

FPGA architecture for fast parallel computation of co-occurrence matrices

D.K. Iakovidis ^{*}, D.E. Maroulis, D.G. Bariamis

Department of Informatics and Telecommunications, University of Athens, Panepistimiopolis, Ilisia, 15784 Athens, Greece

Available online 3 March 2006

Abstract

This paper presents a novel architecture for fast parallel computation of co-occurrence matrices in high throughput image analysis applications for which time performance is critical. The architecture was implemented on a Xilinx Virtex-XCV2000E-6 FPGA using VHDL. The symmetry and sparseness of the co-occurrence matrices are exploited to achieve improved processing times, and smaller, flexible area utilization as compared with the state of the art. The performance of the proposed architecture is evaluated using input images of various dimensions, in comparison with an optimized software implementation running on a conventional general purpose processor. Simulations of the architecture on contemporary FPGA devices show that it can deliver a speedup of two orders of magnitude over software.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Image analysis; Co-occurrence matrix; FPGA; Texture

1. Introduction

The co-occurrence matrix is a powerful statistical tool which has proved its usefulness in a variety of image analysis applications, including biomedical [1,2], remote sensing [3], quality control [4] and industrial defect detection systems [5]. It captures second-order grey-level information, which is mostly related to human perception and the discrimination of textures.

Although the computational complexity of the co-occurrence matrix for an image of $N \times N$ dimensions is only $O(N^2)$, the processing power requirements for the computation of multiple co-occurrence matrices per time unit can be prohibitive for the analysis of large image streams, using software co-occurrence matrix implementations on general purpose processors. Such demanding applications include video analysis [1,6], content-based image retrieval [7], real-time industrial applications [5] and high-resolution multispectral image analysis [2].

Field Programmable Gate Arrays (FPGAs) are low cost, reconfigurable high density gate arrays capable of performing many complex computations in parallel while hosted by conventional computer hardware [8]. Their features enable the development of a hardware system dedicated to performing fast co-occurrence matrix computations, thus meeting the requirements of real-time image analysis applications. On the other hand, the Very Large Scale Integration (VLSI) architectures could be considered as competitive alternatives [9]. However, they are not reconfigurable and they involve high development cost and time-consuming development procedures.

Within the first FPGA architectures, dedicated to co-occurrence matrix computations, was the one presented in [5,10] for the computation of two statistical measures of the co-occurrence matrix. However, these measures were being approximated, without needing to compute the matrix itself. In a later work, Tahir et al. [2] developed an FPGA architecture for the computation of 16 co-occurrence matrices in parallel. The implementation considerations include symmetry, but do not include sparseness. As a result, a large FPGA area is utilized even for small input images.

^{*} Corresponding author. Tel.: +30 210 7275317; fax: +30 210 7275333.
E-mail address: rtsimage@di.uoa.gr (D.K. Iakovidis).

In this paper, we present a novel FPGA architecture for parallel computation of 16 co-occurrence matrices that exploits both their symmetry and sparseness to achieve improved processing times and smaller, flexible area utilization.

2. The co-occurrence matrix

The co-occurrence matrix of an $N \times N$ -pixel image I , comprises of the probabilities $P_{d,\theta}(i, j)$ of the transitions from a grey-level i to a grey-level j in a given direction θ at a given intersample spacing d

$$P_{d,\theta}(i, j) = \frac{C_{d,\theta}(i, j)}{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} C_{d,\theta}(i, j)}, \quad (1)$$

where $C_{d,\theta}(i, j) = \#\{(m, n), (u, v) \in N \times N: f(m, n) = j, f(u, v) = i, |(m, n) - (u, v)| = d, \angle((m, n), (u, v)) = \theta\}$, $\#$ denotes the number of elements in the set, $f(m, n)$ and $f(u, v)$ correspond to the grey-levels of the pixel located at (m, n) and (u, v) , respectively, and N_g is the total number of grey-levels in the image [11]. In accordance with [2], we choose $N_g = 32$ (5-bit representation).

The co-occurrence matrix can be regarded symmetric if the distribution between opposite directions is ignored. The symmetric co-occurrence matrix is derived as $P_{d,\theta}(i, j) = (P_{d,\theta}(i, j) + P_{d,\theta}(i, j)^T)/2$, where symbol T denotes the transpose matrix. Therefore, the co-occurrence matrix can be represented as a triangular structure without any information loss, and θ is chosen within the range of 0° to 180° . Common choices of θ include $0^\circ, 45^\circ, 90^\circ$ and 135° [1,2,6,12]. Moreover, depending on the image dimensions, the co-occurrence matrix can be very sparse, as the number of grey-level transitions for any given distance and direction, is bounded by the number of image pixels.

3. Architecture

The presented architecture was developed in Very high speed integrated circuits Hardware Description Language (VHDL). It was implemented on a Xilinx Virtex-XCV2000E-6 FPGA, which is characterized by 80×120 Configurable Logic Blocks (CLBs) providing 19,200 slices (1 CLB = 2 slices). The device includes 160 256×16 -bit Block RAMs and can support up to 600 kbit of distributed RAM. The host board, Celoxica RC-1000 has four 2 MB static RAM banks. The RAM banks can be accessed by the FPGA and the host computer independently, whereas simultaneous access is prohibited by the board's arbitration and isolation circuits.

An overview of the proposed FPGA architecture is illustrated in Fig. 1. The FPGA includes a control unit, four memory controllers (one for each memory bank) and 16 Co-occurrence Matrix Computation Units (CMCUs). Up to four input images of N_g grey-levels can be loaded in parallel to the available RAM banks. In accordance with [2], a 5-bit grey-level representation was used, i.e., $N_g = 32$. However, in [2] each image is loaded into a corresponding RAM bank using a 5-bit per pixel representation whereas in the proposed architecture a 25-bit per pixel representation is used. Each pixel is represented by a vector $\bar{a} = [a_p, a_0, a_{45}, a_{90}, a_{135}]$ that comprises of five 5-bit components, namely, the grey-level a_p of the pixel and the grey-levels a_0, a_{45}, a_{90} and a_{135} of its neighboring pixels at $0^\circ, 45^\circ, 90^\circ$ and 135° directions.

All FPGA functions are coordinated by the control unit which generates synchronization signals for the memory controllers and the CMCUs. The control unit also handles communication with the host, by exchanging control and status bytes, and requesting or releasing the ownership of the memory banks. Each CMCU is used

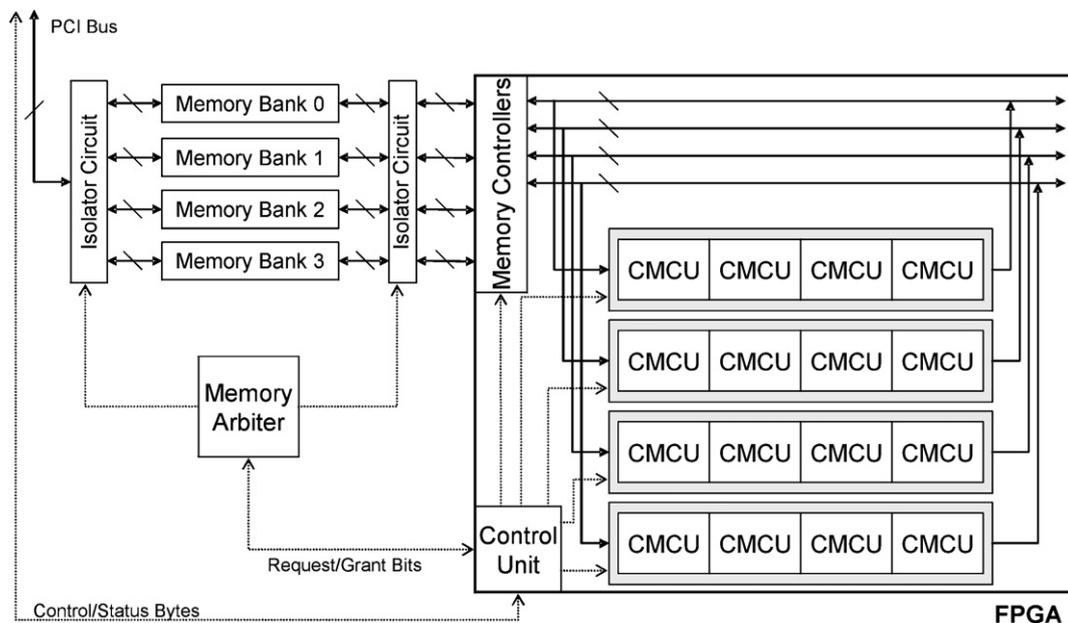


Fig. 1. Overview of the FPGA architecture.

Download English Version:

<https://daneshyari.com/en/article/461007>

Download Persian Version:

<https://daneshyari.com/article/461007>

[Daneshyari.com](https://daneshyari.com)