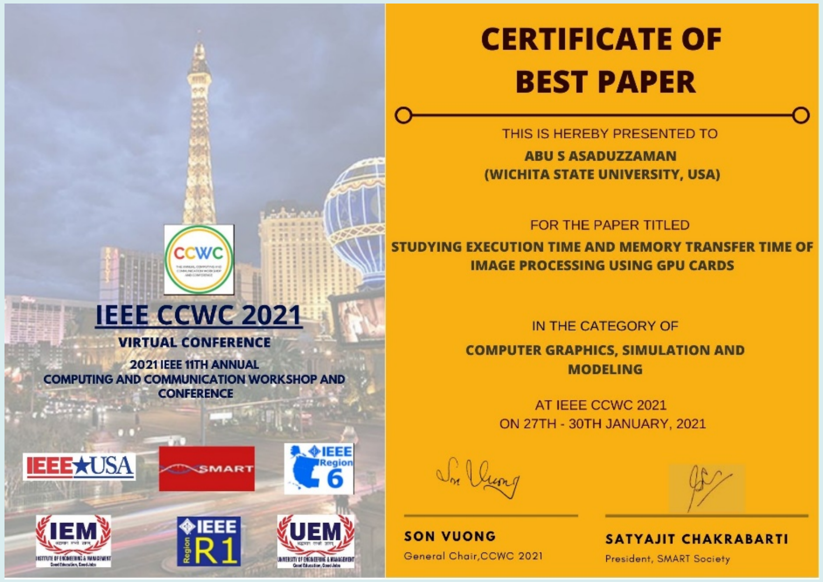
Asaduzzaman, A., Jojigiri, S., Sabu, T., and Tailam, S., “Studying Execution Time and Memory Transfer Time of Image Processing Using GPU Cards,” accepted in IEEE Computing and Communication Workshop and Conference (CCWC-2021), Virtual Conference, USA, Jan. 27-30, 2021. (Received the Best Paper award.)


IEEE CCWC 2021
VIRTUAL CONFERENCE
2021 IEEE 11TH ANNUAL
COMPUTING AND COMMUNICATION WORKSHOP AND CONFERENCE

CERTIFICATE OF BEST PAPER

THIS IS HEREBY PRESENTED TO
ABU S ASADUZZAMAN
(WICHITA STATE UNIVERSITY, USA)

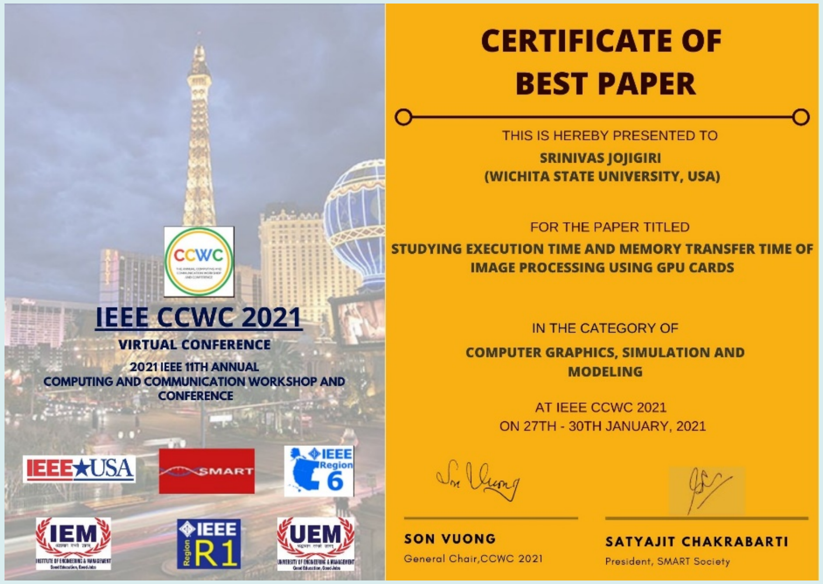
FOR THE PAPER TITLED
STUDYING EXECUTION TIME AND MEMORY TRANSFER TIME OF IMAGE PROCESSING USING GPU CARDS

IN THE CATEGORY OF
COMPUTER GRAPHICS, SIMULATION AND MODELING

AT IEEE CCWC 2021
ON 27TH - 30TH JANUARY, 2021

SON VUONG
General Chair,CCWC 2021

SATYAJIT CHAKRABARTI
President, SMART Society


IEEE CCWC 2021
VIRTUAL CONFERENCE
2021 IEEE 11TH ANNUAL
COMPUTING AND COMMUNICATION WORKSHOP AND CONFERENCE

CERTIFICATE OF BEST PAPER

THIS IS HEREBY PRESENTED TO
SRINIVAS JOJIGIRI
(WICHITA STATE UNIVERSITY, USA)

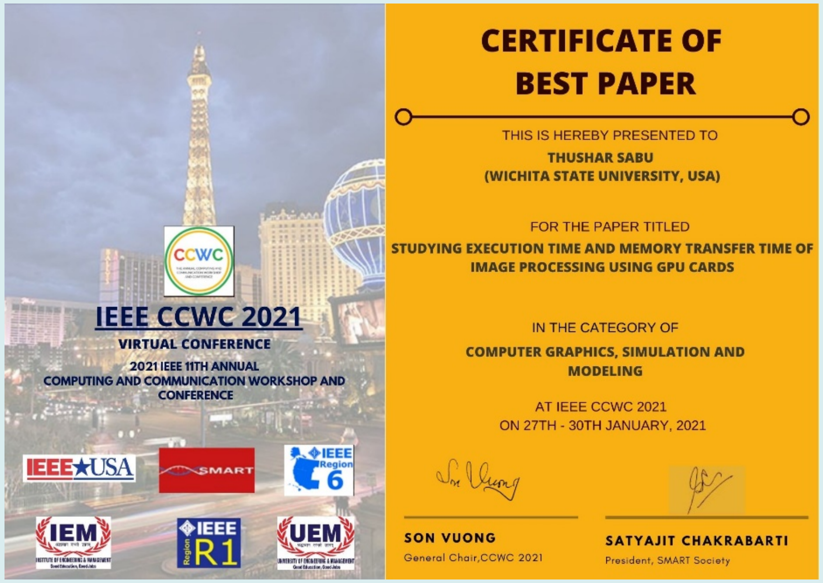
FOR THE PAPER TITLED
STUDYING EXECUTION TIME AND MEMORY TRANSFER TIME OF IMAGE PROCESSING USING GPU CARDS

IN THE CATEGORY OF
COMPUTER GRAPHICS, SIMULATION AND MODELING

AT IEEE CCWC 2021
ON 27TH - 30TH JANUARY, 2021

SON VUONG
General Chair,CCWC 2021

SATYAJIT CHAKRABARTI
President, SMART Society


IEEE CCWC 2021
VIRTUAL CONFERENCE
2021 IEEE 11TH ANNUAL
COMPUTING AND COMMUNICATION WORKSHOP AND CONFERENCE

CERTIFICATE OF BEST PAPER

THIS IS HEREBY PRESENTED TO
THUSHAR SABU
(WICHITA STATE UNIVERSITY, USA)

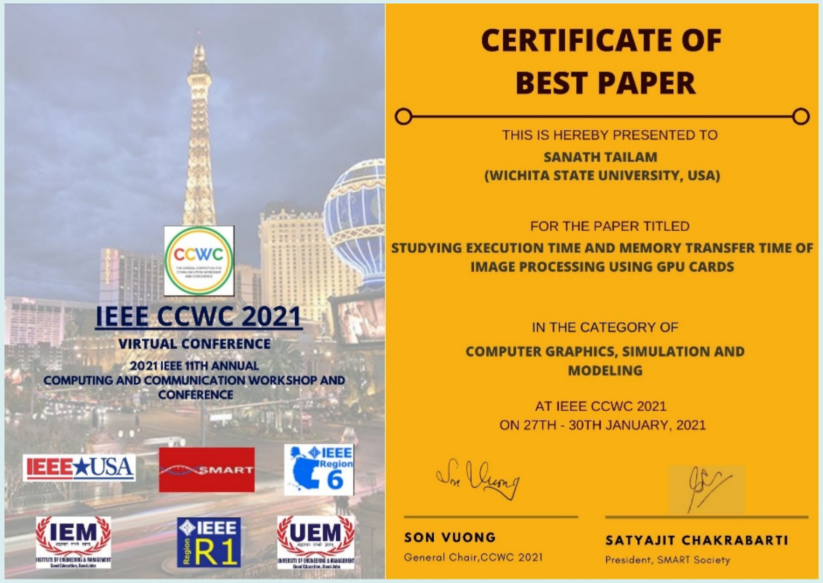
FOR THE PAPER TITLED
STUDYING EXECUTION TIME AND MEMORY TRANSFER TIME OF IMAGE PROCESSING USING GPU CARDS

IN THE CATEGORY OF
COMPUTER GRAPHICS, SIMULATION AND MODELING

AT IEEE CCWC 2021
ON 27TH - 30TH JANUARY, 2021

SON VUONG
General Chair,CCWC 2021

SATYAJIT CHAKRABARTI
President, SMART Society


IEEE CCWC 2021
VIRTUAL CONFERENCE
2021 IEEE 11TH ANNUAL
COMPUTING AND COMMUNICATION WORKSHOP AND CONFERENCE

CERTIFICATE OF BEST PAPER

THIS IS HEREBY PRESENTED TO
SANATH TAILAM
(WICHITA STATE UNIVERSITY, USA)

FOR THE PAPER TITLED
STUDYING EXECUTION TIME AND MEMORY TRANSFER TIME OF IMAGE PROCESSING USING GPU CARDS

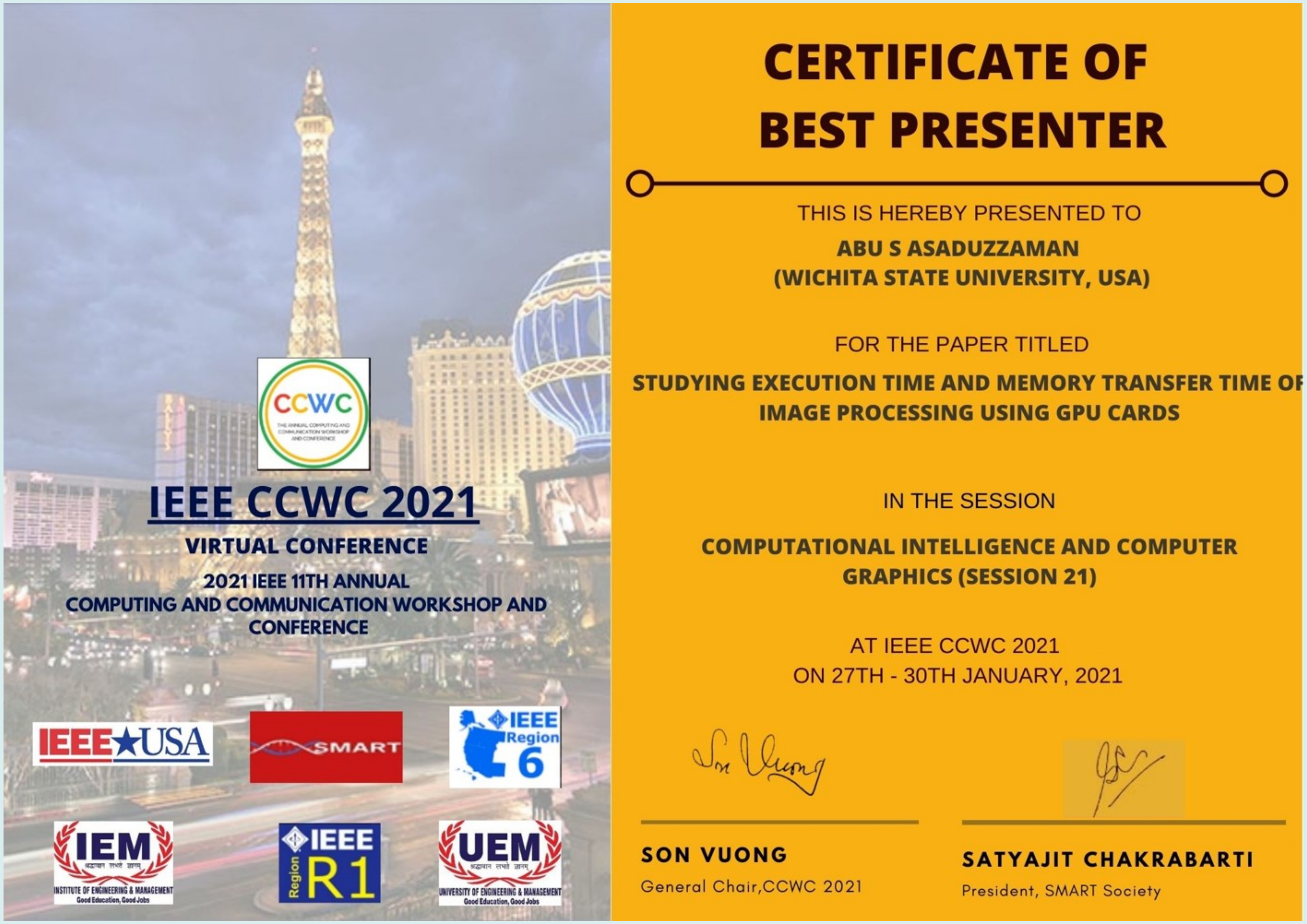
IN THE CATEGORY OF
COMPUTER GRAPHICS, SIMULATION AND MODELING

AT IEEE CCWC 2021
ON 27TH - 30TH JANUARY, 2021

SON VUONG
General Chair,CCWC 2021

SATYAJIT CHAKRABARTI
President, SMART Society

Asaduzzaman, A., Jojigiri, S., Sabu, T., and Tailam, S., “Studying Execution Time and Memory Transfer Time of Image Processing Using GPU Cards,” accepted in IEEE Computing and Communication Workshop and Conference (CCWC-2021), Virtual Conference, USA, Jan. 27-30, 2021. (Received the Best Presenter award.)



IEEE CCWC 2021

VIRTUAL CONFERENCE

2021 IEEE 11TH ANNUAL
COMPUTING AND COMMUNICATION WORKSHOP AND CONFERENCE

CERTIFICATE OF
BEST PRESENTER

THIS IS HEREBY PRESENTED TO

ABU S ASADUZZAMAN
(WICHITA STATE UNIVERSITY, USA)

FOR THE PAPER TITLED

STUDYING EXECUTION TIME AND MEMORY TRANSFER TIME OF
IMAGE PROCESSING USING GPU CARDS

IN THE SESSION

COMPUTATIONAL INTELLIGENCE AND COMPUTER GRAPHICS (SESSION 21)

AT IEEE CCWC 2021
ON 27TH - 30TH JANUARY, 2021

[illegible] [illegible]

SON VUONG
General Chair,CCWC 2021

SATYAJIT CHAKRABARTI
President, SMART Society

IEEE CCWC 2021
Virtual Conference
January 27-30, 2021

"Studying Execution Time and Memory Transfer Time of Image Processing Using GPU Cards"

Presenter:
Abu Asaduzzaman, Associate Professor

Authors:
Abu Asaduzzaman, Srinivas Jojigiri, Thushar Sabu, and Sanath Tailam
Wichita State University, USA

WICHITA STATE UNIVERSITY

CAPPLab
capplab@wichita.edu

"Studying Execution Time and Memory Transfer Time of Image Processing Using GPU Cards"

Outline

- **Introduction**
  - Image Processing Algorithms
  - GPU Cards
  - Problem Description, Contribution
- **Related Work**
  - GPU Memory: Latency Ranking
  - GPU for Image Processing
- **Algorithms Considered**
  - Sobel Filter with Bilateral Filter
  - Canny Edge Detector
- **Results and Discussion**
  - Execution Time
  - Memory Transfer Time
- **Conclusion**

Courtesy of
http://remananr.com/Blog/wp-content/uploads/2018/09/NVIDIA_CUDA_Logo_White.jpg and http://cdn.wccftech.com/wp-content/uploads/2013/07/NVIDIA-Tesla.jpg

CAPPLab
capplab@wichita.edu

“Studying Execution Time and Memory Transfer Time of Image Processing Using GPU Cards”

Introduction

Image Processing Algorithms

- There are many applications of image processing in many fields such as medicine, astronomy, etc.
- Important algorithms used for edge detection (i.e., image processing) include Sobel filter, Bilateral filter, and Canny edge detector [1-2].
- The Sobel filter is less computation intensive because it is based on convolving the image with a small and integer valued filter in horizontal and vertical directions.
- The Bilateral filter is an edge preserving and smoothing filter that replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels.
- Although the Canny edge detector is much more complex than the Sobel and Bilateral filters, the Canny detector is considered as the most popular and most effective edge detecting method.

[1] N. Tsankashvili, “Comparing Edge Detection Methods,” 2018. https://medium.com/@nikatsanka/comparing-edge-detection-methods-638a2919476e

[2] D. Demirovic, E. Skejic, and A. Serifovic–Trbalic, “Performance of Some Image Processing Algorithms in Tensorflow,” IEEE International Conference on Systems, Signals and Image Processing (IWSSIP), 2018.

"Studying Execution Time and Memory Transfer Time of Image Processing Using GPU Cards"

Introduction

GPU Cards | Problem Description

- Due to issues such as data dependency, synchronization, and communication, multicore central processing units (CPUs) cannot speed up as required for image processing.
- After successful implementation of vector and matrix operations on graphics processing units (GPUs), GPU cards gain more popularity in scientific research.
- Studies indicate that the performance of image processing and edge detection techniques varies when the GPU cards are changed [3-6].
- In this work, two imaging algorithms (Sobel/Bilateral filter and Canny edge detector) are implemented using three GPU cards (C2075, GTX-965, and K20) to study execution time and memory overhead.

[3] N. Zhang, Y.-S. Chen, and J.-L. Wang, "Image parallel processing based on GPU," IEEE International Conference on Advanced Computer Control (ICACC), Vol. 3, 2010.

[4] J.H. Jo and S.G. Lee, "Sobel mask operations using shared memory in CUDA environment," IEEE International Conference on New Trends on Information Science and Service Science and Data Mining (ISSDM), pp. 289-292, 2012.

[5] M. Rumanek, T. Danek, and A. Lesniak, "High performance image processing of satellite images using graphics processing units," IEEE International Geoscience and Remote Sensing Symposium, 2011.

[6] A.S. Ladkat, A.A. Date, and S.S. Inamdar, "Development and comparison of serial and parallel image processing algorithms," IEEE International Conference on Inventive Computation Technologies (ICICT), 2016.

“Studying Execution Time and Memory Transfer Time of Image Processing Using GPU Cards”

Related Work

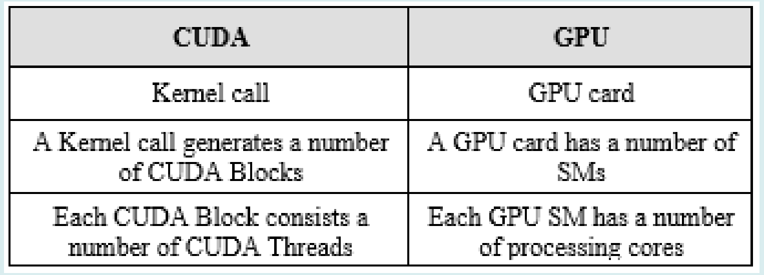
GPU Memory: Latency Ranking

- A kernel call occupies a GPU card as shown in Table I.

  A CUDA kernel → CUDA blocks

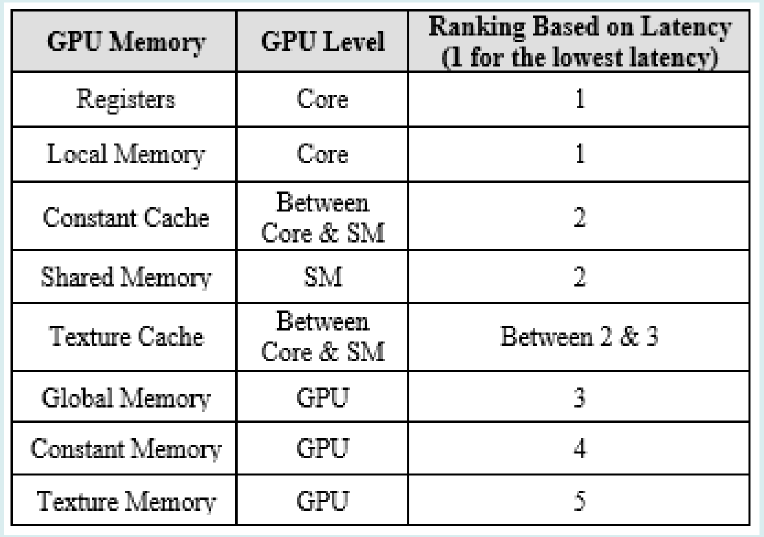
  CUDA blocks → CUDA threads
- Table II summarizes various types of GPU memory, the level (i.e., location) of their existence in the GPU card, and their ranking based on latency.
- It is time-efficient to use data from the shared memory instead of global and/or texture memory.
- It is important to note that the GPU performance depends on how the CUDA program uses the GPU memory system.

Table I. CUDA Kernel and GPU Card



CUDA	GPU
Kernel call	GPU card
A Kernel call generates a number of CUDA Blocks	A GPU card has a number of SMs
Each CUDA Block consists a number of CUDA Threads	Each GPU SM has a number of processing cores

Table II. GPU Memory Latency Ranking



GPU Memory	GPU Level	Ranking Based on Latency (1 for the lowest latency)
Registers	Core	1
Local Memory	Core	1
Constant Cache	Between Core & SM	2
Shared Memory	SM	2
Texture Cache	Between Core & SM	Between 2 & 3
Global Memory	GPU	3
Constant Memory	GPU	4
Texture Memory	GPU	5

“Studying Execution Time and Memory Transfer Time of Image Processing Using GPU Cards”

Related Work

GPU for Image Processing

- There are earlier work that focused on evaluating and improving image processing algorithms using CUDA (i.e., GPU cards) [7-9].
- Jo et al. implemented the Sobel mask operations using GPU shared memory [7]. The implemented algorithm minimized the number of accesses of the global memory and improved the speedup factor about 30% than the conventional algorithms.
- Sanida et al. implemented the Sobel filter using GPU and OpenCL [10]. They achieved speedup improvement up to 1546x with 3×3 convolution kernels.

[7] J.H. Jo and S.G. Lee, “Sobel mask operations using shared memory in CUDA environment,” IEEE International Conference on New Trends on Information Science and Service Science and Data Mining (ISSDM), pp. 289-292, 2012.
[8] E. Young and F. Jargstorff, “Image Processing and Video Algorithms with CUDA,” NVIDIA nvision, 2008. https://developer.download.nvidia.com/presentations/2008/NVISION/NVISION08_ImageVideoCUDA_web.pdf
[9] K. Karimi, N. Dickson, and F. Hamze, “A Performance Comparison of CUDA and OpenCL,” 2010. https://arxiv.org/abs/1005.2581
[10] T. Sanida, A. Sideris, and M. Dasygenis, “A Heterogeneous Implementation of the Sobel Edge Detection Filter Using OpenCL,” IEEE International Conference on Modern Circuits and Systems Technologies (MOCAST), 2020.

"Studying Execution Time and Memory Transfer Time of Image Processing Using GPU Cards"

Algorithms Considered

Sobel Filter with Bilateral Filter

- Sobel/Bilateral filter starts with decoding the image data and taking the pixel values into an array. Figure 1 illustrates the workflow.
- The first kernel call applies the Bilateral filter to the image data that can smooth out the noise without effecting edges.
- The denoised data from the Bilateral filter is used by the Sobel filter.
- The second kernel call applies the Sobel filter to get the edges of the image.
- The result data is written to the buffer in the GPU global memory.
- The execution time and memory transfer time are calculate.

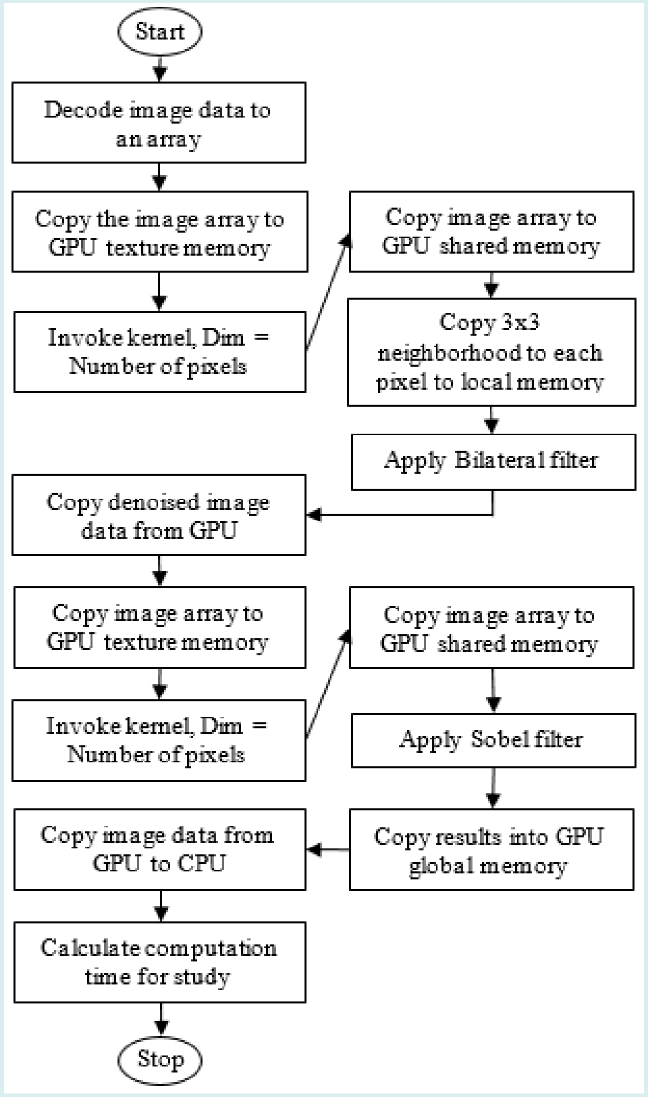

Start
Decode image data to an array
Copy the image array to GPU texture memory
Invoke kernel, Dim = Number of pixels
Copy image array to GPU shared memory
Copy 3x3 neighborhood to each pixel to local memory
Apply Bilateral filter
Copy denoised image data from GPU
Copy image array to GPU texture memory
Copy image array to GPU shared memory
Invoke kernel, Dim = Number of pixels
Apply Sobel filter
Copy image data from GPU to CPU
Copy results into GPU global memory
Calculate computation time for study
Stop

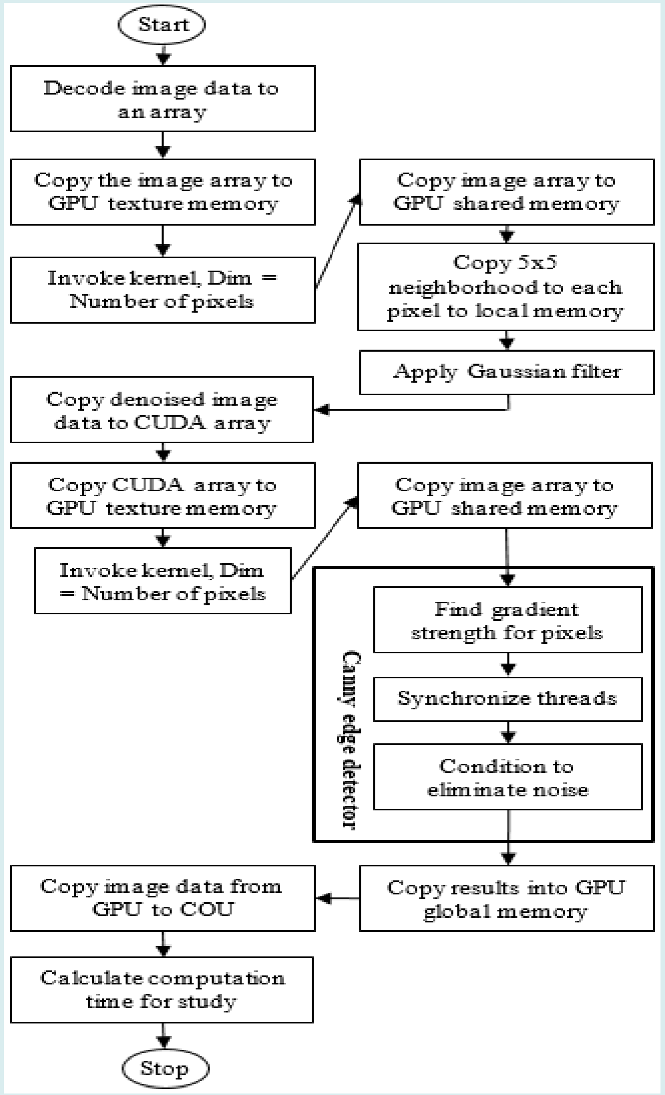
Fig. 1. Flowchart for Sobel/Bilateral filter.

"Studying Execution Time and Memory Transfer Time of Image Processing Using GPU Cards"

Algorithms Considered

Canny Edge Detector

- Canny detector starts with decoding the image data and take the pixel values into an array. Figure 2 illustrates the workflow.
- The first kernel call applies the Gaussian filter to reduce noise without effecting edges.
- The denoised data from the Gaussian filter is used by the Canny detector.
- The second kernel call applies the Canny edge detector to get the edges of the image.
- The result data is written to the buffer in the GPU global memory.
- The execution time and memory transfer time are calculate.



Start
Decode image data to an array
Copy the image array to GPU texture memory
Invoke kernel, Dim = Number of pixels
Copy image array to GPU shared memory
Copy 5x5 neighborhood to each pixel to local memory
Apply Gaussian filter
Copy denoised image data to CUDA array
Copy CUDA array to GPU texture memory
Invoke kernel, Dim = Number of pixels
Copy image array to GPU shared memory
Canny edge detector
Find gradient strength for pixels
Synchronize threads
Condition to eliminate noise
Copy results into GPU global memory
Copy image data from GPU to COU
Calculate computation time for study
Stop

Fig. 2. Flowchart for Canny Edge Detector.

"Studying Execution Time and Memory Transfer Time of Image Processing Using GPU Cards"

Outline

- Introduction
  - Image Processing Algorithms
  - GPU Cards
  - Problem Description, Contribution
- Related Work
  - GPU Memory: Latency Ranking
  - GPU for Image Processing
- Algorithms Considered
  - Sobel Filter with Bilateral Filter
  - Canny Edge Detector
- **Results and Discussion**
  - Execution Time
  - Memory Transfer Time
- **Conclusion**

NVIDIA CUDA

Courtesy of http://remananr.com/Blog/wp-content/uploads/2018/09/NVIDIA_CUDA_Logo_White.jpg and http://cdn.wccftech.com/wp-content/uploads/2013/07/NVIDIA-Tesla.jpg

WSU CAPPLab capplab@wichita.edu

“Studying Execution Time and Memory Transfer Time of Image Processing Using GPU Cards”

Results and Discussion

Execution Time

- For the Sobel filter with 10x10^7 pixels, K20 provides the best speedup (K20 more than 300x, GTX-965 about 260x, and C2075 about 190x) as shown in Figure 3.
- For the Canny detector with 10x10^7 pixels, GTX-965 provides the best speedup (GTX-965 almost 120x, C2075 about 50x, and K20 about 45x) as shown in Figure 4.

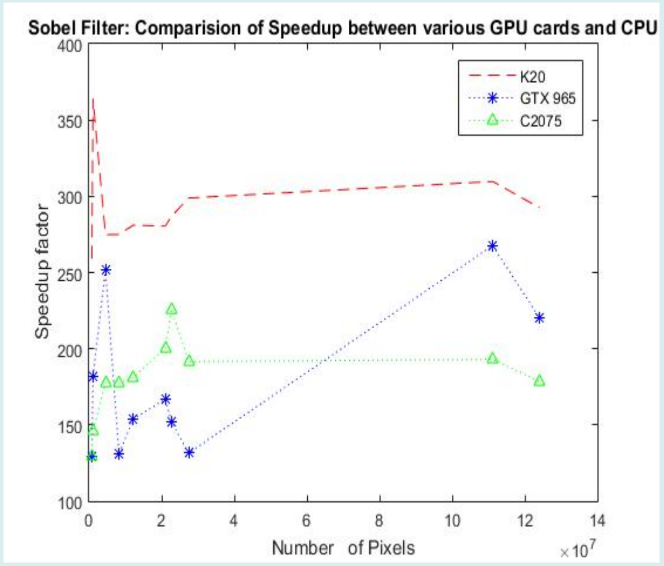


Fig. 3. Speedup due to Sobel filter.

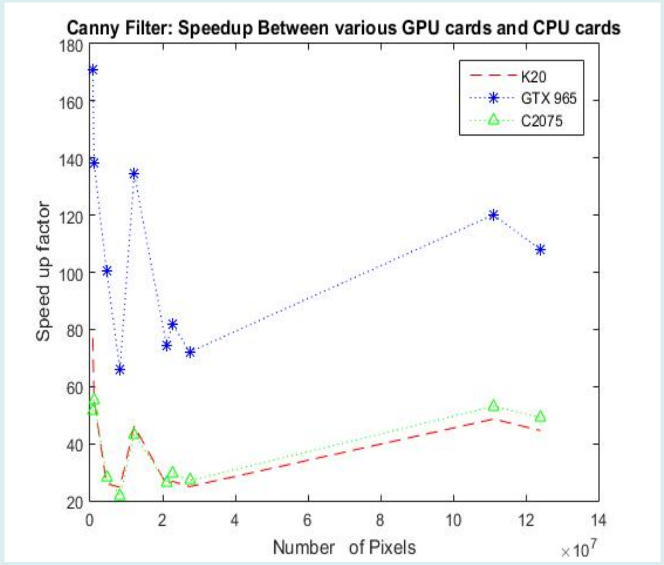


Fig. 4. Speedup due Canny detector.

Note 1: Intel i5-4210H CPU is used. It has Turbo boost technology enabled, up to four CPU threads.
Note 2: GPU cards show significant performance improvements over CPU-only implementations (for 10x10^7 pixels, CPU-only takes more than 73 seconds, C2075 takes 4 seconds, and GTX965/K20 takes about 3 seconds).

"Studying Execution Time and Memory Transfer Time of Image Processing Using GPU Cards"

Results and Discussion

Memory Transfer Time

- For Sobel filter with 10x10^7 pixels, memory transfer time due to GTX-965, C2075, and K20 are 0.21 sec, 0.35 sec, and 0.45 sec, respectively, as shown in Figure 5.

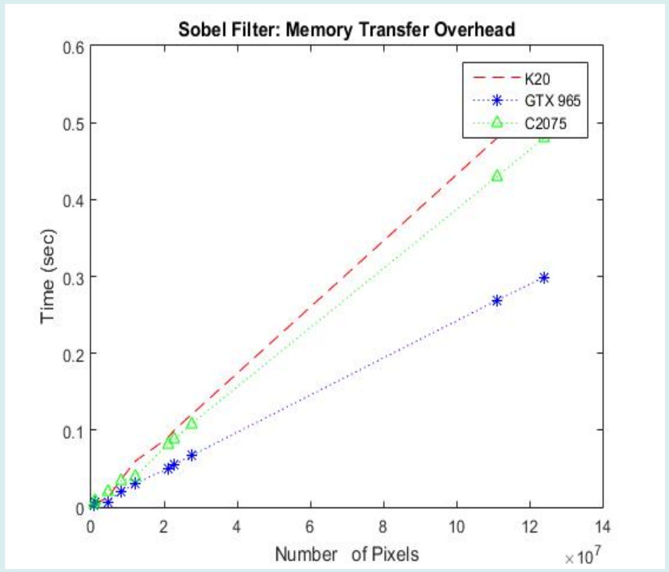


Fig. 5. Memory overhead due to Sobel filter.

- For Canny detector with 10x10^7 pixels, memory transfer time due to C2075, GTX-965, and K20 are 0.22 sec, 0.23 sec, and 0.30 sec, respectively, as shown in Figure 6.

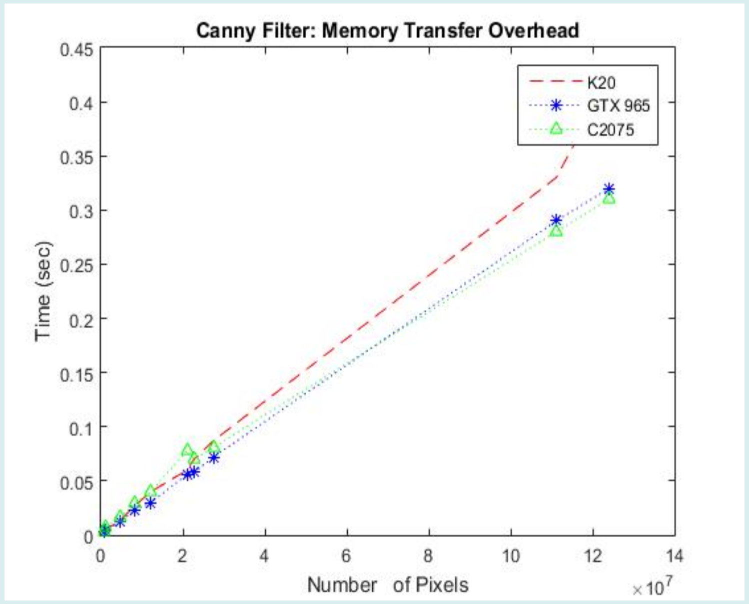


Fig. 6. Memory overhead due to Canny detector.

“Studying Execution Time and Memory Transfer Time of Image Processing Using GPU Cards”

Results and Discussion

(Cont’d…)

- Considering the execution time speedup, the Canny detector does not perform as well as the Sobel filter does. This is because
  - the Canny detector uses data dependent algorithm, and it requires synchronization of threads that requires more time to execute.
  - It should be noted that the Sobel filter does not need thread synchronization.
- Another issue we encounter is that the data transfer time from GPU card to CPU after applying noise filter is different and it has larger impact on Kepler GPU card when compared to that on Maxwell GPU card.

“Studying Execution Time and Memory Transfer Time of Image Processing Using GPU Cards”

Conclusion

- This work compares the execution time and memory transfer time due to three GPU cards (namely Tesla/Fermi C2075, GeForce/Maxwell GTX-965, and Tesla/Kepler K20) on two popular image algorithms (namely Sobel filter along with Bilateral filter and Canny edge detector).
- According to simulation results, K20 provides the best execution time speedup (more than 300x for K20, about 250x for GTX-965, and less than 200x for C2075) for the Sobel filter.
- However, GTX-965 provides the best speedup (almost 120x for GTX-965, about 45x for C2075, and more than 40x for K20) for the Canny detector.
- Simulation results show that GTX-965 performs better or about the same as C2075 for both algorithms (Sobel filter and Canny detector).
- The memory transfer time due to GTX-965 is minimum (GTX-965 takes about 0.21 sec and K20 takes about 0.45 sec) for the Sobel filter and for the Canny detector (GTX-965 takes about 0.23 sec and K20 takes about 0.30 sec).

IEEE CCWC 2021
Virtual Conference
January 27-30, 2021

"Studying Execution Time and Memory Transfer Time of Image Processing Using GPU Cards"

Contact:
Abu Asaduzzaman
Wichita State University
Abu.Asaduzzaman@wichita.edu
+1 (316) 978-5261

Thank You!

Wichita State University

CAPPLab
capplab@wichita.edu