International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 22 (2015) pp 43077-43081

Image Restoration using modified binary particle Swarm Optimization Richardson-Lucy (MBSO-RL) algorithm

Hari Shanker Shukla,
Professor, Department of Computer Science, D. D. U Gorakhpur university, Gorakhpur District profhsshuklagkp@rediffmail.com

Narendra Kumar,
Ph.D. Scholar, Department of Computer Science, D.D.U Gorakhpur university, Gorakhpur District narendra298@gmail.com

Rakesh Prakash Tripathi,
Professor, Department of Mathematics, Graphic Era University, Dehradun, Dehradun District tripathi_rp0231@rediffmail.com

Abstract
The recovery of the image is called restoration that has been degraded by blur and noise. The recovery of an original image from a corrupted image is of importance and can find its application in several area including medical and diagnostics, satellite and astronomical imaging and remote sensing etc.
In this paper, we use modified binary particle Swarm Optimization to optimize the PSF and Richardson-Lucy algorithm is used to restore the corrupted image by pepper-and-salt noise. By using the median filter and MBPSO-RL methods, the de-noising and restoration of the image are performed efficiently. The performance of image denoising and restoration techniques is evaluated by comparing the result of proposed techniques with the existing de-noising filter like Wiener Filter and Bilateral Filter. Comparison result shows a high-quality de-noising the restoration ratio for the noisy images than the existing method, in the terms of second-derivation-like measure of enhancement(SDME)..

Keywords: The second-derivation-like measure of enhancement (SDME), modified binary particle Swarm Optimization (MBPSO), Richardson-Lucy(LR), particle Swarm Optimization(PSO), point spread function(PSF)..

Introduction

Image deblurring or image de-convolution is sometimes known as field image restoration and concerned with the reconstruction or estimation of the uncorrupted image from blurred and noisy one. blurring and noise Image restoration are associated with minimizing or even removing artifacts. Blurring is a linear form of degradation can occur due to camera de-focusing or due to motion.

When such pictures are converted from one form to another by processes such as imaging, scanning, or transmitting, the quality of the output image may be inferior to that of the original input picture. There is thus a need to improve the quality of such images so that the output image is visually more pleasing to human observers from a subjective point of view. It is important to increase the dynamic range of the chosen features in the image to perform the task, which is requirements of image enhancement. Enhancement has another purpose as well, that is to undo the degradation effects which might have been caused by the imaging system or the channel.The growing need to develop automated systems for image interpretation necessitates that the quality of the picture to be interpreted should be free from

Noise and other aberrations. Thus, it is important to perform preprocessing operations on the image so that the resultant preprocessed image is better suited for machine interpretation. Image enhancement thus has both a subjective and an objective role and may be viewed as a set of techniques for improving the subjective quality of an image and also for enhancing the accuracy rate in automated object detection and picture interpretation.

Enhancement refers to accentuation or sharpening of image features, such as contrast, boundaries, edges, etc. Modeling of the degradation process, in general, is not required for enhancement. However, knowledge of the degradation process may help in the choice of the enhancement technique. The realm of image enhancement covers contrast and edge enhancement, noise filtering[1], feature sharpening, and so on. These algorithms are generally interactive, application dependent, and employ linear or nonlinear local or global filters.

A.Noise Model

Since main sources of noise[8] presented in digital images are resulted from atmospheric disturbance and image sensor circuitry, following assumptions can be made:

- The noise model is spatial invariant, i.e., spatial location independent.
- The noise model is uncorrelated with the object function.

Some commonly used noise models can be categorized into two groups: additive noise and multiplicative noise.

i. Additive noise models:

Additive noise can be classified In according to resulted in variation of the image signal. noise distributions are given below:

- Gaussian noise distribution

$$\mathrm{p(z)} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$

- Rayleigh noise distribution

$$\mathrm{p(z)} = \frac{2}{b}(z-a)e^{-\frac{(z-a)^2}{b}}, \text{for } z \geq a$$

- Gamma(a, b) noise distribution

$$p(z) = \frac{a^b z^{b-1}}{(b-a)!} e^{-az}, \text{ for } z \geq 0$$

- Exponential noise distribution

$$p(z) = a e^{-az}, \text{ for } z \geq 0$$

ii. Multiplicative noise models

In this case, the noise is signal dependent and is multiplied to the image. Two commonly discussed multiplicative noise models are:

- Salt-and-Pepper

$$p(z) = P_a \delta(z-a) + P_b \delta(z-b)$$

- Speckle noise

$$a = a_R + j a_I$$

where a_R, a_I are independent Gaussian, with zero mean and named data networking based decision coordination recharging framework has been proposed to optimize the policies of recharging the nodes by sensors under dynamic network conditions. The named data networking is an architecture of network recently proposed for the internet. In NDN, data are addressed by names instead of locations of nodes. Named Data networking has very smart benefits for wireless sensor network. First by sending out new interest packets a mobile receiver can update the routing states in intermediate nodes continuously. Thus, the data can flow through the same reverse paths traversed by the packets. This also guarantees that the last energy information of sensor nodes can reach the sensors in time. Second to scale to larger network size we divide the network into groups, we formed the clusters. The energy information is gathered in aggregated forms.

Thus, we ensure that the data packets are bounded to a group rather than the particular node. Thus, the data can be addressed by the area name.

Restoration model

A. Mean filter

For every pixel in the image, the pixel value is replaced by the mean value of its neighboring pixels ($N \times M$) with a weigh t$w_k = 1/(NM)$. This will result in a smoothing effect in the image.

B. Median filter

For every pixel in the image, the pixel value is replaced by the statistical median of its neighboring pixels ($N \times M$). Although median filter also provides a smoothing effect, it is better in preserving detailed image information, e.g., edges.

C.Homomorphic filter

In the case of multiplicative noise[8], one cannot simply the apply the smooth filter to the observed noisy image $f(x,y)$, as the Fourier transform of the product of two functions is not separable. To overcome this issue, a logarithmic representation of the image model is used instead, i.e.,

$$\ln(f(x,y)) = \ln(o(x,y)) + \ln(n(x,y))$$

The Fourier transform of the logarithmic function is

$$LF(u,v) = LO(u,v) + LN(u,v)$$

D. Wiener Filter

Wiener Filter can be used effectively when the image frequency characteristics and additive noise are known, to at least some degree. Wiener filters are generally applied in the frequency domain. An important advantage of this algorithm is that it removes the additive noise and inverts the blurring Simultaneously. A demerit of the Wiener filters is that they are unable to reconstruct frequency components which have been degraded by noise, but can only suppress them. These filters are comparatively slow since they require working in the frequency domain. The spatially truncated Wiener filter is inferior to the frequency domain version, but may be much faster.

E. Lucy-Richardson algorithm

When the point-spread function PSF (blurring operator) is known we use LR algorithm, but little or no information is available for the noise. The noisy image is restored by the Lucy-Richardson algorithm. The additional optical system such as camera characteristics can be used as input parameters to improve the quality of the image restoration. The algorithm requires a good estimate of the process by which the image is degraded for the accurate restoration. The degradation can be caused in many ways, such as subject movement, out-of-focus lenses, or atmospheric turbulence, and is described by the point spread function (PSF) of the system[3]. The image is assumed to come from a Poisson process and, therefore, is corrupted by signal-dependent noise. There may also be electronic or quantization noise involved in obtaining the image.

PSF optimization model

A.Particle Swarm Optimization

Kennedy and Eberhart in 1995 developed the particle swarm optimization (PSO)[1] algorithm by studying the social and cognitive behavior of ants. The individuals, called particles, are "flown" through a multidimensional search space. Particle swarm requires very simple operations creating a highly efficient algorithm for optimization. Most likely parameters for an optimum solution after selecting PSO, it multiplies these by a uniform random term, which prevents premature convergence.

The movement of the particles is influenced due to two factors using the global particle-to-particle best solution and the local particle’s iteration-to-iteration best solution. As a result of iteration-to-iteration information, the particle stores in its memory the best solution it has visited so far, called “*pbest*”, and experiences an attraction towards this solution as it traverses through the solution search space. As a result of the particle-to-particle information, the particle stores in its memory the best solution visited by any particle, and an attraction towards this solution, called “*gbest*”, results as well. The first, *pbest*, and second, *gbest*, factors are called the cognitive and social components, respectively. After each iteration, the *pbest* and *gbest* are updated for each particle if better. This process continues, iteratively, until either the algorithm achieves the desired result, or it's determined that an acceptable solution cannot be found within computational limits determined by the application. These two factors determine the direction and amount of movement resulting from the particle's velocity. Interestingly, the performance of the two solution points does not affect the direction or amount of motion in traditional PSOs but completely controls the choice of the global and local best solution. A modified PSO, Fitness Distance Ratio PSO, incorporates the solution's performance or fitness into the velocity and does have faster convergence to the globally best answer.

The PSO defines each particle in the D-dimensional[2] space as $X_i = (x_{i1}, x_{i2}, ..., x_{iD})$, where the subscript 'i' represents the particle number and the second subscript is the dimension, number of parameters defining the solution. The memory of the previous best position is represented as $P_i = (p_{i1}, p_{i2}, ..., p_{iD})$, and a velocity for each dimension is independently established as $V_i = (v_{i1}, v_{i2}, ..., v_{iD})$. After each iteration, the velocity term is updated, and the particle is moved with some randomness in the direction of its own best position, *pbest*, and the global best position, *gbest*. This is apparent in the velocity update equation, given by

$$V_{id}^{(t+1)} = \omega \times V_{id}^{(t)} + U[0,1] \times \psi_1 \times (p_{id}^{(t)} - x_{id}^{(t)}) + U[0,1] \times \psi_2 \times (p_{gd}^{(t)} - x_{id}^{(t)}) \quad (1)$$

The position is updated using this velocity and

$$X_{id}^{(t+1)} = X_{id}^{(t)} + V_{id}^{(t+1)} \quad (2)$$

where U [0, 1] samples a uniform random distribution, t is a relative time index, ψ_1 and ψ_2 weights trading off the impact of the local best and global best solutions' on the particle's total velocity.

The particle swarm optimization algorithm is highly efficient in searching complex, continuous solution landscapes. The particle swarm can also be implemented as a parallel algorithm improving its efficiency for real-time applications. The particles can be split up among multiple processors and then the global best solution is shared among the particles.

B.Binary Particle Swarm Optimizer

In the binary valued space, the continuity loses meaning and the interpretation of the fitness function as a function of the position loses its meaning. A binary version of the algorithm [4]transitions particles in a probabilistic space using the velocity of the particle. This has implied that both the binary variables have a probability associated with them. The swarm tries to maximize the probability of a certain binary variable by having a velocity such that its probability is maximized. The algorithm uses the same velocity update equation as in (1) but the values of 'X' are now discrete and binary. For position update, first the velocity is transformed into an [0, 1] interval using the sigmoid function given by

$$S_{id} = sig(V_{id}) = \frac{1}{1+e^{-V_{id}}} \quad (3)$$

where, V_{id} is the velocity of the ith particle's dth dimension. A random number is generated using a uniform distribution which is compared to the value generated from the sigmoid function and a decision is made about the X_{id} in the following manner.

$$X_{id} = u(S_{id} - U[0,1]) \quad (4)$$

u is a unit step function. The decision regarding X_{id} is now probabilistic, implying that higher the value of the V_{id} , higher the value of the S_{id}, making the probability of deciding '1' for X_{id} higher. It should be noted that as $V_{id} \to \infty$, $S_{id} \to 1$ making it impossible X_{id} to return to zero after that point. Until that point, there is some probability of X_{id} returning to zero. Figure 1 shows this property of the binary PSO. The probability of X_{id} =1 increases as V_{id} increases. However, $P(X_{id} = 1)$ is almost equal to 1 for V_{id}>10, but is not exactly equal to 1. This is the key to the design of the discrete binary PSO since particles do not get stuck once they find optima.

***The problem with binary PSO*:**

The main problem about binary PSO [6]is Parameters of binary PSO: in binary PSO small number of V_{max} promotes explorations. for example, if $V_{max} = 4$, then $\text{sig}(V_{max}) = 0.982$ $is\ changed\ to\ bit$ 1.

There is the problem with choosing proper values for inertia weight w. For a value of-1<w<1, $V_{id.}$

Become 0 over time. If w>1 velocity increases over time and $\lim_{n\to\infty} sig(v_{id}(t)) = 1$ so all change to 1.to solve this problem [7]first we need to remove momentum term.according to, as the change in particle position is randomly influence by x_{id}.second approach is to use a random number for w in the range(-1, 1).

Proposed System

A.Image restoration by MBPSO-RL:

After performing de-noise process by the Wiener filter, the de-noise image a(x, y) is given to the image restoration to enhance the image quality**.**Richardson-Lucy algorithm is utilized, in that RL algorithm the restoration performance is improved byMBPSO in RL algorithm by MBPSO.

B.Richardson-Lucy algorithm:

the Richardson-Lucy algorithm is also known as Richardson-Lucy de-convolution, is an iterative procedure for recovery a latent image that has been the blurred by a known PSF.

The LR algorithm can be used effectively when the point-spread function PSF is known.The image is assumed to come from a Poisson process and therefore corrupted by impulse noise.

$$P(A'(x, y)|t) = \prod_z \frac{h(z)*t(z)^{A'(x,y)(z)} e^{-h(z)*t(z)}}{A'(x,y)(z)'} \quad (5)$$

From the equation, a minimized function is L(t^)=-log P(A'(x, y)|t) is obtained for giving the maximum likelihood estimation as

$$L(t') = \sum_z -A(x,y)(z).\log[(n * t^{\wedge})(z)] + (n * k^{\wedge})(z) \quad (6)$$

RL algorithm is given by

$$t_{n+1}(z) = \frac{A(x,y)(z)}{t(z)*h(z)} * h(z) * t^{\wedge}{}_n(z) \quad (7)$$

Algorithm work and stop after finite number of iterations. Finally we have estimate restored image A"(x, y).

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 22 (2015) pp 43077-43081

C.Algorithm for MBPSO-RL.

Step1.initialized the swarm position of a particle and randomly initialize within the hypercube.
Step2.evaluate the performance of each particle using its current position.
Step 3.compare the performance of each individual to its best performance.
Step 4. Compare the performance of each particle to global best particle.
Step5.change the velocity of partitions
Step 6.calculate the velocity of change of bits.
Step 7.generate the random variable in the range(0, 1)
Step 8.go to step 2 and repeat until convergence.

Experimental Result

The image restoration is performed on the noise-free image by using the MBPSO-RL algorithm. This image restoration performance is analyzed and compare with the other optimization algorithm and noise removal filtering methods. The image restoration performance results in term of SDME measure of our MBPSO-RL of different filtering techniques.

Table: Image Restoration Results from proposed MBPSO-RL.

Noise Variance (σ^2)	Images	MBPSO-RL		
		Bilateral Filter	Wiener Filter	Median Filter
0.03	1	21.76	34.53	30.31
	2	37.87	48.00	35.73
	3	21.24	42.91	35.72
	4	31.05	37.73	32.03
0.04	1	15.65	39.10	25.70
	2	24.85	34.62	21.68
	3	18.11	24.69	31.33
	4	21.82	28.95	25.04
0.05	1	21.57	24.35	19.88
	2	23.12	26.27	22.93
	3	26.27	33.74	22.02
	4	22.25	39.61	27.40

Figure 1: Original Image

Figure 2: Noisy Image

Figure 3: Restore image using MBPSO-RL

Conclusion

This paper proposed the MBPSO-RL algorithm to remove the impulse noise from the images. All the filtering techniques have been implemented in MATLAB 7.1 with the Pentium-core2due processor. When we using wiener filter with MBPSO-RL the de-noised image performance is better than the other filter. The impure contaminations include salt and pepper noise.. i.e. better-restored results and another parameter for restoration compared to the existing schemes when impulse noise is considered

References

[1] R. Eberhart, and J. Kennedy, A New Optimizer Using Particles Swarm Theory, Proc. Sixth international Symposium on MicroMachine and Human Science (Nagoya, Japan), IEEE Service Center, Piscataway, NJ, pp. 39-43, 1995.

[2] Aboul-Ella Hassanien, Mariofanna G. Milanova, TomaszG. Smolinski, Ajith Abraham, Computational Intelligence in Solving Bioinformatics Problems: Reviews, Perspectives, and Challenges, zomp. Intel. in Biomed. & Bioinfo rm., SCI, vol. 151, pp. 3-47, 2008.

[3] J. Kennedy, and R. Eberhart, "Particle Swarm Optimization on", IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV, pp. 1942-1948, 1995.

[4] J. Kennedy and R. Eberhart. Swarm Intelligence. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 2001.

[5]. H S Shukla, Narendra Kumar and R P Tripathi. Article: Gaussian Noise Filtering Techniques using New Median Filter. International Journal of Computer Applications 95(12):12-15, June 2014

[6] Kennedy, J.; Eberhart, R.C. "A discrete binary version of the particle swarm algorithm", IEEE international Conference on Systems, Man, and Cybernetics, 1997.

[7] Admore Gota and Zhang Jian Min "Analysis and Comparison on Image Restoration Algorithms UsingMATLAB"International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 12, pp 1350-1360 December-2013

[8] H S Shukla, Narendra Kumar and R P Tripathi. Article: Median Filter based Wavelet Transform for Multilevel Noise. International Journal of Computer Applications, Vol. 107, No. 14, pp 11-14, December 2014.