Heuristic Designs of SAD Algorithm for a Platform-Based Vision System

JunSeong KIM†(a) and Jongsu YI†, Members

SUMMARY Vision sensors provide rich sources of information, but sensing images and processing them in real time would be a challenging task. This paper introduces a vision system using SoCBase platform and presents heuristic designs of SAD correlation algorithm as a component of the vision system. Simulation results show that the vision system is suitable for real-time applications and that the heuristic designs of SAD algorithm are worth utilizing since they save a considerable amount of space with little sacrificing in quality.

key words: vision system, platform-based design, real-time image correlations, SAD algorithm, SoCBase

1. Introduction

Sensor systems are crucial in environmental observations for providing timely and affordable services in various applications including robots, factory automations, intelligent vehicles, and home networks. Especially, vision sensors, as a passive system, are much less sensitive to environmental interference [3], [4]. They are used for providing rich sources of information in scene recognition, motion detection, object tracking, surveillance, and so on. Vision sensors, however, generate high bandwidth data due to the nature of images. Standard PCs are commonly used for image analysis, but processing even small low-resolution images can easily take more than a second in software. This is well below the frame rates obtainable with commodity cameras, which can provide 30 or more images per second, and may be far too slow to provide services in real-time applications. Sensing multiple images simultaneously and processing them in real time, even in a low-resolution, would be a challenging task.

A stereo vision, which is based on two or more images taken from different view points, is able to build three-dimensional maps of its environment [1], [5]. It can provide much more complete information than a 2D image-based vision, but has to process, at least, that much more data. In the past decade, real-time stereo has become a reality. Some solutions are based on reconfigurable hardware and others rely on specialized hardware. For example, a stereo system for household mobile robot using Xilinx XC2V3000 FPGA runs at 60 fps for 640 × 480 images [4]. The PARTS reconfigurable engine using 16 Xilinx 4025 FPGAs runs 42 fps for 320 × 240 images [6]. The FingerMouse using an ASIC runs at 30 fps for 320 × 240 images [7]. Each of them, however, is designed for a specific application, and thus it is difficult to extend its functionality. A platform-based design alleviates the problem allowing a vision system to be easily modified, thus increasing design productivity. This paper introduces an extensible vision system using SoCBase, which is a platform developed in the Center for SoC Design Technology [11], [12], and presents various designs of SAD correlation algorithm, as a component of the vision system, trading off accuracy and resource usages. The vision system works for sensing devices in intelligent home networks since it is small in size and fast to support real-time services.

The remainder of this paper is organized as follows. Section 2 briefly presents the vision system using SoCBase platform. Section 3 provides design details of SAD algorithm and results of our experiments. Conclusions are summarized in Sect. 4.

2. A Platform-Based Vision System

Advances in CMOS technology integrating multi-million transistors in a single chip make it feasible to build complex systems for various applications. As the design complexity of digital systems grows platform-based design methodologies are widely accepted [10]. With a well-defined platform, complex systems can be designed in less effort and time by reusing components IPs.

SoCBase is a generic System-on-Chip design platform developed by the Center for SoC Design Technology, Seoul National University, Seoul, Korea [11], [12]. It is an abstraction, that hides details of possible implementation refinements, containing essential IPs such as bus, peripheral and memory controllers. With the concept of platform modules it allows designers not to consider the detailed modeling of a system increasing design productivity. A platform module is a subsystem integrating many component IPs that are commonly used together.

We implement a stereo vision system using SoCBase 1.0 platform, which is a base platform targeting low-power embedded systems. Figure 1 shows the block diagram of the vision system. Three platform modules are utilized: processor, memory, and peripheral modules. The processor module embeds an ARM922T core and the peripheral module contains low-speed devices including a SAD correlator, a TFT-LCD controller, and a keypad controller. The three platform modules communicate with each other through the AMBA AHB buses. The SAD correlator, as a component...
in the peripheral module, is implemented in hardware considering real-time operations since it is the most compute intensive part in the system. Section 3.1 provides the design details of the SAD correlator. The system controls including initialization, interrupt handling, and depth-map calculations are implemented in software considering flexibility and extensibility.

3. Stereo Image Correlations

For extracting three-dimensional information a stereo vision has to determine, given a pair of stereo images, which parts in one image correspond to which parts in the other image. Since a single pixel has too little to determine its correspondence sets of neighboring pixels, called image windows, are used for practical matching algorithms. That is, one window is fixed in one image and another window is moving in the other image for the matching process. By comparing them image correlation is measured and the matching window, that maximizes the similarity criterion, is determined. Common metrics in area-based correlations are Normalized Cross-Correlation (NCC), Sum of Squared Differences (SSD), Sum of Absolute Differences (SAD), Census, and Rank algorithms [2], [3], [8].

In this study, we use SAD algorithm since its regular structure provides linear data flow and abundant parallelism. The SAD function is defined to be

\[ C(x, y, \delta) = \sum_{y=0}^{wh-1} \sum_{x=0}^{ww-1} I_R(x, y) - I_L(x + \delta, y) \]  

where \( I_R(x, y) \) and \( I_L(x, y) \) represent coordinates in pixel on the right and the left image, respectively. The \( \delta \) is disparity number ranging between 0 and the maximum disparity value \( \Delta \). The \( ww \) and \( wh \) represent the window width and window height. The SAD function \( C(x, y, \delta) \) is evaluated for all possible values of the disparity, \( \delta \), and the minimum is chosen. That is, the criterion for the best match in SAD algorithm is minimization of the sum of absolute differences for corresponding windows.

3.1 A Design of SAD Algorithm

We design the SAD correlator, which fully reflects the operations of the SAD function \( C(x, y, \delta) \). Figure 2 shows the block diagram of the SAD correlator. It consists of three modules: pixel shift register (PSR), disparity calculator (DC), and minimum calculator (MC) modules.

The PSR module is organized as a type of one-dimensional array with the size of “scan line length × (wh – 1) + \Delta”. In order to exploit the parallelism in SAD algorithm, a single window from the left-PSR and (\( \Delta + 1 \)) windows from the right-PSR submodules are transmitted to the DC module simultaneously. The DC module executes the SAD function \( C(x, y, \delta) \). Figure 3 shows the detailed block diagram of the DC module, which is further divided into column-DC, shift-buffer, and window-DC submodules. By separating column-SAD calculations from the SAD function we can eliminate redundant operations among neighboring windows. First, the column-DC submodule calculates column-SAD results for a pair of columns from each window. Next, the window-DC submodule calculates SAD results for a pair of window with the corresponding column-SAD results. The shift-buffer submodule, which is organized as a two-dimensional array of the size “(\( \Delta + 1 \)) × \( ww \)”, is placed in between the
two submodules. It stores the column-SAD results and selectively provides them to the window-DC submodule. Finally, the MC module determines the best fit pair of windows among the \((\Delta + 1)\) SAD results from the DC module. The SAD correlator is implemented with only adders and comparators using tree topology to provide the performance of \(O(\log \Delta + 1 + \Delta)\). The interface of the SAD correlator is AMBA AHB compliant so that it can be easily integrated into the vision system using SoCBase platform.

3.2 Heuristic Designs of SAD Algorithm

There are trade-offs in the SAD correlator design between accuracy and resource usages. Accurate correlations require a large disparity range \([0, \Delta]\) and high-resolution images, which directly increase logic counts of the SAD correlator. Some applications may want small in size or low in price at the cost of accuracy degradation. Others may need rooms for application specific pre- or post-processing in hardware.

We test four heuristic designs of SAD algorithm for this trade-off. The followings are modifications of the original SAD correlator design (ORG) in Sect. 3.1 for approximations:

**Bit Cuts (BC)** In this scheme, we use small size of adders. That is, in the evaluation of the SAD function \(C(x, y, \delta)\) the least significant bits are ignored such that the same size of adders can be used throughout the tree adder topology in DC and MC modules. This may bring many draws in search for the best match and we give a priority to smaller \(\delta\) for simplicity.

**Sparse Windows (SW)** In this scheme, we reduce the number of SAD function evaluations within a window. That is, in the evaluation of the SAD function \(C(x, y, \delta)\) only a subset of the \(x\) and a subset of the \(y\) are considered. Specifically, we make the odd (or even) pixels of \(x\) and \(y\) be involved in SAD calculations.

**Stride Disparity (SD)** In this scheme, we reduce the number of SAD function evaluations with respect to \(\delta\). That is, in the image correlation using the SAD function \(C(x, y, \delta)\) only a subset of the \(x\) is considered. The strides of 2 \((\delta = 0, 2, 4, \ldots, \Delta)\) and 4 \((\delta = 0, 4, 8, \ldots, \Delta)\) are considered. The skipped portions in outputs are filled with the average of their neighboring pixels.

**Window Shape (WS)** In this scheme, we use a rectangular window in SAD calculations. It is unnecessarily common to use a square window \((ww = wh)\) in matching processes. We keep the size of window width reducing the size of window height or vice versa. Both a wide \((ww > wh)\) window and a long \((ww < wh)\) window are considered.

3.3 Experimental Results

We implement the SAD correlators in VHDL and synthesize them to determine their resource requirements and performance. For this experiment, we use Altera Excalibur EPCA10F1020C2 (1 million gates) with the maximum disparity \(\Delta = 64\), and the window size of \(9 \times 9\). Simulation shows the total execution time of \(8,070 \mu s\) to processing a pair of \(320 \times 240\) pixel stereo images. It corresponds to about 123 images per second and can support real-time services even for higher resolution images.

The original SAD correlator design costs 18,550 logic cells, which is about 48.3% of the programmable area of Excalibur and easily fits into a single modern FPGA. Figure 4 shows the relative resource usages of the heuristic designs to that of the original SAD correlator design. From the graph we can see that a considerable amount of space can be saved with the approximations. Note that the SW scheme saves about 40% and the WS scheme about 30% in resource usages. Figure 5 shows a sample of resulting depth-maps of the vision system. All the heuristic designs seem to give similar results in accuracy other than the SD scheme with stride = 4, which saves space about 49%. In order to quantitatively compare the accuracy of the heuristic designs we measure the relative matching rate \((m)\), which is a bit-by-bit comparison of depth-map results to that of the original SAD correlator design.

\[
m = \frac{1}{M \times N} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y)
\]

where \(f(x, y) = \begin{cases} 1 & \text{if } |I_O(x, y) - I_H(x, y)| \leq \theta, \ I_O(x, y) \\ 0 & \text{otherwise} \end{cases}\), \(I_O(x, y)\) and \(I_H(x, y)\) represent coordinates in pixel on the resulting depth-map of the original design and a heuristic design, respectively. \(\theta\) is set 12, which is 5% of the maximum difference (256), so that it can be tolerated by the differences in brightness of a given pair of stereo image. Sets of widely used stereo images from the Middlebury Stereo Page[9] - sawtooth, venus, cone, teddy, art, books, dools, moebius,
aloe, baby, cloth, rock - are considered and the average is taken.

Figure 6 shows the relative matching rate of the SAD correlators. We can see that the heuristic designs except the SD4 scheme produce similar matching quality to the original one: The SW scheme shows about 99.5% and the WS scheme about 98.0% in the matching rate. It results from the fact that the SAD function evaluations are quite redundant bringing unnecessarily accurate results in many cases.

When we look into carefully the matching process a small part of each window determines the correspondence. The other pixels in a window are mainly used to support the matching by reducing noise effects. We can conclude that the approximations in SAD algorithm are worth utilizing. Especially, the SW scheme shows about 40% savings in space with less than 1% in accuracy degradation.

4. Conclusion

Vision sensors provide plentiful information and are the most powerful means to comprehending an environmental situation. However, processing even a small image takes seconds in software. Hardware accelerations are necessary for real-time supports. In this paper, we introduce an extensible vision system using SoCBase. A platform-based design methodology is used to provide flexibility such that derivative designs are easy by modifying or adding few components from an once-established system. Then, we present various heuristic designs of SAD correlation algorithm, as a component of the vision system, with their associated accuracy and resource requirements. The approximations of SAD algorithm with respect to the key parameters of the window size and the maximum disparity value provide good savings in space with little accuracy degradations.

Acknowledgements

This research was supported by the Chung-Ang University Research Grants in 2010.

References