A STUDY ON MOTION ESTIMATION

M. SUSHMA SRI1

1Faculty in ECE, Govt. Institute of Electronics

Abstract- The proliferation of high powered computers, the availability of high quality & inexpensive video cameras & the increasing need for automated video analysis has generated a great deal of interest in object tracking. In this regard object tracking based on Motion Estimation which is a popular technique for computing the displacement vectors between object and motion capture. Motion is very important feature of image sequences. Motion estimation is a challenging and fundamental problem of computer vision and it is demanding field among researchers. With the recent advances in video technology, there is rapid increasing need for a more reliable, efficient and robust for video processing and its analysis. The most general and challenging version of motion estimation is to compute an independent estimate of motion at each pixel, which is generally known as optical or optic flow. In this paper we have provided overview of some basic concepts behind motion estimation, block matching algorithm and optical flow.

Keywords- Block Matching algorithm, Motion Estimation, Optical Flow.

I. INTRODUCTION

With the widespread adoption of technologies such as digital television, Internet streaming video and DVD, video compression has become an essential component of broadcast and entertainment media. Video compression is useful because it helps to reduce the consumption of expensive resources such as data storage on hard disks/servers and transmission bandwidths. Video sequences contain a significant amount of statistical and subjective redundancies within and between frames. The ultimate goal of a video source coding is bit rate reduction for storage and transmission by exploring both statistical (spatial) and subjective (temporal) redundancies. Motion Estimation techniques exploit these unwanted redundancies and achieve video compression [1]. Motion Estimation using block matching techniques is the most widely used method to find motion vector (MV). In this paper, we present three categories of the search algorithms for motion estimation using block matching method.

II. MOTION ESTIMATION

Motion estimation is the process of determining motion vectors that describe the transformation from one 2D image to another; usually from adjacent frames in a video sequence. Motion estimation based on video compression helps in saving bits by sending encoded difference images which have inherently less entropy as opposed to sending a fully coded frame. However the most computationally expensive and resource extensive operation in the entire compression process is motion estimation. Hence, a fast and computationally inexpensive algorithm for motion estimation is a need for video compression. The methods for finding motion vectors can be categorised into pixel based methods direct and feature based methods indirect.

A. DIRECT METHODS

- Block-matching algorithm
- Phase correlation and frequency domain methods
- Pixel recursive algorithms
- Optical flow
B. INDIRECT METHODS

Indirect methods use features, such as corner detection, and match corresponding features between frames, usually with a statistical function applied over a local or global area. In the real world, the objects may move, rotate, or deform. The movements cannot be observed directly, but instead the light reflected from the object surfaces and projected onto an image. The light can be moving, and the reflected back 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 or there might be fog, rain or snow blurring the observed image. This discretization causes noise into the video sequence, from which the video encoder makes its motion estimations. There may also be noises in the image capture device (such as a video camera) or in the electrical transmission lines. A perfect motion model would take all the factors into account and find the motion that has the maximum likelihood from the observed video sequence.

Changes between frames are mainly due to the movement of objects. Using a model of the motion of objects between frames, the encoder estimates the motion that occurred between the reference frame and the current frame. This process is called motion estimation (ME) [2] as shown in “Fig.” 1. The encoder then uses this motion model and information to move the contents of the reference frame to provide a better prediction of the current frame. This process is known as motion compensation (MC), and the prediction so produced is called the motion compensated prediction (MCP) or the displaced-frame (DF) [4]. The most commonly used ME method is the block matching motion estimation (BMME) algorithm.

III. BLOCK MATCHING ALGORITHM

The Block Matching Algorithm (BMA) is a way of locating matching macro blocks in a sequence of digital video frames for Motion Estimation (ME). A typical Block Matching Algorithm, each frame is divided into blocks, each of which consists of luminance and chrominance blocks as shown in “Fig.” 2. Usually, for coding efficiency, motion estimation is performed only on the luminance block. Each luminance block in the present frame is matched against candidate blocks in a search area on the reference frame. These candidate blocks are just the displaced versions of original block. The best candidate block is found and its displacement (motion vector) is recorded. In a typical interframe coder, the input frame is subtracted from the prediction of the reference frame. Consequently the motion vector and the resulting error can be transmitted instead of the original luminance block; thus interframe redundancy is removed and data compression is achieved. At receiver end, the decoder builds the frame difference signal from the received data and adds it to the reconstructed reference frames.

This algorithm is based on a translational model of the motion of objects between frames. It also assumes that all pixels within a block undergo the same translational movement. There are many
other ME methods, but BMME (block matching motion estimation) is normally preferred due to its simplicity and good compromise between prediction quality and motion overhead. This assumption is not strictly valid, since we capture 3-D scenes through the camera and objects do have more degrees of freedom than just the translational one. However, they are still reasonable, considering the practical movements of the objects over one frame and this makes our computations much simpler. There are many other approaches to motion estimation, some using the frequency or wavelet domains, and designers have considered scope to invent new methods since this process does not need to be specified in coding standards. The standards need only specify how the motion vectors should be interpreted by the decoder. Block Matching (BM) is the most common method of motion estimation. Typically each macro block (16 × 16 pels) in the new frame is compared with shifted regions of the same size from the previous decoded frame, and the shift which results in the minimum error is selected as the best motion vector for that macro block. The motion compensated prediction frame is then formed from all the shifted regions from the previous decoded frame [3].

A. BACKWARD MOTION ESTIMATION

The motion estimation generally considered as backward motion estimation, since the current frame is considered as the candidate frame and the reference frame on which the motion vectors are searched is a past frame, that is, the search is backward. Backward motion estimation leads to forward motion prediction.

“Figure 2.1: backward motion estimation with current frame as k and frame (k-1) as the reference frame”

B. FORWARD MOTION ESTIMATION

It is just the opposite of backward motion estimation. Here, the search for motion vectors is carried out on a frame that appears later than the candidates frame in temporal ordering. In other words, the search is “forward”. Forward motion estimation leads to backward motion prediction. It may appear that forward motion estimation is unusual, since one requires future frames to predict the candidate frame. However, this is not unusual, since the candidate frame, for which the motion vector is being sought is not necessarily the current, that is the most recent frame. It is possible to store more than one frame and use one of the past frames as a candidate frame that uses another frame, appearing later in the temporal order as a reference.

“Figure 2.2: Forward Motion estimation with current frame as k and frame (k+1) as the reference frame”

IV. OPTIMAL SEARCH ALGORITHMS

The optimal search techniques are the simplest block matching techniques for motion estimation. We have two optimal search techniques, viz. full search (FS) algorithm and successive elimination algorithm (SEA).
A. FULL SEARCH ALGORITHM

Full Search (FS) or Exhaustive Search (ES) algorithm is a method to find motion vectors from the given block within the search window. The algorithm starts by identifying the best matching block from the previous frame for a given block in the reference frame in a pixel-by-pixel manner and then estimating the motion vectors for the corresponding best matching block. The algorithm searches all the \((2W+1)^2\) candidate block positions in the search window of size \(W\). This algorithm gives the optimum quality of compression, but it is quite slow and requires more computational time.

![Figure 3.1: full search algorithm](image)

B. SUCCESSIVE ELIMINATION ALGORITHM

Successive elimination algorithm (SEA) [2] is a type of fast full search algorithm. This algorithm reduces the complexity and improves the speed of the search.

V. SUB-OPTIMAL SEARCH ALGORITHMS

The sub-optimal search algorithms existing in literature are three-step search (TSS) [6], four-step search (4SS) [4], diamond search (DS) [2], cross-search (CSA) [4] among others. These algorithms use geometric search patterns to determine the motion vector from a given macro-block. The performance of these algorithms depends on the search pattern used and the complexity of video sequence.

A. THREE STEP SEARCH ALGORITHM

The three-step search algorithm (TSS) [5], developed by Koga et al, uses a 9 x 9 grid with eight checking points and an initial step size. This algorithm searches for the point having minimum Block Distortion Measure (BDM) among all the eight points at a distance of step size around the center. After every search the center is moved to the point of minimum BDM and the step size is halved. The search terminates when the minimum BDM point happens to be the center, in which case the motion vectors corresponding to the center are the required vectors for the best matching block.

![Figure 4.1: three step search](image)

B. FOUR STEP SEARCH ALGORITHM

The four-step search algorithm (4SS) was developed by Lai-Man Po and Wing-Chung Ma [3], to resolve the problem encountered by the three-step search algorithm. The algorithm utilizes the center-biased search pattern with a nine checking points in a 5 x 5 window. The center of the search window is then shifted to minimum point is found at the center of the search window, the search will go to the final step, where a 3 x 3 search window is used to obtain the motion vector corresponding to the minimum BDM point. Otherwise, the search window size is maintained 5 x 5 and the search pattern is changed based on the location of minimum BDM point on the search window.
C. DIAMOND SEARCH ALGORITHM

The diamond search (DS) algorithm [3] uses two search patterns, viz. large diamond search pattern (LDSP) with nine checking points and small diamond search pattern (SDSP) with five checking points. In the searching procedure, the LDSP pattern is used repeatedly until the minimum BDM point is obtained at the center. Once the minimum BDM occurs at the center of the pattern, the LDSP switches to SDSP. The point with minimum BDM in SDSP provides the motion vector of the best matching block.

VI. OPTICAL FLOW

Optical flow or optic flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. In simple words: 2D Motion Field can be defined as -2D velocities for all visible points. And Optical Flow Field can be defined as- Estimate of the 2D motion field. Optical flow techniques such as motion detection, object segmentation, time-to-collision and focus of expansion calculations, motion compensated encoding, and stereo disparity measurement utilize this motion of the objects’ surfaces and edges. Optical flow estimation is still one of the key problems in computer vision. Estimating the displacement field between two images, it is applied as soon as correspondences between pixels are needed. In the recent times the quality of optical flow estimation methods has increased dramatically. To determine optical flow we need to track some property of images. Two key problems in optical flow estimation are: 1) Determine what image property to track 2) Determine how to track it. Some features of the images are assumed to stay constant among multiple frames during optical flow estimation.

VII. CONCLUSION

This paper consists of a overview on motion estimation and block matching algorithm which are more applicable for video coding application. In order to employ effect in a limit transmission bandwidth, to convey the most, high quality user information it is necessary to have more advanced compression method in image / video and data. Also in this paper, a novel block matching algorithm based on search algorithms are also discussed to estimate the motion vectors.

REFERENCES

[1] J.R.Jain and A.K.Jain, “Displacement measurement and its application in inter frame image coding,” IEEE Trans. Commun., vol.COM-29,PP. 1799–1808, Dec.1981.
[2] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, “A novel unrestricted center-biased diamond search algorithm for block motion estimation”, *IEEE Trans. Circuits Systems Video Technology*, vol. 8, no. 4, pp. 369-377, Aug 1998.

[3] L. Po and W. Ma, “A novel four-step search algorithm for fast block motion estimation”, *IEEE Trans. Circuits Systems Video Technology*, vol. 6, no. 3, pp. 313-317, Jun 1996.

[4] M. Ghanbari, “The cross-search algorithm for motion estimation”, *IEEE Trans. Communications*, vol. 38, no. 7, pp. 950-953, Jul 1990.

[5] S. Kappagantula and K. R. Rao, “Motion compensated interframe image Prediction”, *IEEE Trans. Commun.*, vol. 33, pp. 1011–1015, Sept, 1985.

[6] R. Li, B. Zeng, and M. L. Liou, “A new three-step search algorithm for fast motion estimation”, *IEEE Trans. Circuits Systems Video Technology*, vol. 4, no. 4, pp. 438-442, Aug 1994.