, and the variety of the population may decline; and this could lead to population stagnation [9].

Population size, number of generations, crossover and mutation rate parameters effect on the GAs algorithms. Great number of generations (i.e. thousands) and greater population size (i.e. hundreds) increase the likelihood of obtaining a global optimum solution, but significantly increase processing time. Crossover amongst parent chromosomes is a common natural process, and the variation of parents' information produces children (offspring). *Versus* crossover, mutation is an uncommon process that resembles a sudden change to a child (offspring). This can be done by randomly selecting one chromosome from the population and then arbitrarily changing some of its information. The profit of mutation is that it randomly introduces new genetic material to the evolutionary process, probably thereby avoiding stagnation around local minima [10]. More details on the mechanism of GAs can be found in Goldberg [11] and Al-Tabtabai and Alex [12]. The flowchart of the genetic algorithm is given in Fig.1.

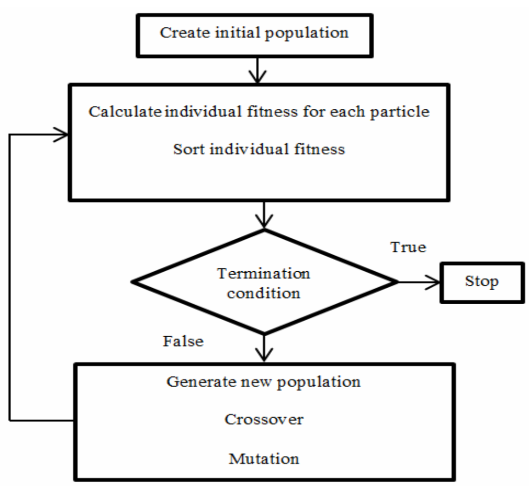

Fig. 1 – Flowchart for genetic algorithm.

In GA every individual in the population gets an evaluation of its adaptation (fitness) to the environment. The selection selects the best gene compositions also referred as individuals, which through crossover and mutation should actuate to better solutions in the next population. Our schemes of Neighbor embedding method via GA can be summarized as follows.

1) Generate initial population (Initialize patch size, overlap and K parameters randomly within their range and corresponding random velocities).

2) For each Particle i=1 to P do
3) Calculate PSNR as a fitness value through neighbor embedding as described in section(2)
4) End for
5) While (Termination condition ≠ true) do
6) Selection – between all individuals in the current population are chose those, who will continue and by means of crossover and mutation will produce offspring population
7) Crossover – the individuals chosen by selection recombine with each other and new individuals will be created. The goal is to get offspring individuals that inherit the best possible combination of the characteristics (genes) of their parents
8) Mutation – by means of random change of some of the genes, it is supported that even if none of the individuals contain the necessary gene value for the extreme, it is possible to reach the extreme
9) New generation – the elite individuals chosen from the selection are combined with those who passed the crossover and mutation, and form the next generation.
10) End while

3.2. Determining Neighbor Embedding Parameters Using Particle Swarm Optimization

PSO was first introduced by Kennedy and Eberhart [13] as an optimization method for continuous nonlinear functions. PSO is a biologically inspired algorithm motivated by a social analogy. The swarm is initialized with a group of random particles and it then searches for optima by updating through iterations. In every iteration, each particle is updated by following two "best" values. The first one is the best value of each particle achieved so far. This value is known as solution. The second one is that, best solution tracked by any particle among all generations of the swarm. The best fitness value is known as solution. These two best values are responsible to drive the particles to move to new better position. After finding the two best values [14], a particle updates its velocity and position with the help of the following equations (4, 5):

$$v_i^{t+1} = W^t \cdot v^t{}_i + c_1 \cdot r_1 \cdot (\mathrm{pbest}_i^t - \mathrm{X}_i^t) + c_2 \cdot r_2 \cdot (g\,\mathrm{best}_i^t - \mathrm{X}_i^t)\,\mathrm{i}^{th}. \tag{4}$$

$$X_i^{t+1} = X_i^t + v_i^{t+1}, \tag{5}$$

where X_i^t and v_i^t denotes the position and velocity of i^{th} particle at time instance t. Maximum and minimum value for *w* is set to two and zero respectively, which is same for all particles. W^t is inertia weight at t^{th} instant of time, c_1 and c_2 are positive acceleration constants in range [0, 2], r_1 and r_2 are random value generated in the range [0, 1], pbest_i is the best solution of i^{th} individual particle, $g\,\mathrm{best}$ is the best particle obtained over all generations so far [14].

Based on the above PSO model, we denote X_i^t as a population matrix. The population with three parameters for i^{th} particle at time instance t includes patch size, overlap and *K* nearest neighbor.

The PSO algorithm searches for the best solution through an iterative process. At every iteration, the fitness of each particle is evaluated using the fitness value (PSNR). If it is the best value the particle has achieved so far, the particle stores that value as 'personal best'. The best fitness value achieved by any particle during current iteration is stored as 'global best'. Our algorithm of Neighbor embedding method *via* PSO can be summarized as follows:

1) Generate initial population with three parameters for each particle includes patch size, overlap and K nearest neighbor. (Initialize parameters patch size, overlap and *K* randomly within their range and corresponding random velocities).
2) For each Particle i = 1 to P do
3) Evaluate PSNR as a fitness value through neighbor embedding as described in section(2)
4) End for

5) While (Termination condition ≠ true) do
6) Update velocity according to the equation (4)
7) Update latent position according to the equation (5)
8) For each Particle i=1 to P do
9) Calculate PSNR as a fitness value as mentioned above
10) If the fitness value is better than the best fitness value (pbest) in history
11) Set current value as new pbest
12) End for
13) Update global best by choosing the particle with the best fitness value of all the particles as the gbest
14) End while

4. EXPERIMENTAL RESULTS AND EVALUATIONS

In this section, we will show the performance of the proposed method and perform comparisons between the PSO and GAs, with the same fixed population, and Bicubic interpolation. For example-based image super-resolution, training set is important for reconstruction quality of high-resolution image. The proposed method is tested on four images (Fig. 2). For all the experiments, when any one image is seen as a testing image, the rest acts as the generation of training samples. To get input LR images, each HR is degraded by blurring, and down-sampled with factor 2 to product a testing input image.

As mentioned above, we use PSO and GAs for finding neighbor embedding parameters. The constant parameters for PSO algorithm used in our experiment can be seen in Table 1. There are three parameters that we explore them by PSO and GAs. We find an optimal value of K for all our experiments. For the low-resolution images, we find $M \times M$ patches with an overlap of N pixels between adjacent patches. If we want to magnify a low resolution image by S times in each dimension, then we use SM * SM patches in the high-resolution image with an overlap of SN pixels between adjacent patches. In our experiment, the range of these parameters is selected for patch size $\in [3, 6]$, overlap $\in [1, 4]$ and $K \in [1, 5]$. Objectively, peak-signal-to-noise ratio (PSNR) is exploited as a fitness value. The PSNR is defined as in the following equation

$$\mathrm{PSNR} = 10 \log_{10} \frac{255^2}{\mathrm{MSE}} . \tag{6}$$

In Table 2, we compare the PSNR value between bicubic interpolation and neighbor embedding using PSO and GAs for four testing images. For all testing images, the results show that PSO and GAs achieve better PSNR value than Bicubic interpolation methods, but the differences between PSO and GAs are not significant. In Table 3, we show details about testing images and optimal value obtained for three parameters.

Moreover, we explore the effect of number of iteration for convergence and show that PSO algorithm has high tendency for premature convergence and GAs has medium tendency, the result can be seen in Table 4. Figure 3 shows the results of applying different super resolution methods to a Lena image to obtain 2× magnification. As it can be seen, our method gives the best results than bicubic interpolation.

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

Fig. 2 – Testing images: a) Lena; b) Head; c) Hat; d) Starfish.

Table 1

Constant parameters for PSO

Parameter Name	Value
Size of the swarm	10
Maximum number of iteration	10
Cognitive scaling parameter(C1)	1.5
Social scaling parameter(C2)	1.5
Fitness value	PSNR

Table 2

Fitness values of the enhanced images

Image	Quantitative comparison (PSNR in db)		
	Bicubic interpolation	Proposed	
		GAs	PSO
Lena	27.9505	30.4272	30.4534
Head	29.8256	30.6940	30.9003
Hat	28.3100	30.2997	30.2997
Starfish	22.7686	26.0750	26.0750

Table 3

Details about original images and optimal value for parameters

Image	Size(M×N)	Method	Low patch size	Overlp	K
Lena	256×256×3	PSO	3	2	5
		GA	3	2	4
Head	280×280×3	PSO	3	2	5
		GA	3	2	1
Hat	256×256×3	PSO	3	2	5
		GA	4	3	5
Starfish	220×220×3	PSO	3	2	5
		GA	3	2	5

Table 4

The effect of number of iteration on convergence

Image/Iteration	Method	Number of Iteration			
		3	5	8	10
Lena	PSO	30.4272	30.4534	30.4534	30.4534
	GA	25.2128	25.3179	30.2997	30.4272
Head	PSO	30.8624	30.9003	30.9003	30.9003
	GA	26.6904	26.6904	30.6940	30.6940
Hat	PSO	30.2613	30.2997	30.2997	30.2997
	GA	26.1635	30.2837	30.2837	30. 2837
Starfish	PSO	26.0093	26.0750	26.0750	26.0750
	GA	22.1594	25.9591	25.9591	26.0750

(a) (b) (c)

(d) (e)

Fig. 3 – The 2× magnification of the head image from a 128 × 128 low-resolution image: a) input low-resolution image; b) true high-resolution image; c) bicubic interpolation; d) our method with PSO; e) our method with GAs.

5. CONCLUSION

The purpose of this paper is to use genetic algorithm and particle swarm optimization solution for determining the three parameters of neighbor embedding super resolution and obtaining HR images. With this goal, we search for optimal value of patch size, overlap and K nearest neighbor for higher PSNR value. Results of the proposed technique are compared with Bicubic interpolation technique. The experimental results show that the proposed algorithm can achieve better results than Bicubic interpolation. Moreover, in PSO, the most important feature is that, it can produce better result with proper tuning of parameters. It is also true for GA based image enhancement. In comparison to GA, PSO takes less time to converge to optimum. Also, we prove that the proposed method increases the average of PSNR 2.2db in comparing with Bicubic interpolation. As a result, we think that a PSO and GAs approaches for super resolution problem are worth generalization and further investigation in other applications.

REFERENCES

1. M. ELAD, and A. FEUER, *Super-resolution reconstruction of image sequences*, IEEE Transactions on Pattern Analysis and Machine Intelligence **21**, *9*, pp. 817–834, 1999.
2. G. H. COSTA, and J.C.M. BERMUDEZ, *Statistical analysis of the LMS algorithm applied to super-resolution image reconstruction*, IEEE Transactions on Signal Processing ED, **11**, pp. 2084–2095, 2007.
3. Y.T. ZHUANG, J. ZHANG, and F. WU, *Hallucinating faces: LPH super-resolution and neighbor reconstruction for residue compensation*, Pattern Recognition, **40**, pp. 3178–3194, 2007.
4. H. CHANG, D. YEUNG, and Y. XIONG, *Super-resolution through neighbor embedding*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, **1**, pp. 275–282, 2004.
5. J.S. PARK, and S.W. LEE, *An example-based face hallucination method for single-frame low-resolution facial images*, IEEE Transactions on Image Processing, **17**, *10*, pp. 1806–1816, 2008.
6. T.M. CHAN, J.P. ZHANG, J. PU, and H. HUANG, *Neighbor embedding based super-resolution algorithm through edge detection and feature selection*, Pattern Recognition Letters, **30**, pp. 494–502, 2009.
7. SHU ZHANG, *Example-based Super-resolution*, Master of Computing, Australian National University, Engineering and Computer Science College, 2013.
8. H. CHANG, D.-Y. YEUNG, and Y. XIONG, *Super-resolution through neighbor embedding*, IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, 2004, pp. 275–282.
9. VORATAS KACHITVICHYANUKUL, *Comparison of Three Evolutionary Algorithms:GA, PSO, and DE*, Industrial Engineering & Management Systems, **11**, *3*, pp.215–223, 2012.

10. EMAD ELBELTAGI, TAREK HEGAZY, DONALD GRIERSON, *Comparison among five evolutionary-based optimization algorithms*, Advanced Engineering Informatics, Elsevier, 2005, pp. 43–53.
11. GOLDBERG D.E., *Genetic algorithms in search, optimization and machine Learning*, Addison-Wesley Publishing Co, 1989.
12. AL-TABTABAI H., ALEX P.A., *Using genetic algorithms to solve optimization problems in construction*, Eng. Constr. Archit. Manage., **6**, *2*, pp.121–32, 1999.
13. J. KENNEDY, R.C. EBERHART, *Particle swarm optimization*, IEEE International Conference on Neural Networks, Perth, pp. 1942–1948, 1995.
14. APURBA GORAI, ASHISH GHOSH, *Gray-level Image Enhancement by Particle Swarm Optimization*, IEEE, Nature & Biologically Inspired Computing, 2009.

Received April 28, 2014