Framework for Multi-View Video Summarization on Many core GPU

Pandurang Matkar
Department of Computer Engineering
Pune Vidhyarthi Griha's College of Engg. & Tech.
Pune, India

Aditya Tajne
Department of Computer Engineering
Pune Vidhyarthi Griha's College of Engg. & Tech.
Pune, India

Sushil Bomane
Department of Computer Engineering
Pune Vidhyarthi Griha's College of Engg. & Tech.
Pune, India

Piyush Bansal
Department of Computer Engineering
Pune Vidhyarthi Griha's College of Engg. & Tech.
Pune, India

Prof. S. A. Saoji
Department of Computer Engineering
Pune Vidhyarthi Griha's College of Engg. & Tech.
Pune, India

***Abstract*— The recent progress of digital media has stimulated the creation, storage and distribution of data, such as digital videos, generating a large volume of data and requiring efficient technologies to increase the usability of these data. Video summarization methods generate concise summaries of video contents and enable faster browsing, indexing and accessing of large video collections, however, these methods often perform slow with large duration and high quality video data. Video summarization using key frames can facilitate to speed up multi-view video processing. We then propose a key frames extraction using wavelet statistics is discussed to use in video summarization with multi-view technique. In extracting key frames, two consecutive frames are firstly DWT transformed and then the differences of the detail components of them are estimated. If different value of a consecutive pair is greater than threshold, the last frame of the pair is considered as a key frame. After extracting key frames we will synthesizes extracted frame to build a summarized video.**

Keywords*— Video summarization (time, video format), NVidia GPU, video & image processing, Key frames, multi-view, video decoding.

I. Introduction

With the advent of digital multimedia, a lot of digital content such as movies, news, television shows and sports is widely available. Also, due to the advances in digital content distribution (direct-to-home satellite reception) and digital video recorders, this digital content can be easily recorded. However, the user may NOT have sufficient time to watch the entire video (Ex. User may want to watch just the highlights of a game) or the whole of video content may not be of interest to the user(Ex. Golf game video).

In such cases, the user may just want to view the summary of the video instead of watching the whole video. Thus, the summary should be such that it should convey as much information about the occurrence of various incidents in the video. Also, the method should be very general so that it can work with the videos of a variety of genre. Our proposed method is key frame extraction from different scenes of a video clip. Each key frame can represent each related scene and also entirely contains all important information of the scene. After the key frame extraction, the key frames are intended to use in video summarization, feature extraction and other processing so key frame extraction algorithm should not be very complex and time consuming. Our objectives for the proposed algorithm are specially focused on speed and precise. In our proposed method, for the detection of key frame from a scene, we have to find the obvious difference value between two successive frames. In a video stream, each video frame is a slightly variation with previous one. However, whenever scenes are changed, visual contents and objects are obviously different between current frame and next one. Hence, we use the difference as a key for the key frame detection.

First, the input video is split into adjacent datacubes. Then DWT is applied to each data cube and statistical features are extracted. This result is used to select pixels of interest in each frame in the data cube. Key frames are identified by Local Maxima and Local Minima. The proposed work outperforms the existing DWT method in terms of identifying all events of interest in the input videos.

Then we apply a multi-view summarized technique in which we take two or more videos and on each video we are going to apply key frame extraction algorithm using DWT method to extract video shots.

II. Related Work

An improved key frame extraction method which measures the similarity of two adjacent frames contents in terms of the information of frame difference and extracts key frames after optimizing the frame difference. The logic behind this method is to begin from the video data; the similarity of inter-frame is computed by the difference of two adjacent frames. Then optimize and select the specific different frame adaptively, export the eligible key frames finally. The key frame extraction algorithm proposed is mainly used to describe the contents of the entire process of a shot. However, the key frames of a particular event are often taken into account. Such as football games, people may pay more attention to the shot content, so frames about the goal should be extracted more under this circumstances. This goal requires a combination of object identification technologies, such as ball and goal identification, locating the position of the shot event to extract key frame. In addition, the key frame extraction evaluation criteria are not perfect, the future work can also be committed to this in order to establish or determine the uniform evaluation criteria.

A method for static video summarization that can produce meaningful and informative video summaries performing an evaluation using over 100 videos in order to achieve a stronger position about the performance of local descriptors in semantic video summarization. Experimental results show, with a confidence level of 99% that of the proposed method using local descriptors and temporal video segmentation produces better summaries than state of the art methods. They also demonstrated the importance of a more elaborate method for temporal video segmentation, improving the generation of summaries, achieving 10% improvement in accuracy. They also acknowledged a marginal importance of color information when using local descriptors to produce video summaries. The logic behind their method, they approached the task of video summarization by considering the semantic information expressed by the video's visual entities. The proposed method elaborates static video summaries and core approach is to use temporal video segmentation and visual words obtained by local descriptors. The proposed method has taken advantage of previous techniques in video summarization and segmentation exploring some of the advantages of GPU like general programming environment, its computing capability, higher bandwidth and instruction operation. Further the authors explained the CUDA hardware architecture model, its compiler procedure and general programming mode of CUDA. They have explored about image processing applications using CUDA like histogram equalization, removing clouds, edge detection and DCT encode and decode. In histogram equalization, original and new histogram distributions are computed where hierarchical structure of threads, block and grid and shared memory concept is used. For removing cloud algorithm, numbers of threads are created so that each thread processes one pixel data. For Fourier Transform, a library of CUFFT is used. For DCT encode and decode, the image is divided into NxN blocks and DCT transform is performed for each block. This paper basically explores how GPU can be used for processing image in parallel nature.

III. PROPOSED WORK

Figure 1 depicts the overall proposed system architecture based on video summarization, framework to be implemented by using CUDA C and GPU.

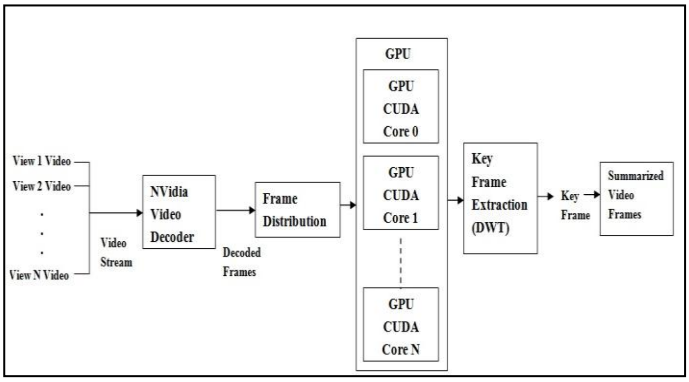

Fig. 1. Overall System Architecture

The overall system design consists of the following modules:

- NVidia Video Decoder
- Frame Distribution on GPU
- Key Frame Extraction
- Summarized Key Frames

The input to the system is multi-view videos whose images will be extracted. The first module will transform video to frames. The second module performs frame distribution on many cores GPU. The third module will generate unrepeated key frames. The forth module will collect all resulted key frames. The output of the system will be summarized video frames stored in the form of binary files.

Proposed implementation will consist following components:

- The video summary must contain high priority entities and events from the video.
- The summary itself should exhibit reasonable degrees of continuity.
- The summary should be free of repetition.
- To create multi view summarization.

A. Multiview Video Summarization

The task of Multi-View Video Summarization is to efficiently represent the most significant information from a set of videos captured for a certain period of time by multiple cameras. The problem is difficult to solve because of the huge size of the data, presence of many unimportant frames with low activity, inter-view dependencies and significant variations in illumination. In this paper, we present a method for summarizing multi-view videos. We propose a novel solution to the above problem within a framework. Multiview video summarization at multiple places or environments like Offices, Campus, Office Lobby, Road and Badminton, capturing both indoor and outdoor environments

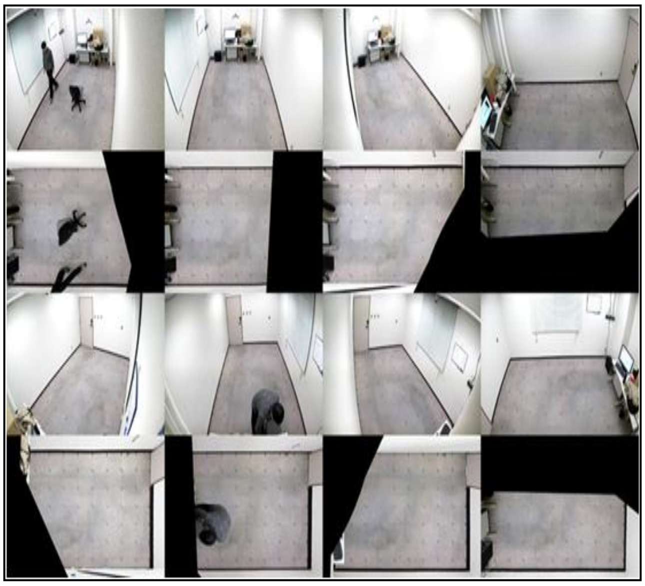

Fig 2. Multiview Video Frames

B. GPU

A Graphics Processing Unit (GPU) is a single-chip processor primarily used to manage and boost the performance of video and graphics.GPU can be used in many applications like computer gaming and graphics, image processing, designing automobiles, Graphical user interface, Construction, movie animation, medical field.

In this framework, GPU used instead of CPU (central processing unit). CPU is not capable to handle large data sets and it requires lots of time to perform large operations. There are lots of benefits of GPU like Faster Processing, Lower capital cost, reduced power consumption; The GPUs extended the performance beyond what is possible with CPUs alone at any cost.

For better performance cuda language will be used for GPU programming. CUDA is well suited for large datasets and highly parallel algorithms. CUDA is the name of NVIDIA's parallel computing architecture in our GPUs. NVIDIA provides a complete toolkit for programming the CUDA architecture that includes the compiler, debugger, profiler, libraries and other information developers need to deliver production quality products that use the CUDA architecture. The CUDA architecture also supports standard languages such as C and Fortran, and APIs for GPU Computing, such as OpenCL and DirectCompute. The CUDA platform is accessible through CUDA-accelerated libraries, compiler directives, application programming interfaces, and extensions to industry-standard programming languages, including C, C++, Fortran, and Python. This project focuses on CUDA C programming. CUDA is also a scalable programming model that enables programs to transparently scale their parallelism to GPUs with varying numbers of cores, while maintaining a shallow learning curve for programmers familiar with the C programming language.

A CUDA program consists of a mixture of the following two parts:

- The host codes runs on CPU.
- The device code runs on GPU.

IV. PROPOSED ALGORITHM

Algorithm: Multi-View Video Summarization using many core GPU.

```
Input: Set of multi-view video frames
 begin
     foreach frame  do
         Apply DWT using multiple parallel threads
         Find difference between current and next frame
         Store the calculated difference in difference vector
         using multiple parallel threads
     end foreach
     foreach frame do
         Calculate mean and S.D. from the difference vector
         on multiple threads
         Update threshold by adding mean and S.D.
          if threshold is less than difference vector then
               second frame is key frame
          end if
     end foreach
     foreach key frame do
         Combine all key frames and store in a video file
         using multiple parallel threads
     end foreach
end
Output: Summarized Video
```

V. MATHEMATICAL MODEL

The overall system architecture can be represented as set of four modular components as below:

SYSTEM={INPUT, FUNCTIONS, OUTPUT, SUCCESS CONDITIONS, FAILURE CASES}

INPUT = {Set of multi-view video frames}

Mathematically, an input video can be represented as a set of frames as follows:

$Invideo = \{view1frame_1 , viewframe_2 , viewframe_3 , \ldots viewframe_n \}$

$inframe_i \subset invideo$ *is a subset of RGB frame*

where invideo is an input video

view1frame$_i$ to viewnframe$_n$ are multiple view video frames

FUNCTIONS = {RGBTO YUV (), MotionEstimation (), Summary (), Difference (), GetKeyFrame ()}

OUTPUT = Summarized meaningful AVI format video

SUCCESS CONDITIONS = { avi video formats }

FAILURE_CASES = {3gp, mp4, mkv, etc video formats}

Functions:

1. RGBToYUV ():

The first step I this process of encoding is transformation of video image of RGB (Red, Green, Blue) format to YUV (Y-Luminance, U&V-Chrominance) format. Each pixel has the three color component R, G and B

2. MotionEstimation ():

Motion estimation is the process in which dynamically motion is estimated in the video with the help of peak signal to noise ratio (PSNR).

3. Summary ():

Combine all results to create video summary.

4. Difference ():

The process of calculating difference is accomplished by following mathematical formulas:

$$dfsfky = pixel_{sy}-pixel_{ky}$$

$$dfsfku = pixel_{su}- pixel_{ku}$$

$$dfsfkv = pixel_{sv} - pixel_{kv}$$

Here, dfsfky, dfsfku, dfsfkv is the difference between sub frame and key frame.

5. GetKeyFrame (): Using this function we extract key frames from the video.

VI. Conclusion

In this project, we propose, to the best of our knowledge, the first attempt at multi-view video summarization on many cores GPU. Video Summarization using proposed method described in this project will definitely reduce time and required less storage space and Parallel implementation of the methods by using CUDA and GPU will help to process lacks of pixels in parallel in less time as compared to time required by sequential implementation. We propose to use multi-view video structure.

Acknowledgment

We express our sincere thanks to Prof. S.A.Saoji, PVG's COET, Pune for their valuable guidance in the overall execution of this project work. We would also like to thanks Prof. Dr. G. V. Garje and Prof. A. M. Bhadgale.

References

[1] Chetan Sharma, "Key Frame Extraction using wavelet transforms – A Video Summarization Technique", IJARCSMS Volume 2, Issue 8, August 2014.

[2] Khin Thandar Tint, Dr. Kyi Soe, "Key Frame Extraction for Video Summarization Using DWT Wavelet Statistics," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) vol. 2, no. 5, May 2013.

[3] Azra Nasreen, Dr. Shobha G "Key Frame Extraction from Videos - A Survey," International Journal of Computer Science & Communication Networks, Vol 3(3), 194-198 ISSN: 2249-5789.

[4] R.H. Evangelio, T. Senst, I. Keller, and T. Sikora, "Video indexing and summarization as a tool for privacy protection", In International Conference on Digital Signal Processing (DSP), pages 1–6, 2013.

[5] Huayong Liu, Lingyun Pan, Wenting Meng, "Key frame extraction from online video based on improved frame difference optimization", IEEE 14th International Conference on Communication Technology (ICCT), vol. 1, no. 1, 9-11 Nov. 2012, pp.940-944.

[6] Geetha, P., and Vasumathi Narayanan. "A survey of content-based video retrieval", 3rd International Conference of Trendz in Information Sciences and Computing (TISC),Chennai, 8-9 Dec 2011, pp. 55-60.

[7] Jason Sanders and Edward Kandrot. "CUDA by Example: An Introduction to General-Purpose GPU Programming", Addison-Wesley Professional, 1st edition, 2010.

[8] Ngai-Man Cheung, Xiaopeng Fan, O.C. Au, and Man-Cheung Kung., "Video coding on multicore graphics processors", IEEE Signal Processing Magazine, 27(2):79–89, 2010.