Journal of Systems Architecture 61 (2015) 639–645

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

A high performance hardware accelerator for dynamic texture segmentation

João P.F. Barbosa [a,b], Antonyus P.A. Ferreira [a,b,*], Rodrigo C.F. Rocha [a], Erika S. Albuquerque [a,b], Josivan R. Reis [a], Djeefther S. Albuquerque [a,b], Edna N.S. Barros [a,b]

[a] Integrated Circuits and Systems Laboratory, Center for Strategic Technologies of the Northeast, Recife, Brazil
[b] Informatics Center, Federal University of Pernambuco, Recife, Brazil

ARTICLE INFO

Article history:
Received 4 May 2015
Revised 26 August 2015
Accepted 25 September 2015
Available online 22 October 2015

Keywords:
Video segmentation
FFT
IFFT
FPGA
Hardware architecture
High performance system

ABSTRACT

Hardware accelerators such as general-purpose GPUs and FPGAs have been used as an alternative to conventional CPU architectures in scientific computing applications, and have achieved good speed-up results. Within this context, the present study presents a heterogeneous architecture for high-performance computing based on CPUs and FPGAs, which efficiently explores the maximum parallelism degree for processing video segmentation using the concept of dynamic textures. The video segmentation algorithm includes processing the 3-D FFT, calculating the phase spectrum and the 2-D IFFT operation. The performance of the proposed architecture based on CPU and FPGA is compared with the reference implementation of FFTW in CPU and with the cuFFT library in GPU. The performance report of the prototyped architecture in a single Stratix IV FPGA obtained an overall speedup of 37x over the FFTW software library.

1. Introduction

Dynamic textures are image sequences of moving scenes with certain stationary spatial and temporal properties, for which there are many potential applications [1]. The detection of dynamic textures is able to assist in identifying people and moving objects in an area within a complex background. The segmentation of moving background blocks is the first step in automated traffic or pedestrian monitoring.

The Fast Fourier Transform (FFT) algorithm [2] is useful for a wide range of engineering applications, almost every solution involving signals, images and videos can be improved using frequency analyses. Two-dimensional Fast Fourier Transform and Inverse Fast Fourier Transform are widely used in digital image processing and may be encountered in image compression, iris recognition, disease diagnosis and detection of heavenly bodies. The three-dimensional FFT that may be used in any type of three-dimensional signal is used more often in video analyses, mainly for motion detection.

Video applications normally process a large amount of data and contain real-time requests in order to achieve live video frame rates (30 frames per second). Therefore, the execution time of the 3-D FFT algorithm may be decisive in making certain applications viable.

The present paper presents a complete system for high-performance video segmentation using a dynamic textures algorithm. The system is a heterogeneous architecture with a general-purpose processor and a hardware accelerator implemented in FPGA. The implemented algorithm was that suggested by Jianghong Li in his paper entitled “Dynamic texture segmentation using Fourier transform”. The accelerator implemented in FPGA is responsible for 3-D FFT calculation and for determining the phase spectrum and the 2-D IFFT, the most costly tasks in the algorithm. We compare our results in FPGA with commercial reference software in CPU and GPU.

This article will explore the use of FPGAs in order to accelerate the dynamic texture video segmentation problem. Thus, the main contributions of this work are as follows

1. Analysis and development of an architecture to accelerate the dynamic video segmentation composed of a general purpose computer and an Altera Field Programmable Gate Array hardware;
2. Hardware implementation on a Gidel ProcStar IV platform [9];
3. Construction of a temporal synthesis of a configurable architecture to process different sizes of video frames;
4. Assessment of the processing speed-up for two different platforms: a CPU + GPU system and a CPU + FPGA system.

The paper is structured as follows: Section 2 presents a brief review of FFT and IFFT, focusing on the symmetry property that will be useful for simplifying computation. Section 3 presents a detailed overview of the dynamic texture segmentation (DTS) of video

---

* Corresponding author. Tel.: +55 08199035583.
E-mail address: apaf@cin.ufpe.br, pyetro.ferreira@cetene.gov.br (A.P.A. Ferreira).

http://dx.doi.org/10.1016/j.sysarc.2015.09.005
1383-7621/

developed by Li and Chen. Section 4 presents the analyses of a number of related works. Section 5 provides a detailed explanation of the architecture proposed for the hardware accelerator in FPGA to solve the video segmentation problem. Section 6 describes the performance speed-up of the system and presents the synthesis results and resource reports for the 3-D FFT, the phase spectrum and the 2-D IFFT hardware accelerator for the EP4SE530H35C2 Altera device [10]. Finally, Section 7 is presented the conclusions and future work.

2. The FFT and IFFT transforms

The Discrete Fourier Transform (DFT) requires on the order of N^2 operations where N is the transform size. The Fast Fourier Transform (FFT) by Cooley and Tukey [2] reduces the algorithm complexity from $O(N^2)$ to $O(Nlog_2N)$. This performance improvement opened a new area in digital signal processing and a large number of FFT algorithm implementations have consequently been developed. The Fast Fourier Transform (FFT) and the Inverse Fast Fourier transform equations are:

$$F(k) = \sum_{n=0}^{N-1} f(n)e^{\frac{-j2\pi kn}{N}} \quad k = 0, 1, , N-1 \tag{1}$$

$$f(n) = \frac{1}{N}\sum_{k=0}^{N-1} F(k)e^{\frac{j2\pi kn}{N}} \quad n = 0, 1, , N-1 \tag{2}$$

Where N is the transform size or the number of sample points in the frame data. $F(k)$ is the frequency output of the FFT at the Kth point where $K = 0, 1, , N-1$ and $f(n)$ is the time sample at the nth point where $n = 0, 1, , N-1$. Amongst the many properties of FFT, there is one which is especially interesting for application in this work: the symmetry property. It states that when all input data are real, the FFT result is symmetrical [8].

$$\forall x(k) \in R \Leftrightarrow X(n-m) = X(m),\ where\ k,\ n\ and\ m\ are\ integers,$$
$$0 \le m \le N/2\ and\ 0 \le m \le N/2.$$

The symmetry property enables the FFT to be calculated on a vector of real numbers much faster than on an array of complex numbers. When a real vector of size N is the input of FFT, only the first $N / 2 + 1$ components of the vector of the result need to be calculated. The other $N/2 - 1$ points of the result may be obtained from the first $N/2 + 1$ points in a data mirroring scheme. As we will observe later in this paper, this property is used to bring about a considerable reduction in the amount of data that is stored and processed by the FPGA while processing the video segmentation algorithm, since its input data are purely real.

3. Dynamic texture segmentation Fourier

Li, Cai and Chen [1] present a simple, efficient implementation of a video segmentation algorithm. This segmentation detects the parts in video frames with movement and applies a mask to those areas that do not move (background). It may be used, for example, to find moving people in video frames of surveillance cameras.

The video segmentation algorithm based on dynamic textures as proposed by Li and Chen may be broken down into seven steps as demonstrated in Fig. 1. In the first step, the video is decomposed into its N frames and, when necessary, each frame is converted into grayscale. The size of the frames is then decreased to obtain the dimension $n \times n$. This is carried out so as to reduce the amount of data to be processed in the operational sequence of the algorithm. This decrease in size provides an acceleration in the processing, but is limited by the visual quality of the results achieved at the end of video segmentation. Thus, it is necessary, through analysis of the results, to establish a limit between the decreased size of the frames and the quality of the results achieved in segmentation. The third step of the process consists of applying a Gaussian filter to each frame so as to reduce the noise that may enter through the process of acquiring the images that make up the video.

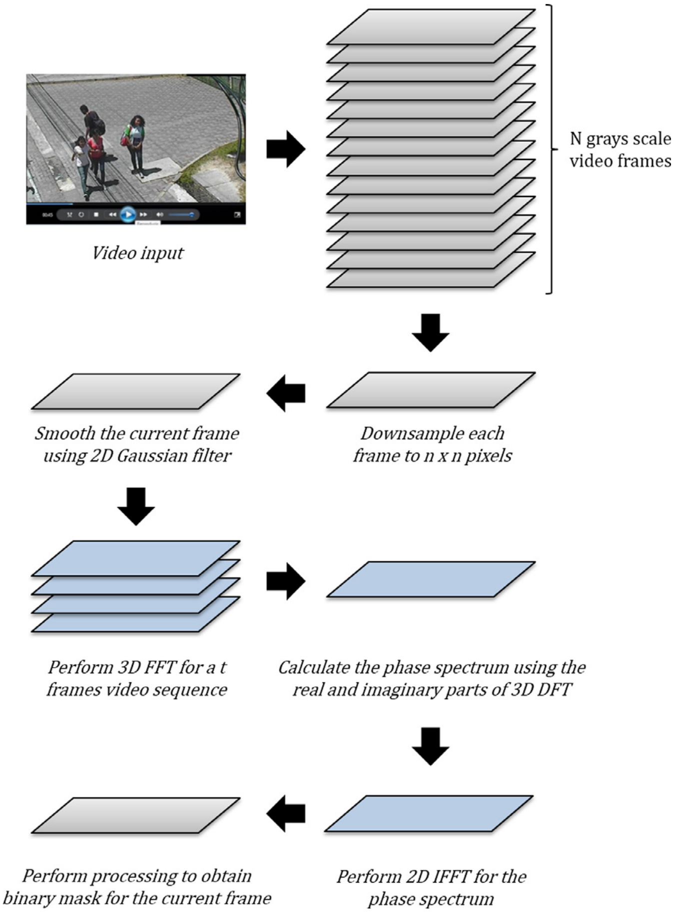

Fig. 1. Dynamic texture segmentation diagram.

The next three steps of video segmentation based on dynamic textures are those that require more computing power. In the fourth step, the 3-D FFT is applied to a t frameset. The value of t determines which temporal depth will be used during segmentation. The higher the t value, the greater the computational capacity expended by the algorithm. However, it is necessary to set a minimum value for t that ensures a balance between the good visual quality of the results of the algorithm and the processing time. Eq. (3) presents the 3-D FFT that is applied to the t set of video frames where each frame has width X and height Y.

$$F(I(x,y,t)) = \sum_{x=0}^{X-1}\left\{W_X^{k1}\sum_{y=0}^{Y-1}\left[W_Y^{k2}\sum_{t=0}^{T-1}W_T^{k3}I(x,y,z)\right]\right\} \tag{3}$$

The fifth and the sixth processing steps are those wherein the result of the 3-D FFT is used to extract the motion information contained within each pixel of an analyzed frame. For this, the fifth stage calculates the phase for each pixel of the frame that has resulted from the previous step, and 2-D IFFT is applied to the phase to finally extract the motion information from each pixel. These two steps may be represented by Eq. (4), where P is the phase obtained from the values of $F(x,y,t)$, and $\hat{I}(x,y,t)$ is the result of applying the IFFT on the values of the phase.

$$\hat{I}(x,y,t) = |F^{-1}\{e^{iP[F(I(x,y,t))]}\}|^2 \tag{4}$$

The seventh and final step of processing involves the construction of the binary mask that will separate the pixels for each frame with