Paper: ASAT-14-204-IP

14th International Conference on
AEROSPACE SCIENCES & AVIATION TECHNOLOGY,
ASAT - 14 – May 24 - 26, 2011, Email: asat@mtc.edu.eg
Military Technical College, Kobry Elkobbah, Cairo, Egypt
Tel: +(202) 24025292 –24036138, Fax: +(202) 22621908

Linear Motion Deblurring from Single Images Using Genetic Algorithms

S. El-Regaily*, H. El-Messiry†, M. Abd El-Aziz‡ and M. Roushdy§

Abstract: One of the key problems of restoring a degraded image from motion blur is the estimation of the unknown linear blur filter from a single blurred input image. Several algorithms have been proposed utilizing image intensity or gradient information. In this paper, we propose an algorithm for restoring the motion-blurred image using Genetic Algorithms. Genetic Algorithms are applied in science and engineering as adaptive algorithms for optimizing practical problems. Certain classes of problem are particularly suited and being tackled effectively with Genetic Algorithm based approach. The direction and the length of the motion blur Point Spread Function (PSF) are used as the parameters of the algorithm. The method assumes a uniform linear camera blur over the image. Experiments on a wide data set of standard images degraded with different directions and blur lengths demonstrate the efficiency of the proposed approach in small blur lengths compared to other algorithms, with a better average Root Mean Squared Error of two values. Experiments also show how ringing artifacts affect the behavior of the algorithm in large blur lengths.

Keywords: Camera Shake, Blind Image Deconvolution, Genetic Algorithms, Ringing Artifacts.

1. Introduction

One of the most common artifacts in digital photography is motion blur caused by the relative motion between the camera and the scene during image exposure time. The problem is particularly apparent in low light conditions when the exposure time can often be in the region of several seconds, and the inevitable result is that many of our snapshots come out blurry and disappointing. Many photographs capture ephemeral moments that cannot be recaptured under controlled conditions or repeated with different camera settings. If camera shake occurs in the image for any reason, then that moment is lost. One solution that reduces the degree of blur is to capture images using shorter exposure intervals. This, however, increases the amount of noise in the image [1].

* Demonstrator, Basic Science Department, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt, E-mail: salsabil.amin@gmail.com,
† Assistant professor, Computer Science Department, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt, E-mail: hmessiry@msn.com,
‡ Assistant professor, Basic Science Department, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt, E-mail: mhaziz@aucegypt.edu,
§ Professor, Dean of Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt, E-mail: miroushdy@hotmail.com.

1.1 Motion Blur Model

Motion blur is usually modeled as a linear convolution of the image intensities, with a blurring kernel that describes the camera motion during exposure, also known as the Point Spread Function (PSF), that describes the amount of time light from a single point in the scene exposes each (x, y) pixel position in the image detector. Motion blur is modeled as

$$B=I \otimes F+ n, \quad (1)$$

where B represents the input blurred image, I: the sharp original image, F: the PSF or the blurring kernel. n represents the sensor noise that is often neglected in most of the algorithms. $\otimes$ represents the convolution operator. To restore the original image I, we need to apply the inverse operation of the convolution, which is the deconvolution between B and F.

Image deconvolution is the process of recovering the unknown image from its blurred version, given a blurring kernel [2]. In most situations, however, the blurring kernel is unknown as well, and the task also requires the estimation of the underlying blurring kernel. Such a process is usually referred to as *blind deconvolution*, which is a problem with a long history in the image and signal processing literature. In the most basic formulation, the problem is under constrained: there are simply more unknowns (the original image and the blur kernel) than measurements (the observed image). Hence, all practical solutions must make strong prior assumptions about the blur kernel, about the image to be recovered, or both.

Motion blur is mainly categorized into two types: linear motion blur and non-linear motion blur. In this paper, we will handle the linear motion blur. To remove linear motion blur we only need to estimate two parameters: the direction and the length of the blur. And then, from these parameters we formulate the PSF as mentioned in [3]:

$$F(x,y) = \begin{cases} \frac{1}{L}, & if \ \sqrt{x^2 + y^2} \leq \frac{L}{2}, \ \frac{x}{2} = -tan(\emptyset), \\ 0, & otherwise \end{cases} \quad (2)$$

1.2 Related Work

Motion blur estimation methods have been greatly advanced recently. Research about blind deconvolution given a single image usually concentrates at cases in which the image is uniformly blurred. A summary and analysis of many deconvolution algorithms can be found in [2]. Levin [4] proposed an algorithm that relies on the observation that the statistics of derivative filters in images are significantly changed by blur and model the expected derivatives distributions as a function of the width of the blur kernel. Fergus [5] proposed a variational Bayesian approach using an assumption on the statistical property of the image gradient distribution to approximate the unblurred image. Moghaddam and Jamzad [3] proposed an algorithm that estimates linear blur parameters using radon transform and fuzzy sets. The angle of motion blur is estimated using three different approaches in [6], the first employs the cepstrum, the second a Gaussian filter, and the third the radon transform.

Related work in genetic algorithms, as in [7], used constrained genetic algorithm for image restoration in. They used the image pixels as the parameters, and assumed that the kernel is known in advance. But this algorithm is computationally expensive as it works on the estimated image as a whole while our work relies on the blur kernel. Moghaddam and Jamzad [8] used genetic algorithms and the wiener filter to estimate the out of focus blur in the frequency domain. Nassar et al. [9] used the genetic algorithms for designing and optimization of an ion-exchanged polarization converter in a similar way to ours.

In this paper, the genetic algorithm is used to estimate the direction and the length of the linear motion blur. The rest of the paper is organized as follows: in section 2 a detailed explanation of the Genetic Algorithm is included. We present our approach to solving linear motion blur using genetic algorithms in section 3, then, in section 4 the goal function and its role in genetic algorithms is explained with the results shown. Section 5 discusses the ringing artifacts and how they affect our results. In section 6 the implementation details and experimental results are included. Finally, in section 7 we present conclusion and future work.

2. Genetic Algorithms

Genetic algorithms (GAs) are now widely applied in science and engineering as adaptive algorithms for optimizing practical problems. Certain classes of problem are particularly suited and being tackled effectively with GA based approach. Next we will present the main operations of the GA [10].

2.1 Creating the First Generation

A first generation consisting of a certain number of entities is found by randomly assigning each parameter one particular value from the set of all possible values for that parameter. Once all entities are determined, the goal function value is calculated for each entity in the generation. Based on the value of the goal function for a certain entity, the probability (or the fitness value) that an entity will be transferred to the next generation is calculated for that entity.

2.2 Reproduction

After calculating the goal function for each entity, entities of the new population are selected by using a roulette selection scheme based on the probabilities. In this scheme, a roulette wheel with slot sized according to fitness is used. We spin the roulette wheel, each time we select one individual for the new population. Obviously, some individuals would be selected more than once as they have the largest probabilities. The best individual gets more copies, the average ones stay even and the worst ones die off. To implement this idea we use a random number generator. See figure 1.

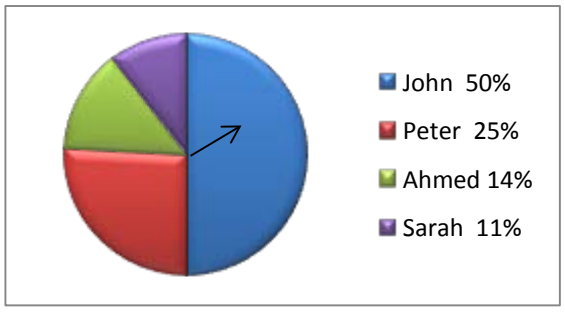

Figure 1 represents the idea of selection to the new generation using a random number generator. If we have a spinning arrow (representing the random number generated), then it is most probable that it will point at John's part most of the times, because he has the biggest portion in the chart.

2.3 Crossover

When two fit entities exchange their parameters, several scenarios might happen. In the best scenario, the best parameters meet together while the worst parameters meet together. This will lead to a more fit new entity, which will take us one further step towards the optimum parameter combination. The other entity with the worst parameters will have no effect on the algorithm since it will be excluded when reproducing the next generation. In other scenarios, some good parameters will meet some bad parameters resulting in almost the same fitness. Such cases do not improve the goal function, but they preserve the good parameters together.

2.4 Mutation

The algorithm explained until now will go towards the optimum, which is required. However, if the goal function has several local extremes, the algorithm would locate only one of them, which might not necessarily be the absolute extreme. In order to scan new regions away from a local extreme, one parameter of an entity is randomly chosen to be given any random value from the set of its allowed values, which is called mutation. This will result in a more or a less fit entity. If it is more fit, the entity will dominate the next generation and redirect the entire algorithm to a completely new path away from the local extreme. On the other hand, if the entity is less fit, it will be excluded when reproducing the next generation. The probability of mutation is the probability of changing a parameter's value to any other value. It is the same for all parameters and all entities.

2.5 Elitist Selection

Since the reproduction of a new generation is a random operation, it might happen that the fittest entity is not included in the new generation. In order to avoid this situation, the fittest entity is exceptionally guaranteed to be transferred at least once to the next generation without being affected by normal reproduction, crossover, or mutation. By preserving good solutions, we can avoid losing some excellent solutions. This operation is called elitist selection.

3. Methodology

The algorithm works as follows: A first generation consisting of 10 to 20 different entities is set. The maximum size of population is defined in the beginning of the algorithm. Each entity, which contains the two parameters, the direction and the length of the blur, is found by randomly assigning any possible value to each of these parameters.

Possible values for the parameters: the direction Φ ranges from 0° to 179° and the length L ranges from 5 to 30 pixels.

For each entity in the first generation, during the goal function call, a kernel with the specified parameters is created. Then deconvolution is applied to the blurred image using the known deconvolution algorithm, Lucy-Richardson [11], to get a candidate restored image. The error is computed, which is the second norm between a reference image and each restored image for each entity, and then the error is transformed to a probability assigned to each entity, such that, the entity with the minimum error has the highest probability. Then, to create the next generation, we ensure that the entities with the maximum probabilities will appear more frequently and the entities with the minimum probabilities will be excluded.

After creating the next generation, two main processes are applied: Crossover and Mutation. Crossover is done by exchanging the parameters (direction and length) of the blur kernels of two random entities within the generation. Mutation is done by exchanging the value of one of these parameters with a random value from the possible set of values. Mutation is

controlled with the probability of mutation, which is set heuristically. The optimum value according to our experiments is 0.08.

The same procedure is repeated for several generations until it is recognized that no better images can be generated, and the algorithm converges till only one entity dominates the whole generation. The process is usually terminated after a fixed number of generations or by reaching a certain value. In our algorithm it was always reaching a certain number of generations since no previous expectations for the minimum value are available for our search. Figure 2 shows the main operations of the GA.

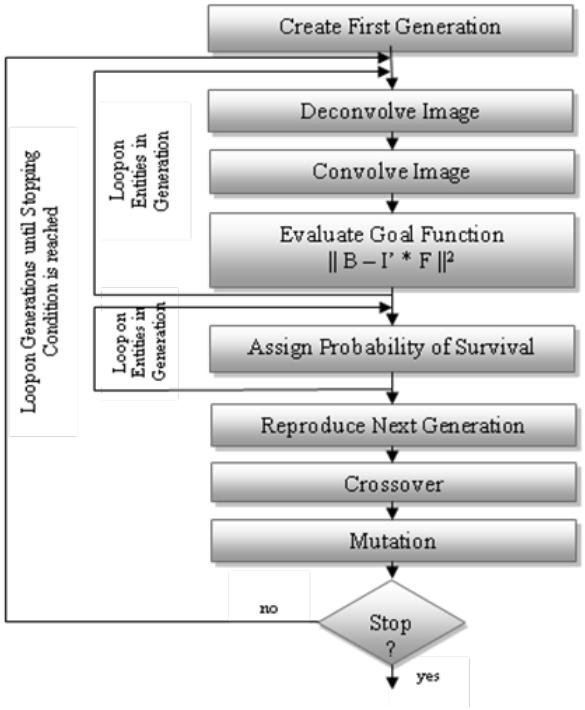

Figure 2: The main operations of the GA.

4. The Goal Function

The goal function is one of the most important parts of the GA, as it assigns a value to each entity, which is converted to a probability, such that the entity with a minimum value (error) has the highest probability among the other entities. The entities with the highest probability are more likely to be chosen in the next generation, and the entities with the lowest probabilities are excluded. At the final generations only one or two entities will dominate the whole generation. The goal function used is:

$$Error = \left\| B - \hat{I} * F \right\|^2 \quad (3)$$

where B represents the blurred image, $\hat{I}$ the original image estimate restored using the current entity and F represents the PSF generated with the direction and the length of the current entity. The Error is computed using the second norm between the blurred image, which is the only input we have, and the blurred version of the estimated image. Only a small patch of the image is used, to reduce the computational time and to reduce the error due to the borders effect. Converting errors into probabilities that act as the roulette selection scheme is summarized in figure 3.

The GA works on a random basis, so different results could be obtained each run, depending on the first generation values. So, as a final solution, the GA is developed to restart itself many times after convergence. After a few iterations, the GA converges to one entity, for example, the same entity appears 14 times out of 16, the size of the population. So, the next generation is initialized randomly from the parameter set similarly like first generation. The restored image with best entity is saved, and all the variables reset to their initial values. Each time the GA converges to an entity, a new random generation is created and the restored image is saved. Finally, at the end, the final image is the average of all the best restored images as shown in figure 4.

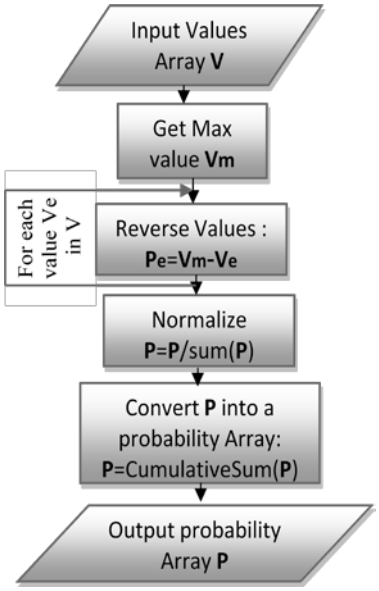

Figure 3: Converting the goal function values (errors) into probabilities.

Figure 4: Left: The barb image blurred with direction 45° and length 5. Right: The final result of GA with direction 44° and length 5

5. Ringing Artifacts

As a result of using the Lucy-Richardson deconvolution algorithm [11], ringing artifacts occur around strong edges due to noise in the blurred image or wrong estimated blur parameters according to the well-known Gibbs phenomenon in the frequency domain [12]. Ringing is aligned along both sides of edges at the distance and angle equal to those of the motion blur. If the number of iterations is increased in the Lucy-Richardson algorithm, the

estimated image will be sharper and clearer, but the ringing effect increases. Also the larger the length of the blur, the more ringing appears. These effects change the estimated images.

Therefore, even if we have an image restored with the correct parameters, it might have more ringing artifacts than other images. That is why the algorithm works perfectly with small blur lengths such as 5 and 7 pixels, as they have less ringing artifacts. But with blur lengths larger than 9, the algorithm tends to choose the correct direction but chooses smaller blur length, as images restored with smaller blur lengths have less ringing than larger blur lengths, even if this length was the correct one; see figure 5.

A ringing removal algorithm [13] is used to reduce the ringing artifacts each iteration, before the goal function computations, but it doesn't affect the results, as it only reduces the ringing and doesn't remove it totally. The ringing is reduced in both the larger length image and the smaller length image, so the algorithm chooses the smaller one as before.

Figure 5: The top row represents the original bird image and the bird image blurred with direction 165° and length 15, respectively. The left bottom represents the bird image restored with the correct parameters, direction 165° and length 15. Notice the ringing effect. The image on the right bottom is the bird image chosen in the algorithm with direction 164° and length 7.

6. Implementation and Experimental Results

The algorithm is implemented using MATLAB 7. A large database is created for linear motion blurred images. 16 different standard images like baboon, cameraman, lena, boat, etc., of size 256×256 are blurred synthetically with different lengths and directions within the range producing a set of 60 different blurred images. The GA produces different but close results for the same input image, so we apply the GA on each image three or four times then take the average as an output. There is a problem that affects the efficiency of the algorithm: the borders effect. To avoid this problem, we had to cut all the affected pixels from the blurred image and used the edgetaper function in MATLAB. However, this happens only with synthetic blur, not with the real blurred images.

For small blur lengths, the algorithm works perfectly with small errors in the direction and the blur length, but for large blurs, the final results are satisfactory in terms of the motion direction, but not as good in the blur length, because of the ringing effect. See Table 1 for results.

The running time depends on the number of generations and the size of population. It varies between 1 minute and 3 minutes, which is faster than any other GA. The reason is that we work on two parameters only, and the kernel may be repeated many times through the generations, so deconvolution is done only once at the beginning and only if there's a change due to crossover or mutation. The usual time taken by a GA could be hours or even days to converge to an entity. By experiment, we need only about 30 generations for convergence and 20 entities per generation is a proper size of population, which takes about 1.8 minutes. See Figure 6.

Table 1: The blur lengths and the algorithm behavior toward each length in terms of the estimated length, the average error of estimated direction, and the average RMSE between the original image and the estimated images

Blur Length	Estimated length	Average direction error	Average RMSE
5	90% chooses 5	5°	6.39
7	70% chooses 7	4.1°	6.7787
9	40% chooses 9 60% chooses 7	5.8°	6.5443
Larger than 9	90% chooses smaller lengths	5.5°	Larger than 7

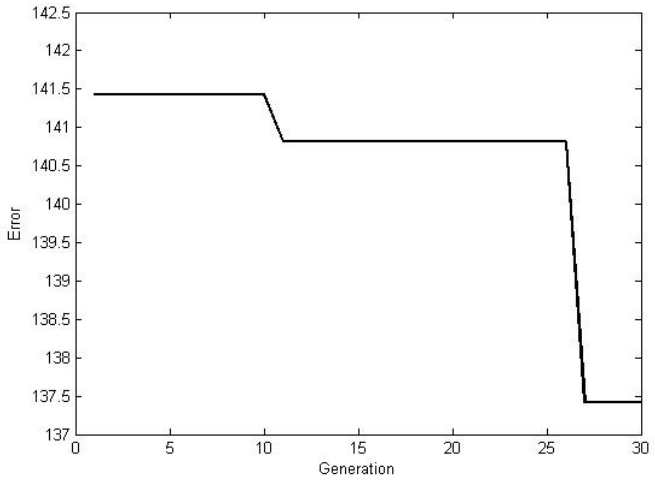

Figure 6: Goal function values (errors) of the best entities over generations describe the convergence of the GA

Table 2 shows some experimental results of our algorithm compared to our implementation of Moghaddam and Jamzad algorithm [3]. The error measurement used is the Root Mean Squared Error (RMSE) between the original image and the restored image.

Table 2: The estimated direction and blur length with the Root Mean Square Errors for our algorithm compared with the algorithm in [3]

	Original direction and length		Our Algorithm			Moghaddam and Jamzad algorithm [3]		
Test images:	L	Θ	L	θ	RMSE	L	θ	RMSE
Girl	5	45º	5	42º	5.1819	7	46º	6.3018
Lena	7	110º	7	111º	6.1265	6.8	136º	7.5846
Pepper	11	60º	5	68º	5.9837	9	46º	5.8932
San	15	55º	7	65º	7.8988	14	46º	7.7567

Some results from the table are shown in Figures 7 and 8. Figure 9 represents a comparison of the RMSE of both algorithms with increase in the blur length, by taking the average RMSE of all the resultant images of our database.

Figure 7: The top row represents the original girl image and the blurred girl image with direction 45º and length 5 respectively. The left bottom image is the result of Moghaddam [3] with direction 46º and length 7, and the right bottom is our result with direction 42 º and length 5.

Figure 8: The top row represents the original san image, and the san image blurred with direction 55º and large blur length 15. The left bottom is the result of Moghaddam [3] with direction 46º and length 14, and the right bottom is our result with direction 65º and length 7.

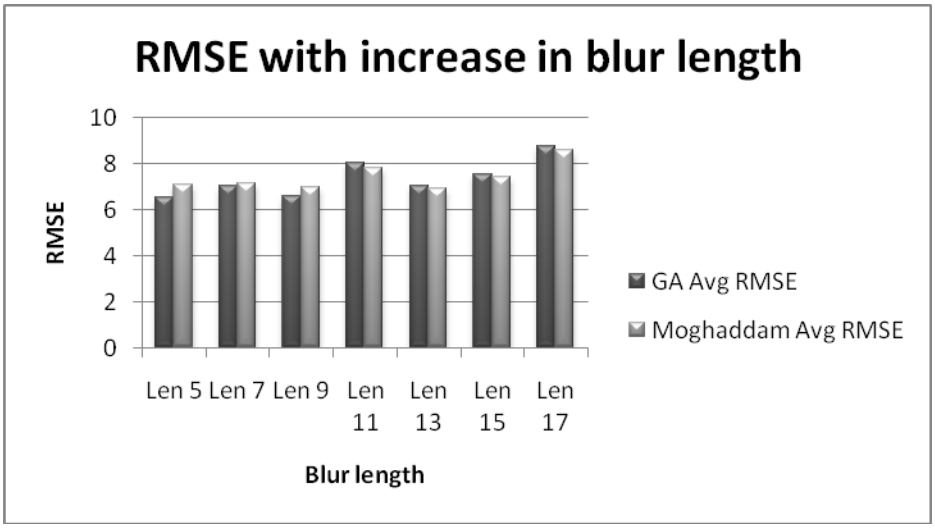

Figure 9: A comparison of RMSE of restored images of GA and the algorithm in [3]

7. Conclusion and Future Work

In this paper we propose an approach to solve linear motion blur in a single image, using GAs. First we estimate the blur parameters, the direction and the length of the blur. Then we use the Lucy-Richardson deconvolution algorithm to restore the image. The GA starts on a random basis then selects the best parameters to restore the image with the minimum error that corresponds to the highest probability.

The ringing artifacts around strong edges due to deconvolution affect the behavior of the algorithm in large blur lengths, and force the algorithm to choose smaller lengths with less ringing. However, the algorithm works perfectly for small blur lengths.

For future work, the aim is to expand the GA to solve non-linear motion blur. Also, many techniques can be applied to enhance the resultant image. The fitness function could be improved by adding other regularization parameters or constraints.

8. References

[1] JEVUSKA, Daniel Cunningham, s0198594, “Image Motion Deblurring” www.jevuska.com/topic/ImageMotionDeblurring.htm.

[2] Kundur, D., Hatzinakos. D., “Blind Image Deconvolution”, IEEE Signal Processing Magazine, 1996.

[3] Moghaddam, M. E., Jamzad, M., “Linear Motion Blur Parameter Estimation in Noisy Images Using Fuzzy Sets and Power Spectrum”, EURASIP Journal on Advances in Signal Processing Volume, Article ID 68985, 8 pages doi:10.1155/2007/68985, 2007.

[4] Levin, A., “Blind Motion Deblurring Using Image Statistics”. In NIPS, 2006.

[5] Fergus, R., Singh, B., Hertzmann, A., Rowies, S. T., Freeman, W., “Removing Camera Shake from a Single Photograph”. ACM Transactions on Graphics 25, 787–794, 2006.

[6] Krahmer, F., Lin, Y. B. McAdoo, K. Ott, J. Wang, D. Widemannk, and B. Wohlberg, “Blind Image Deconvolution: Motion Blur Estimation", Technical Report, Institute of Mathematics and its Applications, University of Minnesota, 2006.

[7] Chen, Y., Nakao Z., Iguchi, M., “Image Restoration by a Constrained Genetic Algorithm”, Bulletin of the Faculty of Engineering University of the Ryukyus No.51, p.67 -71,1996.

[8] Moghaddam, M. E., Jamzad, M., “Out of Focus Blur Estimation Using Genetic Algorithm”, 15th International Conference on Systems, Signals and Image Processing (IWSSIP), pp: 417-420, Bratislava, Slovak, June 2008.

[9] Nassar, I. M., El-Refaei, H., Khalil, D., Omar, O. A., “The Design and Optimization of an Ion-Exchanged Polarization Converter using a Genetic Algorithm”, IEEE Photonics Technology Letters, Vol. 19, No. 16, August 15, 2007.

[10] M. Melanie, "An Introduction to Genetic Algorithms", A Bradford book, the MIT Press, Campridge, Massachusetts, London, England, 1999.

[11] Richardson, L., “Bayesian-Based Iterative Method of Image Restoration”, Journal of Astronomy 79, pages 745-754, 1974.

[12] Atreas N. and Karanikas C., “Book on Gibbs Phenomenon”, Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece, pp. 54-124.

[13] Chalkov, S., Meshalkina, N., Kim, C., “Post-Processing Algorithm for Reducing Ringing Artefacts in Deblurred Images”, 23rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), 2008.