A Review on Detection of Motion in Real Time Images Using Pel Approach

[1]Mr.Parveen Kumar (M.Tech Student), [2]Mrs.Pooja Sharma (Astt.Prof.)

[1,2]Department of Electronics & Communication Engineering, Galaxy Global Educational Trust's Group of Institution, Ambala

Email: parveenporiya@gmail.com, pooja.sharma@galaxyglobaledu.com

Contact No. +919416677456

Abstract— Estimating the motion (or dynamics) manifested in a set of images or an image sequence is a fundamental problem in both image and video processing and computer vision. From a computer vision perspective, much of what is interpretable in this real-world scene is reflected in the apparent motion. For instance, estimating apparent motion in a video sequence provides necessary information for many applications including self-directed navigation, industrial process control, 3-D shape reconstruction, object recognition, robotic motion control, object tracing, and automatic image sequence analysis. In image and video processing, estimation of motion plays a dynamic role in video compression as well as multi-frame image enhancement. Unlike as they may seem, these many applications share one common thread in all such applications, demand is high for accurate estimates of motion requiring negligible computational cost. In this, a new technique called Motion Detection using Pixel Processing is proposed. This technique is based on pattern search algorithm, in which size is dynamically determined, based on the mean of two motion vectors of the neighboring macro-blocks instead of one as is the case with adaptive rood patter search. In this, we present enhancement based motion detection which will be implemented on MATLAB.

Keywords— Motion Detection, Pixel Processing, Real time enhancement, motion estimation.

I. INTRODUCTION

Comparing digitally stored video sequences and those "stored" on celluloid and considering fact that data-storage or data-transmission capacity still is restricted in computer technology, this comparison shows necessity of compressing the video-data: Using the same approach to store videos digitally as they are in classic way on celluloid would require at least 25 still images per second. A high-quality 90 minute movie with a resolution of 720*576 pixels and a 24-bit color depth per pixel could require of over 156 GB of storage capacity. Transmitting this amount of data over Internet is unreasonable, especially when real-time performance is needed. This uncompressed video needs a transmission bandwidth of over 237 M Bit/s. Similar problems occur when storing data to disc – only very few memory devices have the necessary capacity [1].

In video sequences, motion is a key source of information. Motion arises due to moving objects in 3D scene, as well as camera motion. Apparent motion, also known as optical flow, captures resulting spatial-temporal variations of pixel intensities in successive images of a sequence. The purpose of motion estimation techniques is to recover this information by analyzing image content. Efficient and accurate motion estimation is an essential component in domains of image sequence examination, computer vision and video communication.

In context of image sequence analysis and computer vision, the main objective of motion estimation algorithms is exactly and faithfully models motion in the scene. This information is fundamental for video understanding and object tracking. Relevant applications include video surveillance, robotics, autonomous vehicles navigation, human motion investigation, quality control in manufacturing, video search and retrieval, and video restoration. Accurate motion is also important in some video processing tasks such as frame rate conversion or de-interlacing [2].

As far as video coding is concerned, compression is attained by exploiting data redundancies in both spatial and temporal dimensions. Spatial redundancies reduction is largely achieved by transform-coding, e.g. using the Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT), which effectively compacts signal energy into a few significant coefficients. In turn, temporal redundancies are reduced by means of predictive coding. Observing that temporal correlation is maximized along motion trajectories, motion compensated prediction is used for this purpose. In this situation, the main objective of motion estimation is no longer to find 'true' motion in the scene, but rather to maximize compression efficiency. In other words, motion vectors should deliver an exact prediction of signal. Moreover, the motion information should enable a compact representation, as it has to be conveyed as overhead in the compressed code stream. Efficient motion estimation is a key to achieve high compression in video coding applications such as TV broadcasting, Internet video streaming, digital cinema, DVD, Blue-ray Disc [3].

A video sequence can be considered to be a discretized three-dimensional projection of the real four-dimensional continuous space-time. The objects in real world may move, interchange, or deform. The movements cannot be observed directly, but instead the light reflected from object surfaces and projected onto an image. The light source can be moving, and reflected light varies depending on the angle between a surface and a light source. There may be objects occluding the light rays and casting shadows. The objects may be transparent (so that several independent motions could be observed at the same location of an image) or there might be fog, rain or snow blurring the observed image. The discretization causes noise into the video sequence, from which video encoder makes its motion estimations. There may also be noises in the image capture device (such as a video camera) or in electrical transmission lines. A perfect motion model would take all factors into account and find the motion that has maximum likelihood from the observed video sequence [4].

The paper is organized as follows. In section II, we discuss related work with recognition of fingerprint images. In Section III, It describes basic motion detection system, its standards and block matching technique. In Section IV, it describes proposed technique of motion detection. Finally, conclusion is explained in Section V.

II. RELATED WORK

In literature, authors proposed a search technique based on conjugate directions, and another simpler technique called one-at-a-time search based on comparison of two methods, the latter technique is adopted as basis for further research. The accepted technique is compared with brute force search, existing 2-d logarithmic search, and a modified version of it, for motion compensated prediction. To estimate the motion on a block-by-block basis by brute force requires extensive computations. These motion estimation techniques are applied to video sequences, and their larger performance compared to the existing techniques is illustrated based on quantitative measures of the prediction errors [5].

Some Authors proposed that three-step search (TSS) algorithm has been widely used as motion estimation technique in some low bit-rate video compression applications, owing to its simplicity and effectiveness. However, TSS uses a consistently allocated checking point pattern in its first step, which becomes inefficient for estimation of small motions. A new three-step search (NTSS) algorithm is proposed in this paper. The features of NTSS are that it pays a centre-biased checking point pattern in first step, which is derived by making the search adaptive to motion vector distribution, and a halfway-stop technique to reduce computation cost. Simulation results show that, as related to TSS, NTSS is much more robust, produces smaller motion compensation errors, and has a very well-matched computational complexity [6].

Some proposed a new four-step search (4SS) algorithm with centre-biased checking point pattern for fast block motion estimation. Halfway-stop technique is employed in new algorithm with searching steps of 2 to 4 and the total number of checking points is varied from 17 to 27. Simulation results show that proposed 4SS performs better than the well-known three-step search and has similar performance to the new three-step search (N3SS) in terms of motion compensation errors. In addition, the 4SS also reduces worst-case computational requirement from 33 to 27 search points and average computational requirement from 21 to 19 search points as compared with N3SS [7].

Authors proposed a block-based gradient descent search (BBGDS) algorithm to perform block motion estimation in video coding. The BBGDS evaluates values of a given objective function starting from a small centralized checking block. The minimum within checking block is found, and gradient descent direction where the minimum is expected to lie is used to determine search direction and position of new checking block. The BBGDS is compared with full search (FS), three-step search (TSS), one-at-a-time search (OTS), and new three-step search (NTSS). Experimental results show that proposed technique provides competitive performance with reduced computational complexity [9].

III. MOTION ESTIMATION

Motion estimation (ME) techniques have been successfully applied in motion compensated predictive coding for falling temporal redundancies. They belong to class of nonlinear predictive coding techniques. An effective representation of motion is serious in order to reach high performance in video coding. Estimation techniques should on one hand provide good prediction, but on other hand should have low computational load [8].

Changes between frames are mainly due to movement of objects. Using a model of motion of objects between frames, encoder estimates the motion that occurred between reference frame and current frame. This process is called motion detection. The encoder then uses this motion model and information to move contents of reference frame to provide a better prediction of the current frame. This process is called motion compensation (MC), and prediction so produced is called motion-compensated prediction (MCP) or displaced-frame (DF). The purpose of Motion estimation (ME) is indeed to globally diminish the sum of these terms. As a compromise, block matching ME, even though not optimal, has been universally used in inter-frame motion compensated (MC) predictive coding since its computational complexity is much lesser than optical flow recursive methods [11].

In block based ME image is partitioned into blocks and same displacement vector is assigned to all pixels within a block. The motion model undertakes that an image is usually composed of rigid objects in translational motion. Although assumption of translational motion is often considered to be a major drawback in presence of zoom but block matching technique is able to estimate closely the true zooming motion.

A. Video Standards

Since there are endless ways to compress and encode data, and many terminal vendors which each may have a unique idea of data compression, common standards are required, that rigidly define how video is coded in the transmission channel. There are mainly two standard series in common use, both having several versions. International Telecommunications Union (ITU) started developing Recommendation H.261 in 1984, and effort was finished in 1990 when it was approved. The standard is aimed for video conferencing and video phone services over integrated service digital network (ISDN) with bit rate a multiple of 64 kilobits per second [10].

MPEG-1 is a video compression standard developed in joint operation by International Standards Organization (ISO) and International Electro-Technical Commission (IEC). The system development was started in 1988 and finished in 1990, and it was accepted as standard in 1992. MPEG-1 can be used at higher bit rates than H.261, at about 1.5 megabits per second, which is suitable for storing the compressed video stream on compact disks or for using with interactive multimedia systems [3]. The standard covers also audio associated with a video [13].

In 1996 a revised version of standard, Recommendation H.263, was finalized which adopts some new techniques for compression, such as half pixel and optionally smaller block size for motion compensation. As a result it has better video quality than H.261.Recommendation H.261 divides each frame into 16 × 16 picture element (pixel) blocks for backward motion compensation, and H.263 can also take advantage of 8 × 8 pixel blocks. A new ITU standard in development is called H.26L, and it allows motion compensation with greater variation in block sizes.

For motion estimation, MPEG-1 uses same block size as H.261, 16 × 16 pixels, but in addition to backward compensation, MPEG can also apply bidirectional motion compensation. A revised standard, MPEG-2, was approved in 1994. Its target is at higher bit rates than MPEG-1, from 2 to 30 megabits per second, where applications may be digital television or video services through a fast computer network. The latest ISO/IEC video coding standard is MPEG-4, which was approved in beginning of 1999. It is targeted at very low bit rates (832 kilobits per second) suitable for e.g. mobile video phones. MPEG-4 can be also used with higher bit rates, up to 4 megabits per second.

B. Block Matching Technique

In a typical Block Matching Algorithm, each frame is separated into blocks, each of which consists of luminance and chrominance blocks. Usually, for coding efficiency, motion approximation is performed only on the luminance block. Each luminance block in present frame is matched against candidate blocks in a search area on the reference frame. These candidate blocks are just displaced versions of unique block. The best candidate block is found and its displacement (motion vector) is recorded. In a typical inter-frame coder, input frame is subtracted from the prediction of the reference frame. Consequently motion vector and the resulting error can be transmitted instead of the original luminance block; thus inter-frame redundancy is removed and data compression is achieved. At receiver end, decoder builds the frame difference signal from received data and adds it to reconstructed reference frames [15].

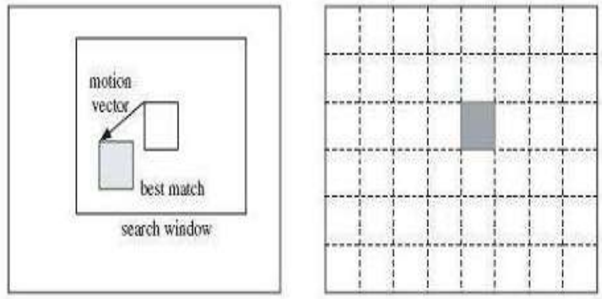

Figure 1: Block-Matching Motion Estimation [15]

This algorithm is based on a translational model of motion of objects between frames. It also assumes that all pels within a block undergo same translational movement.

C. Matching Criteria for Motion Estimation

Inter frame predictive coding is used to eliminate the large amount of temporal and spatial redundancy that exists in video sequences and helps in compressing them. In conventional predictive coding difference between the current frame and the predicted frame is coded and transmitted. The better the prediction, smaller the error and hence the transmission bit rate when there is motion in a sequence, then a pel on same part of the moving object is a better prediction for the current pel [12].

1. Block Size

The important parameter of the BMA is the block size. If the block size is smaller, it attains better prediction quality. This is due to a number of reasons. A smaller block size reduces effect of the accuracy problem. In other words, with a smaller block size, there is less possibility that block will contain different objects moving in different directions. In addition, a smaller block size provides a better piecewise translational approximation to non-translational motion. Since a smaller block size means that there are more blocks (and consequently more motion vectors) per frame, this better prediction quality comes at the expense of a larger motion above. Most video coding standards use a block size of 16 × 16 as a compromise between prediction quality and motion overhead. A number of variable-block-size motion estimation methods have also been proposed in the literature [14].

2. Search Range

The maximum allowed motion displacement dm, also known as the search range, has a direct impact on both the computational complexity and the prediction quality of the BMA. A small dm results in poor compensation for fast-moving areas and consequently poor prediction quality. A large dm, on other hand, results in better prediction quality but leads to an increase in the computational complexity. A larger dm can also result in longer motion vectors and consequently a slight increase in motion overhead [6]. In general, a maximum allowed displacement of dm = ±15 pels is sufficient for low-bit-rate applications. MPEG standard uses a maximum displacement of about ±15 pels, although this range can optionally be doubled with the unrestricted motion vector mode [15].

3. Search Accuracy

Initially, the BMA was designed to estimate motion displacements with full-pel accuracy. Clearly, this limits the performance of the algorithm, since in reality the motion of objects is completely unrelated to the sampling grid. A number of workers in the field have proposed to extend the BMA to sub-pel accuracy. For example, Ericsson demonstrated that a prediction gain of about 2 dB can be obtained by moving from full-pel to 1/8-pel accuracy. Girod presented an elegant theoretical analysis of motion-compensating prediction with sub-pel accuracy. He termed the resulting prediction gain the accuracy effect. He also presented that there is a “critical accuracy” beyond which the possibility of further improving prediction is very small. He concluded that with block sizes of 16 × 16, quarter-pel accuracy is desirable for broadcast TV signals, whereas half-pel accuracy appears to be sufficient for videophone signals [14].

IV. MOTION DETECTION USING PIXEL APPROACH

The proposed steps for motion estimation using pixel approach are:

1. Interface WEBCAM with MATLAB
   a) Install image acquisition device.
   b) Retrieve hardware information.
   c) Create a video input object.

   d) Preview video stream (optional)
   e) Configure object properties.
   f) Acquire image data.
   g) Starting the video input object.
   h) Triggering the acquisition.
2. Image Reading
3. Image Enhancement
4. Image Conversion
5. Image Segmentation
6. Apply thresholding process
7. Feature Extraction
8. Detect the final motion

A. Basic Image Acquisition Procedure

This section illustrates basic steps required to create an image acquisition application:

1. Install Your Image Acquisition Device

Follow the setup instructions that come with your image acquisition device. Setup typically involves installing frame grabber board in your computer. These are supplied by the device vendor by connecting a camera to a connector on the frame grabber board. It verifies that the camera is working properly by running the application software that came with the camera and viewing a live video stream. Generic Windows image acquisition strategies, such as Webcams and digital video camcorders, typically do not require installation of a frame grabber board. You connect these devices straight to your computer via a USB or FireWire port.

2. Retrieve Hardware Information

You may use the imaqhwinfo function to retrieve Adaptor name, Device ID, Video format. You can optionally specify the video format of the video input object. To define which video formats an image acquisition device supports, look in the Supported Formats field of the device info structure returned by the imaqhwinfo function.

3. Create a Video Input Object

In this step it creates video input object that the toolbox uses to represent the connection between MATLAB and an image acquisition device. Using properties of a video input object, you can control many aspects of image acquisition process. To create a video input object, enter the video input function at the MATLAB prompt. The video input function uses the adaptor name, device ID, and video format that you retrieved in step 2 to create the object. The adaptor name is the only required argument; the video input function can use defaults for the device ID and video format. For more info about image acquisition objects, see linking to Hardware.

vid = videoinput('matrox');

4. Image Reading

img= imread (strcat (a,num2str(i),b));

This takes the grey values of all the pixels in the grey scale image and puts them all into a matrix img. This matrix img is now a MATLAB variable, and so we can perform many matrix operations on it. In general the imread function reads pixel values from an image file, and returns a matrix of all pixel values.

5. Image Enhancement

Image quality is an important factor in performance of minutiae extraction and matching algorithms. A good quality image has high contrast between ridges and valleys. A poor quality image is low in contrast, noisy, wrecked, or smudgy, causing spurious and missing minutiae. Poor quality can be due to cuts, creases, or bruises on surface of fingertip, excessively wet or dry skin condition, uncooperative attitude of subjects, broken and unclean scanner devices, low quality fingers (elderly people, manual worker), and other factors [11].

6. Image Segmentation

- Segmentation refers to operation of partitioning an image into component parts, or into separate objects.
- Segmentation refers to process of partitioning a digital image into multiple segments.
- The aim of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyses.
- Image segmentation is normally used to locate objects and boundaries in images.
- More precisely, image segmentation is process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.
- The outcome of image segmentation is a set of segments that collectively cover the entire image.

7. Thresholding

- A grey scale image is turned into a binary (black and white) image by first choosing a grey level in the original image, and then converting every pixel black or white according to whether its grey value is greater than or less than T :
- A pixel becomes : {white if its grey level is > T
  {Black if its grey level is <T
- Thresholding is a vital part of image segmentation, where we wish to isolate objects from background. It is also an important component of robot vision. Thresholding can be done very simply in MATLAB.

B. Implementation

MATLAB is one of a number of commercially accessible, sophisticated mathematical computation tools, which also comprise Maple, Mathematica, and MathCAD. Despite what supporters may claim, no single one of these tools is “the best.” Each has strengths and weaknesses. Each allows you to perform simple mathematical computations. They differ in way they handle symbolic calculations and more complicated mathematical processes, such as matrix manipulation. For example, MATLAB (short for Matrix Laboratory) excels at computations involving matrices, whereas Maple excels at symbolic calculations. At a central level, you can think of these programs as sophisticated computer-based calculators. They can perform same functions as your scientific ‘C’ calculator—and many more. If you have a computer on your desk, you may find yourself using MATLAB instead of your calculator for even the simplest mathematical applications—for example, balancing your check book. In many engineering classes, the use of programs such as MATLAB to perform computations is replacing more traditional computer programming.

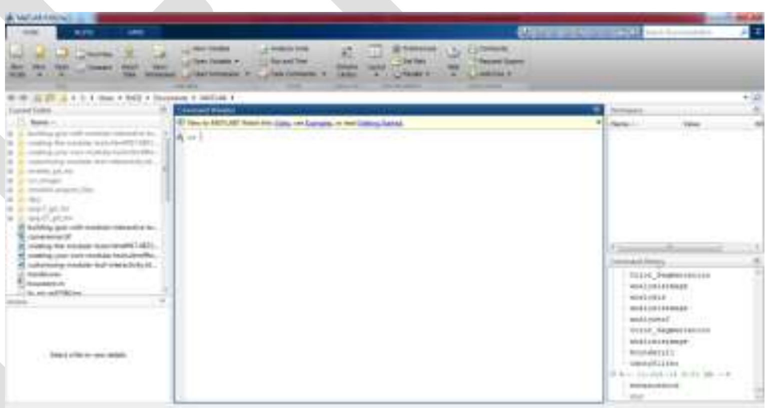

Figure 2: MATLAB Tool

CONCLUSION

Because of the Internet is more and more universal and the technology of multimedia has been progressed, the communication of image data is a part in life. In order to pay effect in a limit transmission bandwidth, to convey most, high quality user information .It is necessary to have more advanced compression method in image and data. Pel-recursive motion estimation is a well-established approach. The proposed algorithm can decrease computational time as compared to block based technique. Motion Estimation (ME) and compensation techniques, which can remove temporal redundancy between adjacent frames effectively, have been extensively applied to popular video compression coding standards such as MPEG- 2, MPEG-4. The displacement of each picture element in each frame forms displacement vector field (DVF) and its estimation can be done using at least two successive frames. The pixel based approaches depend upon intensity of image and its performance is affected by presence of noise. While block based techniques depends upon motion vectors and it has high computation time as compared to pixel approaches.

REFFERENCES:

[1] Ram Srinivasan and K. R. Rao. Predictive Coding Based On Efficient Motion Estimation. IEEE Transactions On Communications, Vol. Com-33, No. 7, August 1985
[2] Renxiang Li, Bing Zeng, And Ming L. Liou. A New Three-Step Search Algorithm For Block Motion Estimation. IEEE Transactions On Circuits And Systems For Video Technology, Vol. 4, No. 3, August 1994.
[3] Lai-Man Po Wing-Chung Ma. A Novel Four-Step Search Algorithm For Fast Block Motion Estimation. IEEE Trans. Circuits Syst. Video Technol., Vol. 6, No. 3, Pp. 312-317,Jun. 1996.
[4] Lurng-Kuo Liu and Ephraim Feig. A Block-Based Gradient Descent Search Algorithm for Block Motion Estimation In Video Coding. IEEE Transactions On Circuits And Systems For Video Technology, Vol. 6, No. 3, August 1996.
[5] Jo Yew Tham, Surendra Ranganath, Maitreya Ranganath, and Ashraf Ali Kassim. A Novel Unrestricted Center-Biased Diamond Search Algorithm for Block Motion Estimation. IEEE Transactions On Circuits And Systems For Video Technology, Vol. 8, No. 3, August 1998
[6] Ce Zhu, Xiao Lin, and Lap-Pui Chau. Hexagon-Based Search Pattern for Fast Block Motion Estimation. Ieee Transactions On Circuits And Systems For Video Technology, Vol. 12, No. 4, May 2002
[7] Chun-Ho Cheung, Member, IEEE, And Lai-Man Po, Member, IEEE. Novel Cross-Diamond-Hexagonal Search Algorithms For Fast Block Motion Estimation. Ieee Transactions On Multimedia, Vol. 7, No. 1, February 2005.
[8] Michael Gallant and Faouzi Kossentini. An Efficient Computation-Constrained Block-Based Motion Estimation Algorithm for Low Bit Rate Video Coding.
[9] Ishfaq Ahmad, Senior Member, Ieee, Weiguo Zheng, Member, IEEE, Jiancong Luo, Member, IEEE, And Ming Liou, Life Fellow, IEEE. A Fast Adaptive Motion Estimation Algorithm. IEEE Transactions On Circuits And Systems For Video Technology, Vol. 16, No. 3, March 2006.
[10] Ka-Ho Ng, Lai-Man Po and Ka-Man Wong. Search Patterns Switching For Motion Estimation Using Rate Of Error Descent. Icme 2007.
[11] Sumeer Goel, Student Member, Ieee Andmagdy A. Bayoumi, Fellow, IEEE. Multi-Path Search Algorithm for Block-Based Motion Estimation. Icip 2006
[12] Alexis M. Tourapis. Enhanced Predictive Zonal Search For Single And Multiple Frame Motion Estimation.
[13] B. Kasi Viswanatha Reddy & Sukadev Meher. Three Step Diamond Search Algorithm for Fast Block-Matching Motion Estimation. International Conference On Electrical, Electronics, Communications And Photonics, ISBN : 978-93-81693-88-19 , Goa, 31st March, 2013
[14] Sven Klomp, Marco Munderloh, Yuri Vatis, Jörn Ostermann, Fellow, Ieee. Decoder-Side Block Motion Estimation For H.264 / Mpeg-4 Avc Based Video Coding. Institut Für Informationsverarbeitung Leibniz Universität Hannover, Appelstr. 9a, 30166 Hannover, Germany2009 IEEE.
[15] Lai-Man Po, Chi-Wang Ting, Ka-Man Wong, And Ka-Ho Ng. Novel Point-Oriented Inner Searches For Fast Block Motion Estimation. IEEE Transactions on Multimedia, Vol. 8, No. 1, January 2007