Proceedings of the International Conference , "Computational Systems for Health & Sustainability"
17-18, April, 2015 - by R.V.College of Engineering,
Bangalore,Karnataka,PIN-560059,INDIA

Implementation of Content Based Video Retrieval in Multimedia Applications Using Histogram Difference

Dr. Nagaraja G. S
Professor
RVCE, BANGALORE

Ramesh. B
Ass. Professor
CEC, BANGALORE

Divya K.S
Research Associate
RVCE,BANGALORE

Abstract

The increasing use of multimedia applications nowadays necessitates the development of effective methodologies for manipulating databases storing video, audio information. Moreover, in its beginning stage, content-based access to video scenes requires parsing of each video scenes into its building blocks. The video stream is constitutes number of shots, each shot is a sequence of frames pictured using a single camera. Switching from one camera to another indicates the transition from a shot to the next one. Therefore, the detection of these changes is known as scene transition or shot transition, is the initial step in any video-analysis system. A number of proposed techniques for solving the problem of shot boundary detection exist, but the major criticisms to them are their inefficiency and lack of reliability. The reliability of the scene change detection stage is a very significant requirement because it is the first stage in any video retrieval system; thus, its performance has a direct impact on the performance of all other stages. On the other hand, efficiency is also crucial due to the voluminous amounts of information found in video streams. This paper proposes a new robust and efficient paradigm capable of detecting scene changes on compressed .AVI video data. A single core processor takes a lot of time to execute single thread execution of algorithms in a very big video database to overcome this drawback a High performance computing is required (HPC) to make use of GPU for multithreaded execution for video processing algorithms.

Keywords: Shot Detection, Histogram, GPU, CUDA.

I. INTRODUCTION

With the growth of internet and development of digital content, content-based video retrieval (CBVR) has become an integral part of information retrieval technology. CBVR technique is searching videos in database similar to the query video, according to the video features related to content. This technique is based on automatic shot detection and shot extraction with features in it, retrieves by automatically comparing the features of query video (such as color, shape, texture, etc.). With the similar features of video feature library, and finally outputs the best matching videos and its corresponding information.

Shot Detection

A shot is an unbroken sequence of frames from one camera. Thus, a movie sequence that alternated between views of two people would consist of multiple shots. A scene is defined as a collection of one or more adjoining shots that focus on an object or objects of interest. For example, a shot consist of a rock and water side by in the next shot bird has standing on it while drinking water. There are a number of different types of transitions or boundaries between shots.

Histograms

In our implementations of shot boundary detection we focused on a histogram-based approach, significant due to its performance and accuracy. Our procedure was on hard cut detection rather than gradual shot transitions such as fades or dissolves. Our technique firstly computes a color histogram for each block of a frame of video, secondly calculates the difference between adjacent frames depending on vector distance and lastly identifies candidate shot transitions by comparing adjacent frame similarities with a threshold. This straightforward technique has been evaluated by us.

GPU Video Processing

Real time video processing with Graphical Processing Unit is parallel programming framework with Combined Unified Device architecture which increases the speedy computations in the video processing.

II. RELATED WORK

The tested methods to a few number of short video sequences and sometimes tuned the methods to work well on the sequences. This total is matched against a second threshold to determine if a shot boundary has been found. Hang, Kankanhalli, and Smoliar1 implemented this method with the additional step of using 333 averaging filter before the comparison to avoid camera motion and noise effects. They found that by selecting a threshold tailored to the input sequence better results were obtained, although the method was slow. We note that manually adjusting the threshold is no practical. St hahraray4 divided the images into 12 regions and found the best match for each region in a neighborhood around the region in the other image. This matching process duplicates the process used to extract motion vectors from an image pair. The pixel differences for the region were sorted, and the weighted sum of the sorted region differences provided the image difference measure. Hampapur, Jain, and Wey mouth computed what the call chromatic images by dividing the change in gray level of each pixel between two images by the gray level of that pixel in the second image. During dissolves and fades, this chromatic image assumes a reasonably constant value. They also computed a similar image that detects wipes. But, this technique is very sensitive to camera and object motion.

Statistical methods expand on the idea of pixel differences by breaking the images into regions and comparing statistical measures of the pixels in those regions. For example, Kasturi and Jain compute a measure based on the mean and standard deviation of the gray levels in of the images, but is slow due the complexity of the statistical formulas. It also generates many false positives ~i.e., changes not caused by a shot boundary.
Histograms are the most common method used to detect shot boundaries. The simplest histogram method computes histograms of the frames from videos. If the bin-wise difference between the histograms is above threshold, a shot boundary is assumed. Ueda, Miyatake, and Yoshizawa7 used the color histogram change rate to find shot boundaries. Nagasaka and Tanaka studied several simple statistics based on gray level and color histograms. Better results are made by breaking the images into 16 regions, using a x2 test on color histograms of those regions, and neglecting the eight largest differences to reduce the effects of object motion and noise. Swanberg, Shu, and Jain8 used gray level histogram differences in regions, weighted by how likely the region was to change in the video sequence. This worked out because their news videos had a very regular spatial structure. They did some simple shot categorization by comparing shots with the known types ~e.g., anchor person shot! from the database. They were also able to group shots into higher level objects such as scenes and segments by matching the shot types with the known temporal structure. Hang, Kankanhalli, and Smoliar1 compared pixel differences, statistical differences, and several loading QoS adjusting to various types of terminals.

Our shot boundary detection and key frame selection procedure work with blocks within video frames. Thus the process of taking a video file and dividing into frames and then blocks.It is a first step in this process which takes a significant proportion of the overall time required for the entire process, and represents a good candidate for acceleration on the GPU. This has already been done by a number of graphics processor vendors, such as in NVidia's Pure Video technology, and ATi's Avivo. Both companies' technologies offload a number of the most computationally intensive aspects of . avi processing on the GPU, in order to speed up the process over the CPU alone so we will concentrate our work on the feature extraction using GPU and CUDA C.

III.PROPOSED METHOD

In this method of processing video clips/data and gives it to the media descriptors the media descriptor performs the feature extraction of that video and then the key frame is chosen from the available frame and indexing is done on that key frame and indexes are stored in database along with indexes and various other features also gets stored on the database.

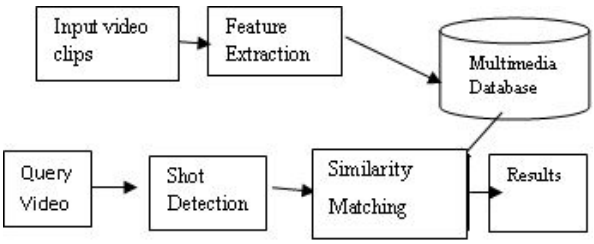

Fig-1

GPU Architecture

GPU is Graphics Processing Unit used for high performance parallel computing. It is a processor which has evolved over few years, from a fixed-function special-purpose processor into a full-fledged parallel programmable processor with few extra fixed-function special-purpose functionality. GPU computing is the use of GPU together with CPU (Central Processing Unit) to accelerate general-purpose scientific and engineering applications.

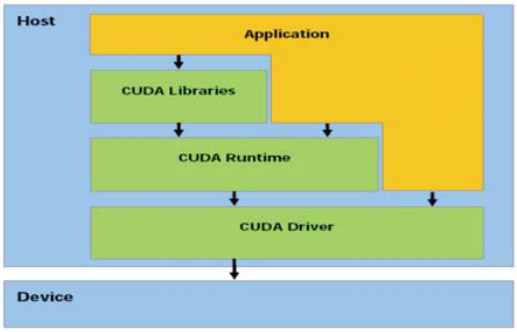

Fig-2

CUDA Architecture

CUDA is Compute Unified Design Architecture. It was introduced by NVIDIA in November 2006. It allows a soft-ware developers to use GPUs through a programming language, 'C. For CUDA .A compiled CUDA program can execute on any number of 366 processor cores. CUDA C is the language mainly designed to provide general purpose computing on GPU.CUDA C adds the global qualifier to standard C.

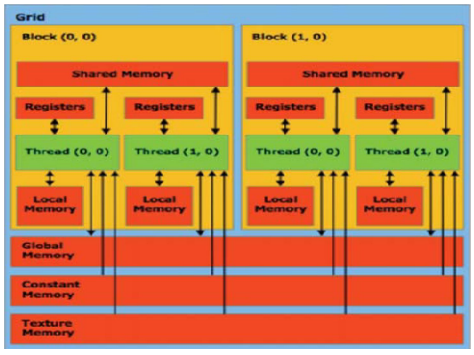

Fig-3

Shot Boundary Detection with GPU

The implementation of a CPU method of shot boundary detection is simple and straightforward. We use a histogram class with functions for generating a histogram based on a provided array of frame data, and for calculating the distance between the frame and a second frame passed to it. After a number of different approaches, we achieved an efficient GPU implementation using an approach which leverages the capability to query the GPU based on a processor that is executing. This querying is exposed by the API, and can be used in rendering graphics to determine, for example, if an object is occluded or not (the application would execute a processor to draw a proxy object in place of the actual more complex version, and use a query to determine if the object was drawn or discarded). This capability can be applied to histogram computation by addressing each bin of the histogram in turn. For each bin, the processor takes the frame as input, and draws it unchanged to another buffer, but first it checks if the pixel to be drawn is within the range of the current bin, whose minimum and maximum values are passed as parameters. If the pixel is within the given range it is drawn, if not it is discarded. The query over this processor simply counts the number of pixels drawn, effectively computing the value for the current bin. Note that in this approach we are still passing over every frame n times for n bins. However we can pass the bin's minimum and maximum values to the processor directly as parameters with each pass, obviating the need for computation of the minimum and maximum values within the processor, without sacrificing parallel speedups.

We calculate the difference between adjacent frames histograms in one processor pass by packing all the histograms into two textures, one containing all textures from 0 to m− 1, where m is the number of histograms, and the other contains all textures from 1 to m. So in short, the second texture is the same as the first, except shifted one histogram to the left. This allows the shader to access a frames 1The word shader has a dual meaning here, referring to processors in the GPU hardware itself, and to software programs that run on them.

Feature extraction with GPU

In content-based video retrieval, feature vector of video frames are automatically stored at index in feature database which describe the content of the video. Features are extracted and stored as feature vectors. The feature vectors of both the query video clip and the video clips from database are compared and thus the required video clips is retrieved. Features used here is color be the query video D, T be the database and t be the threshold distance. So, the distance between both the feature vectors is given as, D (Feature (Q), Feature (T)) ≤ t.

Similarity matching with GPU

In this step the feature vectors of the query video as well as that of the video database is compared for their similarities and thus the suitable match is searched for. Graphical Processors Units (GPU) play important role to speedup processing of database images matching algorithms because it has more inbuilt execution cores. One of the techniques is where the feature space as well as similarity on this space is defined. The similarity measure is done by k -th nearest neighbor search. To speed up the computation 369 time, parallel implementation of the K-Means search on a Graphic Processing Unit (GPU) using CUDA is developed and it is observed that the computation time for one similarity measure between two videos required 0.2s on average .The online image matching can be enhanced by using GPUs.

K-means clustering

Clustering algorithm has been widely used in computer vision such as image segmentation and database organization. The purpose of clustering is to group images whose feature vectors are similar by similarity judgment standard; meanwhile to separate the dissimilar images. Clustering algorithms can be broadly divided into two groups: hierarchical and partitioned. Hierarchical clustering algorithms recursively find nested clusters either in agglomerative mode (starting with each data point in its own cluster and merging the most similar pair of clusters successively to form a cluster hierarchy) or in divisive (top-down) mode (starting with all the data points in one cluster and recursively dividing each cluster into smaller clusters). .

Proposed algorithm can be shown as,

The main steps of k-means algorithm are as follows:

(1) Select an Starting partition with K=k clusters; repeat Steps (2) and (3) until cluster membership stabilizes.

(2) Generate a new partition by assigning each pattern to its nearest cluster center.

(3) Calculate new cluster centers

VI. CONCLUSION

This Paper has been done for the purpose of retrieving the video from the Multimedia Database by using efficient algorithms to increase the performance of the system which is difficult in traditional video retrieving system. We are implementing Content Based Video Retrieval System. Video data storage, searching and indexing large databases efficiently and effectively has become a challenging problem. In order to solve the problem, there exist two distinct approaches in the literature. One solution is to annotate each image manually with keywords or captions and then search images using a conventional text search engine. The system uses a modified k-means clustering algorithm to improve the image segmentation, and uses a new similarity distance measure where object uniqueness is considered during computation. The proposed system performs better when the contrast between the main object and the background is visible in the image and performs worse when the image is complex and the objects have smooth edges. During the implementation, by considering object uniqueness during similarity distance computation improve the accuracy during retrieval.

The following are the advantages of the proposed method,

1) An improvement in video segmentation accuracy, especially for simple videos.

2) An improvement during dissimilarity computation by using the parameter of object uniqueness into consideration by k-means algorithm.

Our future work will involve testing the GPU implementation of these operations using even more recent GPUs compared against faster CPUs.

VII. REFERENCES

[1] Ashok Ghatol "Implementation of Parallel Image Processing Using NVIDIA GPU framework." advances in Computing Communication and Control. Springer Berlin Heidelberg, 2011. 457-464..

[2] O. Chapelle, P. Haffner, and V. Vapnik, —Svms for histogram based image classification,‖ IEEE Trans. Neural Netw., vol. 10, pp. 1055–1064,1999.

[3] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, ''Automatic partitioning of full-motion video,'' Multimedia Systems **1**~1!, 10–28 993!.cs.berkeley.edu/

[4] A. Nagasaka and Y. Tanaka, ''Automatic video indexing and fullvideo search for object appearances,'' in Visual Database Systems II, E. Knuth and L. Wegner, Eds., pp. 113–127, Elsevier Science Publishers~1992.

[5]. Content Based Video Retrieval Using Cluster Overlapping Deepak C R1,Sreehari S2, Gokul M3, Anuvind B4.

[6] G. Cybenko, "Dynamic load balancing for distributed memory multiprocessors,"J. Parallel Distrib. Comput., vol. 7, no. 2, pp. 279–301,1989.

[7] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collectionsin 3D," ACM Trans. Graph., vol. 25, no. 3, pp. 835–846,2006.

[8] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from Internet photo collections," Int. J. Comput. Vision, vol. 80, no. 2, pp. 189–210, Nov.2008.

[9] The National Science Foundation. (2008, Sept. 30). Cyber-physical systems.Program Announcements and Information. NSF, Arlington,

[10] X.-S. Hua and S. Li., "Personal media sharing and authoring on the web," in Proc. ACM Int. Conf. Multimedia, Nov. 2005, pp. 375–378.

[11] S. F. Chang and A. Vetro, "Video adaptation: Concepts, technologies, andopen issues," Proc. IEEE, vol. 93, no. 1, pp. 148–158, 2005.

[12] Origin Digital. (2009, Nov. 17). Video services provider to reduce transcoding costs up to half [Online]. Available: as studies/Case_Study_Detail.aspx?CaseStudyID=4000005952

[13] A Novel Content Based Image Retrieval System using K-means/KNN with Feature Extraction Ray-I Chang , Shu-Yu Lin , Jan-Ming Ho , Chi-Wen Fann 2, and Yu-Chun Wang 2 1 Dept. of Engineering Science and Ocean Engineering National Taiwan University Taipei, Taiwan.

[14] S. Shi, W. J. Jeon, K. Nahrstedt, and R. H. Campbel, "Real-time remote rendering of 3D video for mobile devices," in Proc. ACM Multimedia, 2009, pp. 391–400.

[15] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content based image retrieval at the end of the early years," IEEE Trans. Pattern Anal. Machine Intell., vol. 22, no. 12, pp. 1349–1380, Dec. 2000.

[16]. Kato,T. Database architecture for content-based image retrieval. Image Storage and Retrieval Systems, 112–123. (1999)

[17]. Datta, R., Joshi, D., Li, J., Wang, J. Z.: Image Retrieval: Ideas, Influences, and Trends of the New Age. ACM Computing Surveys, Vol. 40, No. 2, Article 5, April. (2008)

[18] P. Browne, C. Gurrin, H. Lee, K. M. Donald, S. Sav, A. F. Smeaton, and J. Ye. Dublin City University Video Track experiments for TREC 2001. In TREC 2001 - Text Retrieval Conference, MD, USA, 2001. National Institute of Standards and Technology.

[19] I. Buck. A Toolkit for Computation on GPUs. Addison-Wesley, 2004.

[20] A. F. Smeaton, P. Over, andW. Kraaij. Evaluation Campaigns and TRECVid. In MIR 2006 - 8th ACMSIGMM International Workshop on Multimedia Information Retrieval, 2006.

[21] H. Zhang, A. Kankanhalli, and S. Smoliar. Automatic partitioning of full-motion video. Multimedia Systems, 1(1):10–28, 1993.

[22] Avinash N. Bhute, B.B. Meshram —Automated ultimedia Information Retrieval using Color and Texture Feature technique‖ IJECCE Volume 3, Issue 5, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

[23] Ashok Ghatol "Implementation of Parallel Image Processing Using NVIDIA GPU framework." Advances in Computing Communication and Control. Springer Berlin Heidelberg, 2011. 457-464.