VIDEO SURVEILLANCE USING RASPBERRY PI GPU

WONG YAN YIN

UNIVERSITI TEKNOLOGI MALAYSIA

Replace this page with form PSZ 19:16 (Pind. 1/07), which can be obtained from SPS or your faculty.

*Replace this page with the Cooperation Declaration form, which can be obtained from SPS or your faculty. This page is **OPTIONAL** when your research is done in collaboration with other institutions that requires their consent to publish the finding in this document.]*

VIDEO SURVEILLANCE USING RASPBERRY PI GPU

WONG YAN YIN

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Computer and Microelectronic System)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JUNE 2018

This thesis is dedicated to my lovely and supportive family members, lecturers and friends.

ACKNOWLEDGEMENT

At the first place I would like to express my gratitude to my project supervisor, Prof. Madya Muhammad Mun'im bin Ahmad Zabidi. Thank you on his patience, motivation, guidance and the continuous support given throughout the completion of this project. He had provide countless helpful advice and gave handy guidance which resulting to a better project.

Besides my project supervisor, I would like to say thank you to my family. They have been always there to support me and make me feel better when I was under stress. I could not have done this without their support.

ABSTRACT

Today, surveillance system is being utilized and deployed in many places to provide supervision and bring security to people. The most commonly used technology currently is Closed-Circuit Television (CCTV). However, there are several defects with the technology such as anomalies cannot be identified automatically and expensive. This project proposed to use the GPU in Raspberry Pi for video surveillance task. Raspberry Pi is a powerful single-board computer which features an ARM processor and a VideoCore IV graphics processing unit (GPU). It is sufficiently powerful to work as a video surveillance system and relatively cheap compared to CCTV. Furthermore, GPU is optimized for parallel computing of video data. It can theoretically provides better performance and have higher efficiency in video processing compared to CPU. Hence, the GPU in Raspberry Pi should provides large performance gain by porting the algorithm from CPU-only reference to works on GPU. The objective of this project is to explore on how the GPU can be programmed for the purpose of video surveillance.

ABSTRAK

Dalam zaman ini, sistem pengawasan telah dipasang dan digunakan di banyak tempat untuk mengawas dan menjamin keselamatan orang ramai. Teknologi yang paling biasa digunakan ialah Televisyen Circuit Tertutup (CCTV). Walau bagaimanapun, terdapat beberapa kelemahan dengan teknologi tersebut seperti anomali tidak dapat dikenal pasti secara automatik dan mahal. Projek ini mencadangkan penggunaan GPU dalam Raspberry Pi untuk tugas pengawasan video. Raspberry Pi sebenarnya merupakan sebuah komputer yang kuat, ia mempunyai pemproses ARM dan unit pemprosesan grafik VideoCore IV. Raspberry Pi mempunyai performasi yang cukup untuk berfungsi sebagai sistem pengawasan video dan agak murah berbanding dengan CCTV. Selain itu, GPU adalah sistem yang dioptimumkan untuk pengkomputeran data video selari. Ia secara teorinya boleh memberikan prestasi yang lebih baik dan mempunyai kecekapan yang tinggi dalam pemprosesan video berbanding dengan penggunaan CPU. Oleh itu, GPU dalam Raspberry Pi sepatutnya memberikan keuntungan prestasi yang besar dengan mengalihkan algoritma dari rujukan CPU sahaja kepada kerja-kerja GPU. Objektif projek ini adalah untuk meneroka bagaimana GPU boleh diprogramkan untuk tujuan pengawasan video.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CCTV	-	Closed Circuit Television
CMOS	-	Complementary Metal Oxide Semiconductor
CPU	-	Central Processing Unit
DSP	-	Digital Signal Processing
FFT	-	Fast Fourier Transform
FPS	-	Frame Per Second
GFLOPS	-	Giga Floating Point Operations Per Second
GPU	-	Graphics Processing Unit
LBP	-	Local Binary Pattern
PC	-	Personal Computer
PGC	-	Programmable Graphic Core
PTU	-	Pan and Tilt Unit
QPU	-	Quad Processing Unit
ROI	-	Region Of Interest
SDRAM	-	Synchronous Dynamic Random Access Memory
SIMD	-	Single Instruction Multiple Data
SOC	-	System On Chip
XML	-	Extensible Markup Language

CHAPTER 1

INTRODUCTION

1.1 Problem Background

Surveillance system is a system which monitor and observe. The main purpose of surveillance system is to provide supervision and security to people. In this modern era, surveillance system is very common and it can be found easily in many places such as shopping mall, bank and even in certain housing areas [3].

The most commonly used technology for surveillance system is CCTV (Closed Circuit Television) and the most common type of CCTV is analogue CCTV. Analogue CCTV is a well-stable, established technology which is supported by all major CCTV manufacturers. Despite the popularity, analogue CCTV has a few downsides. First of all, it is dumb. It does not mean literally dumb, in this context it is saying the analogue CCTV is lacking some intelligence [4]. Analogue CCTV can only perform surveillance task solely. It cannot do anything other than passive monitoring and recording the video. Secondly, analogue CCTV requires installation and maintenance. Analogue CCTV has to be installed and it involves a lot of wiring work, not to mention it still has to be maintained from time to time.

Because of all the mentioned disadvantages of CCTV, a new trend of low cost embedded video surveillance system using Raspberry Pi emerged [3, 5, 6, 7]. Raspberry Pi is being used for several reasons. First, it has a small form factor. The size of Raspberry Pi is only pocket size large. This makes it suitable for surveillance purpose because it can be hide effortlessly. Second, it has a relatively powerful hardware. As matter of fact, Raspberry Pi is essentially a mini computer and it can

turned into a smart video surveillance system through programming to perform motion or face detection. Third, Raspberry Pi has a cheap pricing. It is affordable for most people.

1.2 Problem Statement

There are a few problem statement in this project. First problem statement is there is very limited exploration on GPU based video surveillance system [8] especially on Raspberry Pi. As of today, not much research has been done on the Raspberry Pi GPU needless to mention those research specialized in video surveillance system using Raspberry Pi GPU.

The second problem statement is high user demand for high performance and intelligent video surveillance system. In computing world, the expectation level of performance for devices is ever increasing, user is constantly expecting devices to run faster and perform better. This applies to Raspberry Pi as well, even though Raspberry Pi already has a powerful enough hardware but as time passed by the computation power of Raspberry Pi may not meet user expectation anymore.

The third problem statement is no previous work has been done to compare the video surveillance system performance on Raspberry Pi GPU and CPU. Currently, there is no research to study the performance difference in Raspberry Pi GPU and CPU based video surveillance system. This is saddening because it should be very interesting to study on the performance gain of switching from CPU implementation to GPU implementation. From the learning, it can determine whether the switch is worthwhile.

1.3 Objective

The objectives to be achieved in this project are:

1. To implement video surveillance algorithms on Raspberry Pi.
2. To improve the performance of Raspberry Pi video surveillance system in terms of FPS or latency.
3. To benchmark the results between Raspberry Pi CPU and GPU platform.

1.4 Scope of Work

There are many different types of Raspberry Pi in the market. In this project, a non-overclock Raspberry Pi model 2B is used. Raspberry Pi 2B has a ARMv7 four cores processors which is running at 900Mhz and a Broadcom VideoCore IV GPU which consists of QPU, VPU and ISP.

In order for Raspberry Pi to have computer vision capability, it needs to be programmed with video surveillance algorithms. The video surveillance algorithms are limited to Local Binary Patterns (LBP) [9], Frame Differencing and Image Convolution.

A normal USB Logitech webcam instead of the official Pi camera module is used for the video capturing purpose.

1.5 Thesis Outline

This thesis consists of five chapters, they are introduction, literature review, research methodology, results and discussion, and conclusion. Chapter 1 gives the motivation and background of the project, raises the problem statement, finding the objective and set the scope for the project. Chapter 2 contains the study from literature reviews from all related field. Chapter 3 describes the design methodology of the project in details. The results and discussion can be found in chapter 4. Lastly, chapter 5 summarize the overall findings and the achievement of this project.

REFERENCES

1. Cocorullo, G., Corsonello, P., Frustaci, F., Guachi, L. and Perri, S. Embedded surveillance system using background subtraction and Raspberry Pi. *AEIT International Annual Conference (AEIT). IEEE*, 2015: 1–5. doi:10.1109/AEIT.2015.7415219.

2. Guachi, L., Cocorullo, G., Corsonello, P., Frustaci, F. and Perri, S. A novel background subtraction method based on color invariants and grayscale levels. 2014.

3. Jyothi, S. N. and Vardhan, K. V. Design and implementation of real time security surveillance system using IoT. *Proceedings of the International Conference on Communication and Electronics Systems, ICCES 2016*, 2016. doi:10.1109/CESYS.2016.7890003.

4. S, K. H., R, G. S., B, K. S. and Sangmesh, P. Smart Video Surveillance. 2015. 3(1): 109–112.

5. Menezes, V. Surveillance and Monitoring System Using Raspberry Pi and SimpleCV. *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, 2015: 1276–1278. doi:10.1109/ICGCIoT.2015.7380661.

6. Shifani, S. A. Security System Using Raspberry Pi. 2017.

7. Jacob, T. P. Motion Activated Security Camera using Raspberry Pi. 2017: 1598–1601.

8. Mijat, R., Computing, V. and Manager, M. Understanding the Role of Integrated GPUs in Vision Applications. 2015. (May).

9. PietikÃinen, M. Local Binary Patterns. *Scholarpedia*, 2010. 5(3): 9775. ISSN 1941-6016. doi:10.4249/scholarpedia.9775. URL `http://www.scholarpedia.org/article/Local{_}Binary{_}Patterns`.

10. Tripathy, R. and Daschoudhury, R. N. Real-time face detection and tracking using Haar classifier on SoC. *International Journal of Electronics and Computer Science Engineering*, 2014. 3(2): 177.

11. Viola, P. and Jones, M. Robust Real-time Face Detection. *Eighth IEEE International Conference on Computer Vision*, 2001. 20: 7695. ISSN 0920-5691. doi:10.1109/ICCV.2001.937709.

12. Huu-Quoc Nguyen, Ton Thi Kim Loan, Bui Dinh Mao and Eui-Nam Huh. Low cost real-time system monitoring using Raspberry Pi. *2015 Seventh International Conference on Ubiquitous and Future Networks*, 2015: 857–859. doi:10.1109/ICUFN.2015.7182665. URL http://ieeexplore.ieee.org/document/7182665/.

13. Sabarinath, S., Shyam, R., Aneesh, C., Gandhiraj, R. and Soman, K. P. Accelerated FFT Computation for GNU Radio Using GPU of Raspberry Pi. 2015. 32: 657–664. doi:10.1007/978-81-322-2208-8. URL http://link.springer.com/10.1007/978-81-322-2208-8.

14. Janard, K. and Marurngsith, W. Accelerating real-time face detection on a raspberry pi telepresence robot. *5th International Conference on Innovative Computing Technology, INTECH 2015*, 2015. (INTECH): 136–141. doi:10.1109/INTECH.2015.7173482.

15. Allusse, Y., Horain, P., Agarwal, A. and Saipriyadarshan, C. GpuCV: A GPU-Accelerated Framework for Image Processing and Computer Vision. *Advances in Visual Computing SE - Lecture Notes in Computer Science*, 2008. 5359: 430–439. doi:doi:10.1007/978-3-540-89646-3_42. URL citeulike-article-id:7187795{%}5Cnhttp://dx.doi.org/10.1007/978-3-540-89646-3{_}42.

16. Semiconductor, F. Fast Image Processing with i . MX 6 Series. 2013: 1–41.

17. Culjak, I., Abram, D., Pribanic, T., Dzapo, H. and Cifrek, M. A brief introduction to OpenCV. *MIPRO, 2012 Proceedings of the 35th International Convention*, 2012: 1725–1730. doi:978-1-4673-2577-6. URL http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6240859.

18. BroadCom. VideoCore ® IV 3D Architecture Reference Guide. *Architecture Guide*, 2013. URL https://www.broadcom.com/docs/support/videocore/VideoCoreIV-AG100-R.pdf.

19. Mn416. QPULib. URL https://github.com/mn416/QPULib.