A Parallel Reconfigurable Architecture for Real-Time Stereo Vision

Lei Chen Yunde Jia
Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, PRC
{chenlei, jiyunde}@bit.edu.cn

Abstract

In this paper, a parallel reconfigurable architecture is proposed for real-time stereo vision computation. The architecture is divided into four components: input port, output port, memory and processor. We use task partition methods to achieve the maximum parallel and full pipeline processing of the algorithm implementation. We also adopt memory management to decrease the latency of memory access time and accelerate the processing speed. Data bandwidth control is employed to reduce the hardware resource consumption while maintaining precision demand of computation. Based on the proposed architecture and design method, we have developed a miniature stereo vision machine (MSVM33) to generate high-resolution dense disparity maps at the video rate for real-time applications.

1. Introduction

Stereo vision, which is inspired by human visual process, computes the disparity between correspondence points in images captured by multiple cameras for distance measurement. Real-time stereo vision has been widely used in intelligent robot navigation, smart human-computer interaction, intelligent surveillance, etc. Stereo vision algorithms involve a large number of regular, repetitive and computationally intensive operations on large sets of structured data. However, traditional software implementation of these algorithms, which runs on CPU-based platforms using sequential instruction execution mode and fixed control mechanism, cannot satisfy the complete video-rate processing demand. In order to sustain high computational load and real-time throughput of these stereo vision tasks, several hardware-based computing solutions have been reported [1 - 10]. The sum of absolute differences (SAD) or sum of square differences (SSD) [1, 3 - 6, 9, 10], census matching [2, 8] and local weighted phase correlation (LWPC) [7] stereo algorithms have been implemented in these systems, which mainly make use of hardware such as digital signal processors (DSPs) [1, 3], field programmable gate arrays (FPGAs) [2, 4 - 7], application specific integrated circuits (ASICs) [8, 9] and graphic processor units (GPUs) [10].

All systems [1 - 10] are advanced for an age, and have been widely used in practical applications. The architecture of these systems can be mainly divided into two categories: PC-based general purpose architecture [10] and customized special purpose architecture [1 - 9]. Systems based on general purpose architecture are usually physically large, high power consumption, and these disadvantages limit the utilization of this architecture in dynamic, variable, high power consumption applications. In contrast, customized special purpose architecture provides an alternative solution to system implementation. ASIC is designed and optimized for specific algorithms and operations, so systems using ASICs can achieve superior performance for a limited set of applications. However, ASICs need long design cycle and restrict the flexibility of the system and exclude any post-design optimizations and upgrades in features and algorithms [11]. Reconfigurable systems with great flexibility reduce the time, cost and expertise requirements in hardware-based algorithm implementation, and can be reprogrammed to facilitate improvement and modification in design at run-time without sacrificing system performance.

In this paper, we present a parallel reconfigurable architecture for stereo vision computation at the video rate. We use task partition methods to achieve the maximum parallel and full pipeline processing of the algorithm implementation. Memory management is adopted to decrease the latency of memory access time and accelerate the processing speed. Data bandwidth control is employed to reduce the hardware resource consumption while maintaining precision demand of computation. We have developed an FPGA-based real-time miniature stereo vision machine (MSVM33) based on this architecture to prove the effectiveness of the architecture.
The remainder of this paper is organized as follows. Section 2 describes the architecture and design methods. Section 3 introduces the implementation of our real-time miniature stereo vision machine. The performance and experimental results of the machine are discussed in Section 4.

2. Architecture

Computer vision algorithms perform a variety of data manipulation, signal processing, mathematical and graph theoretic computations. Typically, these tasks are performed repeatedly in some sequence, and thus are characterized by significant amount of temporal and spatial parallelism [12]. Reconfigurable computing utilizes hardware which can be adapted at run-time to facilitate greater flexibility without compromising performance [11]. Combining the advantages of parallel and reconfigurable mechanism, we present a parallel reconfigurable architecture for real-time stereo vision implementation.

2.1. Architecture Overview

This architecture is intended to cope with image data acquisition, stereo vision computation and data transmission in real-time, and designed to be adapted for a variety of stereo vision algorithms. The architecture contains four components: input port, output port, memory, and processor as illustrated in Fig. 1.

![Figure 1. Architecture of the real-time stereo vision computing](image)

Processor and memory are fundamental components of the architecture and critical for overall performance. The main design methods are:

1. Operations on an image are implemented in sub-images, in column and row direction respectively and in serial pipeline stages to achieve the maximum parallel and full pipeline processing in the processor.

2. Ping-pong mode management decreases the latency time in memory access.

3. Asynchronous FIFO design accelerates the processing speed.

4. Data bandwidth control maintains precision demand of computation and reduces the hardware resource consumption.

Processor servers as the computer’s CPU in this architecture. It deals with data transmission, parallel and pipelining processing, memory access control and communication between each component in the architecture.

Memory includes large, low speed, external memory and small, high speed, internal memory. External memory is used for input/output data stream buffering and intermediate computing results storage. Internal memory provides storage resource for FIFO, look-up-table, IP cores, etc, which are commonly used in code program execution of hardware description languages (HDLs).

The input port mainly consists of multiple CMOS or CCD optical sensors. The function of this component is to control these sensors to capture image data synchronously for stereo matching and transfer acquired data to the memory. The role of output port is to deliver the computing results, which are stored in the memory, out of the whole architecture to the host for high level practical applications using common transmission interfaces such as PCI, USB, IEEE 1394, etc.

2.2. Parallel and Pipelining Processing

As low level of vision, many stereo vision algorithm computations on image pixels are repetitive and only related to the values of local neighborhood pixels. Many computing operations in these algorithms can be precisely decomposed into image row and image column direction respectively or within an acceptable RMS error range. Meanwhile, the operation itself can also be divided into some independent stages. Due to these characteristics, it makes the parallel and full pipeline implementation of these algorithms reasonable. By translating algorithm into proper parallelism level and optimal long pipeline stages, the architecture can achieve outstanding computing performance [13].

As the most timing-critical and hardware resource consumption part, we adopt two kinds of parallelism in the implementation of stereo vision algorithms: temporal and spatial parallelism.

1. Spatial parallelism (processing unit duplication): computations on one image are constructed by copies of the same processing module, and the image is partitioned into sub-images, then these processing modules are executed on sub-images in parallel. As stereo vision deals with multiple images, operations can be computed among multiple images in parallel.
Temporal parallelism (processing time overlapping): computations in the algorithms are translated into serial separate and independent pipeline stages.

Fig. 2 illustrates the parallel and pipeline designs adopted in our work, and these designs can be spatial mapped onto the functional units of processor and memory resource in the architecture.

Figure 2. Parallel and pipeline processing. (a) The whole image is partitioned into n sub-images (for example, if n = 4, each sub-image is a quarter of the whole image), and n processing module which have the same function can execute the computation on the n sub-images in parallel. (b) Operation is computed in parallel among multiple images, and can be decomposed into column and row direction respectively in one image. (c) Operation on one pixel can be implemented as n-stages serial pipeline mode. Pixels move through the pipeline stage by stage so that when the pipeline is full, n separate and independent computation are being executed concurrently, each in a different stage. Furthermore, a final computation result emerges from the pipeline every clock cycle.
2.3. Memory Access and Data Bandwidth Control

The data size is large when multiple sensors are used together in one system. Consider, for instance, when we only use grayscale image data (8 bits per pixel respectively) from three sensors, the throughput is almost 211Mbits per second for 640×480×30fps applications. Data also needs to be stored into and loaded from memory during processing, so the memory access cost is usually one of the critical bottlenecks of the execution time. The main methods for memory access and data bandwidth control are as follows:

(1) Memory access management in ping-pong mode. The data storage in memory is based on ping-pong mode operation to decrease waiting latency time for data storage or load. For example, using two chips of memory in turn, one chip is used for buffering a frame of synchronously captured image data from input ports while the processor loads the previous frame of image data from the other chip for stereo vision computing at the same time.

(2) Asynchronous FIFO design. Large amount of data, such as original image data stream and intermediate computing results, should be stored in big low speed memory without consuming valuable small high speed memory. In order to achieve maximum performance, we design an asynchronous FIFO for efficient data exchange between two kinds of memory in the processing. The input of the FIFO is 16 bits data loaded from external memory every two clock cycle at 60MHz clock rate, and FIFO outputs 8 bit data each clock cycle at 120MHz clock rate to the internal memory. Processing speed is multiplied with this method and the serial output data is suitable for high speed full pipeline processing which is mentioned before.

(3) Data bandwidth control. In differential computing processing, there are a large degree of variation in data bandwidth and the required precision of the operands and operation is not the same. Analyzing the data flow and execution processing, we precisely choose the bandwidth for each data. Even the bandwidth for the same data in different processing stages is exactly chosen respectively. This design not only satisfies the precision demand of computing, but also reduces the hardware resource requirement in algorithm implementation.

The three methods are significant in design, and they make our architecture properly balance the hardware resource consumption, circuit size, peripheral chips versus performance trade-off.

3. Implementation

Based on the architecture described in the previous section, we have designed a trinocular miniature stereo vision machine. This system is composed of three parts: stereo head, processing unit and transmission interface as shown in Fig. 3.

Three triangular configuration COMS optical sensors on the stereo head can synchronously capture 640×480 pixels grayscale or color images at 30fps. Trinocular stereo vision system, with little increasing of hardware cost, can achieve better results than a typical binocular stereo vision system by providing a second pair of image sensors for reducing the ambiguity greatly. Three image sensors are placed at right angle to reduce distortions in trinocular epipolar rectification.

The processing unit mainly includes a high performance FPGA and tightly coupled local SRAMs. FPGA serves as the processing core of the whole system and its principal functions can be summarized as follows: (a) supplying external synchronous signal for image sensors on the stereo head because images for stereo matching must be captured simultaneously; (b) executing stereo matching computation to calculate disparities; (c) managing data access control of SRAMs; (d) outputting grayscale images and corresponding dense disparity maps simultaneously. For high computational requirements and real-time throughput of the system, the function and organization of ten chips of SRAM are as follows: two chips for buffering initial grayscale images captured by CMOS optical sensors on the stereo head in turn, six chips for storing the intermediate results of stereo matching computation and two chips for buffering output grayscale and disparity images in turn.

The transmission interface mainly includes a high performance FPGA, a DSP and IEEE1394 chipsets. FPGA controls the data transmission. DSP and IEEE1394 chipsets implement the IEEE 1394a high serial bus protocol for transferring data to the host computer at a speed of 400Mbps.

Corresponding to the architecture mentioned in Section2, in this system, the stereo head and transmission interface serves as the input port and output port respectively. The processing unit, which is the primary part in the system, works as processor and memory. The block diagram of the miniature stereo vision machine is shown in Fig. 4.
There are five main modules in the FPGA on the processing unit: image data capture module, image data pre-processing module, trinocular epipolar rectification module, stereo vision computing module and disparity extraction and post-processing module.

Image data capture module controls the multiple optical sensors to capture images simultaneously and stores the data in the SRAM chips. Image data pre-processing module mainly performs image filtering algorithms to reduce image noise in initial captured images. Trinocular epipolar rectification module is used to simplify the stereo match computation. The correspondence search is done only along image column or row lines in rectified images [14]. The disparity extraction and post-processing module calculates the right value from candidate disparities and combines the disparity value with intensity value of each pixel for output transmission. For the valid disparities, a simple parabolic interpolation method is adopted for sub-pixel value disparity estimation from candidate disparities.

The core module in the system is the stereo vision computing module, which computes the candidate disparities for each pixel in the reference image with the adaptive aggregation based cooperative stereo vision algorithm [15]. The main procedure of the cooperative algorithm is described in Fig. 5. Let C1, C2 and C3 denote the optical centers of three image sensors respectively. The corresponding images are I1, I2 and I3, and I1 is the reference image. The sum of the absolute difference (SAD) of photometric similarity is used for the disparity space initialization. We define one disparity level as one pixel in row line of rectified image, and the SAD value of two rectified stereo pairs is given by

\[ SAD(c, r, d) = |I_1(c, r) - I_3(c, r)| + |I_1(c, r) - I_2(c, r)|. \]

The initial values of disparity space \( D_0(c, r, d) \) is computed using normalized SAD

\[ D_0(c, r, d) = 1 - \frac{SAD(c, r, d)}{512}. \]

The process of disparity space iteration is given below:

Calculate the sum of support area \( S_n \) centered on \((c, r, d)\) in disparity space

\[ S_n(c, r, d) = \sum_{(x, y, z)} D_n(x, y, z) \]

\[ = \sum_{x \in \phi} \left( \sum_{y \in \phi} \sum_{z \in \phi} D_n(x, y, z) \right) \]

Compute the sum of inhibition area \( I_n \)

![Figure 3. Miniature stereo vision machine (MSVM33) overview](image)

![Figure 4. Block diagram of the system](image)
\[ I_n(c \ r \ d) = \sum_{(x \ y \ z) \in \{c \ r \ d\}} S_n(x \ y \ z) = \sum_{i=1}^{\text{Max}} S_n(c \ r \ i) - S_n(c \ r \ d) \]

where Max is the disparity search range.

Compute the inhibition gain using equations (3) and (4)

\[ G_n(c \ r \ d) = \left[ \frac{S_n(c \ r \ d)}{I_n(c \ r \ d)} \right]^\alpha \]

where \( \alpha \) is the coefficient of inhibition magnification.

Iteratively update the disparity space using equations (2) and (5)

\[ D_{n+1}(c \ r \ d) = D_0(c \ r \ d) \times G_n(c \ r \ d) \]

Repetitious disparity space iteration is implemented by the same iteration module cascaded. Different iterations are performed simultaneously and multiple iterations are executed in parallel with this method.

The values of disparity space will become convergent within limited times of iteration, and stereo vision computing module will output candidate disparities for disparity extraction and post-processing module.

To configure our stereo vision machine for a particular practical application, the user can replace the image data pre-processing module and disparity extraction and post-processing module with different algorithms or processing methods. Users can also modify parameters in the adaptive aggregation based cooperative stereo vision algorithm such as the size of disparity map, the search range of disparities, the size of support area, etc. This reconfigurable feature allows the system to be applied to different specific tasks.

### 4. Performance

The adaptive aggregation based cooperative stereo vision algorithm [15] is implemented entirely with a single chip of Virtex4 Xilinx FPGA on the processing unit. All modules inside FPGA are written in hardware description language VHDL. The integrated development environment of Xilinx ISE is used for synthesis, place and route. The synthesis report for the algorithm is listed in Table 1.

<table>
<thead>
<tr>
<th>Disparity Search Range</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support Area Size</td>
<td>5x5x3</td>
</tr>
<tr>
<td>Number of Iterations</td>
<td>6</td>
</tr>
<tr>
<td>Number of Slice Flip Flops</td>
<td>15381 out of 53248 29%</td>
</tr>
<tr>
<td>Number of 4 input LUTs</td>
<td>19215 out of 53248 36%</td>
</tr>
<tr>
<td>Number of Occupied Slices</td>
<td>15788 out of 26624 59%</td>
</tr>
<tr>
<td>Number of FIFO16/FRAME16</td>
<td>136 out of 163 83%</td>
</tr>
<tr>
<td>Number of DSP48s</td>
<td>19 out of 64 29%</td>
</tr>
<tr>
<td>Number of DCM ADVs</td>
<td>2 out of 8 25%</td>
</tr>
</tbody>
</table>

The global clock frequency is 60MHz. The stereo vision computing module with adaptive aggregation based cooperative stereo vision algorithm runs at 120MHz clock frequency, and up to 187MHz synthesized by Xilinx XST. The stereo vision machine can achieve a throughput of more than 30fps with 320x240 pixels dense disparity maps in 32-pixel disparity searching range and 5-bit disparity precision.

By comparison, the same cooperative stereo vision algorithm running with software programmed in C language takes approximately 3s when runs on a Dell Pentium D 3.40GHz PC with 1.0GB memory.

The significant speedup performance relative to software implementation is mainly due to the parallelism of the architecture, full pipeline programming implementation and optimized data flow path. Fig. 6 shows examples of the system final output images in pair which are synchronously transferred to the host computer.
The resolution of disparity maps, the disparity search range, the frame rate, the clock frequency of the stereo matching computation and power consumption of the whole system are closely related to the stereo vision system overall performance. The performance of our miniature stereo vision machine is characterized in Table 2.

Table 2. Miniature Stereo Vision Machine (MSVM33) Specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera Number</td>
<td>3</td>
</tr>
<tr>
<td>Input Image Size (max)</td>
<td>640×480</td>
</tr>
<tr>
<td>Disparity Search Range (max)</td>
<td>32</td>
</tr>
<tr>
<td>Disparity Map Size (max)</td>
<td>320×240</td>
</tr>
<tr>
<td>Frame Rate (max)</td>
<td>30fps</td>
</tr>
<tr>
<td>Algorithm</td>
<td>Cooperative</td>
</tr>
<tr>
<td>Processor</td>
<td>FPGA</td>
</tr>
<tr>
<td>IO Interface</td>
<td>IEEE 1394</td>
</tr>
<tr>
<td>System Size</td>
<td>9cm×7cm</td>
</tr>
<tr>
<td>Clock Frequency</td>
<td>120MHz</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>&lt;7W</td>
</tr>
</tbody>
</table>

5. Conclusion

We have presented a parallel reconfigurable architecture for real-time stereo vision. This architecture is shown to be flexible and effective for different algorithms implementation, improvement and modification in functionality design. We have also proposed some critical design methods closely related to the overall performance of the architecture. Based on the proposed architecture and methods, a miniature stereo vision machine (MSVM33) has been implemented for real-time dense high-resolution disparity mapping. MSVM33 is portable, low cost, low power consumption, high-speed computation and flexible for different specific tasks. It proves that our architecture and design methods are suitable for real-time stereo vision practical applications.

Acknowledgements

This work was partially supported by the Natural Science Foundation of China (60675021), the Chinese High-Tech Program (2006AA01Z120), and Beijing key discipline program.

References


