IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 6, June 2018

GPU Based Unattended Object Detection in Video

Snehal M. Wagh[1], Prof. Dipti Pawar[2]

Dept. of Computer Eng., Sinhgad College of Engg., Pune, India[1,2]

Abstract: This paper presents GPU based CUDA framework that is efficiently and accurately detect unattended object from surveillance video. The system focuses on the problem of finding the unattended object in public places such as shopping mall, airport, railway station etc. In a recent year, GPU has attracted the attention of many application developers as powerful massively parallel system. CUDA as a general purpose parallel computing architectures that makes GPU is an appealing choice to solve many complex computational problems in a more effective way. The processing of surveillance video is computationally intensive. This paper describes parallel implementation of video object detection algorithms like Gaussian Mixture Model(GMM) for background modeling, morphological operation for post processing. SVM classifier is used for unattended object detection. Experimental evaluation shows that parallel GPU implementation achieves significant speedup for GMM and Morphological operations when compared to sequential implementation running on Intel Processor.

Keywords: GPU, CUDA, Parallel computing, unattended object.

I. INTRODUCTION

In the recent year, the real time information is very important in the video surveillance area such as in military operation, reconnaissance path planning and motion detection. Now people suffer the unpredictable threads, such as terrorism especially explosive and chemicals attack when unattended objects are left repeatedly in public areas such as airports, train and subway station, bus terminal etc. GPU is a parallel computational device which reduces the time for operation performance. The programmable GPU is an exceedingly parallel, multi-thread, many core coprocessors specific for serious profoundly parallel calculation. New models and parallelization techniques are being produced because of the expanded openness of multi-thread, multi-core processors alongside the broadly useful designs handling units (GPUs). The current improvements in the GPU engineering have given a successful device to deal with the workload. Previously, several methods have been found describing on abandoned object detection and their application to public safety and security problem. Progressively moving objects discovery on recordings utilizing GPU in [1]. Many methodologies for foundation subtraction were proposed [2]. Such techniques contrast for the most part in the sort of foundation display and in the methodology used to refresh the model. Among them, a mixture of Gaussian distributions has been used to modeling the pixel intensities. In [3] the author proposed a basic foundation subtraction strategy in light of logarithmic intensities of pixels. They correct to have comes about that are better than customary contrast algorithms and which make the issue of limit determination less basic. A prediction-based online method for modeling moving scenes is proposed in [4]. The approach seems to work well, although it needs a supervised training procedure for the background modeling [5], and requires hundreds of images in absence of moving objects.

Adaptive Kernel density estimation is utilized as a part of [3] for a movement based background subtraction algorithm, the identification of moving objects to deal with complex background [6]; however the computational cost is moderately high. All algorithms proposed in the current past to manage the issue of unattended object detection in video, GMM is relatively best techniques for background subtraction [7] and continuously application, they are giving that much good outcome when contrasted with another strategy. So the off chance that we utilize some element extractions techniques like SVM then we can enhance the productivity of identifying a foreground objects because of their reliance on complex probabilistic arithmetic. The vast majority of the algorithms have failed to perform tastefully continuously situations. In addition, the other difficulty of detecting an abandoned object[8] under occlusion adds to the overall complexity.

In this paper, the parallel implementation of various video surveillance algorithms on GPU architecture is presented. This work is focused on the algorithm like 1.Gaussian Mixture Model for background modeling. 2.Morphological image operations for image noise removal. 3. HOG descriptor is used feature extraction of the object. 4.SVM is used for classification of objects. In each of these algorithms, different memory types and thread configuration provided by the CUDA architecture have been adequately work.

Remaining part of the paper is organized as follows. The proposed methodology is discussed in section II. Result and discussion are described in section III and Section IV explains the conclusion of the work. References are cited at the end.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 6, June 2018

II. PROPOSED METHOD

In this section, we describe a solution to detect unattended objects. Fig. 1 shows overall system flow of the proposed system. In the proposed system the very first step which is preprocessing comprises of a group of some simple image processing jobs that alter the raw input video into a format

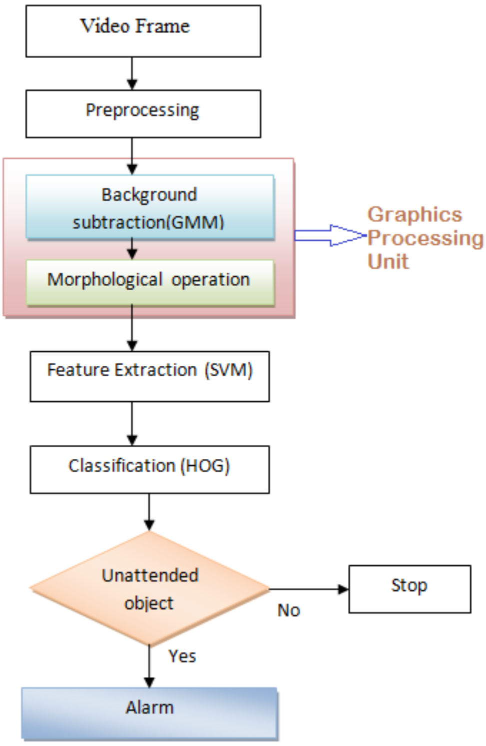

Fig.1 Overall System Flow Diagram of Proposed System

that can be processed by succeeding steps. Background modeling which is the support of the background subtraction algorithm. We utilized pixel level Gaussians mixture background model which has been utilized in a most variety of systems because of its efficiency. In the modeling multi-core distribution of backgrounds and its ability to adapt change which can each be solved independently of the background.

It displays the force of each pixel by a mixture of K Gaussian circulation and consequently turns out to be computationally extremely costly for huge image size and estimation of K. Moreover, there is high level of information parallelism in the algorithm as it includes autonomous operations for each pixel. In this manner, figure escalated trademark and accessible parallelism influences Gaussians Mixture Model (GMM) an appropriate possibility for parallelizing on multi-core processors.

Morphological image operations for image noise removal and in which perform the erosion and dilation operation on foreground image. GPU perform parallel implementation of video object detection algorithms like GMM for background modeling, morphological operation for post processing. SVM is utilized for classification of object and HOG is utilized for feature extraction of unattended objects. Unattended object is found from video surveillance.

This system considers the overall problem of unattended object detection as a collection of smaller problems. Thus it is built using a different subsystem for solving each of these sub-problems. Similar to the formulation stated. It can be divided into five main modules along with two additional modules for pre-processing and for filtering the final results [1]. The brief description of the algorithms that have been implements for each of them follows.

B. Background subtraction modeling:

Background subtraction is a mostly utilized algorithm for detecting moving objects in videos streams from static cameras. It is the general method of motion detection. It is a process that finds the difference of the current image and the background image to detect the motion region, and it is commonly efficient to deliver data included object information[9][10]. It tries to distinguish moving pixels by subtracting the present picture pixel-by-pixel from a reference background picture which is formed by averaging image after some time in an initialization period. The pixels where the variety is beyond a threshold value are categorized as foreground. A not too bad foundation subtraction algorithm must control the moving object that initially submerge out of spotlight and after that progress toward becoming closer view at advanced time.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 6, June 2018

1. Gaussian Mixture Model (GMM):
This method first introduced in [6] and improved significantly in [7]. It assumes that the overall intensity at any pixel at each instant is produced by a combination of background and foreground processes and each such process can be modeled by a single Gaussian probability distribution function. For each pixel in the current frame, the probability of observing the current intensity is given by

$$p(X_t) = \sum_{i=1}^{K} \omega_{i,t}\eta(X_t\, \mu_{i,t}\omega\textstyle\sum_{i,t}) \qquad (1)$$

Here, K is the number of distributions (K=3 here) $\omega_{i.t}$ is the weight associated with the ith distribution at time t while $\mu_{i,t}$ is the mean and $\sum_{i,t}$ is the co-variance matrix of this distribution, η is the exponential Gaussian probability density function given by

$$\eta(X_t\mu_{i,t}\textstyle\sum_{i,t}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\sum_t|^{\frac{1}{2}}} e^{-\frac{1}{2}}(X_t - \mu_t)^T \textstyle\sum_t^{-1}(X_t - \mu_t) \qquad (2)$$

Here, n is the dimensionality of each pixels intensity value (e.g. n=1 for grayscale image and n=3 for RGB image). In order to avoid a costly matrix inversion and decrease computation cost, it is also assumed that the red, green and blue channels in the input images are not only independent but also have the same variance σ_{kt}^2 so that the covariance matrix becomes:

$$\textstyle\sum_{Kt} = \sigma_{kt}^2 I \qquad (3)$$

The K Gaussian distributions are always ordered in the decreasing order of their contribution to the current background model. This contribution is measured by the ratio ω/σ under the assumption that higher is the weight and lower is the variance of a distribution, more is the likelihood that it represents the background process. For each pixel in an incoming frame, its intensity value is compared with the means of the existing distributions the first one and a match is said to be obtained if its Euclidean distance from the mean is less than m standard deviations (m=3 is used here), i.e. it satisfies the following condition:

$$|I_t - \mu_{kt}| \leq m * \sigma_{kt} \qquad (4)$$

Here, it is the pixel intensity while μ_{kt} and σ_{kt} are the mean and standard deviation of the kth distribution at time t. Since the background model is dynamic, it needs to be updated with each frame. While the weights are updated for all distributions, the mean and variance are updated only for the matched distributions. Following are the standard update equations used for this purpose:

$$\mu_t = (1-\rho)\mu_{t-1} + \rho\, X_t \qquad (5)$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \qquad (6)$$

$$\omega_{kt} = (1-\alpha)\omega_{k,t-1} + \omega(M_{kt}) \qquad (7)$$

Here X_t is the pixel intensity while M_{kt} = 1 for matched distributions and 0 for unmatched ones while ρ and σ are learning rates. In the current work ρ and σ are related as:

$$\rho_{kt} = \frac{\alpha}{\omega_{kt}} \qquad (8)$$

Thus, while α is fixed for all distribution (α=0.001 used here), ρ is smaller for higher weighted distributions. If none of the existing distributions match the current intensity, the least probable distribution (i.e. with the smallest value of ω/σ) is replaced by a new distribution with a high initial variance, low prior weight and the new intensity value as its mean.

1. Gaussian Mixture Model Implementation on GPU: GMM offers pixel level information parallelism which can be effortlessly misused on CUDA architecture. Since the GPU comprises of multi-cores which permit autonomous thread scheduling and execution, flawlessly reasonable for free pixel computation. So, an image of size m * n requires m * n threads, implemented using the appropriate size blocks running on multiple cores.

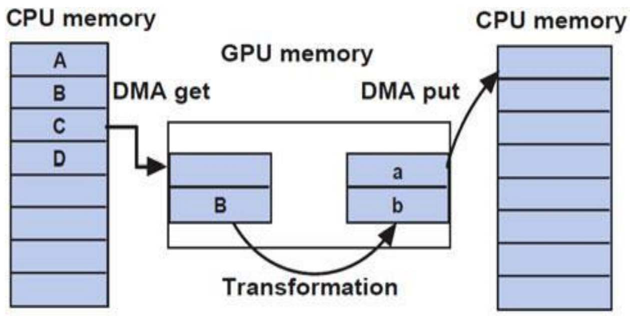

Fig. 2 Streaming (double buffering) mechanism on GPU memory to overlap communication with computation.

Each block has its own shared memory which is accessible (read/write) to all its threads simultaneously, so this most

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 6, June 2018

improves the parallel computation on each thread since memory access time is significantly reduced. In our approach, we have utilized K(number of Gaussians) as 4 which brings about compelling mixing as well as lessens the bank clashes the proficiency of mixing is very noticeable. Our approach for GMM includes Streaming, i.e. we process the info outline utilizing two streams. Thus, the memory duplicates of one stream (a half portion of the picture) cover (in time) with the piece execution of the other stream

By portion execution of a stream [11], we mean to utilized of the GMM approach as examined above to a half portion of the pixels in a casing at any given moment. This is like the prevalent two fold buffering system as appeared in Fig. 2 Streaming gives great outcomes on the grounds that the ideal opportunity for memory duplicates was firmly coordinated to the ideal opportunity for kernel execution.

C. Morphological frame processing:

Both lighting change detection and shadow detection processes are prone to false classifications and often leave behind holes inside valid foreground objects along with some left behind shadow pixels that may be detected as small blobs. These are removed by applying the morphological operations of closing followed by opening. The previous procedure applies expansion took after by erosion by while the last applies erosion took after by dilation. A case of the tidying up impact of morphological preparing is appeared in Fig. 3. As the morphology operation on neighboring pixel utilize information which have spatial area, this advancement lessens get to time extensive without using shared memory. The programmable GPU is a profoundly parallel, multi-thread, many core coprocessors specific for process concentrated exceptionally parallel algorithm. A half twist (16 strings) has a data transmission of 32 bytes/cycle and subsequently 16 strings, each preparing 2 pixels (2 bytes) utilize full transfer speed, while composing back noise free image.

Fig.3 Foreground mask before (left) and after (right) applying morphological processing.

This parts the aggregate number of strings therefore decreasing the execution time altogether. A straightforward convolution was done with one thread running on two neighboring pixels.

D. Feature extraction and classification:

If the detected set of pixels represents unattended object and a human being then objects behavior is tracked for any sort of anomalies. For detecting unattended object being in our system we are utilizing the people detector, which detect unattended objects and moving object in an input image using the Histogram of Oriented Gradient (HOG) features and a trained Support Vector Machine (SVM) classifier. HOG is a mechanism for feature description. The need for a feature descriptor is to generalize a given object in such a way that the object (in this case a unattended object) exhibits a close match to this feature descriptor when observed under various conditions. This makes the classification process simple. In our system we have used Support Vector Machine (SVM), to recognize HOG descriptors of unattended object. It utilizes a sliding detection window which is moved across the image. A HOG descriptor is computed at each position of the detector window. This descriptor is then given to the trained SVM, which classifies it whether it is person or not. Once an unattended object is detected, next task is to track his behavior for abnormalities. It is basically an iterative expectation maximization-clustering algorithm implements within local search regions.

III. RESULT AND DISCUSSION

The parallel implementation of unattended object detection on video is executed on NVIDIA GEFORCE GT 740M GPU on board 2.30 GHz Intel(R) core(TM) i5-4200u 1.60GHz machine with 2 core processor at 4.00 RAM. The GEFORCE GT 740m GPU has a staggering 384 cores at memory data rate 1800MHz. Experimentation is performed on various sizes of images and videos. The system is tested on our own video sequences taken with normal video camera in classroom and railway station. The environment is randomly chosen. No special background condition has been imposed. We have taken five video sequences from different places. It also contains complex scenario with multiple people sitting. People are walking

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 6, June 2018

with variable speed. Some are sitting in very still position. This type of environments is very common in our daily life. Even though most of existing methods so far do not take this realistic situation into account, our proposed method can handle these cases successfully. All videos have instances of various shapes of abandoned object and still people.

We used python version 2.7 for experimentation. In the proposed system, first the video sequences are taken as input. Image processing is done by set of morphological operations. Background subtraction is done by Gaussian Mixture Model (GMM) which is followed by unattended object detection by Support Vector Machine (SVM). Aim of the experiments to evaluate CPU and GPU performance for the proposed System.

A. Unattended Object Detection

Timing Window for CPU and GPU execution for unattended object detection and morphological operation are described in this section. The Fig. 4 shows the result of system execution Time on CPU and GPU, in which we detect the unattended object and shows the morphological operation window with the timing for detection of unattended object from each frame.

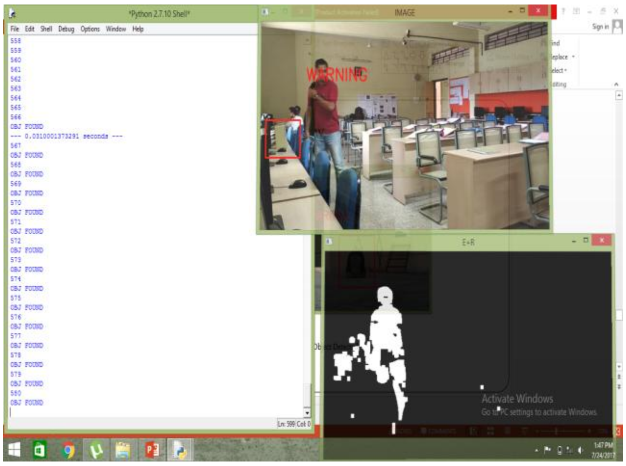

Fig. 4: Output window for unattended object detected

B. Unattended Object Detection from videos

For the experimentation we used 4 videos from different scenario of different sizes. The scenarios are indoor and outdoor and also detect 1-2 object detected. Table 1 shows the comparative timing analysis of CPU Vs GPU. The results clearly indicate that GPU achieves significant speedup compared to CPU.

Table1: Comparison of execution time (in seconds) and frame processing rate of sequential versus parallel GPU implementations system of unattended object detection various videos.

Videos	CPU	GPU	No of Objects detected
Video 1	1800.26	829.26	1
Video 2	3317.41	2135.67	2
Video 3	5144.99	3811.05	1
Video 4	7489.18	6405.12	1

C. Unattended Object Detection for Different Size of Images

The experimentation is also carried on different images of sizes 320 * 240, 720*480, 1024*768, 360*738. It is observed from Table 2 that GPU requires less time for unattended object detection compare to CPU. As the image size increases time required to detect unattended object also increases.

Table 2: Unattended object detection for various frames size

Image size	320*240	720*480	1024*768	1360*738
CPU	2.854	4.361	5.668	7.217
GPU	1.303	2.989	4.132	5.676

D. Unattended Object Detection for Various Algorithms:
We also compare the timing analysis for the implementation of Gaussian Mixture Model and Morphology.

Table 3: Time taken (in seconds) by various algorithms while running for Unattended object detection.

Implementation	GMM	Morphology	Total
CPU	0.016	1.3519	1.3679
GPU	0.014	1.3269	1.3409

1. Parallel GMM Performance

GMM implementation on CPU and GPU is compared in Table 3. Results show significant speedup for parallel GPU implementation compared to sequential execution on Intel core processor. Aside from the picture information, GMM stores the foundation display data of mean, sigma and weight esteems in the memory which should be exchanged forward and backward host to gadget memory for progressive edge calculation. This makes part of postponement due memory duplicate (cuda-Memcpy) operations which backs off the handling and consequently speedup. In any case, by applying the idea of twofold buffering utilizing offbeat moves of information in two streams, we endeavor to cover the correspondence with algorithm time consequently, could get huge speedup esteems. It can be watched that the speedup increments with picture sizes on account of increment in the aggregate number of CUDA strings (every pixel is worked upon by one CUDA string) which keeps the centers occupied.

2. Parallel Morphological Operation

Morphology operations implementation on CPU and GPU is compared in Table 3. We are able to get enormous speedup for parallel GPU implementation compared to sequential execution on Intel core processor. First of all, the input data to this algorithm is binary values for each pixel which needs very less memory for storing the entire frame and thus the time for memory copy operation is reduced to minimal. Moreover, we could upgrade memory get to time amid calculation by putting away the twofold picture in surface memory which is a perused just memory and furthermore stored.
.

IV. CONCLUSION

Through this paper we describe the implementation of unattended object detection from surveillance video on the parallel architecture of NVIDIA GPU. The various algorithms described in the paper are GMM, morphological operation. SVM is utilized to classify unattended object. In experimental results, we compared the performance of sequential and parallel GPU implementation of GMM and morphology operation. The sequential execution was done on GMM and Morphology operation. Parallel GPU implementation was done on GMM and Morphology operation. Experimental evaluations presents that the parallel GPU implementation achieves significant speedup for GMM and for Morphology operation. Our future work will include testing the current implementation on very high resolution video to measure us Scalability and efficiency. We will also focus on to detect the stolen abject and removed object from complex environment.

ACKNOWLEDGMENT

I humbly thank to **Prof. D. R. Pawar** (Professor of Computer Engineering Department, Sinhgad College of Engineering Pune, Maharashtra, India) for lending her valuable expertise.

REFERENCES

[1]Praveen Kumar, Ayush Singhal Sanyam Mehta, Ankush Mittal, “Real-time moving object detection algorithm on high-resolution videos using GPUs”, Minneapolis, USA, Graphics Era University, Dehradun, India, Springer-Verlag Berlin Heidelberg, 2013.
[2]Miss Helly, M Desai, Mr. Vaibhav Gandhi “A Survey: Background Subtraction Techniques”, International Journal of Scientific & Engineering Research, Volume 5, Issue 12, 2014.
[3] Dipti Pawar, “GPU based Background Subtraction using CUDA : State of the Art”, IEEE International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2017.
[4] Dipti Pawar, "Fuzzy Min-Max Neural Network with Compensatory Neuron Architecture for Invariant Object Recognition", IEEE International Conference on Computer, Communication and Control, 2015.
[5] Bane, S., Pawar, D.R.: Survey on Feature Extraction methods in Object Recognition. In: International Journal of Computer Science and Information Technologies, vol, 5, pp. 3224–3226 (2014).
[6]Takashi Toriu, “Unattended Object Intelligent Analyzer for Consumer Video Surveillance”, IEEE 2011.
[7]Medha Bhargava, Chia-Chih Chen, M. S. Ryoo, J. K. Aggarwal, “Detection of object abandoned using temporal logic”, 2009.

[8]G. Mallikarjun Rao, Dr. Ch. Satyanarayana, “Object Tracking System Using Approximate Median Filter”, Kalman Filter and Dynamic Template Matching, I.J. Intelligent Systems and Applications, Vol 05, 83-89, 2014.

[9]C. Lakshmi Devasena, R. Revath, M. Hemalatha, “Video Surveillance Systems-A Survey”, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, 2011.

[10] M. Elhamod and M. D. Levine, “A real time semantics-based detection of suspicious activities in public scenes”, in Proc. 9th Conf. CRV, Toronto, Canada, pp. 268-275, 2012.

[11] N. Bird, S. Atev, N. Caramelli, R. Martin, O. Masoud and N. Papanikolopoulos, “Real Time, Online Detection of Abandoned Objects in Public Areas”, IEEE International Conference on Robotics and Automation, pp. 3775-3780, 2006.