Speeding Up Camera Sabotage Detection on CUDA

Püren GÜLER, Deniz EMEKŞİZ, Alptekin TEMİZEL
Graduate School of Informatics
Middle East Technical University, Ankara, Turkey
e170986@metu.edu.tr, e171000@metu.edu.tr, atemizel@i..metu.edu.tr

Abstract
Camera Sabotage Detection (CSD) algorithms, namely Camera Moved Detection, Camera Out of Focus Detection and Camera Covered Detection, are used to detect tampering attempts on surveillance cameras in real-time. CSD algorithms are required to be run on a high number of cameras, bringing high computational load to the video analytics systems. Importance of speeding up these algorithms is two-fold:

1. Enabling operation on all cameras and hence reducing security lapses,
2. Lowering valuable computational power to other video analytics such as object tracking and activity analysis.

In this work, the CSD algorithms that were previously developed by our group [1] are accelerated by using parallelization methods in CUDA. While different algorithms have different speed-up rates, the overall system test results show that parallelization in GPU makes the system 18 times faster than its CPU counterpart and up to 400 cameras can be supported in real time on a GTX 470.

General Flow of The Algorithms

1. CURRENT FRAME
2. DIFFERENCE IMAGE
3. BACKGROUND SUBTRACTION
4. DIFFERENCE IMAGE CALCULATION
5. MAGIC NUMBER
6. UPDATE H
7. CALCULATE MP
8. CALCULATE LUT

Background Estimation

Background Estimation is the process of calculating the image that keeps track of changing pixels (MP). Differences in the background image from the current frame are used to calculate background change in the current frame. All processes are implemented in an OpenCL version, each pixel is calculated in 8 parallel threads.

1. Background frame
2. Current frame
3. Difference image
4. Difference of current background
5. Current background frame
6. Moving pixel map
7. Simulation of 256 bin histogram of current image
8. Background estimation

Camera Covered Detection

Camera Covered Detection is the process of determining if the current frame is completely covered by a moving object. The result of the difference step is compared with the histogram of the moving object that was calculated previously. This information is used to determine if the current frame is completely covered by a moving object or not.

1. Current frame
2. Previous frame
3. Difference image
4. Difference of current frame
5. Moving object histogram
6. Moving object map
7. Flag for covered frame

Camera Moved Detection

Camera Moved Detection is the process of determining if the camera has moved between two frames. The difference between the two frames is compared with a threshold value. If the difference between the two frames is above the threshold value, then the camera is considered to have moved.

1. Previous frame
2. Current frame
3. Difference image
4. High frequency content
5. Template matching
6. High frequency content of template
7. Comparison

Camera Out of Focus Detection

Camera Out of Focus Detection is the process of determining if the camera is out of focus. The difference between the two frames is compared with a threshold value. If the difference between the two frames is above the threshold value, then the camera is considered to be out of focus.

1. Previous frame
2. Current frame
3. Difference image
4. High frequency content
5. Template matching
6. High frequency content of template
7. Comparison

Results of Used Algorithms

Experiments are performed on a PC having Intel Core i7 CPU and 3.5 GB usable RAM. The GPU algorithms are tested with Quadro 2000 and NVIDIA GTX 470.

Camera Covered Detection

1. GPU
2. CPU
3. Quadro 2000
4. GTX 470

Camera Moved Detection

1. GPU
2. CPU
3. Quadro 2000
4. GTX 470

Camera Out of Focus Detection

1. GPU
2. CPU
3. Quadro 2000
4. GTX 470

References


High Frequency Calculation

High frequency is the frequency of high values. A modified version of the High Frequency calculation is used in our algorithm. It is ensured that movement inside the kernel is calculated accordingly. The CUDA algorithm is developed so that the logical moving block is always the same.

High Frequency Calculation

1. High frequency of binned histogram
2. High frequency of binned histogram
3. High frequency of binned histogram
4. High frequency of binned histogram
5. High frequency of binned histogram
6. High frequency of binned histogram
7. High frequency of binned histogram
8. High frequency of binned histogram
9. High frequency of binned histogram
10. High frequency of binned histogram

Results of the Overall System

1. Speedup
2. Number of cameras supported
3. Frame size
4. Frame size
5. Frame size

Speedup

1. GPU
2. CPU
3. Quadro 2000
4. GTX 470

Number of cameras supported

1. Quadro 2000
2. GPU
3. GTX 470
4. CPU

Frame size

1. 160x120
2. 320x240
3. 640x480
4. 160x120
5. 320x240
6. 640x480
7. 160x120
8. 320x240
9. 640x480