IMPLEMENTATION OF CROSS SEARCH (CS) ALGORITHM IN MATION ESTIMATION IN MATLAB

NORASYIKIN BINTI HJ. MD. SALLEH

This Report Is Submitted In Partial Fulfillment of Requirement For
Bachelor Degree of Electronic Engineering (Telecommunication Electronic)

Fakulti Kejuruteraan Elektronik dan Kejuruteraan Komputer
Universiti Teknikal Malaysia Melaka

April 2007

UNIVERSTI TEKNIKAL MALAYSIA, Melaka | UTeM |

FAKULTI KEJURUTERAAN ELEKTRONIK DAN KEJURUTERAAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : Implementation of Cross Search (CS) Algorithm in Motion Estimation in MATLAB.

Sesi Pengajian : 2006/2007

Saya, NORASYIKIN BINTI HJ. MD. SALLEH

mengaku membenarkan laporan 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)

Alamat Tetap: 1-158 Jalan Rajawali 7,

Taman tanjung Gading, 84000 Muar.

Tarikh: 3 MAY 2007

(COP DAN TANDATANGAN PENYELIA)

REDZUAN B ABDUL MANAP
Pensyarah
Fakulti Kej Elektronik dan Kej Komputer (FKEKK),
Universiti Teknikal Malaysia Melaka (UTeM),
Karung Berkunci 1200,
Ayer Keroh, 75450 Melaka

Tarikh: 3/5/07

"Saya akui laporan ini adalah hasil kerja saya sendiri kecuali ringkasan dan petikan yang tiap-tiap satunya telah saya jelaskan sumbernya."

Tandatangan : ....

Nama Penulis : NORASYIKIN BINTI MD. SALLEH

Tarikh : 3 May 2007

"Saya akui bahawa saya telah membaca karya ini pada pandangan saya karya ini adalah memadai dari skop dan kualiti untuk tujuan penganugerahan Ijazah Sarjana Muda Kejuruteraan Elektronik (Elektronik Telekomunikasi)."

Tandatangan : ....

Nama Penyelia : EN. REDZUAN BIN ANDUL MANAP

Tarikh : MAY 2007

REDZUAN B ABDUL MANAP
Pensyarah
Fakulti Kej Elektronik dan Kej Komputer (FKEKK),
Universiti Teknikal Malaysia Melaka (UTeM),
Karung Berkunci 1200,
Ayer Keroh, 75450 Melaka

To my lovely parents
And
My Fiancée...

ACKNOWLEDGMENTS

I would like to extend my gratitude to all those who helped and supported me while preparing this thesis. Firstly, I would like to thank my supervisor Mr. Redzuan Abd. Manap for his support and guidance for both technical and otherwise over the last one year.

My deepest gratitude, love and affection belong to my parents for their constant love and support. I owe all that I am and all I have ever accomplished.

ABSTRACT

Block matching is a widely used method for stereo vision, visual tracking, and video compression. Among block matching motion estimation algorithms, the full search algorithm is known for its superiority in the performance over other matching techniques. However, this method is computationally very extensive. Thus, many previous fast block matching algorithms have been proposed and developed with aim to reduce computational costs while maintaining desired quality performance. One of these, which are proposed for the implementation in this project, is called Cross Search Algorithm. The aim of this algorithm is to find the optimum motion vector with minimal number of search points along the process. The performance of this algorithm will be compared to other common algorithms in terms of peak signal to noise ratio (PSNR), number of search points required and computational complexity.

ABSTRAK

Penyesuaian blok (Block Matching) biasanya digunakan untuk visi stereo, laluan penglihatan dan kemampatan video. Di antara kaedah pengiraan pergerakan penyesuaian blok adalah pengiraan penuh (Full Search), yang dikenali sebagai algorithm terbaik berbanding teknik pengiraan yang lain. Walaubagaimanapun kaedah ini terlalu rumit dan memerlukan pengaturcaraan yang rumit. Beberapa kaedah penyesuaian blok dibangunkan dengan tujuan untuk mengurangkan kos sementara kualiti persembahan dikekalkan. Salah satu daripada kaedah tersebut dikenali sebagai 'Cross Search Algorithm'. Tujuan utama kaedah ini adalah untuk mencari pengiraan vektor dengan pengurangan jumlah pengiraan sepanjang proses dijalankan. Persembahan kaedah ini akan dibandingkan dengan kaedah lain dari segi peak signal to noise ratio (PSNR), bilangan titik yang diperlukan dan pengiraan yang tepat.

CONTENTS

LIST OF ACRONYMS

BBGDS	Block Based Gradient Descent Search
BMA	Block Matching Algorithm
BMC	Block Matching Compensation
CCF	Cross-Correlation Function
CDS	Cross Conjugate Search
CS	Cross Search
DS	Diamond Search
FS	Full Search
FSS	Four Step Search
LDSP	Large Diamond Search Pattern
MAE	Mean Absolute Error
MC	Motion Compensation
MME	Modified Motion Estimation
MSE	Mean Square Error
MV	Motion Vector
NTSS	New Three Step Search
OSA	Orthogonal Search Algorithm
PSNR	Peak Signal Noise Ratio
SDSP	Small Diamond Search Pattern
TDL	Two-Dimension Logarithm
TSS	Three Step Search
VBSMC	Variable Block Size Motion Compensation

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 Introduction

Video has huge redundant information which must be exploited to be stored and transmitted efficiently. The common technique to achieve this goal is known as motion estimation. In this technique, the current frame is predicted from a previous frame known as reference frame by using motion vectors. With the increasing demand of multimedia applications, considerable efforts are needed for efficient video compressing and encoding algorithms. Motion estimation has proven to be an effective technique for exploiting the temporal redundancy in video sequences and is therefore an essential part of MPEG and H.263 compression standards.

Since motion estimation is the most computationally intensive portion of video encoding, efficient fast motion estimation algorithms are highly desired for video compressors subject to diverse requirement on bit rate, video sequence characteristics and delay. Knowledge of the motion is not available from a video data and must be deduced using computationally intensive algorithms. For efficient handling of motions with variety of contents, the need for adaptive motion estimation methods is inevitable.

Generally, the optimal full search (FS) block matching algorithm results in the best performance with respect to the quality of decoded video sequences, however, it is computationally very intensive. Due to the huge demand of the computational requirement several fast search algorithms have been developed and introduced in recent years including the three step search (TSS), the new three step search (NTSS), the four step search (FSS), the block based gradient descent search (BBGDS), the diamond search(DS), and Cross Search (CS).

In this thesis, the CS Algorithm is proposed. The aim of this algorithm is to find the optimum motion vector with minimal number of search points along the process. The performance of this algorithm will be compared to other common algorithms in terms of PSNR, number of search points required and computational complexity.

1.2 Objectives

i. To investigate and gain knowledge on theories of video coding, Motion Estimation, Block Matching Algorithm (BMA) and CS Algorithm.
ii. To implement the CS Algorithm in MATLAB and to analyze its performances.
iii. To produces working program.

1.3 Problem Statement

Several BMA have been developed in order to achieve high compression ratio in video coding techniques. One of them is CS algorithm. The main objective of this project is to implement the CS algorithm and verify its performances.

1.4 Scopes of Work

Scopes of work of this project are:

i. Acquisition of Motion Estimation, video coding, BMA and CS algorithm theory and background information including the use of MATLAB.
ii. Implementation of CS algorithm in MATLAB.
iii. Performance analysis of the implemented algorithm and comparison to other algorithm.

1.5 Thesis Layout

Chapter 2 provides some background on video motion compensation. It introduces the problem of video motion estimation for general problems in computer vision and video compression. This is followed by Block Matching. This shown Block Matching is the most time consuming part of the encoding process. Motion Estimation also will be introduced. The temporal prediction technique used in MPEG video is based on motion estimation.

Chapter 3 presents a theory of CS Algorithm. It starts by a review of BMA. A set of preliminary concepts in various measurement are introduced. This is followed by a detailed discussion on step for design criteria and search schemes for the proposed motion estimation algorithm.

Chapter 4 is detailed in methodology overview. Project started with data acquisition and literature review. Step one in MATLAB is extracting video sequence into frame. This followed by BMA and implementation of CS algorithm. Then the predicted frame had been constructed. Last step is performance analysis and results.

Chapter 5 presents detailed experimental results. The project work is divided into two stages. For the first stage, only CS algorithm is implemented into the standard video sequence and its PSNR and search point's performance are analyzed. The second stage, CS algorithm is compared to several other common BMA algorithms by using several standard video sequences.

Chapter 6 is discussing the results with respect to the observations from experiments. Discussion presents detailed experimental results of the proposed algorithm in comparison to standard algorithms on various videos with different characteristics. Chapter 7 concludes the thesis and suggests some future research directions.

CHAPTER II

LITERATURE REVIEW

2.1 Video Compression

In video compression, motion compensation (MC) describes a picture in terms of where each section of that picture came from, in a previous picture. This is often employed in video compression. It can be also used for deinterlacing. A video sequence consists of a number of pictures - usually called frames. Subsequent frames are very similar, thus, containing a lot of redundancy. Removing this redundancy helps achieve the goal of better compression ratios.

A first approach would be to simply subtract a reference frame from a given frame. The difference is then called residual and usually contains less energy (or information) than the original frame. The residual can be encoded at a lower bit-rate with the same quality. The decoder can reconstruct the original frame by adding the reference frame again.

A more sophisticated approach is to approximate the motion of the whole scene and the objects of a video sequence. The motion is described by some parameters that have to be encoded in the bit-stream. The pixels of the predicted frame are approximated by appropriately translated pixels of the reference frame. This gives much better residuals than a simple subtraction. However, the bit-rate occupied by the parameters of the motion model must not become too large.

Usually, the frames are processed in groups. One frame (usually the first) is encoded without motion compensation just as a normal image. This frame is called I-frame (intra-coded frame, MPEG terminology) or I-picture. The other frames are called P-frames or P-pictures and are predicted from the I-frame or P-frame that comes (temporally) immediately before it. The prediction schemes are, for instance, described as IPPPP, meaning that a group consists of one I-frame followed by four P-frames.

Frames can also be predicted from future frames. The future frames then need to be encoded before the predicted frames and thus, the encoding order does not necessarily match the real frame order. Such frames are usually predicted from two directions, i.e. from the I- or P-frames that immediately precede or follow the predicted frame.

In global MC, the motion model basically reflects camera motions such as dolly (forward, backwards), track (left, right), boom (up, down), pan (left, right), tilt (up, down) and roll (along the view axis). It works best for still scenes without moving objects. There are several advantages of global MC:

i. It models precisely the major part of motion usually found in video sequences with just a few parameters. The share in bit-rate of these parameters is negligible.
ii. It does not partition the frames. This avoids artifacts at partition borders.

iii. A straight line (in the time direction) of pixels with equal spatial positions in the frame corresponds to a continuously moving point in the real scene. Other MC schemes introduce discontinuities in the time direction.

However, moving objects are not sufficiently represented by global MC. Thus, other methods are preferable.

2.2 Block Motion Compensation

In block motion compensation (BMC), the frames are partitioned in blocks of pixels (e.g. macroblocks of 16×16 pixels in MPEG). Each block is predicted from a block of equal size in the reference frame. The blocks are not transformed in any way apart from being shifted to the position of the predicted block. This shift is represented by a motion vector (MV).

The MV is the parameters of this motion model and has to be encoded into the bit-stream. As the MVs are not always independent (e.g. if two neighboring blocks belong to the same moving object), they are usually encoded differentially to save bit-rate. This means that the difference of the MV and the neighboring MV(s) encoded before is encoded. (The result of this differencing process is mathematically equivalent to global motion compensation capable of panning.) An entropy codec can exploit the resulting statistical distribution of the MV (around the zero vectors).

It is possible to shift blocks by non-integer vectors, which is called sub-pixel precision. This is done by interpolating the pixel's values. Usually, the precision of the MVs is increased by one bit: half-pixel precision. Of course, the computational expense for sub-pixel precision is much higher since the interpolation functions can be quite consuming.

The big disadvantage of BMC is the discontinuities introduced at block borders (blocking artifacts). They have the form of sharp horizontal and vertical edges which, firstly, are highly visual for the human eye and, secondly, produce ringing effects (big coefficients in high frequency sub-bands) in the Fourier-related transform used for transform coding of the residual frames.

BMC divides up the *current* frame into non-overlapping blocks, and the MC vector tells where those blocks come *from*. The source blocks typically have some overlap in the source frame. Some video compression algorithms assemble the current frame out of pieces of several different previously-transmitted frames.

2.3 Variable Block-Size Motion Compensation

Variable block-size motion compensation (VBSMC) is the use of BMC with the ability for the encoder to dynamically select the size of the blocks. When coding video, the use of larger blocks can reduce the number of bits needed to represent the MVs, while the use of smaller blocks can result in a smaller amount of prediction residual information to encode. Older designs such as H.261 and MPEG-1 video typically use a fixed block size, while newer ones such as H.263, MPEG-4 Part 2, H.264/MPEG-4 AVC, and VC-1 give the encoder the ability to dynamically choose what block size will be used to represent the motion.

2.4 Frame Segmentation

The current frame of video to be compressed is divided into equal sized non-overlapping rectangular blocks. Ideally the frame dimensions are multiples of the block size and square blocks are most common. Chan et al. [1] used rectangular blocks of 16 x 8 pixels, claiming that blocks of this shape exploit the fact that motion within image sequences is more often in the horizontal direction than the vertical. Block size affects the performance of compression techniques. The larger the block size, the fewer the number of blocks, and hence fewer MVs need to be transmitted. However, borders of moving objects do not normally coincide with the borders of blocks and so larger blocks require more correction data to be transmitted. Small blocks result in a greater number of MVs, but each matching block is more likely to closely match its target and so less correction data is required. Lallaret et al. [1] found that if the block size is too small then the compression system will be very sensitive to noise. Thus block size represents a trade off between minimizing the number of MVs and maximizing the quality of the matching blocks. The relationship between block size, image quality, and compression ratio has been the subject of much research and is well understood.

2.5 Search Threshold

If the difference between the target block and the candidate block at the same position in the past frame is below some threshold then it is assumed that no motion has taken place and a zero vector is returned. Thus the expense of a search is avoided. Most video codec employ a threshold in order to determine if the computational effort of a search is warranted.

2.6 Block Matching

Block matching is the most time consuming part of the encoding process. During block matching each target block of the current frame is compared with a past frame in order to find a matching block [1]. When the current frame is reconstructed by the receiver this matching block is used as a substitute for the block from the current frame. Block matching takes place only on the luminance component of frames. The colour components of the blocks are included when coding the frame but they are not usually used when evaluating the appropriateness of potential substitutes or *candidate blocks*.