

Providing a New Approach for Modeling and Parameter Estimation of Probability Density Function of Noise in Digital Images

Hanif Yaghoobi¹, Keivan Maghooli², Alireza Ghahramani Barandagh^{3,*}

^{1,2} Department of Biomedical Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

³ young Researchers and Elite Club, Tabriz branch, Islamic Azad University, Tabriz, Iran

Abstract

The main part of the noise in digital images arises when taking pictures or transmission. There is noise in the images captured by the image sensors of the real world. Noise, based on its causes can have different probability density functions. For example, such a model is called the Poisson distribution function of the random nature of photon arrival process that is consistent with the distribution of pixel values measured. The parameters of the noise probability density function (PDF) can be achieved to some extent the properties of the sensor. But, we need to estimate the parameters for imaging settings. If we assume that the PDF of noise is approximately Gaussian, then we need only to estimate the mean and variance because the Gaussian PDF with only two parameters is determined. In fact, in many cases, PDF of noise is not Gaussian and it has unknown distribution. In this study, we introduce a generalized probability density function for modeling noise in images and propose a method to estimate its parameters. Because the generalized probability density function has multiple parameters, so use common parameter estimation techniques such as derivative method to maximize the likelihood function would be extremely difficult. In this study, we propose the use of evolutionary algorithms for global optimization. The results show that this method accurately estimates the probability density function parameters.

Keywords : Noise Probability Density Function, Parameter Estimation, Global Optimization, Evolutionary Algorithms

1 Introduction

Image noise reduction is one of the most important steps in many image processing applications.

Common approach to noise reduction and restoration is the use of various filters such as morphological filters. These filters are suitable for certain types of noise and adjust to them was not possible to eliminate other noises. Median filter is able to remove isolated noise point apart as well as linear noise apart from each other, while at the edges of the image will not change. In this type of filter is considered to be windows to the $(2k + 1) \times (2k + 1)$ or in the form of a cross and the center of each pixel in the image is shifted. In this way, the values level in a window arranged in order from small to large and middle of Central would be as the amount of gray level. Among the advantages of median filter is the ability of the Impulse noise cancelling. But it does not have the ability to reduce Gaussian noise as well. Another advantage of this filter is that this does not create new value of brightness (gray level) in the image. But it is a major weakness in the displacement of the edges in the image to the size of one or two pixels. Also, if the window size of the filter increases, not only noise will be removed but also the small levels of noise-free resolution of the original image.

Another of the most famous filters for noise removal and restoration of the image is Wiener filter. Wiener filter, removes the noise statistically. In this filter, it is assumed that the image noise is a Gaussian distribution with mean zero. A Wiener filter act to reduce the mean

square error between the original image and the image is approximated by the noisy image (Denoised Image). Equation 1 shows this in the following way:

$$|\mathcal{E}|^2 = \|I_{original} - I_{denoised}\|^2 \quad (1)$$

Wiener filter is capable of removing Gaussian noise [1]. But the nature of the noise may not always Gaussian. For example, such a model is called the Poisson distribution function of the random nature of photon arrival process that is consistent with the distribution of pixel values measured. If the noise is unknown nature, it must first be determined probability density function. After that, the probability density function parameters to be estimated. There are many ways to estimate the noise probability density function parameters including maximum likelihood method. This method requires the use of derivative values in order to find optimal parameters, so that to obtain maximum likelihood functions. Derivative methods is quite difficult for a number of parameters, larger than 3. But since this method, an optimization problem, you can use evolutionary algorithms to find the optimal values of the parameters, Such as genetic algorithms. In this paper we present a method that is defined in a general probability density function of several parameters for noise. Then, we estimate the parameters of the density function using Imperialist Competitive Algorithm.

2 Generalized Distribution Function

Generalized probability distribution has been developed as a general formula for some probability distribution known as Weibull, exponential, Riley, gamma, log-normal, Pareto, Maxwell, generalized Laplace and so on. Apart from the above mentioned about the probability distribution, the new family of distributions can be achieved. The statistical properties viz. moments and moment generating function have been worked out for generalized probability distribution (G.P.D). In order to define G.P.D., consider the following form of confluent hyper-geometric function of variable Z as:

$$f(Z; \alpha, \lambda, p, A, C) = \alpha^{\lambda+1} (Z)^\lambda e^{-pZ} M(A, C, -\alpha Z) \tag{2}$$

Where α, A, p and A are constants; $C \neq 0, -1, -2$, and $0 \leq Z \leq \infty$.

$M(A, C, Z)$ is confluent hyper geometric function [2]. A random variable Z follows a G.P.D, if its probability density function (pdf) is given by:

$$f(Z; \alpha, \lambda, p, A, C) = K^{-1} \alpha^{\lambda+1} (Z)^\lambda e^{-pZ} M(A, C, -\alpha Z) \tag{3}$$

Where, the constant:

$$K = \left(\frac{\alpha}{p}\right)^{\lambda+1} \Gamma(\lambda+1) F(A, \lambda+1; C; -\frac{\alpha}{p})$$

$\alpha + p > 0; \lambda > -1$

(4)

$F(a, b, d; -x)$ is Gauss hyper geometric function [2].

Some of the well-known widely used probability distributions both from non-exponential as well as from exponential family can derive from Generalized

Probability Distribution. For example: If random variable $Z = X$ is a G.P.D, then X follows uniform distribution for $A = 0, \lambda = 0$, and if $A = a, C = a + 1, \lambda=0, \alpha=1$, then (3) reduces to incomplete gamma distribution [2]. Also if $Z = X\beta$ is a G.P.D, then X follows weibull distribution for $\lambda=0$.

If you consider the general probability distribution as a probability distribution of noise, we can estimate the parameters and we find the nearest known distribution function for the noise. Thus, it becomes clear statistical characteristics of noise and noise can be eliminated easily with a restoration filter.

3 Parameter Estimation

There are many ways to estimate the probability density function parameters. One of the common methods is the use of maximum likelihood. Suppose there is a sample x_1, x_2, \dots, x_n of n independent and identically distributed observations, coming from a distribution with an unknown probability density function $f_0(\cdot)$. It is however surmised that the function f_0 belongs to a certain family of distributions $\{f(\cdot | \theta), \theta \in \Theta\}$ (where θ is a vector of parameters for this family), called the parametric model, so that $f_0 = f(\cdot | \theta_0)$. The value θ_0 is unknown and is referred to as the true value of the parameter vector. It is desirable to find an estimator $\hat{\theta}$ which would be as close to the true value θ_0 as possible. Either or both the observed variables x_i and the parameter θ can be vectors.

To use the method of maximum likelihood, one first specifies the joint density function for all observations. For

an independent and identically distributed sample, this joint density function is

$$f(x_1, x_2, \dots, x_n | \theta) = f(x_1 | \theta) \times f(x_2 | \theta) \times \dots \times f(x_n | \theta)$$

Now we look at this function from a different perspective by considering the observed values x_1, x_2, \dots, x_n to be fixed "parameters" of this function, whereas θ will be the function's variable and allowed to vary freely; this function will be called the likelihood:

$$L(\theta; x_1, x_2, \dots, x_n) = f(x_1 | \theta) \times f(x_2 | \theta) \times \dots \times f(x_n | \theta) = \prod_{i=1}^n f(x_i | \theta) \quad (6)$$

Denotes a separation between the two input arguments: θ and the vector-valued input x_1, x_2, \dots, x_n . In practice it is often more convenient to work with the logarithm of the likelihood function, called the log-likelihood:

$$\text{Ln}(L(\theta; x_1, x_2, \dots, x_n)) = \sum_{i=1}^n \text{Ln} f(x_i | \theta) \quad (7)$$

Or the average log-likelihood:

$$\hat{L} = \frac{1}{n} \text{Ln}(L) \quad (8)$$

The hat over L indicates that it is akin to some estimator. Indeed, \hat{L} estimates the expected log-likelihood of a single observation in the model.

The method of maximum likelihood estimates θ_0 by finding a value of θ that

maximizes $\hat{L}(\theta; x)$. This method of estimation defines a maximum-likelihood estimator (MLE) of θ_0 ...

$$\{\hat{\theta}_{mle}\} \subseteq \{\arg \max_{\theta \in \Theta} (\hat{L}(\theta; x_1, \dots, x_n))\} \quad (9)$$

If any maximum exists. An MLE estimate is the same regardless of whether we maximize the likelihood or the log-likelihood function, since log is a strictly monotonically increasing function. For many models, a maximum likelihood estimator can be found as an explicit function of the observed data x_1, \dots, x_n . For many other models, however, no closed-form solution to the maximization problem is known or available, and an MLE has to be found numerically using optimization methods. For some problems, there may be multiple estimates that maximize the likelihood. For other problems, no maximum likelihood estimate exists (meaning that the log-likelihood function increases without attaining the supremum value) [3].

In this paper we have used Imperialist Competitive Algorithm to maximize the likelihood and obtain parameters. This algorithm is a global optimization algorithm, which we explain below.

4 Imperialist Competitive Algorithm (ICA)

Imperialist Competitive Algorithm (ICA) is a method that focuses on the field of evolutionary computation algorithms to find the optimal solution of optimization problems. This algorithm

provides an approach for solving optimization problems by modeling the process of social - political evolution. The application of this algorithm is in the category of population-based optimization algorithms such as Genetic Algorithms, Particle Swarm Optimization, Ant Colony Optimization, and Simulated Annealing Metals Algorithm and so on. Like all algorithms under this category, Imperialist Competitive Algorithm consists of an initial set of possible answers. The initial solutions are known as "chromosome" in Genetic Algorithms, "particle" in the Particle Swarm Algorithm and the "Country" in Imperialist Competitive Algorithm. The initial solutions of ICA algorithms can be improved gradually and provides the appropriate solution for the optimization problem (best imperial). The main pillars of this algorithm consist of Assimilation, Imperialist Competition and Revolution. Like other evolutionary algorithms, this algorithm starts with a random initial population, each of them called a "country." Some of the best elements of the population (in issues relating to minimizing the ones that have the lowest cost function) are selected as imperialists. The remaining populations are also considered as a colony. Colonies move towards imperialists by Assimilation operator. This movement is shown in Fig.1 in which a colony moves toward the imperialist by x units.

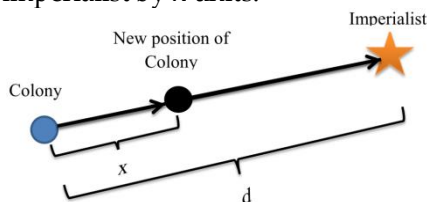
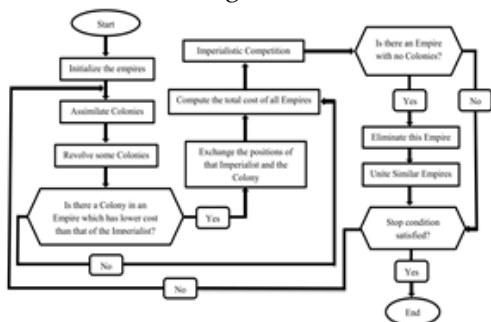


Figure 1

The new position of the colony is shown in a darker color. The direction of the movement is the vector from the colony to the imperialist. In this figure x is a random variable with uniform (or any proper) distribution. Then $x \sim U(0, \beta \cdot d)$, where β is a number greater than 1 and d is the distance between the colony and the imperialist state. $\beta > 1$ causes the colonies to get closer to the imperialist state from both sides. Revolution causes a country to suddenly change its socio-political characteristics. That is, instead of being assimilated by an imperialist, the colony randomly changes its position in the socio-political axis. The revolution rate in the algorithm indicates the percentage of colonies in each colony which will randomly change their position. While moving toward the imperialist, a colony might reach to a position with lower cost than the imperialist. In this case, the imperialist and the colony change their positions. Then the algorithm will continue by the imperialist in the new position and the colonies will be assimilated by the imperialist in its new position. The total power of the empire depends on both its constituent parts, namely, the Imperial (the core) and its colonies.

In math mode, this dependence has been modeled by defining imperial power as the total power of the imperialist country besides a percentage of the average of its colonial power. With the formation of the early empires, imperial competition between them begins. If each empire cannot be succeed in imperialist competition, and increase their power (or at least reduce its influence to prevent), will be removed from the colonial competition. So the survival of an empire

depends on its ability to attract rival colonial empires and bring them under control. As a result, in the imperialist rivalry, the power of larger empires gradually increases, and weaker empires are removed. Empires will be forced to develop their own colonies to increase their power [4]. Fig. 2 shows the flowchart of the algorithm.



Figur.2

5 Parameter Estimation of Noise in Digital Images using ICA and Generalized Probability Density Function

Our proposed methodology of this study is to introduce the generalized probability density function as noise probability density function with unknown parameters and to estimate this parameters using ICA. If the noise in a digital image is cause of destruction, we can achieve noise density function and its parameters to calculate the statistical characteristics such as mean and variance of noise. Usually, when using restoration filters such as Weiner filter, it is necessary to know the total variance of noise. For example, Adaptive Wiener filter needs to estimates the local mean and variance around each pixel.

$$\mu = \frac{1}{NM} \sum_{x,y \in \eta} f(x, y)$$

and

$$\sigma^2 = \frac{1}{NM} \sum_{x,y \in \eta} f(x, y)^2 - \mu^2$$

(10)

Where $f(x,y)$ is grey level of pixel and η is the N-by-M local neighborhood of each pixel in the image. Adaptive Wiener filter then creates a pixelwise Wiener filter using these estimates,

$$g(x, y) = \mu + \frac{\sigma^2 - v^2}{\sigma^2} (f(x, y) - \mu)$$

(11)

Where v^2 is the noise variance [5]. So we can estimate noise parameters to calculate the noise variance. The implementation of our proposed method is as follows:

- 1) Select part of an image that is almost without changes.
- 2) Consider gray level of this part of the image as noise data.
- 3) Estimate the parameters of Generalized Probability Density function using this data and ICA.
- 4) Calculate the noise variance and other statistical properties using these parameters.
- 5) Restoration of the image using the appropriate Restoration Filter. (Such as adaptive Weiner filter)

6 Results

For example, we have an image with a noise that has a Weibull probability density function is destroyed to obtain the image noise. Fig.3 and Fig.4 show the original and noisy image, respectively.



Figure 3

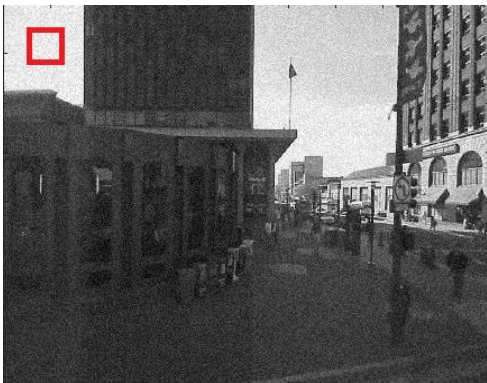


Figure 4 .With 200 pixels from the left corner of the original image that has been marked in Figure 4, the noise parameters are estimated by ICA. Figure 5 shows a histogram of the area.

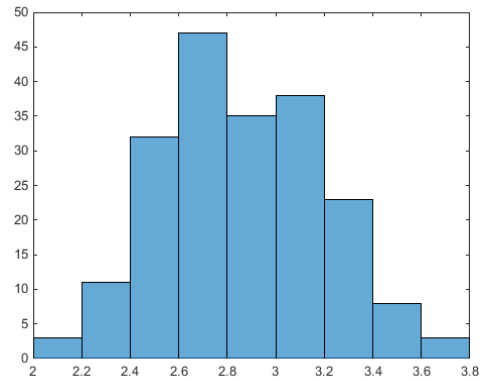


Figure 5

In this algorithm, we have considered the cost function equal to negative logarithmic likelihood function. So:

$$\begin{aligned}
 \text{Cost - function} &= -\text{Ln}(L(\theta; x_1, x_2, \dots, x_n)) = \\
 &= -\sum_{i=1}^n \text{Ln}f(x_i | \theta)
 \end{aligned}
 \tag{12}$$

Therefore, the algorithm finds the values of the parameters so that the cost function would be minimum. Fig.6 shows the values of the cost function at 20 iterations the algorithm.

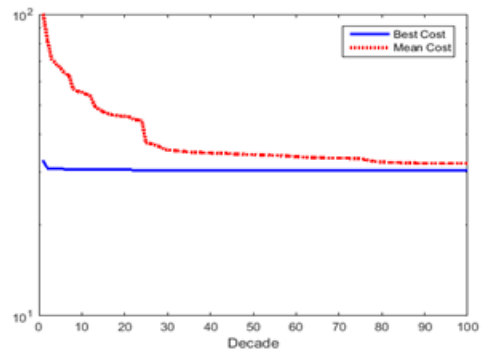


Figure 6

The values obtained for the parameters show that the noise has Weibull nature. These values were obtained respectively as follows: $\alpha =1, \beta =3, \gamma =2$

Therefore, we calculate the variance (dispersion) of noise as follows [6]:

Where Γ is Gamma function. The value obtained for the variance is equal to $\sigma^2=26.85$.

$$\sigma = \alpha^2 \left[\Gamma\left(1 + \frac{2}{\beta}\right) - \Gamma^2\left(1 + \frac{1}{\beta}\right) \right] \quad (13)$$

Figure 7 shows the reconstructed image using Adaptive Weiner Filter cited the amount of variance.



Figure 7

7 Summary

In this paper, we've introduced a new approach to identify the noise and estimate probability density function parameters. The results show that evolutionary algorithms can be easily and accurately estimate these values. Then we calculated the statistical characteristics of the noise, such as variance.

REFERENCES

Rafael C. Gonzalez, Richard E. Woods: Digital Image Processing (3rd Edition), Prentice-Hall, Inc. Upper Saddle River, NJ, USA ©2006, ISBN:013168728X.

S.K. Agarwal, in: On Generalized Probability Distribution and Generalized Mixture Distribution, Jour. Ind. Soc. Ag. Statistics, 50(2), 1997; /92-/98.

Information on http://en.wikipedia.org/wiki/Maximum_likelihood.

Atashpaz-Gargari, E.; Lucas, C (2007). "Imperialist Competitive Algorithm: An algorithm for optimization inspired by imperialistic competition". IEEE Congress on Evolutionary Computation 7. pp. 4661– 4666.

Lim, Jae S., Two-Dimensional Signal and Image Processing, Englewood Cliffs, NJ, Prentice Hall, 1990, p. 548, equations 9.26, 9.27, and 9.29.

Ștefan Cristian CIUCU, Mihai REBENCIUC, Ionuț Alexandru PETRE: Addenta to Weibull Distribution in MATLAB