ECE 5570 High Performance Computing

Project Proposal Specification

Instructor: Anthony Smith

Parallel Video Processing Techniques for Surveillance Applications

Pranoti Adekar

Nikhil Badri

Florida Institute of Technology

ABSTRACT

The main aim of the project is to compute the video processing for the surveillance using the GPU (parallel processing). This incorporates the background subtraction and motion detection. Then triggering an action for the motion detected in the video. This project is done on the NVIDIA graphics GPU, programmed in CUDA using visual studio API.

1. INTRODUCTION

Automated video surveillance is a research area growing in the commercial sector. Technology has reached a stage where exhibiting cameras capture video imagery is inexpensive, but finding available human resources to sit and monitor that imagery is expensive. The surveillance cameras are already prevalent in commercial institutions, with camera output being recorded to tapes that are either rewritten occasionally or stored in video archives. After an incident occurred –a car is stolen or a store is robbed – investigators can go after the fact to see what happened, but of course by then it is too late. The action here needed is continuous 24-hour monitoring and analysis of video surveillance data to alert the operator or security officers to a burglary in progress, or to a suspicious individual strolling in the parking lot, while options are still open for evading the crime.

Problem Statement. To identify the motion in the given video, there are calculations that involve finding the difference in the intensities of the previous frame pixel to the current frame pixel and then comparing with the threshold. This is time taking and slow process. To have faster calculations in order to reduce the processing time we need to incorporate this with Graphical Processing Unit (GPU).

Motivation. We are getting an opportunity to utilize the capabilities of the GPU and get to know how each thread processes a pixel in the frame. It also helps us to identify the adeptness difference between the CPU and the GPU.

Proposed Solution. We came up with a solution to use background subtraction in the CPU for the frame and programming the GPU to calculate the changes in motion vectors which indeed detects the motion and then triggers an alarm.

Contributions. An enumeration of the contributions of the senior design project

This project contributes to the following points:

- We are programming the GPU in such a way that it can detect motion in the video by tracing changes in every pixel of the video.
- We are testing it by comparing the original video with the motion vector video.

1.1 Design Goals

The target audience will be the traffic surveillance people and vehicles. The Audience may benefit by identifying the incidents\accidents in real-time. Most important goals of Security Surveillance are to collect and disseminate realtime information and provide situational awareness to operators and security analysts.

The features and functionality that are implemented in the project are as follows:

- **Action triggered when motion detected**: An alarm or a pop up occurs when motion is detected
- **Real Time Object detection**: It can detect a slight change in intensity value of the pixel
- **Provides object silhouette**: Provides a dark shape outline of the object in motion.

2. RELATED WORK

A framework was developed to explore and implement parallel video effects using a network of workstations distributed computing system. In some research the security management process was defined, structured and analyzed with the goal to further harden security relevant systems designed to be nearly indestructible.

Perhaps we are dealing with above research, we need to implement in parallel processing so that we can have improvement in processing time and performance.

2.1 Example Subsection Header

The bibliography for this project:

- [CLKFDTTEHBW00] "A System for Video Surveillance and Monitoring"
- [MMSKMP07] "Parallel Implementation of Video Surveillance Algorithms on GPU Architecture using CUDA"
- [DA14] "Parallel Video Processing Techniques for Surveillance Applications"
- [GW08] 'Digital Image Processing.
- [CW13] "An Efficient Acceleration of Digital Forensics Search Using GPGPU"
- [S08]" Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction"

3. PROJECT PROPOSAL

The project is about real-time video surveillance using parallel processing (using GPU).The technique of background subtraction is used to eliminate the background which makes object identification easy. The motion detection is done by the GPU as it a time taking process. We are also incorporating spatial domain filtering [Image processing and manipulation in the spatial domain] so each pixel can be processed independently and in parallel.

3.1 Anticipated Approach

Our project consists of two phases of implementation:

1. Background subtraction
2. Motion detection

Background Subtraction. It is the technique to extract the moving foreground from static background .It is a Gaussian Mixture Model based background/foreground segmentation algorithm. There are various algorithms for background subtraction and are as follows:

1. Background subtractor MOG
2. Background subtractor MOG2
3. Background subtractor GMG

Motion Detection. The algorithm for motion detection is coded in GPU as this includes multiple calculation to find changes in motion vectors. The algorithm contains calculation that needed to be done for every pixel in every frame, which is a tedious and time taking process. So by using GPU for this, we can improve the processing time i.e. the process can be made faster.

The algorithm for motion detection is:

Step 1: Keep track of the previous frame of the image to track the change.

Step 2: Calculate square difference (Previous Frame – Current Frame)

Step 3: If the difference is above a set threshold we fire motion detected event and activate sound, alarm, message etc.

The calculation involved for motion detection (i.e. the step 2 in the algorithm) in detail is given below:

1: Differentiate:

    Color_diff = Prev_Color – Current_Color

2: Square difference:

    Color_diff += Color_diff

3: Update previous color:

    Prev_Color = Current_Color

4: Adjust threshold value:

    Threshold = xxxx

5: Color_diff > threshold

    Fire 'Motion detected'

    Current_Color = RED

6: else leave unchanged

    Or make it grey

    Or do nothing

Spatial domain is a plane where a digital image is defined by the spatial co-ordinates of its pixel. It uses neighborhood processing. The digital image is defined by its decomposition into spatial frequency participating in its formation, because of the concept of processing a limited area of the input image, operations can be implemented separately here where parallelism comes into the picture.

Detection of objects, people, people faces, cars and their patterns of movement to enhance automated decision making and alarming. This is difficult to obtain, but it can be overcome by Background Subtraction.

The programming language that is used is an Application Programming Interface (API) provided by NVIDIA named CUDA (Compute Unified Device Architecture). CUDA can be used in simulation, genetic algorithms, processing digital forensics and other fields, each thread perform the same operation on a different set of data.

3.2 Target Platforms

The following are the hardware's that will be used in this project:

1. NVIDIA graphics GPU
2. Intel i7 processor
3. Video formats: MP4, MPEG, AVI etc.
4. 8GB RAM

3.3 Evaluation Criteria

The snapshot below is an example that demonstrates the background subtraction and motion vectors that detects the motion. In this snapshot, the frame is divided into two halves, one with the original frame and other involving the background subtraction. But in our project we are going to implement this for the whole frame. This project incorporates the technique of over the line motion detection this was done considering a scenario of the restricted area. For example: if the right side of the road is restricted and if there is any motion detected, the alarm will be given to the operator.

Specific to this project we have eliminated the concept of over the line motion detection as we need to incorporate motion detection on a complete snapshot of the image rather that particular section of an image, as there might not always be the scenario of the restricted area in the surveillance input. So we have extended our scope rather than narrowing it down to a specific portion of the video.

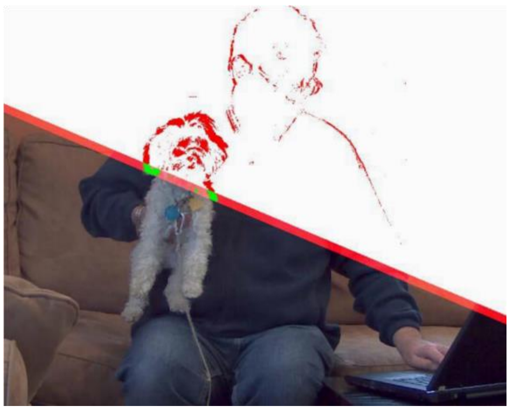


Figure 1: ***Snapshot showing Motion Detected above a user specified line and on the line.***

4. RESEARCH TIMELINE

We have gone through various papers that integrate algorithms that we have mentioned in this document. We have also planned on framework of what should be implemented using what hardware and codes so that the code debugging can go smoothly.

Gantt chart:

Time line chart also called as Gantt chart. It is a type of bar **chart**, developed by Henry **Gantt** in the 1910s, that illustrates a project schedule. **Gantt charts** illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project.

The below figure 2 elaborates on the timeline related to this project.

Project Milestone Report (Alpha Version)

- Completed all background reading – Pranoti & Nikhil
- Proposed software framework is functioning with simple base case – Pranoti & Nikhil
- Collected all necessary data and images – Pranoti & Nikhil

Project Final Deliverables

List what you will deliver at the end of the semester

- **Software having particular features:** Microsoft Visual Studio with CUDA code for motion detection.
- Demonstrating the project (aim, proposal, and solution) and its flow (planning, time line).
- Demonstrating the project with the result by comparing the original video with the output (motion vector) video with background subtraction.
- Documentation of the project with the result and conclusion.

Project Future Tasks

- We would incorporate and test the project with real time surveillance. Also would go with the scalability and implement the edge detection, sharpening the video with high definition.
- Would likely to extend this for the remote surveillance. Face detection in the video.

Method

The above proposed method, algorithms were used for implementing the project. A minor changes from the proposed method were made in order to complete the project as per the schedule. The changes made includes,

A. We decided to mark the pixel in red shade instead of alarm or pop-up for the motion detected.

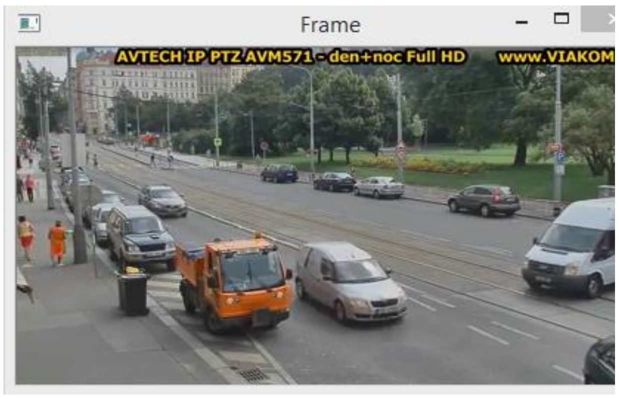
B. Our reference paper implements the surveillance for a part of the frame, but we considered the whole frame and implemented.

The key steps or the main algorithm involved in our project are as below:

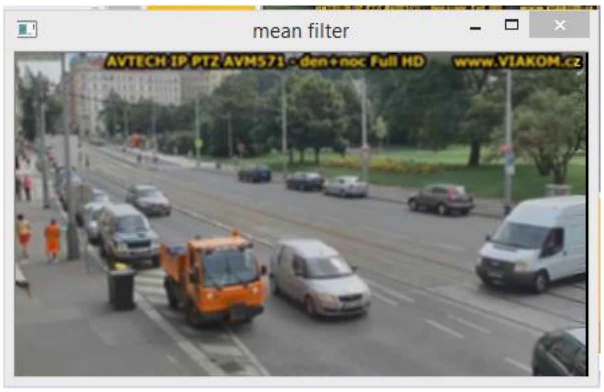
1. Capture a frame from the video.
2. Apply background subtraction using Gaussian mixture model or MOG 2 model on the captured frame. This function is performed on the CPU
3. Spatial filter (Mean Filter) the frame to calculate the pixel intensity.
4. The original image frame is sent to motion detection vector where the above mentioned algorithm is performed. This function is performed on the GPU

Result:

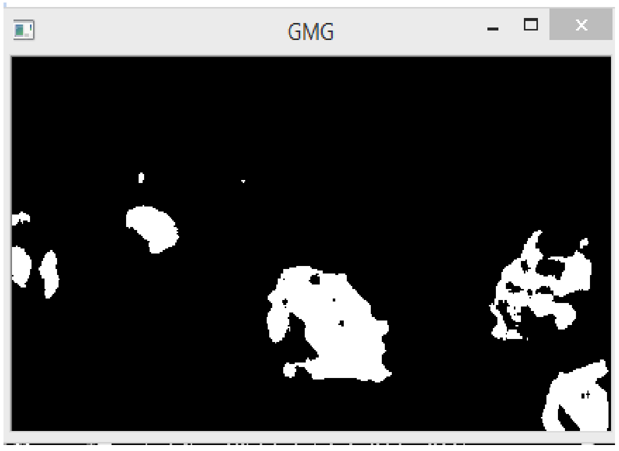
Original Frame:



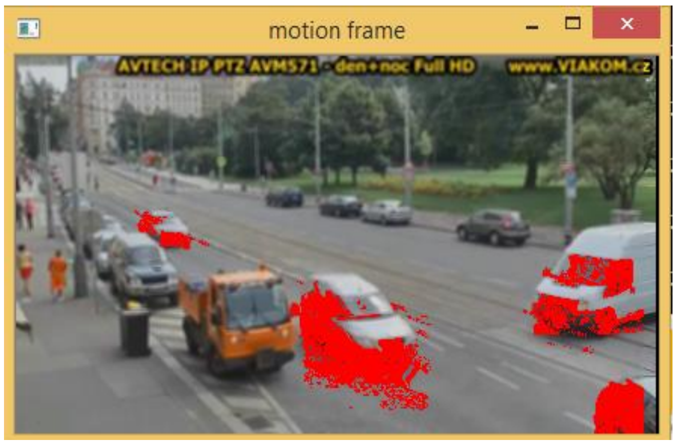
Mean filter:



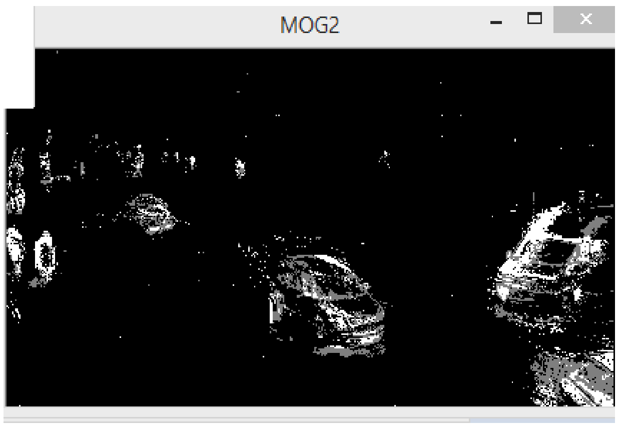
Background Subtraction using GMG:



Motion Detection Frame:



Background Subtraction using MOG 2:



Conclusion And Future work:

In this paper shows the implementation of the parallel video processing for surveillance, we have presented several algorithms that can be easily implemented. The algorithms mentioned above are applicable in security and most importantly in security surveillance. We have implemented the important sections, the background subtraction on CPU and the video filtering and the motion detection on GPU for taking benefits of parallel processing in order to increase the performance and reduce the execution time.

We would like to take this project to the next level by implementing this with live video surveillance and even with some kind of alerts to the operators other than the pixel highlighting.

References:

[CLKFDTTEHBW00] Robert T. Collins, Alan J. Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt1 and Lambert Wixson1 “A System for Video Surveillance and Monitoring”2000.

[MMSKMP07] Sanyam Mehta, Arindam Misra, Ayush Singhal, Praveen Kumar, Ankush Mittal, Kannappan Palaniappan “Parallel Implementation of Video Surveillance Algorithms on GPU Architecture using CUDA” 2007.

[DA14] Leonidas Deligiannidis, Hamid R. Arabnia“Parallel Video Processing Techniques for Surveillance Applications”2014.

[GW08] Rafael C. Gonzalez and Richard E. Woods, “Digital Image Processing (3rd Edition)”. Prentice Hall, ISBN 9780131687288, 2008.

[CW13] Chung-han Chen and Fan Wu “An Efficient Acceleration of Digital Forensics Search Using GPGPU”, In Proc. Of The 2013 International Conference on Security and Management (SAM), 2012.

[Shi08] Daniel Shiffman, "Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction", Morgan Kaufmann Series in Computer Graphics, ISBN 9780123736024, 2008

	OCTOBER				NOVEMBE	
	Week 1	Week 2	Week 3	Week 4	Week 1	Week 2
Initial Project workshop	🟩					
Data Collection		🟩				
Project Plan			🟩			
Research Reference				🟩		
Initial Project Documentation					🟨	
Initial Project Presentation 1					🟨	
Develop Action Plan						🟨
Initialize code						
Meeting and briefing						
Final Report						
Final Presentation						