ACCELERATING EARTH MOVERS DISTANCE WITH INSTRUCTION SET EXTENSION FOR IMAGE RETRIEVAL

Guangyu Yu, Xiaowei Xu, Zeyu Yan and Hu Yu
School of Optical and Electronic Information, HUST, Wuhan, 430074, China
*Corresponding Author's Email: bryanhu@hust.edu.cn

ABSTRACT

Image retrieval is one of the most popular applications for computer vision and pattern recognition, in which similarity computation is the computational bottleneck. Earth Movers Distance (EMD) is one of the most popular similarity measure for image retrieval, which has a high time complexity of $O(n^3 logn)$. Recently, with the explosion of image data, EMD acceleration has been emerging. In this paper, we propose an EMD acceleration architecture based on instruction set extensions for image retrieval. The EMD acceleration architecture achieves a speedup of 1.3x-2.2x over software implementations. The main advantage of the proposed architecture over existing hardware accelerations is that it can support larger histograms. Specifically, the number of supported variables in histograms has 1-2 orders of magnitude improvement.

INTRODUCTION

Earth Movers Distance (EMD) is one of the most popular similarity measure for image retrieval. Today, EMD acceleration has been urgent, which is mainly due to the following two reasons. Firstly, the time complexity of EMD is up to , which cannot meet the processing requirement of many realtime applications. Secondly, huge amounts of image data need to be processed more efficiently with the advent of the era of big data.

Recently, studies related to EMD mainly focus on software acceleration. Assent et al. [4] accelerated EMD algorithm in Filter-and-Refine framework and proposed several lower bound methods for EMD. Wichterich et al. [12] presented a new dimension reduction method, and results showed that EMD distances in the derivative space is a lower bound of EMD in the original space.

Another approach [9] [13] for EMD acceleration research focused on EMD simplification for specific applications. Andoni et al. [3] achieved EMD simplification in high dimensional space. Pele et al. [7] simplified EMD calculation by limiting the number of connections in the graphs. Jang et al. [6] and Shirdhonkar et al. [10] proposed linear EMD methods by simplifying EMD, respectively. Andoni et al. [2] proposed a method for accelerating planar EMD. There exists a hardware acceleration work for simplex algorithm, which can support general EMD computation [5]. However, the work can only support 751 variables, which corresponds to only 27 bins for pair-wise EMD computation.

In this paper, we propose an EMD acceleration architecture based on instruction set extensions. Particularly, max-flow min-cost algorithm [1] is adopted which is specific for the EMD problem. With algorithm analysis, we design extended instructions (dedicated hardware) for EMD bottlenecks. To achieve parallel input/output operation, we design a customized memory for the extended instructions. The main advantage of the proposed architecture over existing hardware accelerations is that it can support larger histograms.

ALGORITHM ANALYSIS

Algorithm Process Analysis

The call graph of max-flow min-cost algorithm for EMD is shown in Fig. 1.

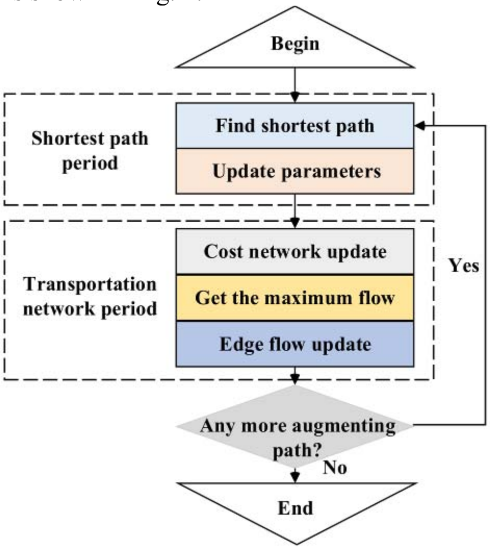


Fig.1. Illustration of max flow min cost algorithm for EMD.

Algorithm Bottleneck Analysis

Based on the analysis of max-flow min-cost algorithm for EMD. We find that finding the shortest cost path and updating the cost of the network operation have high time complexity.

The percentages of runtime of finding shortest cost path and updating network with different experimental configurations are shown in Fig. 2. We find that the two operations of the network account about 75% of the total running time.

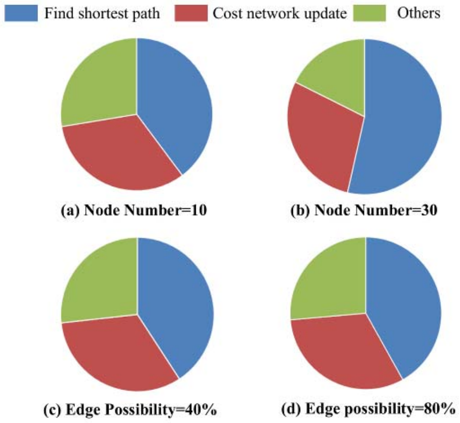


Fig.2. Illustration of EMD bottleneck.

EMD ACCELERATION ARCHITECTURE

Acceleration Architecture

The EMD acceleration architecture based on instruction set extension is shown in Fig. 3. An on-chip RAM is connected to the extended instructions to get higher throughput. We design three extended instructions: *Initial*, *ExtrMin* and *ReduceCost* instruction. And the three custom instructions are integrated into an extended multi-cycle instruction.

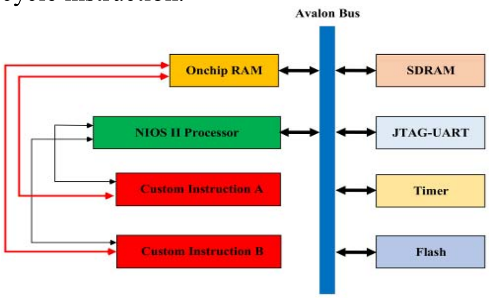


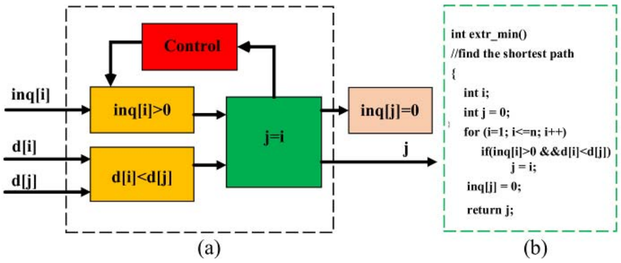
Fig. 3. EMD acceleration architecture based on instruction set extension.

Initialization Instructions Design

Initial instruction mainly realizes the function of loading data to On-chip RAM to store the data in a specific area before calling *ExtrMin* and *ReduceCost* instructions.

Finding the Shortest Cost Path Instruction Design

The hardware implementation of the *ExtrMin* instruction is shown in Fig. 4(a). The corresponding code is shown in Fig. 4(b).



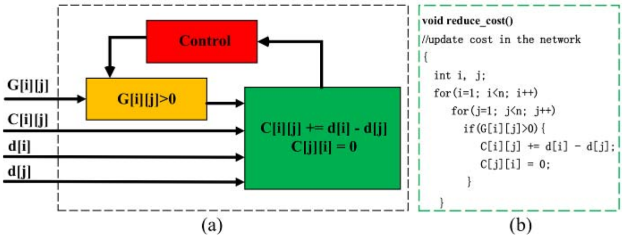
```
int extr_min()
//find the shortest path
{
   int i;
   int j = 0;
   for (i=1; i<=n; i++)
      if(inq[i]>0 &&d[i]<d[j])
            j = i;
    inq[j] = 0;
    return j;
```

Fig. 4. Instruction design of shortest path operation, ExtrMin.

The main calculation is an iteration operation caused by the *for* operation. Three input data can be read parallel, and all the operation can be fully pipelined. The whole calculation can be processed efficiently and can be processed in one clock cycle.

Cost Network Update Instruction Design

The hardware implementation of cost network update instruction *ReduceCost* as shown in Fig. 5(a). The corresponding code is shown in Fig. 5(b).



```
void reduce_cost()
//update cost in the network
{
  int i, j;
  for(i=1; i<n; i++)
     for(j=1; j<n; j++)
       if(G[i][j]>0){
          C[i][j] += d[i] - d[j];
          C[j][i] = 0;
       }
  }
```

Fig. 5. Instruction design of cost update operation, ReduceCost.

The main calculation is a two-layer iteration operation caused by the *for* operation. Multiple input data can be read in parallel.

EXPERIMENTS

Experiment Setup

We implement the acceleration architecture based on ISE on Altera Nios II processor. Particularly, a FPGA chip EP4CE15 is selected. Hardware design are implemented with Quartus II 13.1, and Nios II processor design are realized with built-in Qsys in Quartus II 13.1. The configuration of the Nios II processor is shown in Table I.

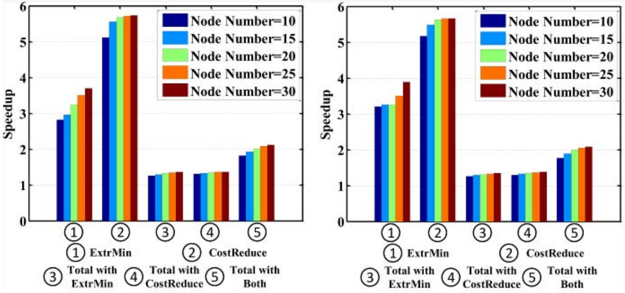
TABLE I. CONFIGURATION OF THE NIOS II PROCESSOR

Module	Parameter	Value
Nios II processor	Performance	II/f
Onchip memory	Dual Port R/W	Enable
	Data width	32bits
	Memory Size	36000bytes
	R/W Delay	1 cycle
Timer	Delay	1us
	Readable snapshot	Enable
SDRAM controller	Data width	16bits
	Memory Size	32Mbytes
Clock	Frequency	100MHz

Experiment Result

As shown in Fig. 6, the average speedup of *ExtrMin* and *ReduceCost* instruction is about 3.2x and 5.5x compared with software implementation, respectively. The speedup of the overall EMD calculation is only about

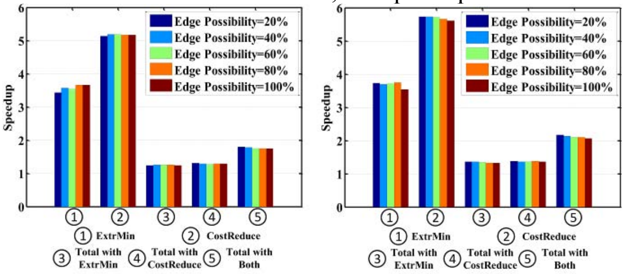
1.3x with only *ExtrMin* instruction or *ReduceCost* instruction. When *ExtrMin* and *ReduceCost* instructions are both adopted, the speedup can reach about 2.0x.



(a) Edge possibility=40% (b) Edge possibility=80%

Fig. 6. Speedup with different node number and edge possibility of 40% (a) and 80% (b).

As shown in Fig. 7, the speedup of the number of bins of 30 is significantly greater than the speedup of that of 10. When the number of bins is 30, the speedup reaches 2.2x.



(a) Node number=10 (b) Node number=30

Fig. 7. Speedup with different edge possibility and node number of 10(a) and 30(b).

SUMMARY AND CONCLUSIONS

In this paper, we propose a EMD acceleration architecture based on instruction set extensions. Particularly, max-flow min-cost algorithm [1] is adopted which is specific for the EMD problem. With algorithm analysis, we design extended instructions (dedicated hardware) for EMD bottlenecks. In order to achieve parallel input/output operation, we design a customized memory for the extended instructions. The main advantage of the proposed architecture over existing hardware accelerations is that it can support larger histograms. The ex-perimental results show that the EMD acceleration architecture has a speedup of 1.3x to 2.2x. Specifically, the supported number of bins in histograms has a 1-2 orders of magnitude improvements.

REFERENCES

[1] R. K. Ahuja. Network flows. PhD thesis, TECHNISCHE HOCHSCHULE DARMSTADT, 1988.

[2] A. Andoni, K. D. Ba, P. Indyk, and D. Woodruff. Efficient sketches for earth-mover distance, with applications. Institute of Electrical and Electronics Engineers, 2010.

[3] A. Andoni, P. Indyk, and R. Krauthgamer. Earth mover distance over high-dimensional spaces. In Proceedings of the nineteenth annual ACM- SIAM symposium on Discrete algorithms, pages 343–352. Society for Industrial and Applied Mathematics, 2008.

[4] I. Assent, A. Wenning, and T. Seidl. Approximation techniques for indexing the earth mover's distance in multimedia databases. In 22nd International Conference on Data Engineering (ICDE'06), pages 11–11. IEEE, 2006.

[5] S. Bayliss, G. A. Constantinides, W. Luk, et al. An fpga implementation of the simplex algorithm. In 2006 IEEE International Conference on Field Programmable Technology, pages 49–56. IEEE, 2006.

[6] M.-H. Jang, S.-W. Kim, C. Faloutsos, and S. Park. A linear-time approximation of the earth mover's distance. In Proceedings of the 20th ACM international conference on Information and knowledge management, pages 505–514. ACM, 2011.

[7] O. Pele and M. Werman. Fast and robust earth mover's distances. In 2009 IEEE 12th International Conference on Computer Vision, pages 460–467. IEEE, 2009.

[8] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. International journal of computer vision, 40(2):99–121, 2000.

[9] B. E. Ruttenberg and A. K. Singh. Indexing the earth mover's distance using normal distributions. Proceedings of the VLDB Endowment, 5(3):205–216, 2011.

[10] S. Shirdhonkar and D. W. Jacobs. Approximate earth movers distance in linear time. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE, 2008.

[11] S. Skiena. Dijkstras algorithm. Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Reading, MA: Addison-Wesley, pages 225–227, 1990.

[12] M. Wichterich, I. Assent, P. Kranen, and T. Seidl. Efficient emd-based similarity search in multimedia databases via flexible dimensionality reduction. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pages 199–212. ACM, 2008.

[13] J. Xu, Z. Zhang, A. K. Tung, and G. Yu. Efficient and effective similarity search over probabilistic data based on earth mover's distance. Proceedings of the VLDB Endowment, 3(1-2):758–769, 2010.