

PERFORMANCE ANALYSIS OF REAL TIME IMAGE
PROCESSING FOR LIGHTNING EVENT USING CYTHON AND
PYTHON PROGRAMMING LANGUAGE

MUN SIU HOU

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

**PERFORMANCE ANALYSIS OF REAL TIME IMAGE
PROCESSING FOR LIGHTNING EVENT USING CYTHON
AND PYTHON PROGRAMMING LANGUAGE**

MUN SIU HOU

**This report is submitted in partial fulfilment of the requirements
for the degree of Bachelor of Electronic Engineering with Honours**

**Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka**

2018

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : Performance Analysis of Real Time Image Processing for Lightning Event Using Cython and Python Programming Language.

Sesi Pengajian : 2017/2018

Saya MUN SIU HOU mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):



SULIT*

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)



TERHAD*

(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan.



TIDAK TERHAD



(TANDATANGAN PENULIS)

Disahkan oleh:
Dr. Mohd Ridwan Bin Ahmad
Penyarah Kanan
Fakulti Kejuruteraan Elektronik Dan Kejuruteraan Komputer
Universiti Teknikal Malaysia Melaka (UTeM)
Hang Tuah Jaya,
76100 Durian Tunggal, Melaka

(COP DAN TANDATANGAN PENYELIA)

Alamat Tetap: 14, Jalan Helang
Siput 4, Kepong
Baru, 52100 Kuala
Lumpur

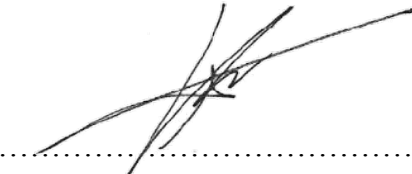
Tarikh : 19 May 2018

Tarikh : 19 May 2018

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this report entitled “Performance Analysis of Real Time Image Processing for Lightning Event Using Cython and Python Programming Language” is the result of my own work except for quotes as cited in the references.

Signature : 

Author : MUN SIU HOU

Date : 24/5/2018

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronic Engineering with Honours.


Dr. Mohd Riduan Bin Ahmad
Pensyarah Kanan
Fakulti Kejuruteraan Elektronik Dan Kejuruteraan Komputer
Universiti Teknikal Malaysia Melaka (UTeM)
Hang Tuah Jaya
76100 Durian Tunggal, Melaka

Signature :

Supervisor Name : DR. MOHD RIDUAN BIN AHMAD

Date : 24/5/2018
.....

DEDICATION

Special thanks to my parents, my love ones and all BBNet members.

ABSTRACT

Due to the uncommon and unpredictable phenomenon, lightning event become one of the challenging research topics for the scientist. The main thing that most capturing their attention is the lightning detection method which can improve the public safety by pinpointing the location of the lightning event and sending out the warning notification. Currently, there are lots of researcher proceed with the study related with the relationship between the pattern of the lightning and the electromagnetic signal received. During the research process, they would like to capture a clear lightning image from the cloud to proceed with the study of characteristic of the pattern of the lightning and electromagnetic field. The most common method for the researcher to detect and capture the lightning image is using the high-speed camera with a flash sensor. But this method is extremely expensive and has its own limitation on mobility part. Therefore, this project is to develop and evaluate the suitable algorithm and programming language onto Raspberry Pi platform and act as an upgraded choice to replace high speed camera with affordable cost, high mobility but less accuracy compares to high speed camera.

ABSTRAK

Kilat adalah satu fenomena yang luar biasa. Oleh itu, peristiwa kilat sering menjadi satu topik penyelidikan yang mencabar dan menarik bagi saintis. Perkara utama yang paling menarik perhatian mereka adalah kaedah pengesanan kilat yang boleh meningkatkan keselamatan awam dengan menentukan lokasi peristiwa kilat dan menghantar pemberitahuan amaran. Pada masa kini, terdapat ramai penyelidik meneruskan kajian yang berkaitan dengan hubungan antara pola petir dan isyarat elektromagnet yang diterima. Semasa proses penyelidikan, mereka ingin menangkap imej kilat yang lebih jelas untuk meneruskan kajian tentang ciri-ciri pola petir dan elektromagnetik. Cara yang sering digunakan oleh penyelidik untuk mengesan dan menangkap imej kilat adalah menggunakan kamera berkelajuan tinggi dengan penerima kilat. Tetapi cara ini adalah cara yang memerlukan kos yang tinggi. Oleh itu, projek ini bertujuan untuk membangun dan menilai algoritma dan bahasa pengaturcaraan yang sesuai bagi platform Raspberry Pi dan bertindak sebagai pilihan dinaik taraf untuk menggantikan kamera berkelajuan tinggi dengan kos yang berpatutan, mobiliti tinggi tetapi dengan ketepatan yang berkurang berbanding dengan kamera berkelajuan tinggi.

ACKNOWLEDGEMENTS

Firstly, I would like to express my feeling to my supervisor Dr. Mohd Riduan Bin Ahmad who always support and guild me with his patience and motivative way. With his guidance, I can easily understand and learn the knowledge regarding to real-time image processing topic. I am truly grateful that my supervisor can give me some helping hand on my project as well as with the thesis.

Besides, I also would like to say thank you to my parents who always give me positive thinking and moral support along the way of completing this final year project. Also, I need to specially shout out to all the BBNets members who had been giving me extra knowledge and ideas. Lastly, thank you to all the friends that giving me positive support.

TABLE OF CONTENTS

Declaration	
Approval	i
Dedication	i
Abstract	i
Abstrak	ii
Acknowledgements	iii
Table of Contents	iv
List of Figures	x
List of Tables	xiv
List of Symbols and Abbreviations	xv
List of Appendices	xvii
CHAPTER 1 INTRODUCTION	18
1.1 PROJECT BACKGROUND	19
1.2 PROBLEM STATEMENT	21

1.3	OBJECTIVES	21
1.4	SCOPE OF WORK	22
1.5	PROJECT SIGNIFICANCE	22
1.6	EXPECTED OUTCOME	23
1.7	SUMMARY	23
	CHAPTER 2 LITERATURE REVIEW	24
2.1	BACKGROUND RESEARCH	25
2.2	THUNDERCLOUD AND LIGHTNING	25
2.2.1	THE STEPPED LEADER	27
2.2.2	THE RETURN STROKE	28
2.2.3	THE DART LEADER	28
2.3	TYPES OF LIGHTNING DISCHARGES	29
2.4	LIGHTNING CAMERAS	29
2.5	RELATED PROJECT	31
2.6	IMAGE PROCESSING TECHNIQUES	32
2.6.1	THRESHOLD METHOD	34
2.6.1.1	OTSU'S METHOD	36
2.6.1.2	BINARY THRESHOLD METHOD	37
2.6.2	EDGE BASED METHOD	38
2.6.2.1	CANNY EDGE METHOD	39

2.6.3	REGION BASED METHOD	41
2.6.3.1	REGION GROWING METHOD	42
2.7	NOISE	44
2.7.1	INTERNAL CAMERA SETTING ERROR	44
2.7.2	TECHNOLOGY ERROR	45
2.7.3	IMAGE PROCESSING ERROR	46
2.8	PROCESSING DELAY	47
2.8.1	SHUTTER SPEED RATE	48
2.8.2	IMAGE RESOLUTION	49
2.8.3	COLOR IMAGE PROCESSING	50
2.9	RAPSBERRY PI PROGRAMMING LANGUAGE	51
2.9.1	COMPARISON BETWEEN PYTHON AND CYTHON	51
2.9.1.1	PROGRAMMING LIBRARY	52
2.9.1.2	OPENCV LIBRARY	53
2.10	SUMMARY	53
	CHAPTER 3 METHODOLOGY	54
3.1	OVERALL PROJECT	55
3.1.1	RESEARCH SECTION	55
3.1.1.1	RELATED INFORMATION	56
3.1.1.2	SOFTWARE REQUIREMENT	56

3.1.1.3	HARDWARE REQUIREMENT	57
3.1.2	IMPLEMENTATION	58
3.1.2.1	INSTALLATION	58
3.1.2.2	HARDWARE CONNECTION	59
3.1.2.3	SOFTWARE CONNECTION	62
3.1.2.4	IMPLEMENTATION TECHNIQUE	63
3.1.3	TROUBLESHOOT	65
3.1.3.1	FAKE ENVIRONMENT TESTING	65
3.1.3.2	REAL ENVIRONMENT TESTING	66
3.1.4	ANALYSIS	68
3.1.4.1	ANALYSIS FRAMES PER SECOND FOR LIGHTNING DETECTION SYSTEM	68
3.1.4.2	ANALYSIS PROCESSING DELAY BASED ON IMAGE PROCESSING METHOD	69
3.2	PROJECT SCHEDULE	70
3.2.1	GANTT CHART	72
3.3	SUMMARY	74
CHAPTER 4 RESULTS AND ANALYSIS		75
4.1	PRE-IMAGE PROCESSING	76
4.1.1	SAVE FILE LOCATION	77
4.1.2	INTERNAL CAMERA SETTING	79

4.1.3	LIGHTNING DETECTION METHOD	79
4.1.3.1	CONVERTING TO GRAYSCALE IMAGE	81
4.1.3.2	FINDING THRESHOLD VALUE	82
4.2	IMAGE PROCESSING	85
4.2.1	CANNY EDGE DETECTION	85
4.2.2	OTSU'S METHOD	88
4.2.3	REGION GROWING METHOD	91
4.3	ANALYSIS AND DISCUSSION	93
4.3.1	ANALYSIS FRAMES PER SECOND BASED ON PROGRAMMING LANGUAGE	94
4.3.2	ANALYSIS FRAMES PER SECOND BASED ON PROGRAMMING LIBRARY USING PYTHON PROGRAMMING LANGUAGE	95
4.3.3	ANALYSIS PROCESSING DELAY BASED ON IMAGE PROCESSING METHOD WITH HIGH RESOLUTION IMAGE USING PYTHON PROGRAMMING LANGUAGE	96
4.3.4	ANALYSIS PROCESSING DELAY BASED ON IMAGE PROCESSING METHOD WITH HIGH RESOLUTION IMAGE USING CYTHON PROGRAMMING LANGUAGE	99
4.3.5	ANALYSIS PROCESSING DELAY BASED ON IMAGE PROCESSING METHOD WITH LOW RESOLUTION IMAGE USING PYTHON PROGRAMMING LANGUAGE	101
4.3.6	ANALYSIS PROCESSING DELAY BASED ON IMAGE PROCESSING METHOD WITH LOW RESOLUTION IMAGE USING CYTHON PROGRAMMING LANGUAGE	103

4.3.7	ANALYSIS THE RELATIONSHIP OF ELECTROMAGNETIC FIELD WITH PATTERN OF LIGHTNING EVENT	105
4.4	SUMMARY	108
	CHAPTER 5 CONCLUSION AND FUTURE WORKS	109
5.1	DISCUSSION	110
5.2	LIMITATION AND WEAKNESS OF THE PROJECT	111
5.3	RECOMMENDATION FOR FUTURE IMPROVEMENT	112
5.4	CONCLUSION	113
	REFERENCES	114
	List of Publications and Papers Presented	118
	APPENDICES	119

LIST OF FIGURES

Figure 1.1: Image of cloud-to-ground lightning ^[1]	19
Figure 1.2: View of Raspberry Pi 3 Model B ^[3]	20
Figure 2.1: Thundercloud Structure ^[4]	25
Figure 2.2: Probable distribution of thundercloud charges	26
Figure 2.3: Leader stroke ^[7]	27
Figure 2.4: Comparison between original image and Otsu method image ^[19]	37
Figure 2.5: The result for histogram value and optimizing value ^[20]	38
Figure 2.6: Comparison between original image and canny edge method image ^[24]	40
Figure 2.7: Example of region growing ^[27]	43
Figure 2.8: ISO scale for Camera Setting ^[29]	45
Figure 2.9: Image Noise ^[31]	45
Figure 2.10: Noise Model ^[33]	46
Figure 2.11: Display in Histogram ^[33]	47
Figure 2.12: Shutter Speed Rate ^[35]	48
Figure 2.13: 8-bit pixel scale ^[41]	50
Figure 2.14: Performance Comparison Between Programming Languages ^[44]	52

Figure 3.1: Flow diagram for project methodology	55
Figure 3.2: Accessories to prevent overheat	60
Figure 3.3: Raspberry Pi 3 Model B GPIO Pin Chart	60
Figure 3.4: Physical connection of Camera Module	61
Figure 3.5: Overall implementation technique flow chart	63
Figure 3.6: Black Box	65
Figure 3.7: Bug Racket Zapper	66
Figure 3.8: CAPPI Radar image from webpage of MMD	67
Figure 4.1: Flowchart for lightning detection system using OpenCV library	76
Figure 4.2: Flowchart for lightning detection system using PiCamera library	77
Figure 4.3: Flow chart for save file location function	78
Figure 4.4: Save file location directory	78
Figure 4.5: Flowchart for internal camera setting function	79
Figure 4.6: Flowchart for lightning detection method function	80
Figure 4.7: Image of Lightning Event Captured	80
Figure 4.8: Sample lightning image	82
Figure 4.9: Sample image of blackout environment	83
Figure 4.10: Histogram for Sample image of blackout environment	83
Figure 4.11: Histogram for Sample Color Lightning Image	84
Figure 4.12: Histogram for Sample Grayscale Lightning Image	84
Figure 4.13: Canny Edge Detection for High Resolution Lightning Image	86
Figure 4.14: Canny Edge Detection for Low Resolution Lightning Image	86
Figure 4.15: After Dilation Process	87

Figure 4.16: Histogram for Canny Edge Image	87
Figure 4.17: Otsu Method for High Resolution Lightning Image	88
Figure 4.18: Otsu Method for Low Resolution Lightning Image	89
Figure 4.19: Histogram for Otsu Method image	89
Figure 4.20: Otsu Method image with threshold to zero control	90
Figure 4.21: Histogram for Otsu image with Threshold to Zero control	91
Figure 4.22: Region Growing Method for High Resolution Image	92
Figure 4.23: Region Growing Method for Low Resolution Image	92
Figure 4.24: Histogram for Region Growing Image	93
Figure 4.25: Graph of the Frames per Seconds based on Programming Language	94
Figure 4.26: Graph of the Frames per Seconds based on Programming Library	96
Figure 4.27: Graph of Processing Delay based on Canny Edge method and Otsu's method using Python Programming Language (High Resolution Image)	97
Figure 4.28: Graph of Processing Delay based on Region Growing Method using Python programming language (High Resolution Image)	98
Figure 4.29: Graph of Processing Delay based on Canny Edge method and Otsu's method using Cython Programming Language (High Resolution Image)	99
Figure 4.30: Graph of Processing Delay based on Region Growing Method using Cython programming language (High Resolution Image)	100
Figure 4.31: Graph of Processing Delay based on Canny Edge method and Otsu's method using Python Programming Language (Low Resolution Image)	101
Figure 4.32: Graph of Processing Delay based on Region Growing Method using Python programming language (Low Resolution Image)	102
Figure 4.33: Graph of Processing Delay based on Canny Edge method and Otsu's method using Cython Programming Language (Low Resolution Image)	103
Figure 4.34: Graph of Processing Delay based on Region Growing Method using Cython programming language (Low Resolution Image)	104

Figure 4.35: Overview of Electrical Field that generated from PicoScope	105
Figure 4.36: The First Return Stroke of the Lightning event	106
Figure 4.37: Image of first return stroke of the lightning event	107

LIST OF TABLES

Table 2.1: Various Display Resolutions with Application ^[39]	49
Table 3.1: Software Requirement	57
Table 3.2: Hardware Requirement	57
Table 3.3: Installation Guidance for Raspbian Jessie	58
Table 3.4: Setup for MobaXterm software	62
Table 3.5: Timetable for FYP 1	70
Table 3.6: Timetable for FYP 2	71
Table 3.7: Gantt Chart for FYP 1	72
Table 3.8: Gantt Chart for FYP 2	73

LIST OF SYMBOLS AND ABBREVIATIONS

DSLR	:	Digital Single-lens Reflex
CPU	:	Central Processing Unit
ISO	:	International Organization for Standardization
FPS	:	Frame Per Second
GPS	:	Global Positioning System
OpenCV	:	Open Source Computer Vision
BOT	:	Robot
BLOB	:	Binary Large Object
NoIR	:	No Infrared
RGB	:	Red, Green and Blue
DSLR	:	Digital Single-Lens Reflex
CMY	:	Cyan, Magenta and Yellow
YIQ	:	Luminance, Inphase and Quadrature
SD	:	Secure Digital
DC	:	Direct Current
LAN	:	Local Area Network

GPIO	:	General-Purpose Input/Output
HDMI	:	High-Definition Multimedia Interface
IP	:	Internet Protocol
MMD	:	Malaysian Meteorological Department
CAPPI	:	Constant Altitude Plan Position Indicator

LIST OF APPENDICES

Appendix A: INSTALLATION SOFTWARE	121
Appendix B: HARDWARE SPECIFICATION	123
Appendix C: INSTALLATION OF OPENCV	127

CHAPTER 1

INTRODUCTION

This chapter explained about the introduction of this project following by the background and the problem statement. For the project background, an explanation about the lightning has been described. Project objectives, scopes and expected results from this project also discussed in this chapter. All the details for each section of the project have been discussed in this chapter.