Parallel Implementation of Souvola's Binarization Approach on GPU

Brij Mohan Singh, Rahul Sharma
Department of CSE,
College of Engineering Roorkee,
Roorkee-247667,Uttarakhand, India

Ankush Mittal
Director (Research),
Graphic Era University, Dehradun-248002, Uttarakhand, India

Debashish Ghosh
Department of E&C,
IIT Roorkee,
Roorkee-247667,Uttarakhand, India

ABSTRACT
Binarization is widely used technique in many of the image processing applications. Fast algorithms are needed for fast and efficient image processing systems. Many algorithms of image processing and pattern recognition have recently been implemented on Graphic Processing Unit (GPU) for faster computational times. GPUs are most prominent hardware in utilizing parallelism and pipelining than general purpose CPUs. Moreover, Speed, programmability, and price become it more productive. In this paper, we proposed a parallel implementation of well known Sauvola's local binarization algorithm for Optical Character Recognition systems. In this experiment, we achieved a computational speedup of parallel implementation on GPU 20.8x times faster than implementation on CPU. The speedup results of GPU are promising.

General Terms
Document Image Analysis, GPU, Parallel Computing.

Keywords
Binarization; CUDA; OCR; GPU; Parallelization.

1. INTRODUCTION
Most document analysis and recognition approaches are developed on taking advantages of the underlying binarised image information [1]. Two level data representation reduced computational overhead and pixel analysis compared to 256 levels of gray scale or color image representation. Document analysis and recognition algorithms are very slower due to requirement of large memory and computational power. These systems require higher implementation performance and enable real time data processing. In the development of fast and accurate OCR systems, computational power reduction and power saving techniques are desirable. GPUs fulfill all the requirement of developing fast OCR systems. Speed, programmability, and price became it more attractive [2-3].

One of the most straightforward strategies for image analysis techniques are based on the use of brightness of regions in the image as a means of identification. It is desirable that the same type of feature will have the same brightness throughout the whole image. Global methods works better when uniform illumination presents in the image [4-9]. Uniform illumination allows easier and more reliable image thresholding and it can be achieved mainly by using artificial light sources and isolating the subject from noises occurred by light reflexes, sunlight etc. However, real-life and real-time document image processing approaches are performed in non-uniform lightning conditions. There might appear different types of degradation such as complexity backgrounds, non-uniform intensity and, shadows. The non-uniformity of light and other kind of degradations can be minimized or even eliminated by using image processing approaches. Local thresholding methods play important role in the processing of document images [10-16].

The era of GPUs were started by many researchers to implement parallel algorithms in various areas such as computational geometry, and scientific computation, as well as computer graphics [17-21]. Parallel implementations on GPUs have been applied to various numerical problems [22-25] to reduce the computation time. Computational cost reduction approaches to handwritten character recognition were proposed in [26-30]. Oh at al. implemented neural networks on GPU, which is one of popular algorithm of pattern recognition, and the GPU was used to implement the matrix multiplication of a neural network to enhance the time performance [31]. Jung [32] proposed a Neural Network based text localization in color images. Recently, Singh et al. proposed parallel implementation of well known profiling based segmentation algorithm for Devanagari character recognition on GPU [33].

In the following sections, we present a detailed description of the proposed methodology as well as experimental results that demonstrate the efficiency of the proposed methodology.

2. INTRODCTION TO NVIDIA CUDA ARCHITECTURE
This paper proposes more quick and efficient parallel implementation on graphics hardware. We use a GPU language CUDA (Compute Unified Device Architecture) Instruction Set Architecture (ISA) and the parallel compute engine in the GPU developed by NVIDIA, as the CUDA code is similar to C language style and has less computational restriction. Other GPU compilers require much special knowledge on computer

graphics. CUDA enabled GPUs have hundreds of cores that can collectively run thousands of computing threads. Each core has shared resources, including registers and memory. The on-chip shared memory allows parallel tasks running on these cores to share data without sending it over the system memory bus [34].

A fundamental building block of CUDA programs is the CUDA kernel function [35]. When launching a CUDA kernel function, a developer specifies how many copies of it to run. We call each of these copies a task. Because of the hardware support of the GPU, each of these tasks can be small, and the developer can queue hundreds of thousands of them for execution at once. These tasks are organized in a two-level hierarchy, block and grid. Small sets of tightly coupled tasks are grouped into blocks. In a given execution of a CUDA kernel function, all blocks contain the same number of tasks. The tasks in a block run concurrently and can easily communicate with each other, which enables useful optimizations such as those of the section shared memory. GPU's hardware keeps multiple blocks in flight at once, with no guarantees about their relative execution order. As a result, synchronization between blocks is difficult. The set of all blocks run during the execution of a CUDA kernel function is called a grid.

3. SOUVOLA'S BINARIZATION APPROACH

Sauvola [12] proposed an algorithm similar to Niblack's [11]. Niblack's method works poorly on noisy and degraded documents. Sauvola made some assumptions based on the distribution of grey values associated with foreground and background pixels. Threshold is computed as:

$$T(x,y) = m(x,y)\cdot\left[1 + k\cdot\left(\frac{s(x,y)}{R} - 1\right)\right] \quad (1)$$

Where m(x,y) and s(x,y) are the same as in Niblack's method. R is the dynamic range of standard deviation. Values of R=128 and k=0.5 were used.

4. IMPLEMENTATION

In this experiment, first we implemented serial code of local binarization approach of souvola's proposed architecture and second, parallel code is implemented. The sequential algorithm is implemented in C++ and making use of C++ Standard Template Library. VC++ version 14.00.50727.42 compiler for 80x86 is used.

The input image is given as a sequence of bytes representing the intensity of a greyscale image. The input image is stored as texture in the device memory. After that block and grid size was calculated according to the dimensions of the input images. A single thread calculates the threshold for a single pixel in the output image. Following section dictates the detailed description of the parallel implementation of the algorithm.

4.1 Parallel Implementation

In CUDA, it is assumed that both host and device maintain their own DRAM. Host memory is allocated using malloc and device memory is allocated using cudaMalloc. CUDA threads are assigned a unique thread ID that identifies its location within the thread, block and grid. This provides a natural way to invoke computation across the image, by using the thread IDs for addressing. The parallel implementation of algorithm of binarization is shown in the form of pseudo code shown in algorithm 1.

Algorithm 1: Parallel Implementation of binarization algorithm

```
Texture grayImage;
Kernal( windowSize, outputImage)
{       int x = blockIdx.x * blockDim.x + threadIdx.x;
        int y = blockIdx.y * blockDim.y + threadIdx.y;
        int sum=0, sqr_sum=0;
        for( i=y-windowSize to y+windowSize)
                for( j=x-windowSize to x+windowSize){
                int v = grayImage.getPixel( j, i);
                        sum += v;
                        sqr_sum += v*v;
                }
        Calculate mean & varience;
        Threshold = mean*(1+k*(varience/R-1));
        if(grayImage.getPixel( x, y) <= threshold)
                outputImage.setPixel( x, y) = BLACK;
        else    outputImage.setPixel( x, y) = WHITE;
}
Main()
{
        dim3 dBlock( BLOCKSIZE, BLOCKSIZE);
        dim3 dGrid( (width+dBlock.x-1)/dBlock.x,
(height+dBlock.y-1)/dBlock.y );
        kernel<<< dGrid, dBlock>>>( windowSize,
outputImage);
}
```

5. RESULTS AND DISCUSSIONS

For the testing of Souvola's approach of local binarization, we collected a data set of handwritten as well as printed documents from newspapers, old books and from different writers. The collected documents are scanned using a scanner at 300 dpi. All the experiments were carried out using the hardware specifications of GPU: GeForce 9500 GT, 1 MB DDR2, No of Processors = 4, No of core =32, RAM 1 GB, Frequency 1.35 GHz, DDR2 and CPU: Intel Core 2 Duo, 2.66 GHZ, No of cores available =2, No of thread=1, No of thread/core=1, Physical Memory =2 GB, DDR2.

To make faster the method, we parallelized Souvola approach of binarization on CUDA and achieved an average speedup of 20.778803x over the serial implementation when running on a GPU. The comparison of execution time of serial implementation over parallel is shown in table 1. Table 1 also shows that execution time depends on the window size and image size in megapixels. As shown in table 1, average speed-up is 22.89901941 when window size is 7, average speed-up is 20.96931534 when window size is 11, average speed-up is 20.31803128 when window size is 15, when window size is 19 the average speed-up is 19.98818541, and when window size is 23 the average speed-up is 19.71946374. The observation is that when window size increases, the execution time decreases. Fig. 1 shows the graph of execution time of CPU in seconds vs. window size. Fig. 2 shows the graph of execution time of GPU

in seconds vs. window size. Fig. 3 shows the graph of speedup vs. window size.

On the basis of visual observation, Souvola's method of binarization completely recovers text from degraded document images. The promising results of approach are shown in fig. 4.

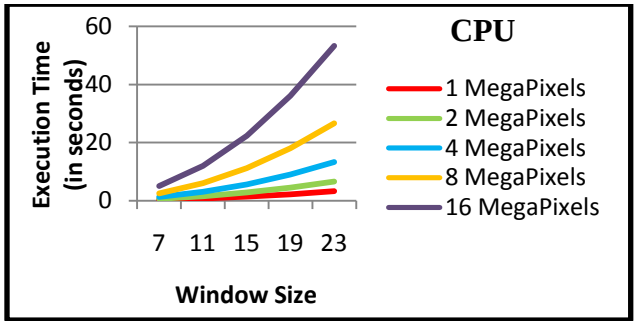


Figure 1: Execution time of CPU in seconds vs. window size

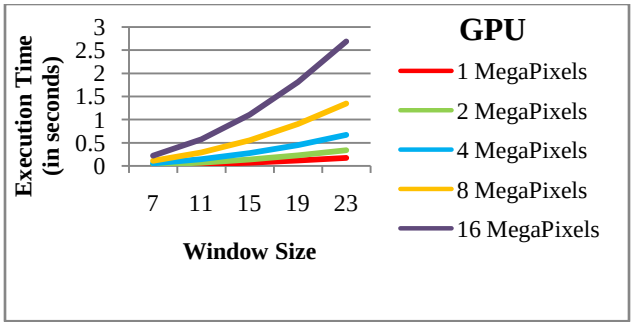


Figure 2: Execution time of GPU in seconds vs. window size

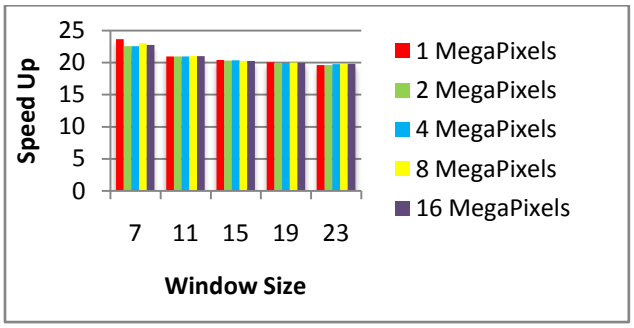
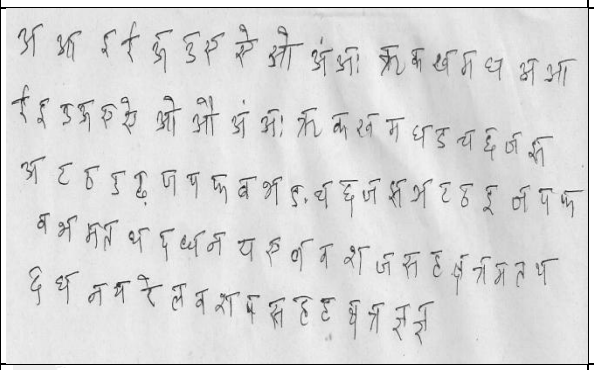
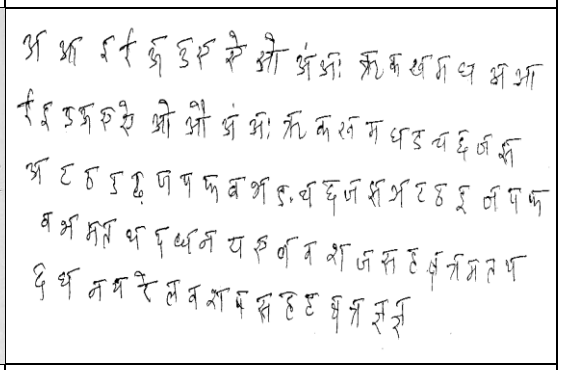
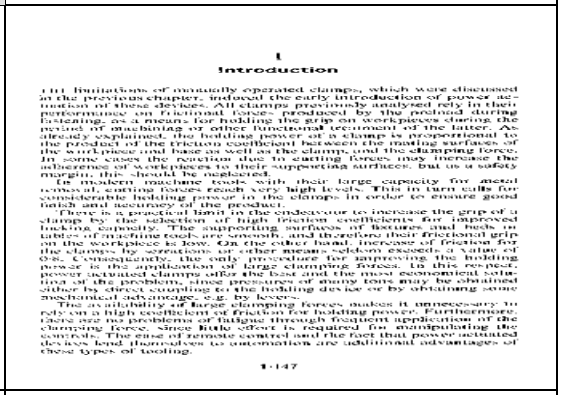
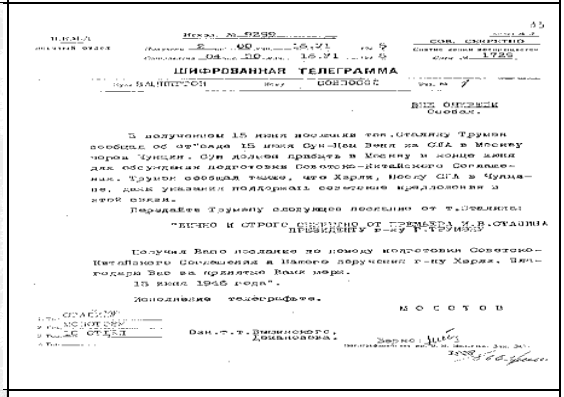
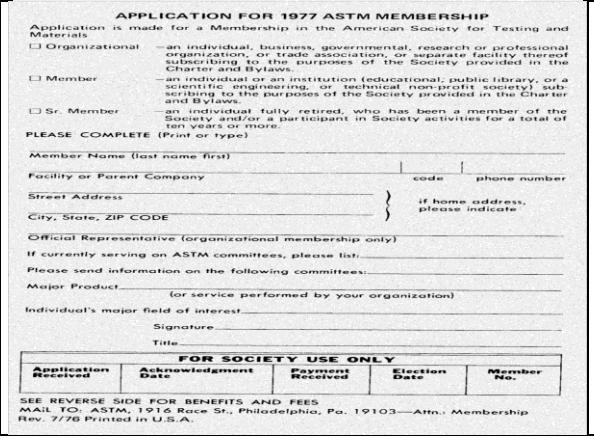
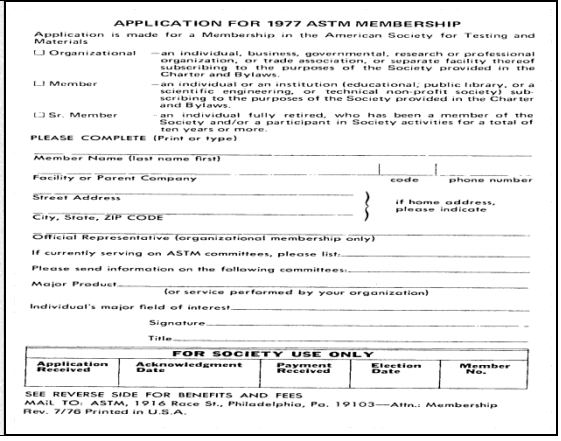


Figure 3: Speedup vs. Window size

Window Size	Mega-pixels	Serial	Parallel	Speed-Up	Speed-Up Average
7	1	0.328	0.01388	23.6311239	22.89901941
	2	0.625	0.0277	22.5631769	
	4	1.25	0.05545	22.5428314	
	8	2.532	0.11016	22.9847495	
	16	5.047	0.22162	22.7732154	
11	1	0.75	0.0358	20.9497207	20.96931534
	2	1.5	0.0716	20.9497207	
	4	3	0.14318	20.952647	
	8	6.016	0.28642	21.0041198	
	16	12.03	0.57312	20.9903685	
15	1	1.39	0.0681	20.4111601	20.31803128
	2	2.797	0.1376	20.3270349	
	4	5.594	0.2751	20.3344238	
	8	11.172	0.5515	20.2574796	
	16	22.359	1.1036	20.260058	
19	1	2.25	0.1123	20.0356189	19.98818541
	2	4.5	0.2255	19.9556541	
	4	9.015	0.4512	19.9800532	
	8	18.047	0.9023	20.0011083	
	16	36.125	1.8091	19.9684926	
23	1	3.296	0.1679	19.6307326	19.71946374
	2	6.61	0.3368	19.6258907	
	4	13.281	0.6724	19.7516359	
	8	26.594	1.3445	19.7798438	
	16	53.265	2.6889	19.8092157	
		Avg. Speed-Up			20.778803

Table 1: Execution time of serial over parallel implementation

Image No.	Degraded Image	Souvola's Output
1.		
2.		
3.		
4.		

1

Introduction

THE limitations of manually operated clamps, which were discussed in the previous chapter, induced the early introduction of power actuation of these devices. All clamps previously analysed rely in their performance on frictional forces produced by the preload during fastening, as a means for holding the grip on workpieces during the period of machining or other functional treatment of the latter. As already explained, the holding power of a clamp is proportional to the product of the friction coefficient between the mating surfaces of the workpiece and base as well as the clamp, and the clamping force. In some cases the reaction due to cutting forces may increase the adherence of workpieces to their supporting surfaces, but as a safety margin, this should be neglected.

In modern machine tools with their large capacity for metal removal, cutting forces reach very high levels. This in turn calls for considerable holding power in the clamps in order to ensure good finish and accuracy of the product.

There is a practical limit in the endeavour to increase the grip of a clamp by the selection of high friction coefficients for improved locking capacity. The supporting surfaces of fixtures and beds or tables of machine tools are smooth, and therefore their frictional grip on the workpiece is low. On the other hand, increase of friction for the clamps by serrations or other means seldom exceeds a value of 0·8. Consequently, the only procedure for improving the holding power is the application of large clamping forces. In this respect, power actuated clamps offer the best and the most economical solution of the problem, since pressures of many tons may be obtained either by direct coupling to the holding device or by obtaining some mechanical advantage, e.g. by levers.

The availability of large clamping forces makes it unnecessary to rely on a high coefficient of friction for holding power. Furthermore, there are no problems of fatigue through frequent application of the clamping force, since little effort is required for manipulating the controls. The ease of remote control and the fact that power actuated devices lend themselves to automation are additional advantages of these types of tooling.

1·147

ШИФРОВАННАЯ ТЕЛЕГРАММА

Нехэд. № 9299

СОВ. СЕКРЕТНО

Куда ВАШИНГТОН Кому СОВПОСОЛ Экз. № 7

ВНЕ ОЧЕРЕДИ
Особая.

В полученном 15 июня послании тов.Сталину Трумэн сообщил об отъезде 15 июня Сун-Цзы Вэня из США в Москву через Чунцин. Сун должен прибыть в Москву в конце июня для обсуждения подготовки Советско-Китайского Соглашения. Трумэн сообщил также, что Харли, посол США в Чунцине, дал указания поддержать советские предложения в этой связи.

Передаём Трумэну следующее послание от т.Сталина:

"ЛИЧНО И СТРОГО СЕКРЕТНО ОТ ПРЕМЬЕР-МИНИСТРА И.В.СТАЛИНА ПРЕЗИДЕНТУ Р-НУ Г.ТРУМЭНУ

Получая Ваше послание по поводу подготовки Советско-Китайского Соглашения и Вашего поручения г-ну Харли, выполняю Ваше пожелание Вам мира.

15 июня 1945 года".

Исполнение телеграфьте.

М О Л О Т О В

СТАЛИНУ
МОЛОТОВУ
10 ОТДЕЛ

Ован.Т.Т-Вижинского, Домановова.

Верно: [illegible]

APPLICATION FOR 1977 ASTM MEMBERSHIP

Application is made for a Membership in the American Society for Testing and Materials

☐ Organizational — an individual, business, governmental, research or professional organization, or trade association, or separate facility thereof subscribing to the purposes of the Society provided in the Charter and Bylaws.

☐ Member — an individual or an institution (educational; public library, or a scientific, engineering, or technical non-profit society) subscribing to the purposes of the Society provided in the Charter and Bylaws.

☐ Sr. Member — an individual fully retired, who has been a member of the Society and/or a participant in Society activities for a total of ten years or more.

PLEASE COMPLETE (Print or type)

Member Name (last name first)

Facility or Parent Company | code | phone number

Street Address } if home address, please indicate

City, State, ZIP CODE }

Official Representative (organizational membership only)

If currently serving on ASTM committees, please list:

Please send information on the following committees:

Major Product (or service performed by your organization)

Individual's major field of interest

Signature

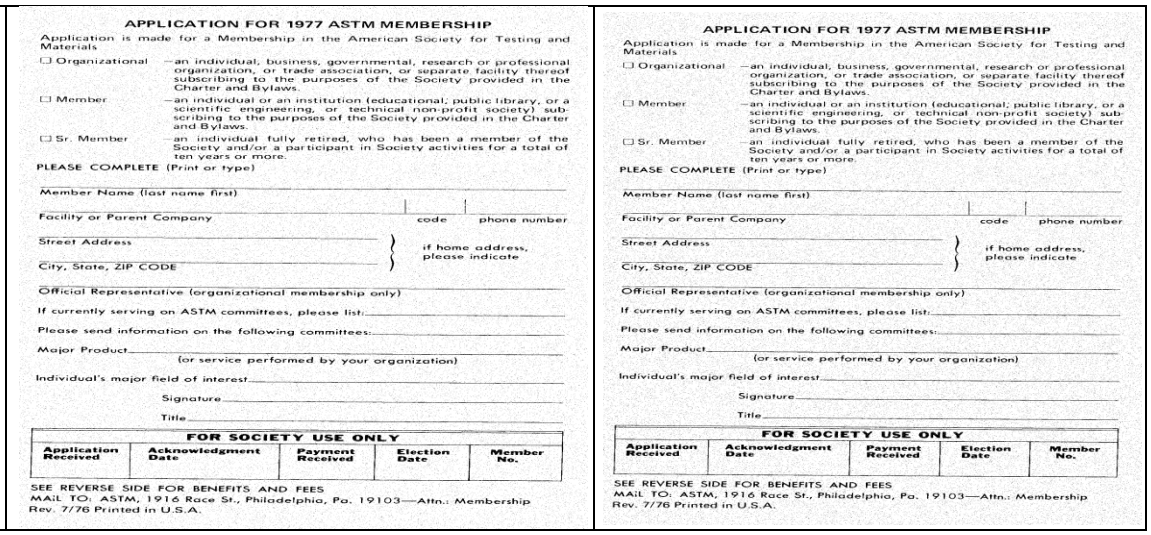
Title

FOR SOCIETY USE ONLY

Application Received	Acknowledgment Date	Payment Received	Election Date	Member No.

SEE REVERSE SIDE FOR BENEFITS AND FEES
MAIL TO: ASTM, 1916 Race St., Philadelphia, Pa. 19103—Attn.: Membership
Rev. 7/76 Printed in U.S.A.

5.



APPLICATION FOR 1977 ASTM MEMBERSHIP

Application is made for a Membership in the American Society for Testing and Materials

☐ Organizational — an individual, business, governmental, research or professional organization, or trade association, or separate facility thereof subscribing to the purposes of the Society provided in the Charter and Bylaws.

☐ Member — an individual or an institution (educational; public library, or a scientific engineering, or technical non-profit society) subscribing to the purposes of the Society provided in the Charter and Bylaws.

☐ Sr. Member — an individual fully retired, who has been a member of the Society and/or a participant in Society activities for a total of ten years or more.

PLEASE COMPLETE (Print or type)

Member Name (last name first)

Facility or Parent Company — code — phone number

Street Address } if home address, please indicate

City, State, ZIP CODE }

Official Representative (organizational membership only)

If currently serving on ASTM committees, please list:

Please send information on the following committees:

Major Product (or service performed by your organization)

Individual's major field of interest

Signature

Title

FOR SOCIETY USE ONLY

Application Received	Acknowledgment Date	Payment Received	Election Date	Member No.

SEE REVERSE SIDE FOR BENEFITS AND FEES
MAIL TO: ASTM, 1916 Race St., Philadelphia, Pa. 19103—Attn.: Membership
Rev. 7/76 Printed in U.S.A.

APPLICATION FOR 1977 ASTM MEMBERSHIP

Application is made for a Membership in the American Society for Testing and Materials

☐ Organizational — an individual, business, governmental, research or professional organization, or trade association, or separate facility thereof subscribing to the purposes of the Society provided in the Charter and Bylaws.

☐ Member — an individual or an institution (educational; public library, or a scientific engineering, or technical non-profit society) subscribing to the purposes of the Society provided in the Charter and Bylaws.

☐ Sr. Member — an individual fully retired, who has been a member of the Society and/or a participant in Society activities for a total of ten years or more.

PLEASE COMPLETE (Print or type)

Member Name (last name first)

Facility or Parent Company — code — phone number

Street Address } if home address, please indicate

City, State, ZIP CODE }

Official Representative (organizational membership only)

If currently serving on ASTM committees, please list:

Please send information on the following committees:

Major Product (or service performed by your organization)

Individual's major field of interest

Signature

Title

FOR SOCIETY USE ONLY

Application Received	Acknowledgment Date	Payment Received	Election Date	Member No.

SEE REVERSE SIDE FOR BENEFITS AND FEES
MAIL TO: ASTM, 1916 Race St., Philadelphia, Pa. 19103—Attn.: Membership
Rev. 7/76 Printed in U.S.A.

Figure 4: Output images of Souvola's approach of binarization

6. CONCLUSION

In this research work, a well known Souvola's binarization algorithm for optical character recognition has been parallelized and achieved an average speed-up of 20.8x. The implementation of binarization algorithm on the graphics device is promising with large two dimensional degraded document images.

CUDA itself has been shown to be an excellent framework to accelerate computational problems of OCR systems for handling large size documents. A fast OCR can be designed using the parallel implementation on GPUs.

7. REFERENCES

[1] He, J., Do, Q. D. M, Downton, and Kim, J. H. 2005. A comparison of binarization methods for historical archive documents. In proceeding of Eighth International Conference on Document Analysis and Recognition (ICDAR'05), 538-542.

[2] Fung, J. and Man, S. 2005. OpenVIDIA: Parallel GPU computer vision. In Proceedings of ACM International Conference on Multimedia, 849-852.

[3] Fernando, R and Kilgard, M. J. 2003. The Cg tutorial the definitive guide to programmable real-time graphics. Addison-Wesley.

[4] Otsu, N. 1979. A threshold selection method from gray level histograms. IEEE Trans. on Systems, Man and Cybernetics, Vol. 9, 62-66.

[5] Yu, B., Jain, A. and Mohiuddin, M. 1997. Address block location on complex mail Pieces," In Proceeding of International Conference of Document Analysis and Recognition, IEEE, 897-901.

[6] Rosenfeld, A. and Kak, A.C. 1982. Digital picture processing, second ed., Academic Press, New York.

[7] Kittler J. and Illingworth J. 1985. On threshold selection using clustering criteria. IEEE Trans. Systems Man Cybernetics, Vol. 15, 652–655.

[8] Brink, A.D. 1992. Thresholding of digital images using two-dimensional entropies. Pattern Recognition, Vol. 25, 803–808.

[9] Yan, H. 1996. Unified formulation of a class of image thresholding techniques. Pattern Recognition, Vol. 29, 2025–2032.

[10] Bernsen, J. 1986. Dynamic thresholding of grey-level images. In Proceeding of International Conference of Pattern Recognition, 1251-1255.

[11] Niblack, W. 1986. An Introduction to digital image processing, Prentice-Hall, Englewood Cliffs, NJ, 115–116.

[12] Sauvola, J. and Pietikainen, M. 2000. Adaptive document image binarization. Pattern Recognition, Vol. 33, 225–236.

[13] Kim, I.K., Jung, D.W. and Park, R.H. 2002. Document image binarization based on topographic analysis using a water flow model. Pattern Recognition, Vol. 35, 265–277.

[14] Gatos, B., Pratikakis, I. and Perantonis, S. J. 2006. Adaptive degraded document image binarization. Pattern Recognition, Vol. 39, 317–327.

[15] Chang, Y.F., Pai, Y.T. and Ruan, S.J. 2008. An efficient thresholding algorithm for degraded document images based on intelligent block detection. In Proceeding of IEEE International Conference on Systems, Man, and Cybernetics,667-672.

[16] Valizadeh, M., Komeili, M., Armanfard, N. and Kabir, E. 2009. Degraded document image binarization based on combination of two complementary algorithms. In Proceeding of International Conference of Advances in Computational Tools for Engineering Applications, IEEE, 595-599.

[17] Moravanszky, A. 2003. Linear algebra on the GPU, in: W.F. Engel (Ed.), Shader X 2, Wordware Publishing, Texas.

[18] Manocha, D. 2003. Interactive geometric & scientific computations using graphics hardware, SIGGRAPH 2003 Tutorial Course #11.

[19] Moreland, K. and Angel E. 2003. The FFT on a GPU. In Proceedings of SIGGRAPH Conference on Graphics Hardware, 112-119.

[20] Mairal, J., Keriven, R. and Chariot, A. 2006. Fast and efficient dense variational Stereo on GPU. In Proceedings of International Symposium on 3D Data Processing, Visualization, and Transmission, 97-704.

[21] Yang, R. and Welch, G. 2002. Fast image segmentation and smoothing using commodity graphics hardware. Journal of Graphics Tools, Vol. 17, (4), 91-100.

[22] Owens, J. D., Luebke, D., Govindaraju, N., Harris, M., Kruger, J., Lefohn, A. E. and Purcell, T. J. 2005. A survey of general-purpose computation on graphics hardware. In proceeding of Eurographics, State of the Art Reports, 21–51.

[23] Larsen, E. S., McAllister, D. 2001. Fast Matrix Multiplies using Graphics Hardware. In Proceeding of International Conference for High Performance Computing and Communications, 159-168.

[24] Trendall C. and Stewart, A. J. 2000. General calculations using graphics hardware with applications to interactive caustics. Rendering Techniques 2000: 11th Eurographics Workshop on Rendering, 287-298.

[25] Li, Wei, Wei, Xiaoming, A. and Kaufman, 2001. Implementing lattice boltzmann computation on graphics hardware. In proceeding of the International Conference for High Performance Computing and Communications.

[26] Mizukami, Y., Koga, K. and Torioka, T. 1994. A handwritten character recognition system using hierarchical extraction of displacement. IEICE, J77-D-II(12):2390–2393.

[27] Kruger, J. and Westermann, R. 2003. Linear operators for GPU implementation of numerical algorithms. In Proceedings of SIGGRAPH, San Diego, 908- 916.

[28] Steinkraus, D., Buck, I., and Simard, P. Y. 2005. GPUs for machine learning algorithms. In proceeding of International Conference of Document Analysis and Recognition, 1115-1120.

[29] Mizukami, Y. and Koga, K. 1996. A handwritten character recognition system using hierarchical displacement extraction algorithm. In Proceeding of International Conference of Pattern Recognition, volume 3,160–164.

[30] Ilie, A. Optical character recognition on graphics hardware. Downloaded from www.cs.unc.edu/~adyilie/IP/Final.pdf

[31] Oh, K.S. and Jung, K. 2004. GPU implementation of neural networks. Pattern Recognition, Elsevier, 1311-1314.

[32] Jung, K. 2001. Neural Network-based text localization in color images. Pattern Recognition Letters, Vol. 22, (4), 1503- 1515.

[33] Singh, B.M., Mittal A., and Ghosh, D. 2011. Parallel implementation of Devanagari text line and word segmentation approach on GPU. International Journal of Computer Applications 24(9):7–14.

[34] NVIDIA CUDA Programming Guide Version 2.0, available at www.nvidia.com/object/cuda_develop.html.

[35] NVIDIA Corporation: NVIDIA CUDA programming guide. Jan 2007, available at http://developer.download.nvidia.com/compute/cuda/2_0/docs/NVIDIA_CUDA_Programming_Guide_2.0.pdf