스테레오 내시경 영상의 GPU 기반 고속 깊이맵 추정 기법

윌리엄*, 장지호**, 박인규*

*인하대학교 정보통신공학과, **한국전자통신연구원

{williem.pao@gmail.com, changjh@etri.re.kr, pik@inha.ac.kr}

Abstract

In this paper, we proposed a novel framework to perform fast depth map computation for a stereo endoscopic image pair. Conventional stereo matching approach performs slowly when it runs on CPU. To resolve this problem, we implement the GPU version of the method. However, naïve GPU implementation encounters speed limitation due to slow global memory access and non-parallelizable algorithm. It is observed that the bottlenecks are the cost computation and cost aggregation steps due to large number of computation based on the number of disparity labels. Thus, optimized GPU implementation is introduced to overcome the limitation. Experimental results show that the proposed method can achieve much faster performance than CPU and naïve GPU implementations.

1. Introduction

For the last few decades, computer vision techniques have been utilized by various medical hardware. One of the examples is DaVinci^{TM} Surgical system which utilizes surgical robots to minimize invasive surgery. It is useful to reduce the incision site and the pain compared to the conventional surgery. Thus, patients can obtain faster recovery speed and shorter hospitalization period. The system captures image through the endoscopic camera and displays it on the monitor to be observed. However, the information is limited due to the narrow vision. To obtain more information, a stereo endoscopic camera is utilized. Therefore, 3D information of human organ can be obtained accurately. Using the 3D information, we can build the 3D model of human organ so that each doctor can do some rehearsal before performing real surgeon to minimize the human error.

First, we implement the stereo matching method on CPU to confirm the depth accuracy. We follow the framework in [1] as the guidance. Note that the stereo matching framework consists of four steps: pre-processing, cost computation, cost aggregation, and post processing [3]. Method in [1] has achieved real-time performance on FPGA so that we believe that the method is suitable for parallelizable on GPU. Note that we utilize CUDA library as the parallelization toolkit. CUDA has been known as the state-of-the-art toolkit for GPU implementation for NVIDIA hardware. After the CPU implementation, we refine the performance by implementing the naïve GPU implementation. It is observed that there are bottlenecks on cost computation and cost aggregation steps, especially due to slow memory access, large number of disparity label, etc. Thus, we optimize those steps so that we can achieve faster performance. Experimental results show that the proposed method overcome the performance on both CPU and naïve GPU implementations.

2. Proposed Method

In the proposed framework, the stereo matching consists of four steps. At the beginning, bilateral filter is performed to remove the noise in the images captured in the real world environment. For matching cost computation, we utilize AD-Census model which is the mixture of Absolute of Difference (AD) and Census data costs. As the cost aggregation, modified information permeability filter is chosen [1]. To obtain the disparity map, Winner-Take-All (WTA) is performed. Several post-processing methods, such as left-right consistency check, subpixel enhancement, and adaptive median filter, are used to refine the current disparity map. As the bottlenecks are the cost computation and cost aggregation, we optimize those steps as described in the following subsections.

2.1 Cost Computation Optimization

To optimize the GPU implementation, we do some optimization scenarios. The first scenario is to find the thread block size used for the GPU implementation. It is important to find the best thread block size that results in the fastest computational time. To find the best size, we evaluate the computational time for each candidate and select the fastest one empirically. It shows that 64 x 1 thread block size obtains the fastest computational time. Then, we utilize shared memory and register memory to reduce the computational time. Both memories are known to be faster than global memory on GPU device but their size is limited. Thus, we should design the usage well. To utilize the register memory as much as possible, the number of local variable used inside the GPU kernel should be reduced. Note that the local variable should be called many times so that it is saved in register memory for temporary period.

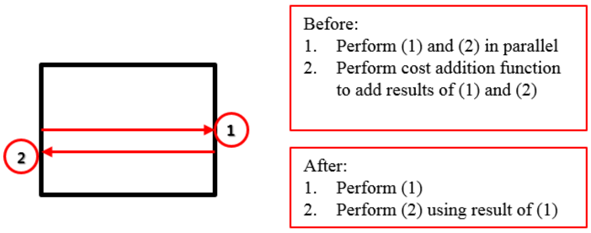

1
2
Before:
1. Perform (1) and (2) in parallel
2. Perform cost addition function to add results of (1) and (2)
After:
1. Perform (1)
2. Perform (2) using result of (1)

Fig. 1. Overview of modified cost aggregation method

While we cannot explicitly use the register memory, we can utilize the shared memory freely. To obtain the best performance, both memories should be fully utilized. When there is a variable inside an global memory array that is called many time, it is recommended to use the shared memory. On the other, register memory is used for a local variable. To improve the performance, we reduce the number of insufficient computation by ignoring boundary pixels and loop unrolling. It removes branching (IF) computation that takes high computation. After the optimization, we reduce the computational time from 0.0934 ms (naïve) to 0.046 ms (optimized).

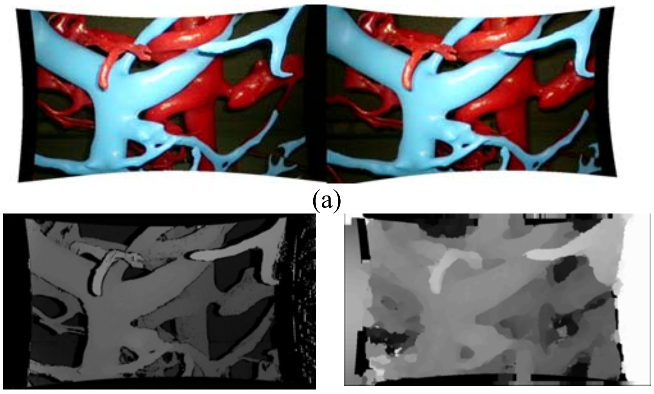
2.2 Cost Aggregation Optimization

On the beginning, we utilize the optimization scenario as described in Subsection 2.1. However, it does not improve the performance as much as the cost computation step because there is sequence dependency in the modified information permeability filter [1]. Note that the algorithm is an iterative algorithm that is not parallelizable friendly. Therefore, we modify the cost aggregation by removing insufficient computation instead of optimizing the CUDA kernel. We observe that by removing the cost addition computation, we can reduce the computational by half. In addition, we found that the quality of the disparity map is not different. Fig. 1 shows the overview of the original and modified cost aggregation method. The computational time is reduced from 0.485 ms (naïve) to 0.224 ms.

3. Experimental Results

The proposed algorithm is implemented on Intel core i7-6700 processor @ 3.4 GHz. We use a dataset that is provided in [2]. The dataset consists of two different images: 3D printed human organ models and actual organ of a pig. Fig. 2 shows the qualitative result of a 3D printed human organ model. To confirm the flexibility of our GPU implementation, we perform the evaluation on two different GPUs: NVIDIA GTX 980M (1536 CUDA cores) and NVIDIA GTX 1080 (2560 CUDA cores).

Table I show the computational time comparison between the CPU implementation and two GPUs. Even though the GPU implementation has additional computation on GPU memory allocation and memory free, it shows that the performance still much faster than CPU implementation. This is due to high load on cost computation and cost aggregation. GTX 1080 achieves faster computational time than GTX 980M because it has more CUDA cores denoting that it can perform more computation in parallel.



(a)

(b) (c)

Fig. 2. Results of the proposed framework. (a) Input stereo endoscopic image; (b) Ground truth; (c) Output depth map.

Table I. Comparison of computational time

	CPU	GPU [GTX 980M]	GPU [GTX 1080]
GPU Memory Allocation	0	0.049	0.0449
Pre-Processing	1.374	0.0153	0.006
Cost Computation	16.573	0.1654	0.046
Cost Aggregation	10.621	0.2773	0.224
Post-Processing	0.845	0.0968	0.032
GPU Memory Free	0	0.0039	0.0298
Total Time	29.413	0.6077	0.3827

4. Conclusion

We introduced a stereo matching framework on GPU to achieve fast computational speed. Our framework dealt with stereo endoscopic camera. GPU optimization were done on cost computation and cost aggregation step because it has high computational complexity. Experimental results show that the proposed framework can obtain faster speed than CPU implementation.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2013R1A2A2A01069181). This work was supported by the ETRI R&D Program (16ZC1400, The Development of a Realistic Surgery Rehearsal System based on Patient Specific Surgical Planning) funded by the Government of Korea.

References

[1] J. Chang, *et al.*, "Real-time hybrid stereo vision system for HD resolution disparity map," Proc. of BMVC, September 2014.

[2] J. Chang, *et al.*, "Performance evaluation of depth map generation algorithm for stereo endoscopic camera," Proc. of ICCIP, November 2016.

[3] D, Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," IJCV, vol. 47, no. 1, pp. 7-42, May 2002.