Parallelization and Performance Evaluation of an Edge Detection Algorithm on a Streaming Multi-Core Engine

Hashir Karim Kidwai, UAE University, UAE
Fadi N. Sibai, UAE University, UAE
Tamer Rabie, UAE University, UAE

ABSTRACT

In the world of multi-core processors, the STI Cell Broadband Engine (BE) stands out as a heterogeneous 9-core processor with a PowerPC host processor (PPE) and 8 synergic processor engines (SPEs). The Cell BE architecture is designed to improve upon conventional processors in graphics and related areas by integrating 8 computation engines each with multiple execution units and large register sets to achieve a high performance per area return. In this paper, we discuss the parallelization, implementation and performance evaluation of an edge detection image processing application based on the Roberts edge detector on the Cell BE. The authors report the edge detection performance measured on a computer with one Cell processor and with varying numbers of synergic processor engines enabled. These results are compared to the results obtained on the Cell’s single PPE with all 8 SPEs disabled. The results indicate that edge detection performs 10 times faster on the Cell BE than on modern RISC processors.

Keywords: Cell Broadband Engine, Edge Detection, Multi-Core Computing, Roberts Edge Detector

INTRODUCTION

The Cell Broadband Engine (BE) is a 3.2GHz heterogeneous multi-core microprocessor designed by STI (Sony, Toshiba, IBM) to achieve high performance computation for graphics, imaging and visualization applications, and generally to a wide scope of data parallel applications. It is composed of one 64-bit PowerPC Processing Element (PPE) serving as host processor, eight specialized co-processors called Synergistic Processing Elements (SPE) each with multiple integer and floating point execution units with vectorization capability, and one internal high speed bus called Element Interconnect Bus (EIB) which links PPE and SPEs together (IBM Corp.). The host PPE supports the 64-bit PowerPCAS instruction set architecture,
and the VMX (AltiVec) vector instruction set architecture to parallelize arithmetic operations. Each SPE consists of a Synergistic Processing Unit (SPU), and a SMF unit providing DMA (direct memory Access), memory management, and bus operations. A SPE is a RISC processor with an in-order dual-issue medium-size pipeline and a 128-bit SIMD organization for single and double precision instructions. Each SPE contains a 256KB instruction and data local memory area, known as the local store or LS for short, which is visible to the PPE and can be addressed directly by software. The local store does not operate like a superscalar CPU cache since it is neither transparent to software nor does it contain hardware structures that predict what data to load.

The EIB is a high bandwidth circular bus made of two channels in opposite directions (precisely, 1 address bus and 4 16B rings, each pair of rings runs in opposite directions) and allows for communication between the PPE, SPEs, memory and I/O. The Cell BE can handle 10 simultaneous threads and over 128 outstanding memory requests.

The following sections describe the edge detection application which we parallelized and developed on the Cell BE (Kidwai, Sibai, & Rabie, 2008), and the programming model chosen for it. Implementations on both PPE only (serial) and PPE-SPE (parallel or embedded) with varying number of SPEs are described. This is followed by the presentation of the execution times in each implementation and speedup analysis.

**EDGE DETECTION**

Detecting edges is a fundamental application in image processing and early computer vision. The edges of objects appearing in an image hold much of the information in the image. The image edges give the locations of where items are, their size, shape, and something about their texture. An edge is where the gray level of the image moves from an area of low values to high values or vice versa (Canny, 1986; Derière, 1987; Lindeberg, 1998; Lindeberg 1994; Zhang & Bergholm. 1997; Ziou & Tabbone, 1998). The edge itself is at the center of this transition. An edge detector should return an image with gray levels like those shown in the lower part of Figure 1. An edge is defined by a discontinuity in gray-level values.

To illustrate why edge detection is a non-trivial task, let us consider the problem of detecting edges in the one-dimensional signal of Figure 2. Here, we may intuitively say that there should be an edge between the 4th and 5th pixels due to the large difference in values between the gray level intensity of pixel 4 (4) and pixel 5 (152). If the intensity difference was smaller between the 4th and 5th pixels, it would not be as easy to say that there should be an edge in this corresponding region (Wikipedia).

The detected edge is a bright spot at the edge and dark areas everywhere else. Edge detection is often referred to as image differentiation.

The problem in edge detection is how to calculate the derivatives (the slope) of an image in all directions. Convolution of the image with masks is the most often used technique for doing this Faler (1990). These are basic masks that amplify the slope of the edge. The idea is to take a 3x3 array of numbers and multiply it point by point with a 3x3 section of the image. The sum of the products will then replace the center point of the image (the current pixel under consideration). The number of masks used for edge detection is almost limitless. Researchers have used different techniques to derive masks and then experimented with them to discover more masks.

Figure 3 shows examples of three different masks; Roberts, Prewitt and Sobel masks. There are two basic principles for each edge detector mask. The first is that the numbers in the mask sum to zero. If a 3x3 area of an image contains a constant value (such as all ones), then there are no edges in that area. The result of convolving that area with a mask should be zero. If the numbers in the mask sum to zero, convolving the mask with a constant area will result in the correct answer of zero. The second basic principle is that the mask should approximate
Related Content

Information Technology Systems Deliver Competitiveness for ABC Parcel Services
www.irma-international.org/article/information-technology-systems-deliver-competitiveness/3224/

Inventing the Future of E-Health
www.irma-international.org/chapter/inventing-future-health/13893/

Information Systems Redesign in a State Social Services Agency: A Case Study
www.irma-international.org/article/information-systems-redesign-state-social/44623/

Community Issues in American Metropolitan Cities: A Data Mining Case Study
www.irma-international.org/article/community-issues-in-american-metropolitan-cities/109515/

GIS and Remote Sensing in Environmental Risk Assessment
www.irma-international.org/chapter/gis-remote-sensing-environmental-risk/13799/