

**Military Technical College
Kobry El-Kobbah,
Cairo, Egypt**



**8th International Conference
on Electrical Engineering
ICEENG 2012**

Non-linear motion deblurring in single images using genetic algorithms

By

Salsabil El-Regaily * H. El-Messiry * M. Abd El-Aziz * M. I. Roushdy*

Abstract:

One of the key problems of restoring a degraded image from motion blur is the estimation of the unknown non-linear blur filter from a single input blurred image. Many blind deconvolution methods typically assume frequency-domain constraints on images, simplified parametric forms for the motion path during camera shake or use multiple input images with specific characteristics. This paper proposes an algorithm for removing non-linear motion blur from a single input blurred image using Genetic Algorithms (GAs), by finding the proper parameters and goal function. Also recent research in natural image statistics is exploited, which shows that photographs of natural scenes typically obey heavy-tailed distribution. The Point Spread Function entries are used as the parameters of the GA. Experiments on a wide data set of standard images degraded with different kernels of different sizes demonstrate the efficiency of the proposed approach especially in small blur lengths compared to other algorithms with reasonable running times for a GA.

Keywords:

Camera Shake, Blind Image Deconvolution, Genetic Algorithm, Non-Linear Motion Blur

* Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt

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].

The main contribution of this paper is to apply GA to motion deblurring by adapting the parameters of the algorithm and finding a proper goal function that takes advantage of the heavy-tailed distribution of image gradients.

1.1. Motion Blur Model:

Motion blur is usually modeled as linear convolution of the image intensities, with the 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.

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

Where B represents the input blurred image, I the sharp original image, F the PSF or the blurring kernel and 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. Non-linear kernels have no specific behavior. It could be in any shape (rotational, affine, non-parametric) simple or complicated, and of any size.

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.

1.2. Related Work:

Motion blur estimation methods have been greatly advanced recently. Some algorithms use only a single input blurred image to estimate the blur kernel. Fergus *et al.* [3] proposed a variational Bayesian approach using an assumption on the statistical property of the image gradient distribution to approximate the unblurred image. Shan *et al.* [4] incorporated spatial parameters to enforce natural image statistics using a local ringing suppression. J. Jia [5] used transparency maps to get cues for object motion to recover blur kernels by performing blind-deconvolution on the alpha matte, with a prior on the alpha-matte. Chai *et al.* [6] proposed an algorithm that removes motion blur from a single image by formulating the blind blurring as a new joint optimization problem, which simultaneously maximizes the sparsity of the blur kernel and the sparsity of the clear image under certain suitable redundant tight frame systems. Qiu [7] proposed a procedure for estimating non-parametric PSFs, on condition that the image contains at least one line edge.

There had been many trials in image restoration in general using GAs, but not particularly in motion deblurring. Chen *et al.* [8] used constrained GA for image restoration. He used the image pixels itself 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. Moghaddam *et al.* [9] used GAs and the wiener filter to estimate the out of focus blur in the frequency domain. A parallel GA is developed by Chen *et al.* [10]. The algorithm divides a large population into smaller subpopulations and executes the main loop of the traditional GA on each processor with its own subpopulation in parallel. Qiu *et al.* [11] restored images based on the GA mixed with kalman filtering. Nassar *et al.* [12] used the GAs for designing and optimization of an ion-exchanged polarization converter in a similar way to ours.

This paper is organized as follows: in section 2 the main processes of GA are explained. In section 3 we present our approach to solving non-linear motion blur using GAs. Then in section 4 the goal function used is explained with a detailed description of the heavy-tailed gradients distribution of natural images, and how it is included in the goal function. In section 5 the implementation details and experimental results are presented. Finally conclusion and future work are included in section 6.

2. 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 [13].

In GA, A first generation consisting of a certain number of entities is found by randomly assigning to each parameter one particular value from the set of all possible values for that parameter. Then the goal function value is calculated for each entity in the generation, and transformed into probabilities. Entities of the new population are selected by using a roulette selection scheme based on their probabilities. *Crossover* is applied on each two fit entities to exchange their parameters, which might take us one further step towards the optimum parameter combination. 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. This operation is called *Mutation*. 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. This process is called *Elitist Selection*. For further details see our previous work in [14].

3. Non-Linear Motion Deblurring using GAs:

The non-linear kernels unlike the linear kernels do not have any specific parameters, can follow any convoluted path and can be of any size. The algorithm works as follows: A first generation consisting of 10 to 25 different entities is set. The maximum size of population is defined in the beginning of the algorithm. Each entity represents a candidate kernel of a size specified at the beginning of the algorithm by the user as the maximum size of the kernel. The parameters of each entity are the kernel pixels. Possible values of the kernel pixels are {0 1 2 4} where each number represents a different speed for the blur motion. Zeros are repeated many times among the other values to insure sparsity of the kernel.

A kernel can be initialized with two different ways:

- 1 – Assign random values from the set of possible values to random locations in the kernel.
 - 2 – Divide $180/\text{size of population}$ to get a set of different directions, and then initialize the kernels in the generation as linear kernels in these different directions.
- Or kernels can be initialized using both ways. The first four kernels in the generations are initialized with the basic four directions: horizontal, vertical, left diagonal and right

diagonal, then rest of the kernels are initialized randomly. The mixed initialization technique is used in the proposed algorithm.

To reduce the huge computational time needed by GA, only a small patch of the blurred image is used. Deconvolution is done using the Lucy-Richardson algorithm to get the estimated images, whose fitness values are calculated through the goal function and transformed into probabilities of survival. For crossover and mutation the proposed algorithm exploits the features of the matrix representation of blur kernels, so each matrix entry can be crossed over with the same entry of another randomly chosen entity (kernel) and can be mutated by randomly exchanging it with a random value from the set of possible values. Crossover and mutation are applied according to probabilities of crossover and mutation, which are defined at the beginning of the algorithm. See the structure of the proposed GA in Figure (2).

The same procedure is repeated for the next generations until it is recognized that there are 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 the proposed algorithm it was always reaching a certain number of generations since no previous expectations for the minimum value are available for our search.

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 the minimum error has the highest probability among the other entities. The entities with the highest probabilities 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 is flexible as any number of terms can be added, and searching for the effective terms is considered the real challenge in GA as they affect the behavior of the whole algorithm. In the proposed algorithm two terms are chosen: a *data relevance term* and a *gradient histogram term*.

4.1. Gradients Histogram:

It is found that image gradients have a heavy-tailed distribution [3], when plotting the image gradients against log number of pixels. Recent research in natural image statistics has shown that, although images of real-world scenes vary greatly in their absolute color distributions, they obey *heavy-tailed* distributions in their gradients. The distribution of gradients has most of its mass on small values but gives significantly more probability

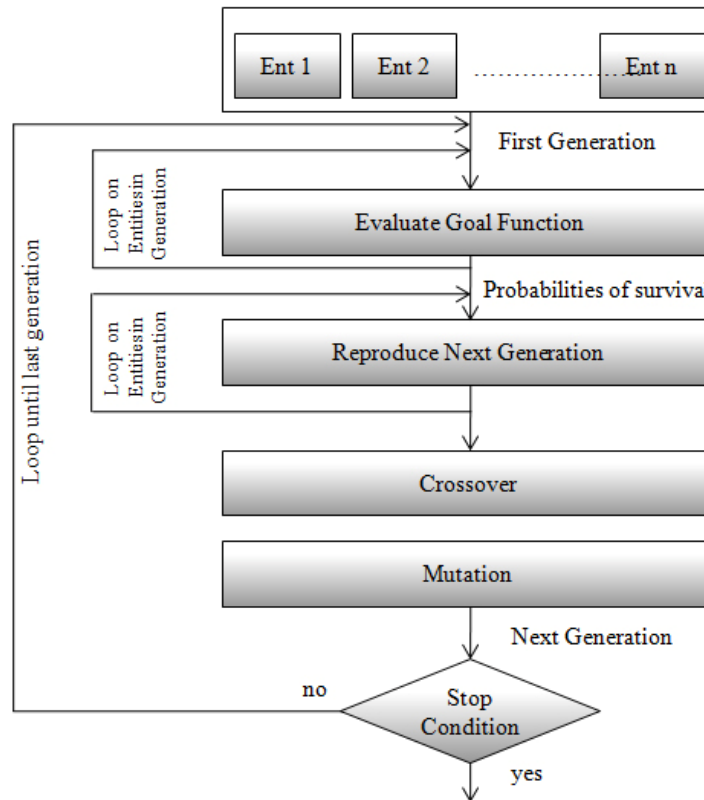


Figure (2): Flowchart of the proposed GA

to large values than a Gaussian distribution. This corresponds to the intuition that images often contain large sections of constant intensity or gentle intensity gradient interrupted by occasional large changes at edges or occlusion boundaries. For example, Figure (3) shows a sharp image and a histogram of its gradient magnitudes.

The distribution shows that the image contains primarily small or zero gradients, but a few gradients have large magnitudes. The original image has more edges and details, thus has a wider range of distribution, while the blurred image has less edges and details, thus the distribution is centered around zero. We aim to expand the gradient histogram of the blurred image using a new goal function, to approach the gradient histogram of the original image. See Figure (4) for comparisons.

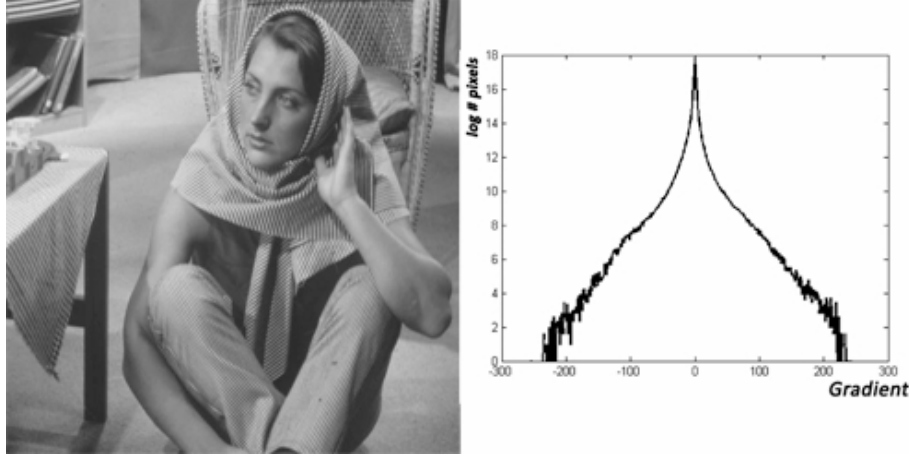


Figure (3): The heavy-tailed distribution of an original sharp image

The final used goal function is:

$$\text{Error} = \alpha \cdot \text{RMSE} (B - Y * F) + \beta \cdot \text{RMSE} (R_g - Y_g), \quad (2)$$

Where:

- B is the blurred Image,
- Y is the estimated image,
- F is the estimated PSF,
- R_g is the gradient histogram of a reference non-blurred image,
- Y_g is the gradient histogram of the estimated image Y ,
- α and β represent weights that are used to balance the values of the two terms.

The first term in the goal function is a “*data relevance term*”, that relates the characteristics of the restored image to the characteristics of the blurred image, where y is the result of deconvolving B with F using Lucy-Richardson algorithm. The expression $Y * F$ represents the blurred estimate of the restored image Y . The other term is the “*gradients histogram term*” that searches for the image with the sharpest edges and the nearest range of image gradients to a reference image gradients.

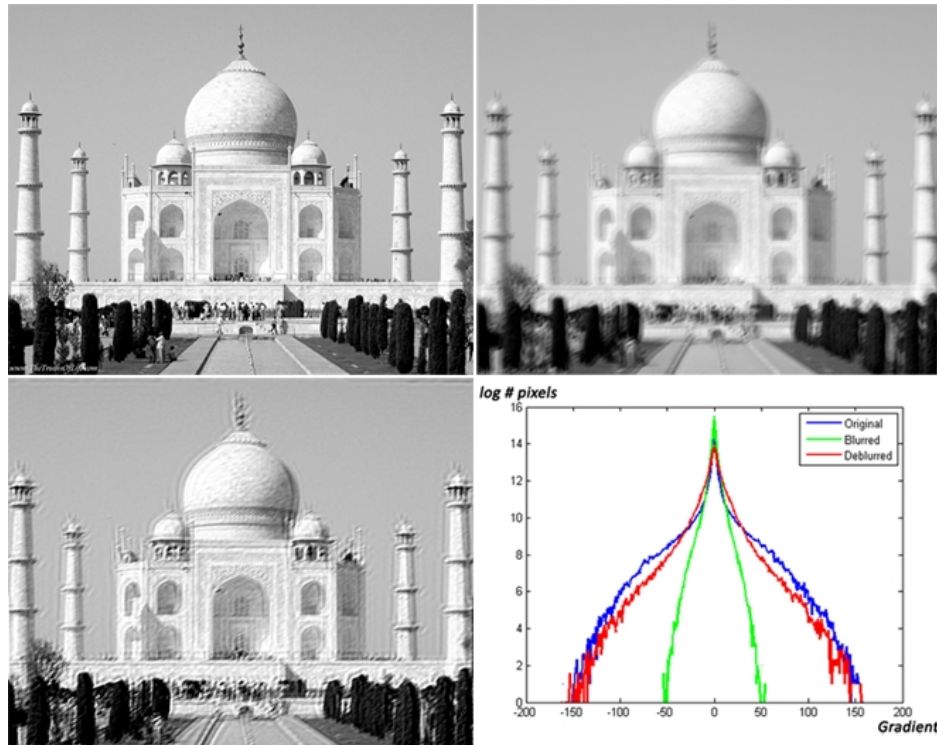


Figure (4): The top row represents the original taj image and the blurred taj image, respectively. The bottom left image represents the restored image using the new gradient histogram term. The bottom right graph represents a comparison between gradient histograms of the original, blurred and restored images

5. Experimental Results and Comparisons:

The proposed algorithm is implemented using MATLAB 2011a. A large database is created of non-linearly motion blurred images. 33 different standard images like baboon, cameraman, lena, boat, etc., of different sizes that varies from 256 to 1024, are blurred synthetically with different generated kernels in different shapes. If the input image is of size 256×256 , then the whole image is used during running time. For larger size images, a small patch of the image defined by the user is used. A reference image is chosen from the images database that has a gradient histogram in the average range and can be used as R_g in (2). For small sized images, the gradient histogram of the “San” image of size 256 is used. For larger images, the gradients of a small patch extracted from the 1024×1024 goldhill image are used. From experiment, probability of mutation is set as 0.09.

Running time depends on the number of generations, the size of population and the size of the blurring kernel, as deconvolution is performed on each entity in the generation. It

varies between 10 minutes and 60 minutes. By experiment, only about 30 generations are needed for convergence, and 20 is a proper size of population, which takes about 20 minutes. See results in Figure (4) and Figure (5). Figure (6) represents the convergence of the GA through generations.

To compare the proposed algorithm to another algorithm, the algorithm proposed by Fergus et al. [3] is chosen for many reasons. It has many similarities with the GA. Both algorithms are iterative algorithms that handle non-linear motion blur, in similar running times and exploit the heavy-tailed property of image gradients histogram. Also, both algorithms use Lucy-Richardson algorithm as the final deconvolution technique to get the final estimated image with the estimated kernel. There may be many other algorithms that restore images in a better and faster way than Fergus [3], but they do not have same common points with the GA.

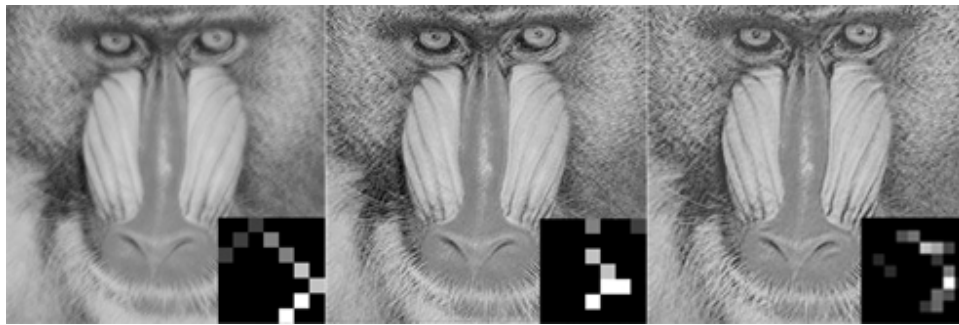


Figure (4): Left: The blurred baboon image and the blur kernel. Middle: GA result with and the final estimated kernel. Right: The result of Fergus [3] and the final estimated kernel



Figure (5): Left: The blurred lena image and the blur kernel. Middle: GA result and the final estimated kernel. Right: The result of Fergus [3] and the final estimated kernel

Fergus [3] estimates the blur kernel by forming an objective function using Baye's rule,

and then minimizes it using a variational Bayesian approach. In the objective function, the image prior exploits the heavy tailed gradients distribution and models it by a mixture of zero-mean Gaussians. The kernel prior is represented as a mixture of exponentials that forces sparsity and positivity and control kernel values. The algorithm runs in a multi-scale approach starting with a 3×3 kernel to avoid local minima.

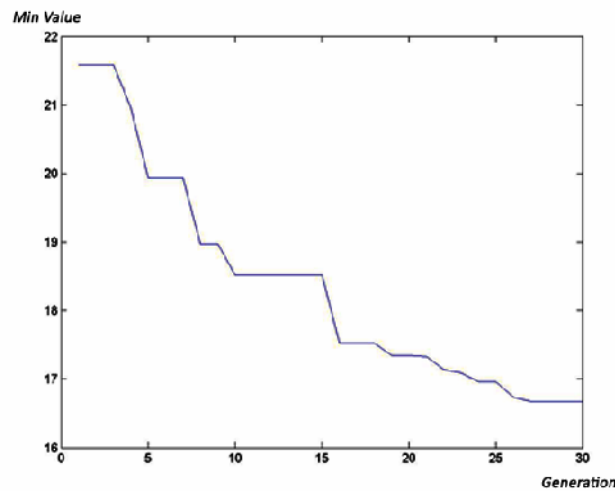


Figure (6): Goal function minimum values (errors) of the best entities over generations describe the convergence of the GA

A comparison of the Root Mean Squared Error (RMSE) and the running time between the proposed GA and Fergus [3] is presented in Figure (7) and Figure (8), based on running the two algorithms on 33 differently blurred images from the generated database.

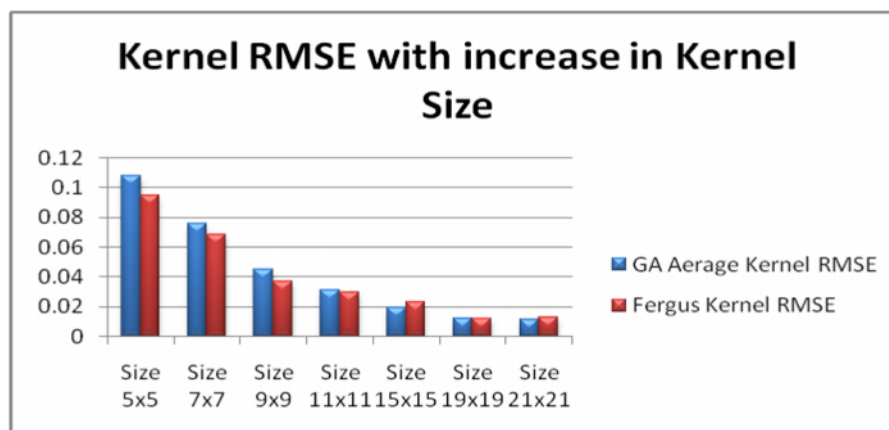


Figure (7): Change of estimated kernels RMSE with increase in kernel size of the GA and Fergus [3] algorithm

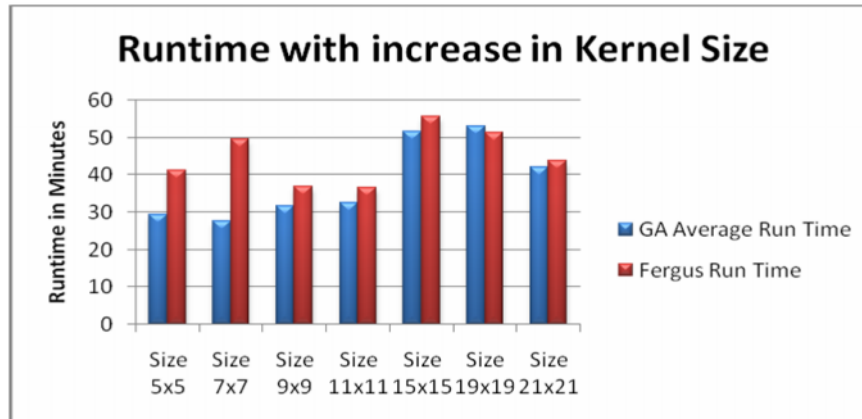


Figure (8): Change of running times with increase in kernel Size of the GA and Fergus [3] algorithm

From these figures we conclude that RMSE of GA approaches the results of Fergus [3] with small differences, especially in large kernel sizes where results of the two algorithms are so close to each other. The GA has better RMSE in 30% of the tested images. But as an advantage to the GA, the algorithm runs faster than Fergus [3] in about 65% of the tested images. See Figure (9) for more results.



Figure (9): Top left: The 504 x504 blurred boat image blurred with a 9x9 kernel. Top right: The result of Jia [4]. Bottom left: The result of Cho [15]. Bottom right: GA result.

6. Conclusions and Future Work:

This paper proposed an approach to non-linear motion deblurring in a single input blurred image, using GAs. It was taken into consideration that the algorithm should always be flexible in order to add any number of constraints to the goal function. The Lucy-Richardson deconvolution algorithm is used to find candidate restored images within the algorithm. The GA starts on a random basis then converges to the best entity that restores the blurred image with the minimum error that corresponds to the highest probability. After many trials, a gradient histogram difference term is added to the goal function that searches for the image with the sharpest edges and the nearest range of image gradients to the original image. Results are comparable to other algorithms, and the running time is reasonable compared to other complex motion deblurring algorithms or to other GAs, that could take hours to converge to the right solution. The fact that the proposed GA doesn't depend on the input image size is an advantage to the algorithm, as it only works on a small selected patch not on the whole image.

For future work, the aim is to improve the goal function, by adding other regularization terms, better constraints, or new image quality measures that searches for the best estimated image through the generations. Also, many techniques can be applied to enhance the resultant image, like reducing ringing artifacts around the strong edges. Different GA techniques can be investigated, like adding other crossover or mutation techniques, or including a line process to the algorithm, to save more time and find better results. Larger running time is also a big disadvantage of the GA, so we will try to run the GA on Graphics Processing Unit (GPU) to accelerate the process.

References:

- [1] JEVUSKA, Daniel Cunningham, s0198594, "Image Motion Deblurring" www.jevuska.com/topic/ImageMotionDeblurring.htm.
- [2] D. Kundur, D. Hatzinakos, *Blind image deconvolution*, IEEE Signal Processing Magazine, 1996.
- [3] R. Fergus, B. Singh, A. Hertzmann, S. T. Rowies and W. Freeman, *Removing camera shake from a single photograph*, ACM Transactions on Graphics 25, 787–794, 2006.
- [4] Q. Shan, J. Jia, and A. Agarawala, *High quality motion Deblurring from a single image*, ACM transitions on graphics, SIGGRAPH, 2008.
- [5] J. Jia, *Single image motion deblurring using transparency*, in CVPR, 1–8, 2007.

- [6] J. Cai, H. Ji, C. Liu and Z. Shen, *Blind motion deblurring from a single image using sparse approximation*, IEEE Conference on Computer Vision and Pattern Recognition, 104-111, 2009.
- [7] P. Qiu, *A nonparametric procedure for blind image Deblurring*, Elsevier, Computational Statistics and Data Analysis 52, 4828-4841, 2008.
- [8] Y. Chen, Z. Nakao and M. Iguchi, *Image Restoration by a Constrained Genetic Algorithm*, Bulletin of the Faculty of Engineering University of the Ryukyus No.51, 67 -71,1996.
- [9] M. E. Moghaddam and M. Jamzad, *Out of focus blur estimation using genetic algorithm*, 15th International Conference on Systems, Signals and Image Processing (IWSSIP), June, 417-420, Bratislava, Slovak, 2008.
- [10] Y. Chen, Z. Nakao, X. Fang and S. Tamura, *A parallel genetic algorithm for image restoration*, IEEE proceedings of ICPR, 1996.
- [11] F. Qiu, Y. Wang, M. Jiang and D. Yuan, *Adaptive image restoration based on the genetic algorithm and kalman filtering*, Communications in Computer and Information Science, Volume 2, Part 18, 742-750, 2007.
- [12] I. M. Nassar, H. El-Refaei, D. Khalil and O. A. Omar, *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.
- [13] M. Melanie, *An introduction to genetic algorithms*, A Bradford book, the MIT Press, Cambridge, Massachusetts, London, England, 1999.
- [14] S. A. El-Regaily, H. El-Messiry, M. H. Abd El-Aziz, and M. I. Roushdy, *Linear Motion Deblurring from Single Images Using Genetic Algorithms*, 14th International Conference on Aerospace Sciences & Aviation Technology (ASAT – 14), 204-IP, 2011.
- [15] S. Cho and S. Lee, *Fast Motion Deblurring*, ACM Transactions on Graphics (SIGGRAPH ASIA 2009), vol. 28, no. 5, article no. 145, December 2009.