Real-time High-quality Stereo Vision System in FPGA *

Wenqiang Wang 1,2, Jing Yan 1, Ningyi Xu 1, Yu Wang 2, Feng-Hsiung Hsu 1
1 Microsoft Research Asia, Microsoft Building 2, HaiDian, Beijing, China
2 Electronic Engineering Department, TNLIST, Tsinghua University, Beijing, China

Abstract—Stereo vision is a well-known technique for acquiring depth information. In this paper, we present an FPGA-based real-time high-quality stereo vision system. By using AD-Census cost initialization, cross-based aggregation and semi-global optimization, the system provides high-quality depth results for high-definition images. This is the first complete real-time hardware system that supports both cost aggregation on cross-based regions and semi-global optimization on FPGA. The system can adjust image resolution, parallelism degree, and support region size to achieve maximum efficiency flexibly during the implementation. We test the accuracy of the system on the Middlebury benchmark and some real-world scenarios with different image resolutions. The results show the accuracy is among the best of FPGA-based stereo vision systems and competitive with current top-performing software implementations. We demonstrate the system using an Altera Stratix-IV FPGA board, processing 1024 × 768 pixel images at 30 frames per second.

I. INTRODUCTION

Stereo vision is an active research area in computer vision, as it is widely used in many applications. Usually, the stereo vision system has two cameras to capture two different images. Stereo matching is a key function of a stereo vision system. The purpose of stereo matching is to search for the disparity between corresponding points in two images to make the cost function achieves minimum results among all disparities. Then the depth could be calculated from the inverse of this disparity. Stereo matching is a complicated and time-consuming procedure, which makes it hard to process in real time on CPU. Current research efforts on stereo vision focus on depth accuracy and processing speed of stereo matching.

Stereo matching algorithms have two approaches: the local methods and the global methods [1]. The local method computes depth at local region, usually on a fixed support window; and the global method computes depth based on a global cost optimization [2]. Because the local method uses only local information to minimize the cost function, the accuracy is bad in low-texture and occlusion regions. While the global method shows better results on these regions [2]. However, the global method is not easily implemented using dedicated hardware because it causes huge pressure on the hardware resources due to huge intermediate computing results and large volume of irregular data access. This is why the majority of existing hardware implementations use local methods [3]. Typical global methods include the Semi-Global Matching (SGM), Dynamic Programming (DP), and Graph-Cuts (GC). GC optimizes an energy function over the entire image. This is hard to put on dedicated hardware platforms. SGM performs better than DP [2] and is a proven effective global method.

Most of current acceleration works for stereo matching have just evaluated their accuracy on Middlebury benchmark [4]. The resolution of the benchmark is not high. And they have avoided to discuss the depth quality when increasing the image resolution. However, the implementation could not always maintain the good depth quality for all resolutions. We explored current state-of-the-art stereo matching algorithms extensively and try to find an efficient design to achieve a better quality for the high-definition images on FPGAs. We are motivated by the AD-Census algorithm [5], which is ranked 1st on the Middlebury benchmark [4] for more than one year until the middle of 2012 and currently ranks 2nd. The current top algorithm has not been published until now (Sep. 2013). AD-Census is a combination of several existing state-of-the-art technologies. We examine the major hypotheses of these key technologies and further optimize them for hardware implementation on FPGA, which will be discussed in Section II and Section III. We will discuss the parameter setting and the quality on different image resolutions in Section IV.

In this paper, we propose a hardware-friendly stereo matching algorithm, and optimize it for high depth quality and real-time processing at high resolutions. This is the first complete real-time hardware system that supports both cost aggregation on cross-based regions and semi-global optimization on FPGA. We implement the whole algorithm on FPGA in a scalable way. The design is parameterized and can be scaled up easily. The prototype we developed is based on Altera EP4SGX230 FPGA. The system achieves real-time processing for 1024 × 768 pixel images and the average error rate on the Middlebury benchmark is 6.17%, which is among the best of all hardware implementation works.

II. STEREO MATCHING ALGORITHM

Cost initialization, cost aggregation, disparity computation and post processing are four common steps of stereo matching algorithm. To achieve high accuracy stereo matching, we aim at an algorithm with the improved cost aggregation technology.

---

* This work was supported by 973 project 2013CB329000, National Science and Technology Major Project (2013ZX03003013-003), National Natural Science Foundation of China (No. 61373026), and Tsinghua University Initiative Scientific Research Program.
and the global optimization. Our final algorithm is composed of AD-Census cost initialization, cross-based cost aggregation, semi-global optimization, disparity selection and post processing. The algorithm is further tuned to be more hardware friendly, which reduces resource utilization with acceptable accuracy loss. The details of our algorithm and the hardware implementation are discussed in the following subsections.

A. Cost calculation

AD-Census cost initialization is proven to improve accuracy in [5]. The information of absolute difference (AD) and census transform [6] is combined in initial cost. Then the initial costs in the support region are aggregated as the final cost. We use the cross-based cost aggregation proposed in [7] in this step. Firstly, we construct a cross for each pixel. The cross consists of four adaptive support arms, which is composed of pixels similar with the central pixel in color. The support region of pixel \( p \) is composed of all horizontal arms of the pixels in vertical arm of \( p \). As an advantage of this special structure, aggregation can be divided into two steps: the horizontal aggregation and then the vertical aggregation. The computing complexity is greatly reduced with this technology.

However, the straightforward implementation of the cross-based aggregation still costs too many logic resources in FPGA. This problem has been discussed in [8] and their inverted aggregation sequence is adopted in our implementation.

B. Semi-global optimization

Semi-global optimization proposed in [9] is to optimize the smoothness of the disparity map along different directions separately. For the optimization direction \( r \), the optimized cost is computed as follows:

\[
L_r(p, d) = C(p, d) + \min \{ L_r(p - r, d),
L_r(p - r, d + 1) \pm P_1, \min_k L_r(p - r, k) + P_2 \}
- \min_k L_r(p - r, k)
\]

The left term \( L_r(p, d) \) represents the path costs in direction \( r \). \( C(p, d) \) represents the aggregated costs. \( P_1 \) and \( P_2 \) are penalties for disparity discontinuities. The final optimized costs are the sum of all the path costs.

The original semi-global method optimizes the costs along 16 directions [9]. However, the directions opposite to the scan line data flow are hard to implement on FPGA. For example, if we want to optimize the costs along the up direction, the recursive optimizing operation must start at the bottom pixels of the image. This will destroy a pipeline design. In our system, only 4 directions, right, bottom, right bottom, and left bottom, are chosen for the semi-global optimization.

III. HARDWARE IMPLEMENTATION

We design a hardware structure for the proposed algorithm, as shown in Fig. 1. This is a fully parameterized and pipelined structure. We buffer the RGB data and census vectors of the stereo images to compute AD-Census costs. The initial costs of these pixels at a certain number of disparities are computed and sent to the aggregation module. After aggregating costs at these disparities, the semi-global optimization is applied to improve the accuracy. The raw disparity result is gotten through the WTA module. At last, the post-processing module is implemented to refine the disparity map.

In the following discussion, we note that the image size is \( H \times W \), the maximum arm length is \( L_{\text{max}} \), and the number of disparity levels is \( N_D \), which means the maximum disparity is \( d_{\text{max}} = N_D - 1 \).

![Fig. 1. Overall structure of the proposed stereo matching module.](image)

Parallel processing is important for accelerating stereo matching on FPGA. We adopt the hybrid-D parallel computing structure proposed in [10], which combines the disparity-level parallelism with the pixel-level parallelism. In our system, \( P_R \) neighboring pixels along the column direction are processed in parallel and \( P_D \) disparities are processed in parallel for each pixel. The total parallelism degree is \( P_D \times P_R \) and \( P_D \) is not necessary to be equal to \( N_D \). Each group of \( P_R \) rows is processed for \( K = N_D/P_D \) passes. In each pass, we can process \( P_D \) disparities, and totally we go through \( P_D \times K = N_D \) disparities.

A. Cost calculation

The cost initialization module provides the initial costs for the aggregation module. For the inverted aggregation module, the initial costs of the up and down arms are needed at the same time. To get these initial costs in parallel, we design line buffers to fetch multiple pixels, as shown in Fig. 2. The line buffer is composed of multiple BRAMs to build a wide output port. RGB and census data are written to line buffers progressively. As show in Fig. 2, the colored BRAMs are ready to output. A column of the image could be read from the output port of the buffer. Then the needed image and census data are selected out by the multiplexer. The read address of the left buffer and the right buffers can be different so as to realize multiple passes in disparity-level parallelism.

We adopt the inverted aggregation method proposed in [8] to implement the aggregation module. A total of \( P_D \) aggregation modules are generated to deal with \( P_D \) disparities in parallel. In each module, \( P_R \) pixels are processed in parallel. The initial

costs are aggregated first vertically and then horizontally. During the aggregation, the pixel number in the support region is recorded. All aggregated costs need to be divided by this pixel number for normalization.

B. Semi-global optimization

The semi-global optimization optimizes the cost along several paths separately. [3] has proposed a hardware structure which processes multiple pixels at one disparity in each cycle. However, this structure doesn’t support the disparity-level parallelism. If we use this structure in our design, the highest processing speed of semi-global optimization is \( P_R \) costs per cycle, which is far below the processing speed of the aggregation module. Thus the semi-global optimization module will be the bottleneck of the whole system.

To improve the data consuming bandwidth, we propose a buffer-based structure with both the disparity-level parallelism and the row-level parallelism, as shown in Fig. 3. This structure could process \( P_R \) pixels at all disparities in parallel.

There are some challenges to implement the disparity-level parallelism and the row-level parallelism. The aggregation costs for each pixel are generated in multiple passes, thus we could not get the aggregated costs at all disparities for one pixel at the same cycle. To solve this problem, we implement a cost buffer to rearrange the sequence of aggregated costs shown in Fig. 3. To implement row-level parallelism, the optimized costs \( L_r(p,d) \) of the upper row are used as inputs to current row in the directions bottom, right bottom, and left bottom.

IV. EXPERIMENTAL RESULTS

A. Processing speed and resource utilization

We build a prototype of real-time stereo vision system on Altera EP4SGX230 FPGA. The processing speed and resource utilization are determined by the image size, the disparity range, the disparity-level parallelism \( P_D \), and the row-level parallelism \( P_R \). These are parameterized to build a scalable design. We could tune parameters to make tradeoff between processing speed and resource utilization. Table I shows the resource utilizations with different parameters. With this platform, we could achieve a processing speed of 31.79fps for \( 1024^*768 \) images with 96 disparities. The system could process higher resolution images with a bigger FPGA.

<table>
<thead>
<tr>
<th>Image resolution</th>
<th>Settings</th>
<th>Processing speed</th>
<th>Resource utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( P_D )</td>
<td>( P_R )</td>
<td>( T_{CPU} )</td>
</tr>
<tr>
<td>640*480@964 disparities</td>
<td>8</td>
<td>1</td>
<td>61.09fps</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>122.11fps</td>
<td>38,085</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>244.41fps</td>
<td>79,314</td>
</tr>
<tr>
<td>1024*768@96 disparities</td>
<td>8</td>
<td>1</td>
<td>15.89fps</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>31.79fps</td>
<td>110,268</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>63.58fps</td>
<td>161,999</td>
</tr>
</tbody>
</table>

B. Quality Evaluation

To discuss the quality of our implementation, we will give the evaluation results on the benchmark and real-world scenarios. The results show our implementation is effective and scalable.

We firstly test our system on the Middlebury benchmark [4]. The average percentage of bad pixels in the result is 6.17%. We compare our implementation with some state-of-art stereo matching implementations and list results in Table II. The accuracy of our system is among the best in hardware accelerated stereo matching systems. The 1st is the original AD-Census implementation on GPU. AD-Census uses many promising technologies to achieve the best accuracy on the Middlebury. It requires large resources to implement all functions for AD-Census. The 2nd implementation uses a local method and the accuracy may drop with the increasing of the image resolution.

Then we test our implementation on some high-definition scenarios, as shown in Fig. 4. The first row is one frame of a 3D video. The original video [15] is captured by Sony HDR-TD10 Video Camera in a Natural History Museum and the resolution is 1280 \( \times \) 720. The second row is from Middlebury data set Art. The resolution of the data set is 1390 \( \times \) 1110. The first column is the original image and the second column is the result of our system. We could see that our system still provides clear depth results in high-definition scenarios. The good quality on high-definition images is because we
use the variable cross-based support region and the semi-global optimization. Compared to a fixed support region, the variable cross-based support region is self-adaptive and useful to identify the edge area. The semi-global optimization could improve the smoothness of the whole disparity map.

To validate the effect of the semi-global optimization, we remove the semi-global optimization module and then process the high-definition images again. The results are shown in the third column of Fig. 4. The quality of the Middlebury benchmark changes little. However, when dealing with the images captured by Sony HDR-TD10 Video Camera, we could see that the semi-global optimization improves the disparity results in large non-texture regions and makes the whole disparity map smooth. These improvements are important when dealing with high-definition images. So we believe our system could perform better in high-definition scenarios.

![Fig. 4. Results of high-definition images. The 1st column is the original image. The 2nd column is the result of our system. The 3rd column is the result without semi-global optimization.](image)

**V. CONCLUSION AND FUTURE WORK**

Our work focuses on the efficient hardware implementation with high quality and high resolution. The algorithm is from top-performing stereo matching algorithms on Middlebury benchmarks. We propose a scalable architecture for the key functions implementation and build a prototype using an Altera Stratix-IV board. This is the first complete work on FPGA that supports the aggregation on a cross-based region and semi-global optimization. The depth quality is evaluated on Middlebury benchmarks and real-world scenarios. The results show our implementation is among best performing stereo matching accelerators on both depth accuracy and processing ability. There is still space for further optimization for the post-processing stage. We will also start its ASIC design after the optimization work on current prototype.

**REFERENCES**


