This Report Is Submitted In Partial Fulfillment Of Requirements For The Bachelor Degree of Electronic Engineering (Computer Engineering)

Signature : ....[illegible]........

Supervisor's Name : Mr.Redzuan bin Manap

Date : May 5, 2006

REDZUAN BIN ABDUL MANAP
Pensyarah
Fakulti Kejuruteraan Elektronik dan Kejuruteraan Komputer
Kolej Universiti Teknikal Kebangsaan Malaysia
Karung Berkunci 1200
Ayer Keroh, 75450 Melaka

MODELING A NEW DIAMOND SEARCH ALGORITHM FOR MOTION ESTIMATION IN MATLAB

RANJIT SINGH s/o SARBAN SINGH

This Report Is Submitted In Partial Fulfillment Of Requirements For The Bachelor Degree of Electronic Engineering (Computer engineering)

Fakulti Kejuruteraan Elektronik dan Kejuruteraan Komputer
Kolej Universiti Teknikal Kebangsaan Malaysia

May 2006

I hereby declare that this thesis is the result of my own effort except certain notes that I clearly stated as references and sources.

Signature :..........[illegible]..........

Name : Ranjit Singh s/o Sarban Singh

Date : May 5, 2006

ACKNOWLEDGMENTS

Thanks go to my friends Leng Huey Ley, Suzanna Liu, Mohd Nazri and Isuwan. Everything I created and produced is a credit to them. Thanks go to my brother Baljit Singh in helping me to get information from University Kebangsaan Malaysia (UKM). My special thanks go to my lecturers, in particular my supervisor Mr. Redzuan bin Manap for helping me in understanding about my project and share all his knowledge with me. A thanks goes to my PSM 1 panels, Ms. Anis Suhaila and Ms. Fauziah, for inspiring me and giving thoughts for modification on my project. Together, thanks go to the entire 4BENC class member for all the collaboration.

ABSTRACT

Due to limited channel bandwidth, video coding is an important process for visual communication applications and always requires very high compression ratio. To achieve that, a method called Block Matching Motion Estimation is widely used in various coding standards. Many fast block matching algorithms have been proposed and developed in this method. One of these, which is proposed for the study in this project, is called A New Diamond Search Algorithm. This algorithm will exploit diamond shape search patterns to find the optimum motion vector with minimal number of search points along the process. The technique will be implemented using the MATLAB system and it's performances will be compared to the others algorithms on term of peak signal noise to ratio (PSNR), computational complexity and processing time.

ABSTRAK

Disebabkan kebatasan lebar jalur, kod video adalah satu proses yang penting bagi pengaplikasian kamunikasi visual dan sentiasa memerlukan nisbah kemampatan yang tinggi. Untuk mencapai matlamat tersebut, satu cara yang dikenali sebagai " Block Matching Motion Estimation" digunakan dengan meluas dalam pelbagai kod piawai. Banyak algoritma "block matching" telah dicadangkan and dibangunkan. Salah satu cara telah dicadangkan dalam projek ini ialah "A New Diamond Search Algorithm ". Algoritma ini akan membentuk bentuk berlian untuk mencari "motion vector" yang paling optimum dengan pencarian yang minimum sepanjang proses. Teknik ini akan dilaksanakan menggunakan MATLAB system dan keputusannya akan dibandingkan dengan algoritma yang lain berdasarkan kepada "peak signal noise to ratio (PNSR)", kekompleksan penggiraan dan pemprosesan masa.

CONTENTS

CONTENTS

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES

LIST OF FIGURES

LIST OF FIGURES

LIST OF ACRONYMS

BMA – Block Matching Algorithm

DS – Diamond Search

EBMA – Exhaustive Block Matching Algorithm

EJO – Early Jump Out

FS – Full Search

FSS – Four Step Search

FPS – Frame Per Second

ISDN – Integrated Service Digital Network

ITU – International Telecom Union

LDSP – Large Diamond Shape Pattern

LTMCP – Long Term Memory Motion Compensation Prediction

MAD – Mean Absolute Difference

MAE – Mean Absolute Error

MB – Macroblocks

MBD – Minimum Block Distortion

ME – Motion Estimation

MPEG – Moving Picture Expert Group

MSE – Mean Squared Error

NTSS – New Three Step Search

PCM – Pulse Code Modulation

PSNR – Point Signal Noise to Ratio

QCIF – Quarter Common Inter format

SAD – Sum of Absolute Differences

SAE – Sum of Absolute Error

SDSP – Small Diamond Shape Pattern

TSS – Three Search Step

VLSI – Very Large Scale Integration

CHAPTER 1

OBJECTIVE, SCOPE OF WORK AND METHODOLOGY

1.1 OBJECTIVE

Basically, many algorithms are available and developed recently to maintain accuracy and precision in achieving high compression ratio in video coding technique. Each technique has its own advantages and disadvantages. Motion Estimation or video coding are widely being used but user's are not aware about the function and the most suitable algorithm that can be used. Thus, this project is proposed to implement one of the algorithms which are available. This is the Diamond Search technique. This fast Block Matching Algorithm will exploit different search patterns and strategies for finding the optimum motion vector with drastically reduced number of search points compared to some other algorithms.

- To reduced the computation cost and compare between available algorithms (advantages and disadvantages).
- Gain the error of image processing. Besides that also producing a fast processing algorithm to process a quality image.
- To gain knowledge on image processing area:
    - This includes Motion Estimation (ME)
    - Block Matching Algorithm (BMA)
    - Diamond Search (DS)

- To model the algorithm using MATLAB and subsequently analyze the performance.
- To produce a simple, working program code for the algorithm.

1.2 WORK SCOPE

There are few types of algorithm available in block matching ME when study is carried out the ME of the displacement or velocity of image structures. This is applied to one frame to another in a time sequence of 2-D images.

Besides that, coding that being used is repeatedly and based on blocks. The blocks are divided into 15 x 15 sizes of blocks to implement this algorithm. This is known as matrix's division. For matrix division MATLAB system software is the right choice to implement this algorithm.

MATLAB system is used to execute this algorithm and the performances of this algorithm are analyzed and to verify the less computation complexity, reduce the time processing and others related.

1.3 METHODOLOGY

1. Data acquisition and literature review of.
    a. Video Coding
    b. ME
    c. BMA
    d. Algorithms available.

2. Extraction video sequence into frame.
    a. Learn how the video sequence can be divided into frame construction.
    b. How it is does in MATLAB system.

3. Block Matching Development
    a. Study how block is constructed.
    b. How the search window operates
    c. Diamond Search development in MATLAB

4. Reconstruction of frames
    a. Get reconstructed image quality and average number of points
    b. Compared with other algorithms
    c. Check on the performances.

6. Final theses and Demonstration (Seminar).

CHAPTER 2

BACKGROUND STUDY

2.1 VIDEO CODING

Video coding is the field in computer science that deals with finding efficient coding formats for digital video. Video data usually not only contains visual information but also audio. Therefore, it is often referred to as multimedia. Modern video coding standards even include other multimedia data such as synthetic computer graphics, text and meta information for searching/browsing and digital rights management. They also often provide mechanisms for user interaction. However, the most intensive parts of video data in terms of data size (memory demand, transmission bandwidth) remain (visual) video and audio data. These parts have to be compressed. Unfortunately, this can hardly be done without loss of quality (lossy compression), because of the enormous size of a lossless video stream. There are two special research areas that deal with multimedia compression: video compression and audio compression. Video coding has two distinct goals: *storing* and *transmission* of video data. These two goals have much in common. Therefore, video file formats usually have the same structure as streaming video formats with just a little header information added. A computer program that encodes and decodes video data is called video codec. (The part that only deals with audio data is called audio codec.)Almost all successful techniques developed for video coding have been integrated in the Moving Picture Expert Group (MPEG) standards. Therefore, the

MPEG standards represent a comprehensive knowledge base for video coding. Most video coding standards and commercial formats are modifications of MPEG.[1,2]

2.2 H.261 [2]

H.261 is video coding standard published by the ITU (International Telecom Union) in 1990. It was designed for datarates which are multiples of 64Kbit/s, and is sometimes called p x 64Kbit/s (p is in the range 1-30). These datarates suit Integrated Service Digital Network (ISDN) lines, for which this video codec was designed for. The coding algorithm is a hybrid of inter-picture prediction, transform coding, and motion compensation. The datarate of the coding algorithm was designed to be able to be set to between 40 Kbits/s and 2 Mbits/s. The inter-picture prediction removes temporal redundancy. The transform coding removes the spatial redundancy. Motion vectors are used to help the codec compensate for motion. To remove any further redundancy in the transmitted bitstream, variable length coding is used. H.261 supports two resolutions, Quarter Common Inter format (QCIF).

Picture Formats Supported							
Picture format	**Luminance pixels**	**Luminance lines**	**H.261 support**	**Uncompressed bitrate (Mbit/s)**			
				10 frames/s		**30 frames/s**	
				Grey	**Colour**	**Grey**	**Colour**
QCIF	176	144	Yes	2.0	3.0	6.1	9.1
CIF	352	288	Optional	8.1	12.2	24.3	36.5

Table 1.1: Picture Formats Supported

2.3 MOTION ESTIMATION ALGORITHMS

ME is a process to estimate the pels or pixels of the current frame from reference frame(s). The temporal prediction technique used in video is based on ME. The basic premise of ME is that in most cases, consecutive video frames will be similar except for changes induced by objects moving within the frames. ME can be done using the block matching technique which exploit different search patterns and search strategies for finding the optimum motion vector for particular ME while reducing the number of search points. It efficiently removes the temporal redundancy between successive frames by BMA. Block-based ME is the most practical approach to obtain motion compensated prediction frames. It divides frames into equally sized rectangular blocks and finds out the displacement of the best-matched block from previous frame as the motion vector to the block in the current frame within a search window. The benefits of long-term memory motion compensated prediction (LTMCP) [3] have been emphasized in recent years. Consequently, this tool has been adopted by several recent standards like H.263+ and H.264iMPEG-4 AVC [3,4]. As continuously dropping the costs of semiconductors, notably higher prediction gain can be achieved by estimating more reference frames in the memory buffer. Nevertheless, an obvious drawback is the complexity will increase proportionally. Extra data are also needed to describe the reference indices.

In the early 1980s, some conventional Fast Block Matching algorithms were proposed, which provides the optimal result by matching all possible candidates within the search window (pixels). Such as the Three Steps Search (TSS), the 2D Logarithmic Search, Full Search (FS), Four Step Search (FSS) and Diamond Search (DS). Among the algorithms, DS is a simple, robust and efficient fast Block Matching Motion Estimation algorithm [3].

Example of ME operation:

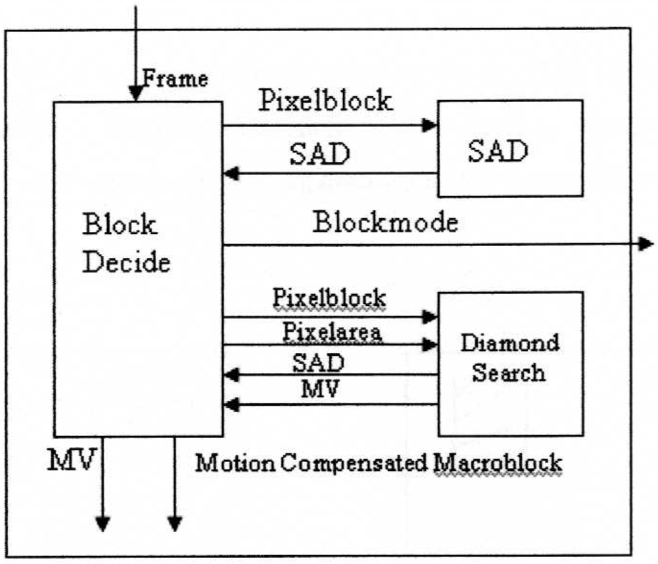

Figure 2.1 : ME Subnetwork

In both ME nodes, ME is performed in the subnetwork. Both the current and reference frame and enters the Decide node, where the current frame is split up in pixel blocks of a certain size in the luminance field. Then the Sum of Absolute Differences (*SAD*) is applied per pixel block on the same pixel blocks in the reference frame, and *DS* is performed on the current pixel block and an area around the same location in the reference frame. The location found with the lowest *SAD* is used as the best match for 'most similar' block. Based on this minimum SAD the Decide node decides if the pixel block is an Intra- or Inter- or NotCoded-block. When the minimum SAD is too high, the pixel block is typed as an Intra-block. If the SAD is not too high, the pixel block is typed as an Inter-block. If the SAD is almost equal to zero, the pixel block is typed as a NotCoded-block.

2.4 BLOCK MATCHING ALGORITHM

BMA are based on the matching of macroblocks (MB) between two Motion Estimation subnetwork images. For each MB, a motion vector is evaluated by matching the MB within a search area, according to the Mean Absolute Error (MAE) criterion.

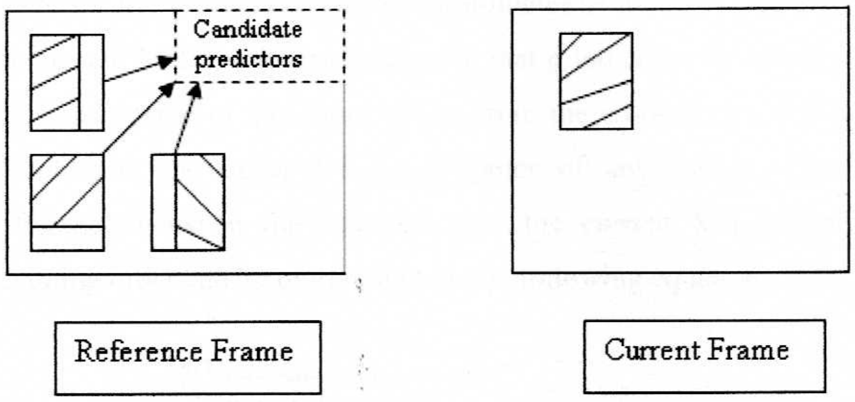

Figure 2.2: Similarity between MB of the current and reference frames

In Figure 2.2, it is clear that the dissimilar areas (represented as white patches in the reference frame) of candidate macro block predictors, can be faster identified and potentially eliminated by different scanning orders than the one used by the standard. In the cases depicted in the figure, a scanning order from the right-most column to the left-most, from the left-most column to the right-most and from the bottom-most row to the top-most would be ideal for fast rejection of candidate predictors. The above figure also suggests that joint exploitation of horizontal and vertical SAD information may speed-up the rejection of candidate predictors that are not the best matches. This idea will be further exploited in the design of the scanning orders. The design of scanning orders for Full search motion estimation algorithms can be subdivided in zero overhead and in limited pre-processing schemes. In zero overhead schemes, the order of SAD computation between pixels in the current and reference MB is usually fixed and the choice of the order tries to compensate for the blindness of the estimation. There are 3 different scanning methods: [4]

i. **Spiraling inward scanning order:** This scanning order is based on the idea that the SAD value between pixels located on corresponding positions on the sides of squares of decreasing size inside the current and reference MB, may be used to reject candidate predictors faster than the raster scan order used by MPEG-2. For a 16*16 pixel area there are 8 such squares with sides 16,14,12,10,8,6,4 and 2 pixels respectively. Formally, let's assume that (,) *n* *I i j* is the intensity of pixel *(i,j)* inside a frame *n*, *dx* and *dy* are the motion vector co-ordinates for a candidate best match of the MB to be encoded in the reference frame and *(k,l)* are the co-ordinates of the upper left hand corner of the current MB. Furthermore, assume that *q* indicates the offset of the upper left hand corner of any inner square from the co-ordinates *(k,l)* and *m* is an offset from the upper left hand corner of any square. Then the SAD difference between the reference and the current MB according to the scanning order can be represented by the following equation:

$$SAD_{reference\_macroblock-current\_macroblock} =$$
$$\sum_{q=0}^{q=7}\left[\begin{array}{l} \displaystyle\sum_{m=0}^{m=15-2\times q} \begin{array}{l}|I_t(k+q+m,l+q)- \\ I_{t-1}(k+q+m+dx,l+q+dy)| +\end{array} \\ \displaystyle\sum_{m=0}^{m=15-2\times q} \begin{array}{l}|I_t(k+(15-q),l+q+m)- \\ I_{t-1}(k+(15-q)+dx,l+q+m+dy)| +\end{array} \\ \displaystyle\sum_{m=0}^{m=15-2\times q} \begin{array}{l}|I_t(k+(15-q)-m,l+(15-q))- \\ I_{t-1}(k+(15-q)-m+dx,l+(15-q)+dy)| +\end{array} \\ \displaystyle\sum_{m=0}^{m=15-2\times q} \begin{array}{l}|I_t(k+q,l+(15-q)-m)- \\ I_{t-1}(k+q+dx,l+(15-q)-m+dy)|\end{array} \end{array}\right]$$

The following figure 2.3 shows the scanning order:

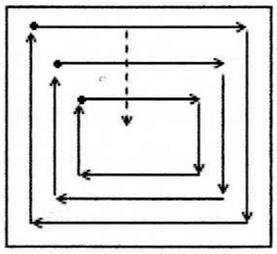

Figure 2.3: Spiraling inward scanning order

In the above figure, SAD computation starts from the points depicted as diamonds and potential Early Jump Out (EJO) points are at the corners of the squares of decreasing size for both the reference and current MB. The direction of the SAD