MEMOCODE 2013 Hardware/Software Co-design Contest: Stereo Matching

Armin Ahmadzadeh*, HatefMadany*, kianoshjafary*
*IPM- Institute for Research in Fundamental Sciences

Abstract
Belief propagation is the common method for solving stereo matching problem. Prevalent implantation of this method needs a high memory bandwidth and is normally serial. We benefit from current techniques to reduce running time of algorithm and parallelize the algorithm in full extent. The quality evaluation according to the given evaluation code in the Memocode 2013 Design Contest shows 15% error in pixels, comparing the ground truth in Tsukuba image set. The implementation runs four orders of magnitude faster than the reference code.

I. Introduction

We started our implementation based upon the given reference code. In order to improve the performance, we have used [1] that reduces the time complexity from $O(L^2)$ to $O(L)$. Furthermore, using the existing techniques proposed in the mentioned paper, like multi-grid and bipartite, we have optimized our implementation.

The rest of this paper is organized as follows: In the second section, we describe our method in which several optimization techniques are deployed to gain the maximum speed up possible. The third section shows our experimental results on different platforms. Finally we present the conclusion of this work in the last section.

II. Our Solution

To achieve the best possible performance, we have applied several methods to the aforementioned algorithm. We introduce a new method called Bidirect, which combines two phases of message updating based on the observation that opposite direction message passing can be computed simultaneously. The multi-grid method consumes considerable memory. We reduce the memory consumption by using only two HBP layers; First layer includes data costs and the second layer contains the compressed data costs. The windows size of the compression is changed by the constant value LSCALE. In order to fully utilize memory access, the arrays used in these two levels are separated.
Separating memory arrays causes the whole array cache accommodation in each level The BP algorithm is executed on the compressed level. BP_iteration is defined as the number of iteration of BP algorithm in the code

Because of our optimization techniques, the second level is automatically stored in the CPU cache and reduces memory access latency. At the end of the execution of BP on the compressed level, the result will be expanded to the next level which contains the uncompressed data. In order to make sure that the second level is fully stored in the CPU cache, compiler derivatives are used.

Finally, we have used OpenMP API to automatically exploit multithread functionalities of the Intel processors.

III. Experimental results

To evaluate our method, three platforms were used. An Intel Xeon x5650 with six cores, at 2.67GHz with hyper threading [4]. Another platform we used, is Intel Core i7 920 with four cores, 2.67GHz exploiting hyper threading. The other platform is an Intel Core i5 M460 with two cores clocked at 2.57GHz with hyper threading. Executing the code on the Tsukuba results in 13900 mismatches on all platforms. Further details on average runtime and price are given in Table1.

Table1. Time and Performance on different platforms

Platform	Exec. Time	Cost (Proc. + 256MB Mem.)
Intel Xeon x5650	615µs	$305 + $10
Intel Core i7 920	890µs	$1003 + $10
Intel Core i5 M460	2990µs	$45 + $10

VI. Conclusion

In this paper, we have proposed a modified version of the BP algorithm to reduce running time and improve data accuracy. The experimental result on Tsukuba test bench data shows 4 orders of magnitude speed up comparing to the reference code. In terms of accuracy, 15% of the pixels differ with the ground truth disparity map that is even better than reference code.

References :

[1] P.F,Felzenszwalb and D.P Huttenlocher .Efficient belief propagation for early vision

[2] Middlebury MRF benchmark, Available online at:http://vision.middlebury.edu/MRF/code/

[3] Memocode contest, Available online at:http://memocode.irisa.fr/2013/

[4] Intel product information, Available online at:http://ark.intel.com/products/