Faster PCA for Face Detection Using Cross Correlation in the Frequency Domain

Hazem M. EL-Bakry
Faculty of Computer Science & Information Systems, Mansoura University, EGYPT

Summary
In this paper, a new technique for fast painting with different colors is presented. The idea of painting relies on applying masks with different colors to the background. Fast painting is achieved by applying these masks in the frequency domain instead of spatial (time) domain. New colors can be generated automatically as a result from the cross correlation operation. This idea was applied successfully for faster specific data (face, object, pattern, and code) detection using neural algorithms. Here, instead of performing cross correlation between the input input data (e.g., image, or a stream of sequential data) and the weights of neural networks, the cross correlation is performed between the colored masks and the background. Furthermore, this approach is developed to reduce the computation steps required by the painting operation. The principle of divide and conquer strategy is applied through background decomposition. Each background is divided into small in size sub-backgrounds and then each sub-background is processed separately by using a single faster painting algorithm. Moreover, the fastest painting is achieved by using parallel processing techniques to paint the resulting sub-backgrounds using the same number of faster painting algorithms. In contrast to using only faster painting algorithm, the speed up ratio is increased with the size of the background when using faster painting algorithm and background decomposition. Simulation results show that painting in the frequency domain is faster than that in the spatial domain.

Key words:
Fast Painting, Cross Correlation, Frequency Domain, Parallel Processing.

1. Introduction

Painting in real time is an important issue for many different applications [57]. In this article, a new idea for fast painting is presented. Painting with colored masks is equivalent to cross correlation with these masks and background. It was proved that performing cross correlation in the frequency domain is faster that time domain [1-54]. Fast cross correlation has been applied successfully for many different applications [1-53]. A faster searching algorithm for face/object detection using neural networks and fast cross correlation was presented in [6,27,29,30,32,33,34,35,38,42,48,50,51,52,53]. Faster sub-image detection was achieved using fast cross correlation [16,20,25,36,39]. Very fast iris detection using fast cross correlation was introduced in [31,41,43,44,45,46,47,49]. A general fast pattern detection using fast cross correlation was presented in [2,7,8,9,10,14,15,21,22,23,24]. Furthermore, a real time fast code detection for communication applications using fast cross correlation was introduced in [18,37,40]. In addition, a new time delay artificial neural network was invented using fast cross correlation as introduced in [1,5,12,19]. As well as, an interesting mathematical application by using fast cross correlation was presented [17,26,28]. A quick algorithm for fast Principle Component Analysis (PCA), applied in many artificial intelligent algorithms, using fast cross correlation was introduced in [3]. Moreover, an Internet application for fast searching on web pages using fast cross correlation was presented in [4]. Finally, high speed data processing using fast cross correlation was introduced in [12].

Here, fast painting is implemented in the frequency domain using different masks. Each mask can have a single color or mixed colors. In addition, new colors can be generated after performing the cross correlation operation. In section 2, a new fast painting algorithm using cross correlation is presented. A faster painting algorithm that reduces the number of the required computation steps through background decomposition is presented in section 3. Accelerating the new approach using parallel processing techniques is also introduced.

2. A New Theory for Fast Painting with Colored Masks Using Cross Correlation in the Frequency Domain

In this new approach, we are interested in increasing the speed of the painting process. By the words “fast painting” we mean reducing the number of computation steps required to perform the painting process. In the proposed

Manuscript revised January 2006.

model, each sub-background in the background is painted with one or more colors. At each position in the background each sub-background is dot multiplied by a mask (window) of a certain color or mixed colors, which has the same size as the sub-background. This multiplication was done in the spatial (time) domain and is performed here in the frequency domain.

In this section, a fast algorithm for automatic painting with colored masks based on two dimensional cross correlations that take place between the background and the sliding window is described. Such window is represented by a mask contains a certain color or mixed colors. The convolution theorem in mathematical analysis says that a convolution of *f* with *h* is identical to the result of the following steps: let *F* and *H* be the results of the Fourier transformation of f and h in the frequency domain. Multiply *F* and *H* in the frequency domain point by point and then transform this product into spatial domain via the inverse Fourier transform [1-54]. As a result, these cross correlations can be represented by a product in the frequency domain. Thus, by using cross correlation in the frequency domain a speed up in an order of magnitude can be achieved during the painting process.

In the painting process, a sub-background *X* of size nxn is extracted from the background, which has a size *P*x*T*. Let W_i be the vector of colored masks (sliding window) that will be applied to this sub-background. This vector has a size of nxn and can be represented as nxn matrix. The resulted new generated colors *h(i)* can be calculated as follows:

$$h_i = \sum_{j=1}^{n} \sum_{k=1}^{n} W_i(j,k)X(j,k) \qquad (1)$$

Equation (1) represents the output of applying each mask for a particular sub-background *X*. It can be computed for the whole background Ψ as follows:

$$h_i(u,v)= \sum_{j=-n/2}^{n/2} \sum_{k=-n/2}^{n/2} W_i(j,k)\,\Psi(u+j,v+k) \qquad (2)$$

Equation (2) represents a cross correlation operation. Given any two functions *f* and *g*, their cross correlation can be obtained by [1-53]:

$$f(x,y)\otimes g(x,y)= \left(\sum_{m=-\infty}^{\infty} \sum_{z=-\infty}^{\infty} f(x+n,y+n)g(m,z) \right) \qquad (3)$$

Therefore, Eq. (2) can be written as follows:

$$h_i = \Psi \otimes W_i \qquad (4)$$

where h_i is the resulted new generated colors matrix *(i)* and $h_i(u,v)$ is the new generated color *(i)* when the colored mask (sliding window) is located at position *(u,v)* in the background Ψ and $(u,v)\in[P-m+1,T-n+1]$.
Now, the above cross correlation can be expressed in terms of the Fourier Transform:

$$\Psi \otimes W_i = F^{-1}\left(F(\Psi)\bullet F^*\left(W_i\right)\right) \qquad (5)$$

(*) means the conjugate of the *FFT* for the weight matrix.
The complexity of cross correlation in the frequency domain can be analyzed as follows:
1. For a background of *NxN* (elements) area, the *2D-FFT* requires a number equal to $N^2\log_2N^2$ of complex computation steps. Also, the same number of complex computation steps is required for computing the *2D-FFT* of the mask matrix. Assume that we have a total number of q masks. Then the number of computation steps required to compute *2D-FFT* for all masks will be ($q\,N^2\log_2N^2$).
2. For each mask, the inverse *2D-FFT* is computed. So, *q* backward and *(q+1)* forward transforms have to be computed. Therefore, for a processed background, the total number of the *2D-FFT* to compute is $(2q+1)N^2\log_2N^2$.
3 The background and the colored masks should be multiplied in the frequency domain. Therefore, a number of complex computation steps equal to qN^2 should be added.
4. The number of computation steps required by the fast cross correlation is complex and must be converted into a real version. It is known that the two dimensional Fast Fourier Transform requires $(N^2/2)\log_2N^2$ complex multiplications and $N^2\log_2N^2$ complex additions [55,56]. Every complex multiplication is realized by six real floating point operations and every complex addition is implemented by two real floating point operations. So, the total number of computation steps required to obtain the *2D-FFT* of an *NxN* background is:

$$\rho=6((N^2/2)\log_2N^2) + 2(N^2\log_2N^2) \qquad (7)$$

which may be simplified to:

$$\rho=5N^2\log_2N^2 \qquad (8)$$

Performing complex dot product in the frequency domain also requires $6qN^2$ real operations.
5. In order to perform cross correlation in the frequency domain, the mask matrix must have the same size as the background. Therefore, a number of zeros = (N^2-n^2) must be added to the weight matrix. This requires a total real number of computation steps = $q(N^2-n^2)$ for all masks. Moreover, after computing the *2D-FFT* for the mask matrix, the conjugate of this matrix must be obtained. So, a real number of computation steps $=qN^2$ should be added in order to obtain the conjugate of the mask matrix for all masks. Also, a number of real computation steps equal to *N* is required to create butterflies complex numbers

$(e^{-jk(2\Pi n/N)})$, where $0<K<L$. These $(N/2)$ complex numbers are multiplied by the elements of the background or by previous complex numbers during the computation of the *2D-FFT*. To create a complex number requires two real floating point operations. So, the total number of computation steps required for the fast cross correlation becomes:

$$\sigma=(2q+1)(5N^2 log_2 N^2) +6qN^2+q(N^2-n^2)+qN^2 +N \quad (9)$$

which can be reformulated as:

$$\sigma=(2q+1)(5N^2 log_2 N^2) +q(8N^2-n^2) +N \quad (10)$$

6. Using a mask (sliding window) of size nxn for the same background of *NxN* elements, $q(2n^2-1)(N-n+1)^2$ computation steps are required when using conventional cross correlation for automatic color generation. The theoretical speed up factor η can be evaluated as follows:

$$\eta=\frac{q(2n^2-1)(N-n+1)^2}{(2q+1)(5N^2 log_2 N^2)+q(8N^2-n^2)+N} \quad (11)$$

The theoretical speed up ratio (Eq. (11)) with different sizes of background and different in size mask matrices is listed in Table 1. Practical speed up ratio for manipulating backgrounds of different sizes and different in size weight matrices is listed in Table 2 using 700 MHz processor and *MATLAB ver 5.3*. An interesting property with fast cross correlation is that the number of computation steps does not depend on either the size of the sub-background or the size of the weight matrix (n). The effect of (n) on the number of computation steps required for fast cross correlation is very small and can be ignored. This is incontrast to conventional cross correlation in which the number of computation steps is increased with the size of both the sub-background and the mask matrix (n).

In practical implementation, the multiplication process consumes more time than the addition one. The effect of the number of multiplications required for conventional cross correlation in the speed up ratio (Eq. 11) is more than the number of of multiplication steps required by the fast cross correlation. In order to clear this, the following equation (η_m) describes relation between the number of multiplication steps required by conventional and fast cross correlation:

$$\eta_m=\frac{qn^2(N-n+1)^2}{(2q+1)(3N^2 log_2 N^2)+6qN^2} \quad (12)$$

Eq. (12) proves that the effect of the number of multiplication steps in case of conventional cross correlation is more than fast cross correlation and this is the reason why the practical speed up ratio is larger than theoretical one.

For general fast cross correlation the speed up ratio (η_g) is in the following form:

$$\eta_g = \frac{q(2n^2-1)N^2}{(2q+1)(5(N+\tau)^2 log_2 (N+\tau)^2)+q(8(N+\tau)^2 - n^2)+(N+\tau)} \quad (13)$$

where τ is a small number depends on the size of the weight matrix. General cross correlation means that the process starts from the first element in the input matrix. The theoretical speed up ratio for general fast cross correlation (η_g) defined by Eq. 13 is shown in Table 3. Compared with *MATLAB* cross correlation function *(xcorr2),* experimental results show that the proposed algorithm is faster than this function as shown in Table 4.

3. A New Faster Algorithm for Automatic Painting and New Color Generation Based on Background Decomposition

In this section, a new faster algorithm for automatic painting and new color generation is presented. The number of computation steps required for fast cross correlation with different background sizes is listed in Table 5. From this table, we may notice that as the background size is increased, the number of computation steps required by fast cross correlation is much increased. For example, the number of computation steps required for a background of size (50x50 elements) is much less than that needed for a background of size (100x100 elements). Also, the number of computation steps required for a background of size (500x500 elements) is much less than that needed for a background of size (1000x1000 elements). As a result, for example, if a background of size (100x100 elements) is decomposed into 4 sub-backgrounds of size (50x50 elements) and each sub-background is processed separately, then a speed up factor for the painting process can be achieved. The number of computation steps required by this faster cross correlation to process a background after decomposition can be calculated as follows:

1. Assume that the size of the background is (NxN elements).

2. Such background is decomposed into α (*LxL* elements) sub-backgrounds. So, α can be computed as:

$$\alpha=(N/L)^2 \quad (14)$$

3. Assume that, the number of computation steps required for processing one (*LxL* elements) sub-background is β. So, the total number of computation steps (T) required for processing these sub-backgrounds resulting after the decomposition process is:

$$T = \alpha \beta \tag{15}$$

The speed up ratio in this case *(η_d)* can be computed as follows:

$$\eta_d = \frac{q(2n^2 - 1)(N - n + 1)^2}{(q(\alpha + 1) + \alpha)(5N_s^2 \log_2 N_s^2) + \alpha q(8N_s^2 - n^2) + N_s^2 + \Delta} \tag{16}$$

where,

Ns: is the size of each small sub-background.

Δ: is a small number of computation steps required to obtain the results at the boundaries between sub-backgrounds and depends on the size of the sub-background.

To paint an area of size 20x20 elements in a background of any size by using faster cross correlation after background decomposition into sub-backgrounds, the optimal size of these sub-backgrounds must be computed. From Table 5, we may conclude that, the most suitable size for the sub-background which requires the smallest number of computation steps is 25x25 elements. Also, the fastest speed up ratio can be achieved by using this sub-background size (25x25) as shown in Figure 1. It is clear that the speed up ratio is reduced when the size of the sub-background *(L)* is increased. A comparison between the speed up ratio for fast cross correlation and faster cross correlation after background decomposition with different sizes of the backgrounds is listed in Tables 6 and 7. It is clear that the speed up ratio is increased with the size of the background when using faster cross correlation and background decomposition. This is in contrast to using only fast cross correlation. As shown in Figure 2, the number of computation steps required by faster cross correlation is increased rapidly with the size of the background. Therefore the speed up ratio is decreased with the size of the background. While in case of using faster cross correlation and background decomposition, the number of computation steps required by faster cross correlation is increased smoothly. Thus, the linearity of the computation steps required by faster cross correlation in this case is better. As a result, the speed up ratio is increased. Increasing the speed up ratio with the size of the background is considered an important achievement. Furthermore, for very large size matrices, while the speed up ratio for fast cross correlation is decreased, the speed up ratio still increase in case of using faster cross correlation and matrix decomposition. Moreover, as shown in Figure 3, the speed up ratio in case of faster cross correlation and background decomposition is increased with the size of the applied mask. For example, it is clear that the speed up ratio is for a mask of size 30x30 is larger than that of size 20x20. Simulation results for the speed up ratio in case of using faster cross correlation and background decomposition is listed in Table 8. It is clear that simulation results confirm the theoretical computations and the practical speed up ratio after background decomposition is faster than using only faster cross correlation. In addition, the practical speed up ratio is increased with the size of the background.

Also, to process small in size masks such as 5x5 or 10x10 using only fast cross correlation, the speed ratio becomes less than one as shown in Table 9. On the other hand, from the same table, it is clear that using fast cross correlation and background decomposition, the speed up ratio becomes higher than one and increased with the dimensions of the background. The dimensions of the new sub-background after background decomposition *(L)* must not be less than the dimensions of the colored mask. Therefore, the following equation controls the relation between the length of the sub-background and the mask size in order not to loss any information in the background.

$$L \geq n \tag{17}$$

For example, in a mask of size 5x5, the background must be decomposed into sub-backgrounds of size not less than 5x5.

To further reduce the running time as well as increase the speed up ratio of the painting process, a parallel processing technique is used. Each sub-background is processed using a faster cross correlation simulated on a single processor or a separated node in a clustered system. The number of operations (ω) performed by each processor / node (sub-backgrounds processed by one processor/node) =

$$\omega = \frac{The\ total\ number\ of\ sub - images}{Number\ of\ Processors\ /\ nodes} \tag{18}$$

$$\omega = \frac{\alpha}{Pr} \tag{19}$$

where, *Pr* is the number of processors or nodes.

The total number of computation steps (γ) required to process a single background by using this approach can be calculated as:

$$\gamma = \omega \beta \tag{20}$$

By using this algorithm, the speed up ratio in this case *(η_{dp})* can be computed as follows:

$$\eta_{dp} = \frac{q(2n^2 - 1)(N - n + 1)^2}{ceil(((q(\alpha + 1) + \alpha)(5N_S^2 \log_2 N_S^2) + \alpha q(8N_S^2 - n^2) + N_S)/pr)} \tag{21}$$

where, *ceil(x)* is a *MATLAB* function rounds the elements of x to the nearest integers towards infinity.

As shown in Table 10, using a symmetric multiprocessing system with 16 parallel processors or 16 nodes in either a massively parallel processing system or a clustered system, the speed up ratio (with respect to conventional cross correlation) for automatic painting is increased. A further reduction in the computation steps can be obtained by dividing each sub-background into groups. For each group, the dot product operation (multiplication by masks and summation) is performed for each group by using a single processor. This operation is done for all of these groups as well as other groups in all of the sub-backgrounds at the same time. The best case is achieved when each group consists of only one element. In this case, one operation is needed for multiplication of this element by its corresponding element in the mask matrix and also a small number of operations (ε) is required to obtain the over all summation for each sub-background. If the sub-background has n^2 elements, then the required number of processors will be n^2. As a result, the number of computation steps will be $\alpha q(1+\varepsilon)$, where ε is a small number depending on the value of n. For example, when $n=20$, then $\varepsilon=6$ and if $n=25$, then $\varepsilon=7$. The speed up ratio can be calculated as:

$$\eta=(2n^2-1)(N-n+1)^2/\alpha(1+\varepsilon) \qquad (22)$$

Moreover, if the number of processors = αn^2, then the number of computation steps will be $q(1+\varepsilon)$, and the speed up ratio becomes:

$$\eta=(2n^2-1)(N-n+1)^2/ (1+\varepsilon) \qquad (23)$$

Furthermore, if the number of processors = $q\alpha n^2$, then the number of computation steps will be $(1+\varepsilon)$, and the speed up ratio can be calculated as:

$$\eta=q(2n^2-1)(N-n+1)^2/ (1+\varepsilon) \qquad (24)$$

In this case, as the length of each group is very small, then there is no need to apply cross correlation between the background and the colored masks in frequency domain.

4. Conclusion

A new technique for high speed painting has been presented. It has been proved mathematically and practically that the speed of the painting process becomes faster than using conventional cross correlation. This has been accomplished by applying cross correlation in the frequency domain between the background and the colored masks. A new general formula for fast cross correlation as well as the speed up ratio has been given. Furthermore, a faster cross correlation approach for automatic painting has been introduced. Such approach has decomposed the background into many small in size sub-backgrounds. Simulation results have shown that, by using faster cross correlation and background decomposition, the speed up ratio of the painting process has been increased with the size of the background. In addition, large values of speed up ratio for the painting process have been achieved by using a parallel processing technique. Moreover, this algorithm can be used for generating new colors automatically as a result from the cross correlation operation.

References

[1] Hazem M. El-Bakry, “New Fast Time Delay Neural Networks Using Cross Correlation Performed in the Frequency Domain,” Accepted and Under Publication in Neurocomputing Journal.

[2] Hazem M. El-Bakry, and Qiangfu Zhao, “Fast Normalized Neural Processors For Pattern Detection Based on Cross Correlation Implemented in the Frequency Domain,” Accepted and Under Publication in the Journal of Research and Practice in Information Technology.

[3] Hazem M. El-Bakry, “Faster PCA for Face Detection Using Cross Correlation in the Frequency Domain,” the International Journal of Computer Science and Network Security, vol.6, No. 2A, Feb. 2006, pp.69-74.

[4] Hazem M. El-Bakry, “New High Speed Normalized Neural Networks for Fast Pattern Discovery on Web Pages,” the International Journal of Computer Science and Network Security, vol.6, No. 2A, Feb. 2006, pp.142-152.

[5] Hazem M. El-Bakry, and Qiangfu Zhao, “Fast Time Delay Neural Networks,” the International Journal of Neural Systems, vol. 15, No.6, December 2005, pp.445-455.

[6] Hazem M. El-Bakry, and Qiangfu Zhao, “Speeding-up Normalized Neural Networks For Face/Object Detection,” Machine Graphics & Vision Journal (MG&V), vol. 14, No.1, 2005, pp. 29-59.

[7] Hazem M. El-Bakry, and Qiangfu Zhao, “A New Technique for Fast Pattern Recognition Using Normalized Neural Networks,” WSEAS Transactions on Information Science and Applications, Issue 11, Vol. 2, November 2005, pp. 1816-1835.

[8] Hazem M. El-Bakry, and Qiangfu Zhao, “Fast Pattern Detection Using Normalized Neural Networks and Cross Correlation in the Frequency Domain,” EURASIP Journal on Applied Signal Processing, Special Issue on Advances in Intelligent Vision Systems: Methods and Applications—Part I, Vol. 2005, No. 13, 1 August 2005, pp. 2054-2060.

[9] Hazem M. El-Bakry, “Fast Neural Networks for Pattern Detection Using 2D-FFT,” Proc. of the 13th European Signal Processing Conference, 4-8 September 2005, Antalya, Turkey.

[10] Hazem M. El-Bakry, “Pattern Detection Using Fast Normalized Neural Networks,” Proc. of ICANN, 11-15 September 2005, Warsaw, Poland, pp 447-454.

[11] Hazem M. El-Bakry, “Human Face Detection Using New High Speed Modular Neural Networks,” Proc. of ICANN, 11-15 September 2005, Warsaw, Poland, 543-550.

[12] Hazem M. El-bakry, and Qiangfu Zhao, “Modified Time Delay Neural Networks For Fast Data Processing,” Proc. of

IEEE Eighth International Symposium on Signal Processing and its Applications, Sydney, Australia, August 28-31, 2005.

[13] Hazem M. El-bakry, and Qiangfu Zhao, “Fast Co-Operative Modular Neural Networks for Fast Human Face Detection,” Proc. of IEEE Eighth International Symposium on Signal Processing and its Applications, Sydney, Australia, August 28-31, 2005.

[14] Hazem M. El-bakry, and Qiangfu Zhao, “Fast Pattern Detection Using Neural Networks and Cross Correlation in the Frequency Domain,” Proc. of IEEE IJCNN’05, Quebec, Montreal, Canada, July 31- August 4, 2005, pp. 1900-1905.

[15] Hazem M. El-bakry, and Qiangfu Zhao, “Normalized Neural Networks for Fast Pattern Detection,” Proc. of IEEE IJCNN’05, Quebec, Montreal, Canada, July 31- August 4, 2005, pp. 1889-1894.

[16] Hazem M. El-bakry, and Qiangfu Zhao, “A Modified Frequency Domain Cross Correlation Implemented in MATLAB for Fast Sub-Image Detection Using Neural Networks,” Proc. of IEEE IJCNN’05, Quebec, Montreal, Canada, July 31- August 4, 2005, pp. 1794-1799.

[17] Hazem M. El-Bakry, and Qiangfu Zhao, “New Faster Normalized Neural Networks For Sub-Matrix Detection Using Cross Correlation in the Frequency Domain and Matrix Decomposition,” International Journal of Signal Processing, vol.2, no.3, 2005, pp. 183-202.

[18] Hazem M. El-Bakry, and Qiangfu Zhao, “A Fast Neural Algorithm for Serial Code Detection in a Stream of Sequential Data,” International Journal of Information Technology, vol.2, no.1, pp. 71-90, 2005.

[19] Hazem M. El-Bakry, and Qiangfu Zhao, “Fast Complex Valued Time Delay Neural Networks,” International Journal of Computational Intelligence, vol.2, no.1, pp. 16-26, 2005.

[20] Hazem M. El-Bakry, “Fast Sub-Image Segmentation Using Parallel Neural Processors and Image Decomposition,” Proc. of the SPIE Vol. 5960, the International Symposium on Visual Communication and Image Processing (VCIP), 12-15 July, 2005, Beijing, China**,** pp. 1712-1722.

[21] Hazem M. El-Bakry, “A Fast Neural Algorithm for Patten Detection Using Cross Correlation in the Frequency Domain,” Proc. of the 19th European Conference on Modeling and Simulation (ESM2005), The 2005 High Performance Computing & Simulation (HPC&S) Conference, 1-4 June 2005, Riga, Latvia.

[22] Hazem M. El-Bakry, “Fast Pattern Detection Using Parallel Neural Processors and Image Decomposition,” Proc. of the 19th European Conference on Modeling and Simulation (ESM2005), Simulation of Intelligent Systems, 1-4 June 2005, Riga, Latvia.

[23] Hazem M. El-Bakry, and Qiangfu Zhao, “A Modified Neural Algorithm Implemented in MATLAB for Fast Pattern Detection,” Proc. of the Eighth International Conference on Pattern Recognition and Information Processing PRIP, 18-20 May 2005, Minsk, Belarus.

[24] Hazem M. El-Bakry, and Q. Zhao, “Fast Pattern Detection Using Neural Networks Realized in Frequency Domain,” Proc. of the International Conference on Pattern Recognition and Computer Vision, The Second World Enformatika Congress WEC'05, Istanbul, Turkey, 25-27 Feb., 2005, pp. 89-92.

[25] Hazem M. El-Bakry, and Q. Zhao, “Sub-Image Detection Using Fast Neural Processors and Image Decomposition,” Proc. of the International Conference on Pattern Recognition and Computer Vision, The Second World Enformatika Congress WEC'05, Istanbul, Turkey, 25-27 Feb., 2005, pp. 85-88.

[26] Hazem M. El-Bakry, and and Qiangfu Zhao “Fast Sub-Matrix Detection Using Neural Networks and Cross Correlation in the Frequency Domain,” Second Workshop of Tohoku Branch, IPSJ, (Information Processing Society of Japan), University of Aizu, Aizuwakamatsu, Japan, Jan. 21, 2005.

[27] Hazem M. El-Bakry, and Qiangfu Zhao, “Face Detection Using Fast Neural Processors and Image Decomposition,” International Journal of Computational Intelligence, vol.1, no.4, pp. 313-316, 2004.

[28] Hazem M. El-Bakry, and Qiangfu Zhao, “A Modified Cross Correlation in the Frequency Domain for Fast Pattern Detection Using Neural Networks,” International Journal on Signal Processing, vol.1, no.3, pp. 188-194, 2004.

[29] Hazem M. El-Bakry, and Qiangfu Zhao, “Fast Object/Face Detection Using Neural Networks and Fast Fourier Transform,” International Journal on Signal Processing, vol.1, no.3, pp. 182-187, 2004.

[30] Hazem M. El-Bakry, ”Face detection using fast neural networks and image decomposition,” Neurocomputing Journal, vol. 48, 2002, pp. 1039-1046.

[31] Hazem M. El-Bakry, ”Human Iris Detection Using Fast Cooperative Modular Neural Nets and Image Decomposition,” Machine Graphics & Vision Journal (MG&V), vol. 11, no. 4, 2002, pp. 498-512.

[32] Hazem El-Bakry, “Face Detection Using Fast Neural Networks and Image Decomposition,” Proc. of INNS-IEEE International Joint Conference on Neural Networks, 14-19 May, 2002, Honolulu, Hawaii, USA.

[33] H. M. El-Bakry, "Comments on Using MLP and FFT for Fast Object/Face Detection," *Proc. of IEEE IJCNN’03*, Portland, Oregon, pp. 1284-1288, July, 20-24, 2003.

[34] H. M. El-Bakry, and H. Stoyan, "Fast Neural Networks for Object/Face Detection," *Proc. of the 30th Anniversary SOFSEM Conference on Current Trends in Theory and Practice of Computer Science*, 24-30 January, 2004, Hotel VZ MERIN, Czech Republic.

[35] H. M. El-Bakry, and H. Stoyan, "Fast Neural Networks for Sub-Matrix (Object/Face) Detection," *Proc. of IEEE International Symposium on Circuits and Systems*, Vancouver, Canada, 23-26 May, 2004.

[36] H. M. El-Bakry, "Fast Sub-Image Detection Using Neural Networks and Cross Correlation in Frequency Domain," *Proc. of IS 2004: 14th Annual Canadian Conference on Intelligent Systems*, Ottawa, Ontario, 6-8 June, 2004.

[37] H. M. El-Bakry, and H. Stoyan, "Fast Neural Networks for Code Detection in a Stream of Sequential Data," *Proc. of CIC 2004 International Conference on Communications in Computing*, Las Vegas, Nevada, USA, 21-24 June, 2004.

[38] H. M. El-Bakry, "Fast Neural Networks for Object/Face Detection," *Proc. of 5th International Symposium on Soft Computing for Industry with Applications of Financial Engineering*, June 28 - July 4, 2004, Sevilla, Andalucia, Spain.

[39] H. M. El-Bakry, and H. Stoyan, "A Fast Searching Algorithm for Sub-Image (Object/Face) Detection Using Neural Networks," *Proc. of the 8th World Multi-Conference*

on Systemics, Cybernetics and Informatics, 18-21 July, 2004, Orlando, USA.

[40] H. M. El-Bakry, and H. Stoyan, "Fast Neural Networks for Code Detection in Sequential Data Using Neural Networks for Communication Applications," *Proc. of the First International Conference on Cybernetics and Information Technologies*, Systems and Applications: CITSA 2004, 21-25 July, 2004. Orlando, Florida, USA, Vol. IV, pp. 150-153.

[41] H. M. El-bakry, "Fast Iris Detection using Cooperative Modular Neural Nets," *Proc. of the 6th International Conference on Soft Computing*, 1-4 Oct., 2000, Japan.

[42] H. M. El-Bakry, "Automatic Human Face Recognition Using Modular Neural Networks," *Machine Graphics & Vision Journal (MG&V)*, vol. 10, no. 1, 2001, pp. 47-73.

[43] H. M. El-bakry, "Fast Iris Detection Using Cooperative Modular Neural Networks," *Proc. of the 5th International Conference on Artificial Neural Nets and Genetic Algorithms*, pp. 201-204, 22-25 April, 2001, Sydney, Czech Republic.

[44] H. M. El-bakry, "Fast Iris Detection Using Neural Nets," *Proc. of the 14th Canadian Conference on Electrical and Computer Engineering*, pp.1409-1415, 13-16 May, 2001, Canada.

[45] H. M. El-bakry, "Human Iris Detection Using Fast Cooperative Modular Neural Nets," *Proc. of INNS-IEEE International Joint Conference on Neural Networks*, pp. 577-582, 14-19 July, 2001, Washington, DC, USA.

[46] H. M. El-bakry, "Human Iris Detection for Information Security Using Fast Neural Nets," *Proc. of the 5th World Multi-Conference on Systemics, Cybernetics and Informatics*, 22-25 July, 2001, Orlando, Florida, USA.

[47] H. M. El-bakry, "Human Iris Detection for Personal Identification Using Fast Modular Neural Nets," *Proc. of the 2001 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences,* pp. 112-118, 25-28 July, 2001, Monte Carlo Resort, Las Vegas, Nevada, USA.

[48] H. M. El-bakry, "Human Face Detection Using Fast Neural Networks and Image Decomposition," *Proc. the fifth International Conference on Knowledge-Based Intelligent Information & Engineering Systems*, 6-8 September 2001, Osaka-kyoiku University, Kashiwara City, Japan, pp. 1330-1334.

[49] H. M. El-Bakry, "Fast Iris Detection for Personal Verification Using Modular Neural Networks," *Proc. of the International Conference on Computational Intelligence*, 1-3 Oct., 2001, Dortmund, Germany, pp.-269-283.

[50] H. M. El-bakry, "Fast Cooperative Modular Neural Nets for Human Face Detection," *Proc. of IEEE International Conference on Image Processing*, 7-10 Oct., 2001, Thessaloniki, Greece.

[51] H. M. El-Bakry, "Fast Face Detection Using Neural Networks and Image Decomposition," *Proc. of the 6th International Computer Science Conference*, Active Media Technology, Dec. 18-20, 2001, Hong Kong – China, pp. 205-215, 2001.

[52] H. M. El-Bakry, "Face Detection Using Fast Neural Networks and Image Decomposition," *Proc. of INNS-IEEE International Joint Conference on Neural Networks*, 14-19 May, 2002, Honolulu, Hawaii, USA.

[53] H. M. El-bakry, M. A. Abo-elsoud, and M. S. Kamel, "Fast Modular Neural Networks for Human Face Detection," *Proc. of IEEE-INNS-ENNS International Joint Conference on Neural Networks*, Como, Italy, Vol. III, pp. 320-324, 24-27 July, 2000.

[54] R. Klette, and Zamperon, *Handbook of image processing operators*, John Wiley & Sonsltd, 1996.

[55] James W. Cooley and John W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput*. 19, 297–301 (1965).

[56] J.P. Lewis, “Fast Normalized Cross Correlation”, Available from:<http://www.idiom.com/~zilla/ Papers/nvisionInterface/nip.html >

[57] Aaron Hertzmann, " Fast Painting Texture," Proc. of ACM Symposium on Non-Photorealistic Animation and Rendering, Annecy, France, June 03 - 05, 2002, pp. 91-98.

Hazem Mokhtar El-Bakry (Mansoura, EGYPT 20-9-1970) received B.Sc. degree in Electronics Engineering, and M.Sc. in Electrical Communication Engineering from the Faculty of Engineering, Mansoura University – Egypt, in 1992 and 1995 respectively. Since 1997, he has been an assistant lecturer at the Faculty of Computer Science and Information Systems – Mansoura University – Egypt.

His research interests include neural networks, pattern recognition, image processing, biometrics, cooperative intelligent systems and electronic circuits. In these areas, he has published more than 41 papers as a single author in major international journals and conferences. He is the first author in 19 refereed international journal papers and more than 71 refereed international conference papers.

Eng. El-Bakry has the patent No. 2003E 19442 DE HOL / NUR, Magnetic Resonance, SIEMENS Company, Erlangen, Germany, 2003. He is a referee for IEEE Transactions on Signal Processing, the International Journal of Machine Graphics & Vision, Enformatika Journals, and many IEEE international conferences.

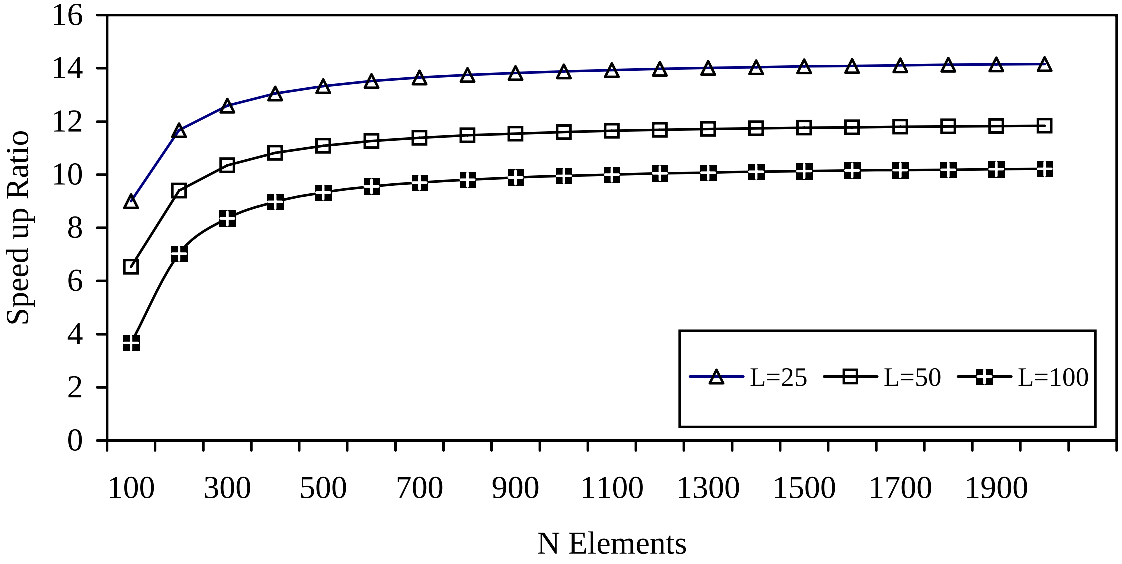

Fig. 1. The speed up ratio for different in size backgrounds decomposed into different in size sub-backgrounds (L).

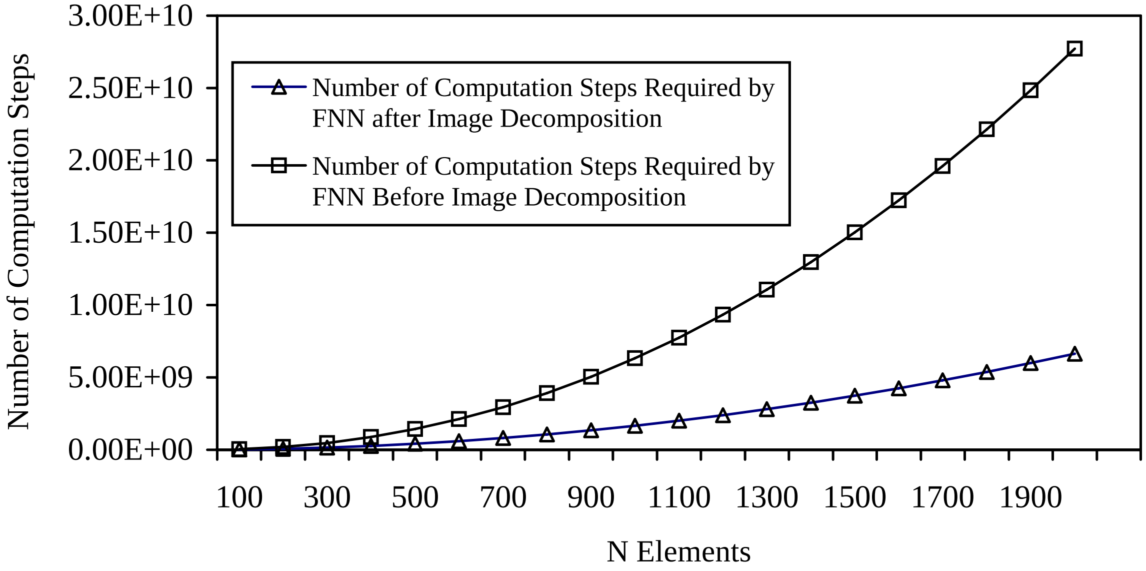

Fig. 2. A comparison between the number of computation steps required by fast cross correlation before and after background decomposition.

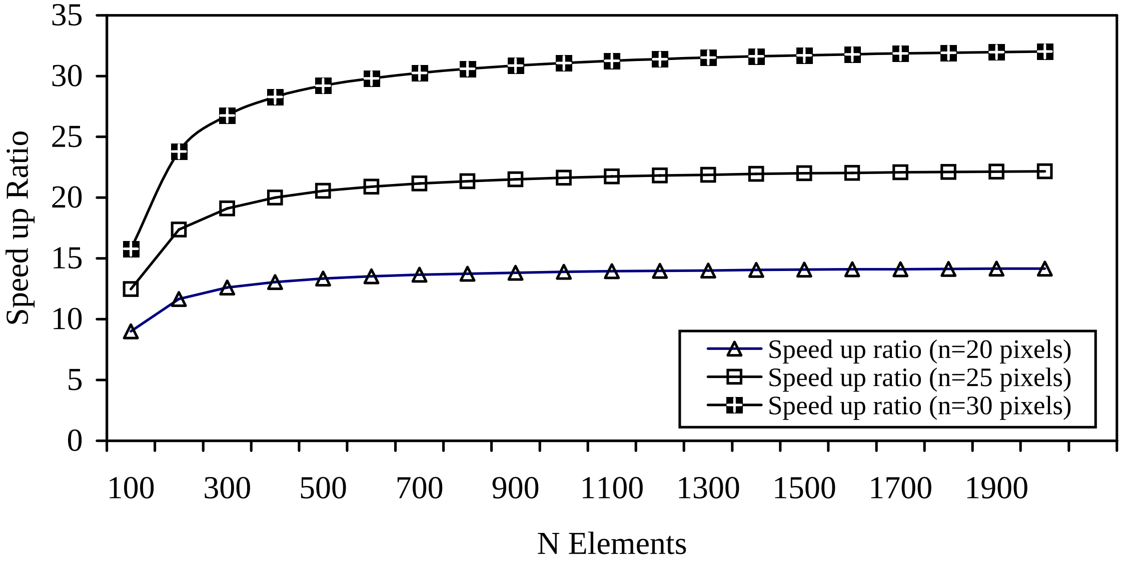

Fig. 3. The speed up ratio in case of background decomposition and different window size (n), (L=25x25).

Table 1: The theoretical speed up ratio for backgrounds with different sizes.

Background size	Speed up ratio (n=20)	Speed up ratio (n=25)	Speed up ratio (n=30)
100x100	3.67	5.04	6.34
200x200	4.01	5.92	8.05
300x300	4.00	6.03	8.37
400x400	3.95	6.01	8.42
500x500	3.89	5.95	8.39
600x600	3.83	5.88	8.33
700x700	3.78	5.82	8.26
800x800	3.73	5.76	8.19
900x900	3.69	5.70	8.12
1000x1000	3.65	5.65	8.05
1100x1100	3.62	5.60	7.99
1200x1200	3.58	5.55	7.93
1300x1300	3.55	5.51	7.93
1400x1400	3.53	5.47	7.82
1500x1500	3.50	5.43	7.77
1600x1600	3.48	5.43	7.72
1700x1700	3.45	5.37	7.68
1800x1800	3.43	5.34	7.64
1900x1900	3.41	5.31	7.60
2000x2000	3.40	5.28	7.56

Table 2: Practical Speed up ratio for backgrounds with different sizes using MATLAB ver 5.3.

Background size	Speed up ratio (n=20)	Speed up ratio (n=25)	Speed up ratio (n=30)
100x100	7.88	10.75	14.69
200x200	6.21	9.19	13.17
300x300	5.54	8.43	12.21
400x400	4.78	7.45	11.41
500x500	4.68	7.13	10.79
600x600	4.46	6.97	10.28
700x700	4.34	6.83	9.81
800x800	4.27	6.68	9.60
900x900	4.31	6.79	9.72
1000x1000	4.19	6.59	9.46
1100x1100	4.24	6.66	9.62
1200x1200	4.20	6.62	9.57
1300x1300	4.17	6.57	9.53
1400x1400	4.13	6.53	9.49
1500x1500	4.10	6.49	9.45
1600x1600	4.07	6.45	9.41
1700x1700	4.03	6.41	9.37
1800x1800	4.00	6.38	9.32
1900x1900	3.97	6.35	9.28
2000x2000	3.94	6.31	9.25

Table 3: The theoretical speed up ratio for the general fast cross correlation algorithm used in painting and automatic color generation.

Background size	Speed up ratio (n=20)	Speed up ratio (n=25)	Speed up ratio (n=30)
100x100	5.39	8.36	11.95
200x200	4.81	7.49	10.75
300x300	4.51	7.03	10.16
400x400	4.32	6.73	9.68
500x500	4.18	6.52	9.37
600x600	4.07	6.35	9.13
700x700	3.99	6.21	8.94
800x800	3.91	6.10	8.77
900x900	3.84	6.00	8.63
1000x1000	3.78	5.91	8.51
1100x1100	3.74	5.85	8.4 3
1200x1200	.3 70	5.78	8.3 3
1300x1300	3.6 6	5.72	8.2 4
1400x1400	3.62	5.66	8.1 6
1500x1500	3.59	5.61	8.08
1600x1600	3.56	5.5 7	8.0 2
1700x1700	3.53	5.52	7.95
1800x1800	3.5 1	5.48	7.89
1900x1900	3.48	5.44	7.8 4
2000x2000	3.45	5.40	7.79

Table 4: Simulation results of the speed up ratio for the general fast cross correlation compared with the MATLAB cross correlation function (xcorr2).

Background size	Speed up ratio (n=20)	Speed up ratio (n=25)	Speed up ratio (n=30)
100x100	10.14	13.05	16.49
200x200	9.17	11.92	14.33
300x300	8.25	10.83	13.41
400x400	7.91	9.62	12.65
500x500	6.77	9.24	11.77
600x600	6.46	8.89	11.19
700x700	5.99	8.47	10.96
800x800	5.48	8.74	10.32
900x900	5.31	8.43	10.66
1000x1000	5.91	8.66	10.51
1100x1100	5.77	8.61	10.46
1200x1200	5.68	8.56	10.40
1300x1300	5.62	8.52	10.35
1400x1400	5.58	8.47	10.31
1500x1500	5.54	8.43	10.26
1600x1600	5.50	8.39	10.22
1700x1700	5.46	8.33	10.18
1800x1800	5.42	8.28	10.14
1900x1900	5.38	8.24	10.10
2000x2000	5.34	8.20	10.06

Table 5: The number of computation steps required by fast cross correlation for backgrounds of sizes (25x25 - 550x550 elements), q=30, n=20.

Background size	No. of computation steps in case of using fast cross correlation	Background size	No. of computation steps in case of using fast cross correlation
25x25	1.9085e+006	1050x1050	7.0142e+009
50x50	9.1949e+006	1100x1100	7.7476e+009
100x100	4.2916e+007	1150x1150	8.5197e+009
150x150	1.0460e+008	1200x1200	9.3306e+009
200x200	1.9610e+008	1250x1250	1.0180e+010
250x250	3.1868e+008	1300x1300	1.1070e+010
300x300	4.7335e+008	1350x1350	1.1998e+010
350x350	6.6091e+008	1400x1400	1.2966e+010
400x400	8.8203e+008	1450x1450	1.3973e+010
450x450	1.1373e+009	1500x1500	1.5021e+010
500x500	1.4273e+009	1550x1550	1.6108e+010
550x550	1.7524e+009	1600x1600	1.7236e+010
600x600	2.1130e+009	1650x1650	1.8404e+010
650x650	2.5096e+009	1700x1700	1.9612e+010
700x700	2.9426e+009	1750x1750	2.0861e+010
750x750	3.4121e+009	1800x1800	2.2150e+010
800x800	3.9186e+009	1850x1850	2.3480e+010
850x850	4.4622e+009	1900x1900	2.4851e+010
900x900	5.0434e+009	1950x1950	2.6263e+010
950x950	5.6623e+009	2000x2000	2.7716e+010
1000x1000	6.3191e+009	2050x2050	2.9211e+010

Table 6: The speed up ratio in case of using fast cross correlation and faster cross correlation after background decomposition into sub-backgrounds (25x25 elements) for backgrounds of different sizes (from N=100 to N=1000, n=25, q=30).

Background size	Speed up ratio in case of using fast cross correlation	Speed up ratio in case of using faster cross correlation after background decomposition
50x50	2.7568	4.5871
100x100	5.0439	8.9997
150x150	5.6873	10.7600
200x200	5.9190	11.6707
250x250	6.0055	12.2228
300x300	6.0301	12.5923
350x350	6.0254	12.8565
400x400	6.0059	13.0547
450x450	5.9790	13.2088
500x500	5.9483	13.3320
550x550	5.9160	13.4328
600x600	5.8833	13.5168
650x650	5.8509	13.5878
700x700	5.8191	13.6487
750x750	5.7881	13.7014
800x800	5.7581	13.7475
850x850	5.7292	13.7881
900x900	5.7013	13.8243
950x950	5.6744	13.8566
1000x1000	5.6484	13.8857

Table 7: The speed up ratio in case of using fast cross correlation and faster cross correlation after background decomposition into sub-backgrounds (25x25 elements) for backgrounds of different sizes (from N=1050 to N=2000, n=25, q=30).

Background size	Speed up ratio in case of using (FNN)	Speed up ratio in case of using FNN after background decomposition
1050x1050	5.6234	13.9120
1100x1100	5.5994	13.9359
1150x1150	5.5762	13.9577
1200x1200	5.5538	13.9777
1250x1250	5.5322	13.9961
1300x1300	5.5113	14.0131
1350x1350	5.4912	14.0288
1400x1400	5.4717	14.0434
1450x1450	5.4528	14.0570
1500x1500	5.4345	14.0696
1550x1550	5.4168	14.0815
1600x1600	5.3996	14.0926
1650x1650	5.3830	14.1030
1700x1700	5.3668	14.1129
1750x1750	5.3511	14.1221
1800x1800	5.3358	14.1309
1850x1850	5.3209	14.1391
1900x1900	5.3064	14.1470
1950x1950	5.2923	14.1544
2000x2000	5.2786	14.1615

Table 8: The practical speed up ratio in case of using fast cross correlation and faster cross correlation after background decomposition into sub-backgrounds (25x25 elements) for backgrounds of different sizes (from N=100 to N=1000, n=25, q=30).

Background size	Speed up ratio in case of using fast cross correlation	Speed up ratio in case of using faster cross correlation after background decomposition
100x100	10.75	34.55
200x200	9.19	35.65
300x300	8.43	36.73
400x400	7.45	37.70
500x500	7.13	38.66
600x600	6.97	39.61
700x700	6.83	40.56
800x800	6.68	41.47
900x900	6.79	42.39
1000x1000	6.59	43.28

Table 9: The speed up ratio in case of using cross correlation and faster cross correlation after background decomposition into sub-backgrounds (5x5 elements) for backgrounds of different sizes (from N=100 to N=1000, n=25, q=30).

Background size	Speed up ratio in case of using fast cross correlation	Speed up ratio in case of using faster cross correlation after background decomposition
100x100	0.3141	1.4543
200x200	0.2872	1.5177
300x300	0.2716	1.5388
400x400	0.2610	1.5493
500x500	0.2531	1.5557
600x600	0.2469	1.5599
700x700	0.2418	1.5629
800x800	0.2375	1.5652
900x900	0.2339	1.5669
1000x1000	0.2306	1.5683

Table 10: The speed up ratio in case of using fast cross correlation after background decomposition into sub-backgrounds (25x25 elements) for backgrounds of different sizes (from N=50 to N=1000, n=25, q=30) using 16 parallel processors or 16 nodes.

Background size	Speed up ratio
50x50	73.3934
100x100	143.9953
150x150	172.1592
200x200	186.7311
250x250	195.5652
300x300	201.4760
350x350	205.7032
400x400	208.8745
450x450	211.3405
500x500	213.3126
550x550	214.9255
600x600	216.2689
650x650	217.4051
700x700	218.3786
750x750	219.2219
800x800	219.9596
850x850	220.6102
900x900	221.1883
950x950	221.7054
1000x1000	222.1707