UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI KEJURUTERAAN ELEKTRONIK DAN KEJURUTERAAN KOMPUTER
BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : IMPLEMENTATION OF FOUR STEP SEARCH ALGORITHM FOR MOTION ESTIMATION USING MATLAB
Sesi Pengajian : 2008/2009

Saya ELYA IRWANA BT ZULLKIFLY

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 ( √ ) :
   (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

Disahkan oleh:

(TANDATANGAN PENULIS)

Alamat Tetap: D113 Panggampuri Subang Impian,
Jln Zulha 15/177, SS U5
40106 Shah Alam
Selangor Derul Ehsan

Tarikh: 28 APRIL 2009

(TANDATANGAN PENGETUA)

Universiti Teknikal Malaysia Melaka
Karang Serunjung No 1752
Pejabat Pos Darian Tunggal
78109 Darian Tunggal, Melaka

Tarikh: 29/4/2009
IMPLEMENTATION OF FOUR STEP SEARCH ALGORITHM FOR MOTION ESTIMATION USING MATLAB

ELYA IRWANA BT ZULLKIFLY

This report is submitted in partial fulfillment of requirements for the award of Bachelor Degree of Electronic Engineering (Telecommunication Electronics) With Honors

Faculty of Electronic Engineering and Computer Engineering
Universiti Teknikal Malaysia Melaka

APRIL 2009
“I hereby declare that this report is the result of my own work except for quotes as cited in the references.”

Signature: ..............................
Name: Elya Irwana bt Zulkifly
Date: ..............................
“I hereby declare that I have read this report and in my opinion this report is sufficient in terms of the scope and quality for award of Bachelor of Electronic Engineering (Telecommunication Electronics) With Honours.”

Signature: ........................................
Name: Redzuan b Abdul Manap
Date: .................................
Dedicated to my dearest family and friends.
ACKNOWLEDGEMENTS

First and foremost, I would like to praise God for His blessing. He gave me physical and mental strength to complete my final year project.

I would like to express my gratitude and thanks to my supervisor, Mr. Redzuan b Abdul Manap for his help, encouragement and sharing his knowledge throughout this project. His valuable advice and guidance are truly appreciated. I am honored to have such a knowledgeable and experienced supervisor.

I wish to express my sincere gratitude and appreciation to Raudzatul Adawiah, Irwan Zilah, Siti Hajar and Nur Hashela for their co-operation, support and help that they gave to me in order to complete this project.

Finally, my deepest gratitude goes to my beloved family members for their constant support, encouragement and prayer.

Thank you all. May The Almighty repay you in blissfully ways.
This project presents an overview of Block Matching for Motion Estimation in video coding. The best solution of block matching implementation is Full Search (FS) Algorithm. Due to high computational workload required in this algorithm, various types of block matching algorithms have been proposed and developed to overcome this problem. The Four Step Search (4SS) Algorithm has been proposed for the study in this project. This algorithm features are based on the centre-biased motion vector distribution characteristic and contents halfway-stop techniques to reduce computation process. This proposed algorithm contains 1 to 4 searching step and 17 to 27 checking point. From the simulation result using Matlab, most of performance in term of PSNR of this algorithm better than others; Cross Search (CS) Algorithm, New Three Step Search (NTSS) Algorithm, Diamond Search (DS) Algorithm and Cross Diamond Search (CDS) Algorithm, and has close to FS Algorithm.
## CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>TITLE</th>
<th>PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TITLE OF PROJECT</td>
<td>i</td>
</tr>
<tr>
<td></td>
<td>DECLARATION</td>
<td>ii</td>
</tr>
<tr>
<td></td>
<td>DEDICATION</td>
<td>iv</td>
</tr>
<tr>
<td></td>
<td>ACKNOWLEDGMENT</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td>ABSTRACT</td>
<td>vi</td>
</tr>
<tr>
<td></td>
<td>ABSTRAK</td>
<td>vii</td>
</tr>
<tr>
<td></td>
<td>CONTENT</td>
<td>viii</td>
</tr>
<tr>
<td></td>
<td>LIST OF TABLE</td>
<td>xi</td>
</tr>
<tr>
<td></td>
<td>LIST OF FIGURE</td>
<td>xii</td>
</tr>
<tr>
<td></td>
<td>LIST OF ACRONYMS</td>
<td>xv</td>
</tr>
<tr>
<td></td>
<td>LIST OF APPENDIX</td>
<td>xvii</td>
</tr>
</tbody>
</table>

### I INTRODUCTION

1.1 Project Introduction 1
1.2 Objective of Project 2
1.3 Problem Statement 2
1.4 Scope of Project 2
1.5 Thesis Structure 3
## LITERATURE REVIEW

2.1 Overview of Video Compression and Coding 4
2.2 Motion Estimation 7
2.3 Block Algorithm 8
  2.3.1 Full Search Algorithm 12
  2.3.2 Cross Search Algorithm 12
  2.3.3 New Three Step Search Algorithm 15
  2.3.4 Diamond Search Algorithm 17
  2.3.5 Cross Diamond Search Algorithm 20
2.4 Matlab 23
2.5 Video Feature 23

## PROJECT METHODOLOGY

3.1 Introduction 25
3.2 Flow chart 26
3.3 Methodologies 27
  3.3.1 Literature Review 27
  3.3.2 Development and Implementation in Matlab 27
  3.3.3 Performance Comparison 28
  3.3.4 Seminar and Presentation 28
  3.3.5 Thesis Writing 28

## FOUR STEP SEARCH ALGORITHM

4.1 Introduction 29
4.2 The Algorithm Steps 29
4.3 Previous Simulation Result 34
4.4 Advantage and Disadvantage 36
VRESULT

5.1 Simulation Result 37
   5.1.1 First Stage 37
   5.1.2 Second Stage 46
5.2 Algorithm Predicted Frame 56

VICONCLUSION 57

REFERENCES 58

APPENDIX 59
LIST OF TABLE

<table>
<thead>
<tr>
<th>TABLE</th>
<th>TITLE</th>
<th>PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Type of video and its specification</td>
<td>23</td>
</tr>
<tr>
<td>2.2</td>
<td>Type of video and its categories</td>
<td>24</td>
</tr>
<tr>
<td>4.1</td>
<td>Average MSE of the first 90 frames</td>
<td>35</td>
</tr>
<tr>
<td>4.2</td>
<td>Average search points per motion vector estimation for the first 90 frames</td>
<td>35</td>
</tr>
<tr>
<td>5.1</td>
<td>Average PSNR (dB) for the chosen BMAs</td>
<td>37</td>
</tr>
<tr>
<td>5.2</td>
<td>Average search points for the chosen BMAs.</td>
<td>38</td>
</tr>
<tr>
<td>5.3</td>
<td>Speedup ratio for the chosen BMAs.</td>
<td>38</td>
</tr>
<tr>
<td>5.4</td>
<td>Simulation time (second) for the chosen BMAs.</td>
<td>38</td>
</tr>
<tr>
<td>5.5</td>
<td>Average PSNR (dB) for the first 30 frames simulation</td>
<td>46</td>
</tr>
<tr>
<td>5.6</td>
<td>Average search points for the first 30 frames simulation</td>
<td>47</td>
</tr>
<tr>
<td>5.7</td>
<td>Speedup ratio for the first 30 frames simulation</td>
<td>47</td>
</tr>
<tr>
<td>5.8</td>
<td>Simulation time (second) for the first 30 frames simulation</td>
<td>47</td>
</tr>
</tbody>
</table>
## LIST OF FIGURE

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>TITLE</th>
<th>PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>An example of MPEG-1 GOP</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>MPEG-1 coded video structure</td>
<td>6</td>
</tr>
<tr>
<td>2.3</td>
<td>Encoder video compression</td>
<td>7</td>
</tr>
<tr>
<td>2.4</td>
<td>Motion vector in motion estimation</td>
<td>8</td>
</tr>
<tr>
<td>2.5</td>
<td>The current and previous frames in a search window</td>
<td>9</td>
</tr>
<tr>
<td>2.6</td>
<td>Example frame size 176 x 144 divided into 16 x 16 macroblock</td>
<td>10</td>
</tr>
<tr>
<td>2.7</td>
<td>Block Matching a macro block of side 16 pixels and a search parameter w of size 7 pixels.</td>
<td>11</td>
</tr>
<tr>
<td>2.8</td>
<td>The CS algorithm block diagram</td>
<td>13</td>
</tr>
<tr>
<td>2.9</td>
<td>An example of the CSA search for w=8pixelsframe</td>
<td>14</td>
</tr>
<tr>
<td>2.10</td>
<td>Block diagram and example of NTSS</td>
<td>16</td>
</tr>
<tr>
<td>2.11</td>
<td>DS search pattern</td>
<td>17</td>
</tr>
<tr>
<td>2.12</td>
<td>DS type search</td>
<td>18</td>
</tr>
<tr>
<td>2.13</td>
<td>The block diagram of the DS algorithm</td>
<td>19</td>
</tr>
<tr>
<td>2.14</td>
<td>Flowchart of the CDS algorithm</td>
<td>21</td>
</tr>
<tr>
<td>2.15</td>
<td>CDS algorithm examples</td>
<td>22</td>
</tr>
<tr>
<td>3.1</td>
<td>Project Methodology</td>
<td>26</td>
</tr>
<tr>
<td>4.1</td>
<td>4SS algorithm search path</td>
<td>30</td>
</tr>
<tr>
<td>4.2</td>
<td>4SS searching points</td>
<td>32</td>
</tr>
<tr>
<td>4.3</td>
<td>Two different search paths of 4SS</td>
<td>32</td>
</tr>
</tbody>
</table>
4.4 The 4SS algorithm block diagram

5.1 Average PSNR (dB) for the chosen BMAs in Akiyo.avi (2 frames).
5.2 Average search points for the chosen BMAs in Akiyo.avi (2 frames).
5.3 Average PSNR (dB) for the chosen BMAs in Claire.avi (2 frames).
5.4 Average search points for the chosen BMAs in Claire.avi (2 frames).
5.5 Average PSNR (dB) for the chosen BMAs in Coastguard.avi (2 frames).
5.6 Average search points for the chosen BMAs in Coastguard.avi (2 frames)
5.7 Average PSNR (dB) for the chosen BMAs in Foreman.avi (2 frames).
5.8 Average search points for the chosen BMAs in Foreman.avi (2 frames).
5.9 Average PSNR (dB) for the chosen BMAs in News.avi (2 frames).
5.10 Average search points for the chosen BMAs in News.avi (2 frames).
5.11 Average PSNR (dB) for the chosen BMAs in Salesman.avi (2 frames).
5.12 Average search points for the chosen BMAs in Salesman.avi (2 frames).
5.13 Average PSNR (dB) for the chosen BMAs in Tennis.avi (2 frames).
5.14 Average search points for the chosen BMAs in Tennis.avi (2 frames).
| 5.15 | Average PSNR (dB) for the chosen BMAs in Akiyo.avi (30 frames). |
| 5.16 | Average search points for the chosen BMAs in Akiyo.avi (30 frames). |
| 5.17 | Average PSNR (dB) for the chosen BMAs in Claire.avi (30 frames). |
| 5.18 | Average search points for the chosen BMAs in Claire.avi (30 frames). |
| 5.19 | Average PSNR (dB) for the chosen BMAs in Coastguard.avi (30 frames). |
| 5.20 | Average search points for the chosen BMAs in Coastguard.avi (30 frames). |
| 5.21 | Average PSNR (dB) for the chosen BMAs in Foreman.avi (30 frames). |
| 5.22 | Average search points for the chosen BMAs in Foreman.avi (30 frames). |
| 5.23 | Average PSNR (dB) for the chosen BMAs in News.avi (30 frames). |
| 5.24 | Average search points for the chosen BMAs in News.avi (30 frames). |
| 5.25 | Average PSNR (dB) for the chosen BMAs in Salesman.avi (30 frames). |
| 5.26 | Average search points for the chosen BMAs in Salesman.avi (30 frames). |
| 5.27 | Average PSNR (dB) for the chosen BMAs in Tennis.avi (30 frames). |
| 5.28 | Average search points for the chosen BMAs in Tennis.avi (30 frames). |

The 30<sup>th</sup> predicted frame for (a) FS, (b) 4SS, (c) NTSS, (d) DS, (e) CS and (f) CDS algorithm from “Tennis” sequence.
LIST OF ACRONYMS

4SS – Four Step Search
BDM – Block Distortion Measure
BMA – Block Matching Algorithm
CCB – Cross Centre Biased
CCITT – International Telegraph & Telephone Consultative Committee
CDS – Cross Diamond Search
CS – Cross Search
DCT – Discrete Cosine Transform
DS – Diamond Search
FS – Full Search
GOP – Group Of Picture
IDCT – Inverse Discrete Cosine Transform
JPEG – Joint Photographic Experts Group
LDSP – Large Diamond Search Pattern
LSI – Large Scale Integration
MAC – Media Access Control
MAD – Mean Absolute Difference
MAE – Mean Absolute Error
MBD – Minimum Block Distortion
ME – Motion Estimation
MPEG – Moving Picture Expert Group
MSE – Mean Square Error
MV – Motion Vector
NTSS – New Three Step Search
PC – Personal Computer
PSNR – Peak Signal To Noise Ratio
SDSP – Small Diamond Search Pattern
VLC – Video LAN Client
<table>
<thead>
<tr>
<th>APPENDIX</th>
<th>TITLE</th>
<th>PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Presentation Slide</td>
<td>59</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

1.1 Project Introduction

In order to achieve high compression ratio in video coding, a technique known as Block Matching Motion Estimation has been widely adopted in various coding standards such as CCITT H.261, MPEG-1, MPEG-2 and many more. This technique is implemented conventionally by exhaustively testing all the candidate blocks within the search window. This type of implementation, called Full Search (FS) Algorithm, gives the optimum solution. However, substantial amount of computational workload is required when using this algorithm. To overcome this drawback, many fast Block Matching Algorithms (BMA’s) have been proposed and developed. Different search patterns and strategies are exploited by these algorithms in order to find the optimum motion vector with minimal number of required search point. One of these fast BMA’s, which is proposed to be implemented in this project, is called Four Step Search (4SS) Algorithm.
1.2 Objective of Project

The objective of this project is to implement the 4SS algorithm in MATLAB and to compare its performance to FS Algorithm, CS Algorithm, NTSS Algorithm, DS Algorithm and CDS Algorithm.

1.3 Problem Statement

FS Algorithm is implemented typically by comprehensively testing all the candidate blocks within the search window in order to find the block with minimum distortion. But as large the search window becomes, the more computation it requires. Due to substantial amount of computational workload required during the execution of FS algorithm, one of fast BMAs is proposed to be implemented which is 4SS Algorithm.

1.4 Scope of Project

This project will focus on 3 main areas; the first been literature review on video coding, BMAs and 4SS. Next is the development and implementation of 4SS algorithm using MATLAB platform. Last but not least is the performance analysis of 4SS to FS Algorithm, CS Algorithm, NTSS Algorithm, DS Algorithm and CDS Algorithm.
1.5 Thesis Structure

The thesis structure of the Implementation of Four Step Search (4SS) Algorithm for Motion Estimation Using MATLAB is as follows.

Chapter I The introduction and objectives of this project, method used to complete this project and report structure are discussed.

Chapter II The literature review of this project is described here. This chapter contains research and information on several important concepts, types and techniques used in video coding, ME, BMA and Matlab.

Chapter III The method and steps used to complete this project are explained in this chapter. The flow of implementation, performance analysis, comparison and discussion of the 4SS Algorithm are discussed.

Chapter IV The details of 4SS Algorithm including the algorithm step, flow chart, previous simulation result, advantage and disadvantage of the algorithm are discussed.

Chapter V The analysis and result argumentation from this project which based with graph, figure and table are included in this chapter.

Chapter VI In this chapter, an overview of this project and the conclusion can be composed, including the recommendation for future improvement of this project are outline.
LITERATURE REVIEW

2.1 Overview of Video Compression and Coding Technique

Digital video compression techniques have played an important role in the world of telecommunication and multimedia systems where bandwidth is still a valuable commodity. Hence, video coding techniques are of prime importance for reducing the amount of information needed for a picture sequence without losing much of its quality, judged by the human viewers. Modern compression techniques involve very complex electronic circuits and the cost of these can only be kept to an acceptable level by high volume production of Large Scale Integration (LSI) chips. Standardisation of the video compression techniques is therefore essential. H.261, H.263, Moving Picture Expert Group (MPEG) 1, 2 and 4 are some of standard video codecs.

These days, MPEG-1 decoders/players are becoming common place for multimedia on computers. MPEG-1 decoder plug-in hardware boards (e.g. MPEG magic cards) have been around for a few years, and now software MPEG-1 decoders are available with the release of new operating systems or multimedia extensions for PC and MAC platforms. Since in all standard video codecs the decoders only have to comply
with proper syntax, software-based coding has added extra flexibility that might even improve the performance of MPEG-1 in the future [1].

In MPEG-1, a preprocessing step is required to reorder the input pictures for coding. This is called picture reordering. Because of the conflicting requirements of random access and highly efficient coding, the MPEG suggested that not all pictures of a video sequence should be coded in the same way. They identified four types of picture in a video sequence [1]. The first type is called I-pictures, which are coded without reference to the previous picture. They provide access points to the coded sequence for decoding. These pictures are intraframe coded as for Joint Photographic Experts Group (JPEG), with a moderate compression. The second type is the P-pictures, which are predicatively coded with reference to the previous I or P-coded pictures. They themselves are used as a reference (anchor) for coding of the future pictures. The third type is B-pictures, or bidirectional coded pictures, which may use past, future or combinations of both pictures in their predictions. This increases the motion compensation efficiency, since occluded parts of moving objects may be better compensated for from the future frame. B-pictures are never used for predictions.

Due to the existence of several picture types, a group of pictures (GOP) is the highest level of the hierarchy. A GOP is a series of one or more pictures to assist random access into the picture sequence. The first coded picture in the group is an I-picture. It is followed by an arrangement for P and B-pictures, as shown in Figure 2.1.

![Figure 2.1 An example of MPEG-1 GOP [1].](image)
Then each picture is divided into a group of macroblocks, called slices. The reason for defining a slice is to namely resetting the variable length code to prevent channel error propagation into the picture. Slices can have different sizes within a picture, and the division in one picture need not be the same as the division in any other picture. The slices can begin and end at any macroblock in a picture, but with some constraints. The first slice must begin at the top left of the picture (the first macroblock) and the end of the last slice must be the bottom right macroblock (the last macroblock) of the picture. Thus, slices are divided into macroblocks of $16 \times 16$ pixels. Macroblocks in turn are divided into blocks, for coding.

Finally, the smallest part of the picture structure is the block of $8 \times 8$ pixels, for both luminance and chrominance components. Discrete Cosine Transform (DCT) coding is applied at this block level. Figure 2.2 illustrates the whole structure of partitioning a video sequence, from its GOP level at the top to the smallest unit of block at the bottom.

Figure 2.2 MPEG-1 coded video structure [1].