ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

Heterogeneous Computing for Real-Time Stereo Matching

A. AL-Marakeby[1], M. Zaki[2]

Systems and Computers Engineering Dept., Faculty of Engineering, Al-Azhar University, Cairo, Egypt[1,2]

Abstract: Stereo matching is used in many computer vision applications such as 3D reconstruction, robot navigation, robotic surgery, 3-D video surveillance, and tracking object in 3D space. Real time stereo matching is difficult due to the heavy computation required for matching algorithms. In this paper a CPU/GPU heterogeneous computing platform is used to accelerate the processing and run the system in real time. The availability of GPUs (Graphics Processing Units) with hundreds of parallel processing cores, increases the speed of matching algorithms. In this work a combination between feature based and area based matching techniques is used. Feature based matching is fast while area based is robust and the used technique takes the advantages of the both. The stereo matching algorithms run over GeForce GPU and a real time processing is achieved with speed up to 15 frames/sec.

Keywords: Parallel Processing, GPU, Stereo matching, Heterogeneous Computing.

I. INTRODUCTION

Stereo vision is the extraction of depth information from 2D scenes. The stereo matching is to determine the pixels of the same object point between left and right images [1]. Stereo matching can be classified into two classes, edge-based and area-based. The main difference between these two categories is the nature of the reconstructed data. Edge-based Stereo produces a sparse disparity map, while the area-based stereo produces a dense disparity map [2]. The main problem in stereo matching is the large time consumed in the calculation of correspondence between points in the right and left images. Many algorithms have been developed to overcome the speed problem especially for real time applications such as robot navigation. The reduction of the search region in the stereo correspondence can increase the performance of the matching process, in the context of the execution time and the accuracy [6]. The addition of geometric constraints and heuristics such as Uniqueness, Projection, Back Projection, and Epipolar Constraints increases the speed and accuracy of matching process [2]. For area based stereo matching, the selection of window size affects both accuracy and speed. Adaptive window size algorithms have been developed which give better performances than fixed size window [7][4][3]. With all these improvements the speed of the stereo matching is still slow, and many applications require faster processing. The spread of GPUs (Graphics Processing Units) which originally designed to accelerate the graphics have attracted researchers with their parallelism power. GPUs consists of tens or hundreds of parallel processor cores with strong floating point capabilities. Developing parallel stereo matching on GPUs, gives very faster execution time than sequential processing algorithms. Jedrzej et.al has used GPU to implement a real-time stereo matching based on an iterative refinement method for adaptive support-weight correspondences [4]. Mikhail et.al has implemented the stereo matching algorithm on CUDA platform available on Nvidia GPUs based on a coarse-to-fine architecture[5]. In this paper a combined technique has been used which use both edge-based and area-based algorithms. An edge-based scanning is applied to detect feature points, then a window based search is used to detect the correspondence between the two images. The system is implemented on NVIDIA GPU using CUDA. This paper is organiezd as follow : section 2: the stereo matching process, section 3: the heterogeneous CPU/GPU architecture , section 4: experiments and results, and section 5: the conclusion.

II. THE STEREO MATCHING PROCESS

The area based matching depends on selecting a window in the reference image and searching for the best match window in the second image. This search depends on some correlation based match functions such as SSD and SAD. If the color or intensity of this window is uniform, the matching is inaccurate. In this application we don't need the dense disparity map, but we search for matching the feature points. Hence we used a combination of area-based and feature-based techniques to overcome the fault matching errors caused by uniform color windows. The proposed algorithm consists of the following stages: feature

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

extraction , point selection, and window matching as shown in fig.1.

Fig.1 Stereo matching stages

The feature extraction depends on edge information based on some masks which detects corners and vertical lines. The corners and vertical lines are useful information to increase the accuracy of the matching process. The following masks are used and the result of applying the feature extraction process is shown in fig.2

$$M1 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad M2 = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \\ 0 & -1 & 1 \end{bmatrix}$$

Fig.2 feature extraction

After the extraction of the features some points are selected as feature points. The selection depends on a simple threshold process for the number of vertical edges and corners in a specified window. The matching process starts by searching for the correspondent window in the second image based on cost function as shown in fig.3. The SAD for RGB colors is used as cost function as illustrated in the following equation and detailed in [1]

$$f(x,y) = \frac{1}{n^2} \sum_{i=-n/2}^{n/2} \sum_{j=-n/2}^{n/2} dist_c(Right(x+i, y+j), Left(x+i+d, y+j)) \quad (1)$$

$$dist_c(c_L, c_R) = (R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2 \quad (2)$$

$$Left(x,y) = (R_{left}(x,y), G_{left}(x,y), B_{left}(x,y))^2 \quad (3)$$

$$Right(x,y) = (R_{right}(x,y), G_{right}(x,y), B_{right}(x,y))^2 \quad (4)$$

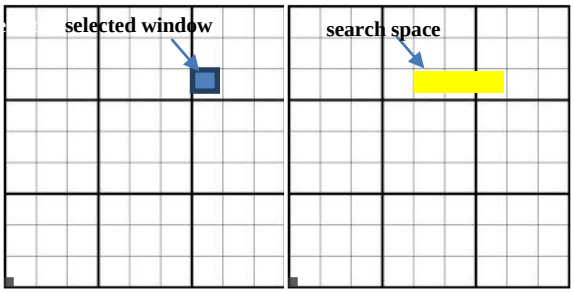

Fig.3 Searching space for window matching

In fig.4 the selected feature points and the matching process is illustrated.

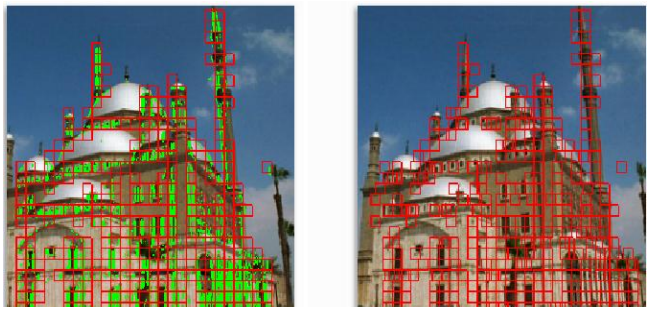

Fig.4 feature points selection and matching process

III. HETEROGENEOUS CPU/GPU PLATFORM

The nature of algorithms parallelism in most of image processing applications, makes the GPU a very effective tool for increasing the speed of processing. The GPU consists of hundreds of cores works in parallel as shown in fig.5 [9].

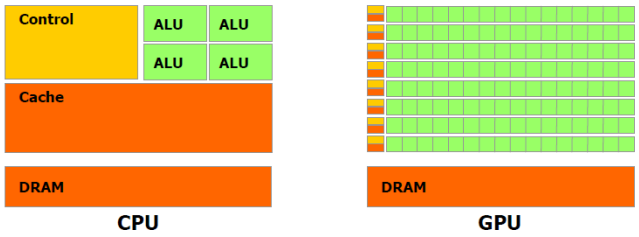

Fig.5 the CPU and GPU architectures

Heterogeneous computing divides the processing load between CPU and GPU utilizing the power of the both. Many parallel programming tools are developed to allow the developers to program on parallel architecture isolating them from many hardware details and memory problems. CUDA is a software and hardware system which treats GPU as a data parallel computing device. Program codes developed on CUDA can be divided into two types in actual execution. One is the host code which runs on CPU, and the other is the device code which runs on GPU as shown in fig.6 [8].

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

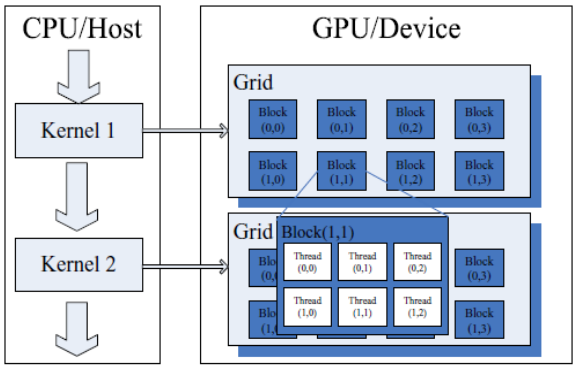

Fig.6 CUDA platform

To implement the proposed stereo matching algorithm on the CUDA platform, the image is divided into small windows.

A 2D grid is used with a size proportional to the image dimensions where

$$Grid_{size} = \left(\frac{Image_{width} + Window_{size} - 1}{Window_{size}}, \frac{Image_{h} + Window_{size} - 1}{Window_{size}}\right)$$

And the grid is divided into 2D blocks with the same size of window size. These blocks run on parallel and the processed window coordinates is given by

$$Window_x = BlockID_x * BlocDim_x + ThreadID_x$$
$$Window_y = BlockID_y * BlocDim_y + ThreadID_y$$

IV. EXPERIMENTS AND RESULTS

The NVIDIA GeForce 610M GPU is used for the implementation of the stereo matching algorithm. This graphics processing unit consists of 48 cores and 1 GB memory. The initialization of the system, and memory transfer is executed on the CPU, while feature extraction and matching process is executed on GPU as described in section 2. Different window sizes and search spaces values have been used to test the speed of the algorithm. Table 1 illustrate the measured speed and the effect of different parameters on this speed.

Table.1 matching speed

Image size	Window size	Search space	Speed frames/sec
280 X 240	8 X 8	30 pixel	15
280 X 240	8 X 8	40 pixel	11
280 X 240	10 X 10	30 pixel	10
280 X 240	10 X 10	40 pixel	8

V. CONCLUSION

The speed of real time stereo matching process is increased and improved using the parallel GPU. The combination of area-based and feature-based stereo techniques increase the accuracy of stereo matching. CUDA platform is an efficient parallel programming tool which isolate the developer from hardware details and facilitate the utilization of parallel capabilities.

REFERENCES

[1] Chang-Il Kim,Soon-Yong Park, "Fast Stereo Matching of Feature Links" , International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission 2011.

[2] Elsayed E. Hemaycd Michacl S. Hrown, Aly A. Farag, W. Brent Scalos, "Cooperative Stereo: Combining Edge- and Area-Based Stereo" Aerospace Conference,. Proceedings. IEEE , 1999

[3] Jargalsaikhan Iveel and Sumam David, SMIEEE "A novel adaptive support window based stereo matching algorithm for 3D reconstruction from 2D images", 11th International Conference on ITS Telecommnincation 2011.

[4] Jedrzej Kowalczuk, , Eric T. Psota, , and Lance C. P´erez, "Real-Time Stereo Matching on CUDA Using an Iterative Refinement Method for Adaptive Support-Weight Correspondences" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 23, NO. 1, JANUARY 2013

[5] Mikhail Sizintsev, Sujit Kuthirummal, Supun Samarasekera, Rakesh Kumar ,"GPU Accelerated Realtime Stereo for Augmented Reality", International Symposium 3D Data Processing, Visualization, and Transmission 2010.

[6] Payman Moallem, Karim Faez, "FAST EDGE-BASED STEREO MATCHING ALGORITHM BASED ON SEARCH SPACE REDUCTION"

[7] Yingyun Yang, Huabing Wang, Bo Liu," A New Stereo Matching Algorithm Based on Adaptive Window" International Conference on Systems and Informatics (ICSAI 2012).

[8] Yu Liu , Longjiang Guo, Jinbao Li, Meirui Ren, and Keqin Li, "Parallel Algorithms for Approximate String Matching with k Mismatches on CUDA , IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum 2012

[9] CUDA programming guide , http://docs.nvidia.com/cuda/cuda-c-programming-guide/

BIOGRAPHY

Ashraf Al-Marakeby has received BSc in Systems and Computer Engineering from Al-Azhar University at 1999 MSc from the same university at 2004. He has received PhD as a channel between Mie University and Al-Azhar University in 2008. He is interested in image processing, computer vision, character recognition and Embedded systems.

M. Zaki is the professor of software engineering, Computer and System Engineering Department, Faculty of Engineering, Al-Azhar University at Cairo. He received his B.Sc. and M.Sc. degrees in electrical engineering from Cairo University in 1968 and 1973 respectively. He received his Ph. D. degrees in computer engineering from Warsaw Technical University, Poland in 1977. His fields of interest include artificial intelligence, soft computing, and distributed systems.