IMPLEMENTATION OF NEW THREE STEP SEARCH ALGORITHM FOR MOTION ESTIMATION USING MATLAB

KONG KHEE KIEN

This report is submitted in partial fulfillment of the requirements for award of Bachelor of Electronic Engineering (Computer Engineering) With Honours

Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka

April 2010

UNIVERSTI TEKNIKAL MALAYSIA MELAKA
FAKULTI KEJURUTERAAN ELEKTRONIK DAN KEJURUTERAAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek	:	Implementation of New Three Step Search Algorithm for Motion Estimation Using MATLAB
Sesi Pengajian	:	2009/2010

Saya KONG KHEE KIEN 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**	

Disahkan oleh:

(TANDATANGAN PENULIS) (COP DAN TANDATANGAN PENYELIA)

Alamat Tetap: 22 Kampung Bemban
31000 Batu Gjah
Perak

Tarikh: .... Tarikh: ....

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

Signature	:	
Author	:	KONG KHEE KIEN
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 the award of bachelor of Electronic Engineering (Computer Engineering) With Honours."

Signature	: ....
Supervisor"s Name	: ....
Date	: ....

This work is dedicated to my family, lecturer and also to all my friends.

ACKNOWLEDGEMENT

.

I would like to express gratitude and thanks to my supervisor, Mr. Redzuan Bin Abdul Manap for his support and unfailing patience throughout the duration of the project. His encouragement and guidance are truly appreciated. Otherwise, this project would not been possible. I have learnt a lot under his guidance. In addition to that, I also would like to thanks my friends: Siow Chan Hoel and Wong Cheong Lun. They are always help me when I face problems in this project. I am also grateful to my all friends who help me and giving me opinion during implementation of this project.

ABSTRACT

To achieve high compression ratio in video coding, a technique known as Block Matching Motion Estimation has been widely adopted in various coding standards. 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 in this algorithm. To overcome this drawback, many fast Block Matching Algorithms (BMAs) have been proposed and developed. Different search patterns and strategies are exploited in these algorithms in order to find the optimum motion vector with minimal number of required search points. One of these fast BMAs, which is proposed to be implemented in this project, is called New Three Step Search (NTSS) Algorithm. This project requires the algorithm to be implemented in MATLAB and then its performance is compared to FS algorithm as well as to other fast BMAs in terms of the peak signal-to-noise ratio (PSNR), number of required search points and computational complexity.

ABSTRAK

Untuk mencapai nisbah mampatan yang lebih tinggi dalam pengekodan video, satu teknik yang dikenali sebagai *Block Matching Motion Estimation* telah digunakan secara luas dalam pelbagai piawaian pengekodan. Secara konvensksyennya, teknik ini menguji setiap blok dalam tingkap pencarian. Ini dikenali sebagai algoritma *Full Search* (FS) yang dapat memberi penyelesaian yang optimum. Tetapi, algoritma ini mewujudkan beban pemprosesan yang lebih dan seterusnya melambatkan pemprosesan. Bagi mengatasi masalah ini, banyak algoritma *Block Matching* telah dibangunkan. Pelbagai corak dan strategi telah dieksploitasi dalam algoritma-algoritma ini untuk mencari vector pergerakan yang optima dengan titik pencarian yang minima. Salah satu algoritma tersebut, dikenali sebagai *New Three Step Search* (NTSS), telah dicadangkan untuk dilaksanakan dalam projek ini. Projek ini memerlukan algoritma ini dibangunkan dan dilaksanakan dalam MATLAB. Prestasi algoritma ini akan dianalisis dan dibezakan dengan algoritma FS serta algoritma-algoritma yang lain.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

BDM	-	Block Distortion Measure
BMAs	-	Block Matching Algorithms
CIF	-	Common Intermediate Format
DS	-	Diamond Search
FS	-	Full Search
FSS	-	Four Step Search
LDSP	-	Large Diamond Search Pattern
MAD	-	Mean Absolute Difference
MATLAB	-	Matrix Laboratory
MPEG	-	Moving Picture Experts Group
MSE	-	Mean Squared Error
NTSS	-	New Three Step Search
PSNR	-	Peak Signal-To-Noise Ratio
QCIF	-	Quarter Common Intermediate Format
SDSP	-	Small Diamond Search Pattern
TSS	-	Three Step Search

LIST OF APPENDIX

CHAPTER I

INTRODUCTION

1.1 Introduction

A technique known as Block Matching Motion Estimation has been widely adopted in various coding standards to achieve high compression ratio in video coding. 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 in this algorithm. To overcome this drawback, many fast Block Matching Algorithms (BMAs) have been proposed and developed. Different search patterns and strategies are exploited in these algorithms in order to find the optimum motion vector with minimal number of required search points.

One of these fast BMAs, which is proposed to be implemented in this project, is called New Three Step Search (NTSS) Algorithm. This project is requires the algorithm to be implemented in MATLAB and its performance is compared to FS algorithm as well as to other fast BMAs in terms of the peak signal-to-noise ratio (PSNR), number of required search points and computational complexity.

1.2 Problem Statements

In recent years, several video compression standards had been proposed for different applications such as CCITT H.261, MPEG-1 and MPEG-2. Generally, video data constitutes most of the multimedia data. Efficient coding of video is important for effectual usage of limited bandwidth and storage medium. Temporal correlation between successive image frames enables high amount of compression. Motion estimation is an important tool for exploiting temporal correlation. Block based motion estimation with non-overlapping rectangular blocks is used in many video coding standards. In this case, image frames are divided into non-overlapping blocks and the best match is searched around a pre-defined search range using all possible positions for each block.

Though this FS method provides optimal quality it significantly suffers from computational load. FS method matches all possible displaced candidate block within the search area in the reference frame in order to find the block with minimum distortion, so this FS algorithm have large motion and more searching point to do the blocks matching and thus the computational may be too complex.

1.3 Objective

The main objective of this project is to implement one of the available fast BMAs, namely NTSS algorithm to overcome the problem encountered by FS algorithm. Besides, the aims are also:

a) To develop and implement NTSS algorithm in MATLAB
b) To compare and analyze the performance of NTSS algorithm to FS algorithm as well as other common fast BMAs.
c) To produce a functional MATLAB program code.

1.4 Scopes of Work

The scopes of works in this project are:

a) Data and theory acquisition on image processing, motion estimation, BMAs and NTSS algorithm.
b) Implementation of NTSS algorithm on MATLAB.
c) Performance comparison of the NTSS algorithm to other available BMAs.

1.5 Brief Explanation of Methodology

There is a guideline in order to complete this project by setting a systematic study of methods which provide guidance when problem is encountered. In addition, this guideline could be an explanation to other people regarding the way this project is being completed. There are nine steps in this project guideline which are stated in chapter 3.

1.6 Thesis Structure

The thesis consists of 5 chapters. Following is the description of each chapter in the thesis.

Chapter 1 gives reader a general description on the project idea. The chapter contains introduction, problem statement, objective of the project, scopes of work, brief methodology, and report structure.

Chapter 2 is a literature review on theoretical concepts applied in this project. The chapter includes the image processing, motion estimation and BMAs. Besides that, this chapter also explains the different type of fast BMAs.

Chapter 3 introduces the methodology of the project. The project is divided to nine steps and each step is being described. .

Chapter 4 will cover all the result from the project. It will also include a discussion part and then analysis is made based on the result.

Chapter 5 will be the conclusion of the project. The chapter is concluded with some recommendation that can be implemented in the future.

CHAPTER II

LITERATURE REVIEW

2.1 Video Compression

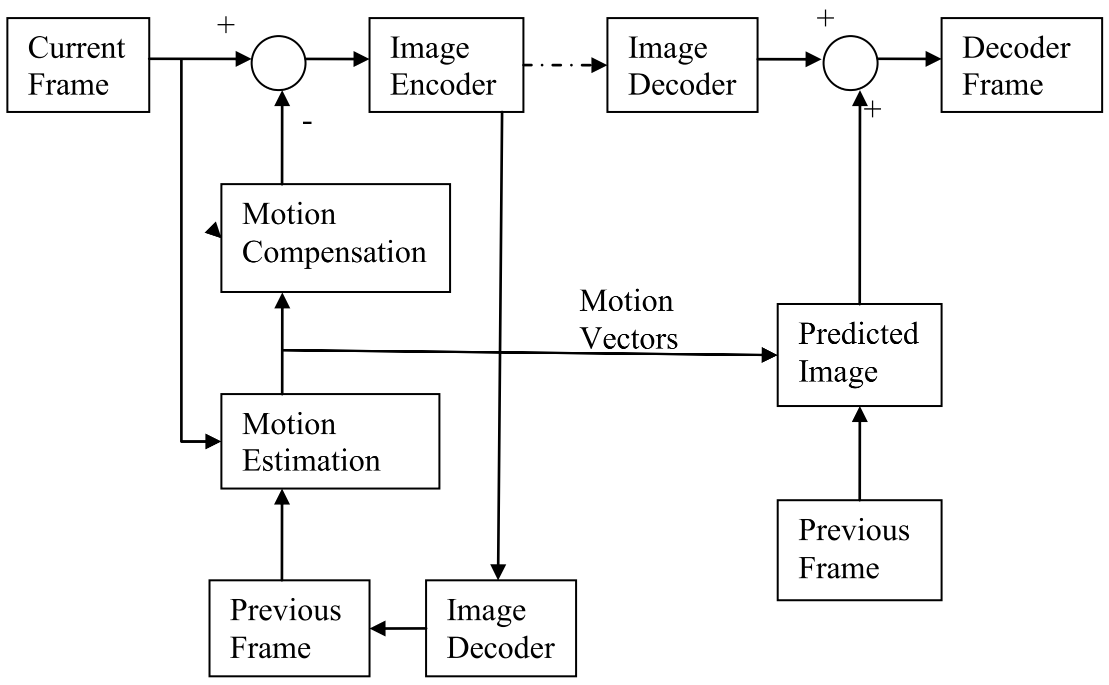

Figure 2.1 MPEG/H.26x Video Compression Process Flow

Compression is the process of compacting data into a smaller size in terms of number of bytes in digital media. Data such as text file, pictures, voice and other data that contains redundancy can be made smaller by employing compression. Since an uncompressed video scene can occupy a large amount of storage space, video compression has an important role in the digital world. [1]

The compression system involves an encoder (compression unit) and a decoder (decompression unit). The encoder exploits the redundancy among the given data and converts it to a compressed data stream. The decoder interprets the compressed data stream and restores it into the original format.

In Figure 2.1, the encoding side estimates the motion in the current frame with respect to a previous frame. A motion compensated image for the current frame is then created that is built of blocks of image from the previous frame. The motion vectors for blocks used for motion estimation are transmitted, as well as the difference of the compensated image with the current frame is encoded and sent. The encoded image that is sent is then decoded at the encoder and used as a reference frame for the subsequent frames. The decoder reverses the process and creates a full frame.

2.2 Motion Estimation

Motion estimation is the process of determining motion vectors that describe the transformation from one two-dimensional (2D) image to another usually from adjacent frames in a video sequence. The idea of motion estimation based video compression is to save on bits by sending encoded difference images which inherently have less energy and can be highly compressed as compared to sending a full frame.

The motion in the current frame is estimated with respect to a previous frame. Motion information is used in video compression to find best matching block in reference frame to calculate low energy residue. The aim is to obtain motion vector which may relate to the whole image or specific parts, such as rectangular blocks, arbitrary shaped patches or even per pixel. This technique eliminates the temporal redundancy due to high correlation between consecutive frames.

The motion estimation module will create a model for the current frame by modifying the reference frames such that it is a very close match to the current frame. This estimated current frame is then motion compensated and the compensated residual image is then encoded and transmitted.

2.3 Block Matching Algorithm

BMA is the block-based search technique and the idea behind BMA is to divide the current frame into a matrix of macro blocks that are then compared with corresponding block and its adjacent neighbors in the previous frame to create a vector that stipulates the movement of a macro block from one location to another in the previous frame. This movement calculated for all the macro blocks comprising a frame, constitutes the motion estimated in the current frame.

A search window with size equal to the rectangular block is placed on those equally divided block to find out the displacement of the best matched block from previous frame as the motion vector to the block in the current frame. Usually the macro block is taken as a square of side 16 pixels.

$$MAD = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left| C_{ij} - R_{ij} \right| \qquad (2.1)$$

$$MSE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{i=0}^{N-1} (C_{ij} - R_{ij})^2 \qquad (2.2)$$

The macro block that results in the least cost is the one that matches the closest to current block. There are various cost functions, of which the most popular and less computationally expensive is Mean Absolute Difference (MAD) given by equation (2.1). Another cost function is Mean Squared Error (MSE) given by equation (2.2) where N is the side of the macro bock, C_{ij} and R_{ij} are the pixels being compared in current macro block and reference macro block, respectively.[2][3]

Block distortions or block-matching error form an error surface over the search window, and the minimum matching error point over the search window corresponds to the motion vector where the best matching (or the least error) incurs. When the best match is found, the motion vectors and residues between the current block and reference block are computed. The process of getting the residues and motion vectors is known as motion compensation. The residues and motion vectors of best match are encoded by the transform unit and entropy unit and transmitted to the decoder side. At decoder side, the process is reversed to reconstruct the original picture.

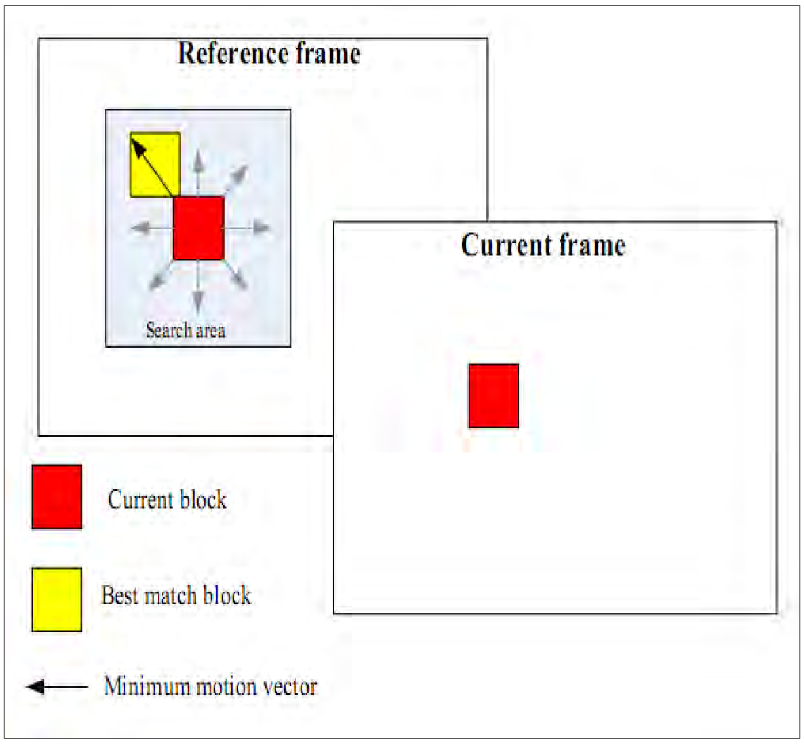

Figure 2.2: Block Based Motion Estimation [1]

Many fast BMAs have been developed, for example, Three Step Search (TSS), New Three Step Search (NTSS), Four Step Search (FSS), Diamond Search (DS), etc. These fast BMAs exploit different search patterns and search strategies for finding the optimum motion vector with drastically reduced number of search points as compared with the FS algorithm.