Motion Estimation and Video Compression

Waseem Khatri*, Sihesh Khapare*, Jarjit Tandel**

Department of Electrical Engineering, Rochester Institute of Technology, Rochester, NY 14623
{wik5717*, ssk1874*, jjt1506**}@rit.edu

ABSTRACT

Our objective is to develop an understanding of 2-D motion estimation and use it towards video compression. First, we estimate the motion between two successive frames using a three step search block matching technique using minimum mean absolute difference (MAD) or minimum mean square error criteria (MSE). We further introduce the concept of motion compensation and attempt to construct the next frame (predictive frame) using the motion vectors. Results indicate that the predicted frame is not exactly as the original frame. However, the quality of reconstructed video can further be improved if the prediction error is added to the predicted frame. Quantizing the prediction error leads to more compression.

Index Terms— Minimum Mean Squared Error, Minimum Absolute Difference, Block Matching, Three Step Search, Motion Compensation, Quantization

1. INTRODUCTION

Motion estimation is mainly used for video compression. A video sequence consists of series of frames. To achieve compression, the temporal redundancy between two adjacent frames can be exploited. That is, a frame is selected as a reference, and subsequent frames are predicted from the reference, using a technique known as motion estimation.

The basic premise of using motion estimation is that in most cases, consecutive video frames will be similar except for the changes induced by objects moving within the frame. In a trivial case of zero motion between two frames, it is easy for the encoder to efficiently predict the current frame as a duplicate of the prediction frame. When this is done, the only information necessary to transmit to the decoder becomes the syntactic overhead necessary to reconstruct the picture from the original reference frame.

When using motion estimation, we assume that the objects in the scene have only translational motion. This assumption holds as long as there is no camera pan, zoom, changes in luminance or rotational motion. Occlusion of one object by another and uncovered background are also neglected.

The paper is organized as follows: Section 2 briefly discusses the block matching techniques using MAD and MSE. Three step search method is discussed in Section 3. Motion compensation and results of predictive video are presented in Section 4. Section 5 contains the results when prediction error is added to the motion compensation process. Conclusions are drawn in Section 6, followed by some important references.

2. BLOCK MATCHING

2-D motion also called ‘projected motion,’ refers to the perspective or the orthographic projection of 3-D motion into the image plane. The movement of a point in 3-D can be represented using a 3-D displacement vector. The movement of the same point in the 2-D domain is captured by consecutive image frames, and can be represented by 2-D displacement vectors. Similarly, a displacement of all such points in an image will result in a motion field.

In block matching technique, we determine the motion vector for a particular block of pixels rather than every pixel. This method is used for most practical applications, as it involves less computation. Dedicated hardware implementation in VLSI can reduce computational cost.

The basic idea of block matching is depicted in the fig.1, where the displacement for a pixel \((n_1, n_2)\) in frame \(k\) (the present frame) is determined by considering an \(N_1 \times N_2\) block centered at \((n_1, n_2)\), searching frame \(k+1\) (the search frame) for the location of the best-matching block of same size. The search is limited to a \((N_1 + 2M_1) \times (N_2 + 2M_2)\) region called the search window.

There are several methods that can quantify the matching of the blocks. Two of the most common methods used are minimum mean squared error (MSE) and minimum absolute difference (MAD).

The equation for MSE is given below,

\[
MSE = \frac{1}{N_1 N_2} \sum_{(n_1, n_2) \in B} [s(n_1, n_2, k) - s(n_1 + d1, n_2 + d2, k + 1)]^2
\]

(13)
where $B$ denotes an $N_1 \times N_2$ block, for a set of candidate motion vectors $(d_1, d_2)$ which minimizes the MSE, that is,

$$
\begin{bmatrix}
\hat{d}_1 \\
\hat{d}_2
\end{bmatrix} = \arg \min_{(d_1, d_2)} \text{MSE} (d_1, d_2)
$$

(14)

The equation for MAD is defined as,

$$
\text{MAD} = \frac{1}{N_1 N_2} \sum_{(n_1, n_2) \in B} |s(n_1, n_2, k) - s(n_1 + d_1, n_2 + d_2, k + 1)|
$$

(15)

MAD is widely used for VLSI implementations. The displacement estimate is given by,

$$
[\hat{d}_1, \hat{d}_2]^T = \arg \min_{(d_1, d_2)} \text{MAD}(d_1, d_2)
$$

(16)

However, the performance of MAD criterion deteriorates as the search area becomes larger due to the presence of several local minima.

3. THREE STEP SEARCH

To determine the best matching block requires optimizing the matching criterion over all possible candidate displacement vectors at each pixel $(n_1, n_2)$. Evaluating such a matching criterion for all values of $(d_1, d_2)$ at each pixel is extremely time-consuming.

The three step search procedure is explained with the help of the Fig. 2, where only the search frame is depicted with the search window parameters $M_1 = M_2 = 7$. The “0” marks the pixel in the search frame that is just behind the present pixel. The criterion function is evaluated at nine points, the pixel “0,” and the pixels marked as “1”.

If the lowest MSE or MSAD is found at the pixel “0,” then we have “no motion”. In the second step, the criterion function is evaluated at 8 points that are marked by “2” centered about the pixel chosen as the best match in the first stage (denoted by circled “1”). Since each step is smaller than the previous step, we achieve finer resolution in our estimates. We stop the search after three steps.

The distance vector is computed from the new center and the center “0”. Similarly, we generate distance vectors for all such blocks in the image and plot the motion field.

4. MOTION COMPENSATION

Motion compensation is used as part of the predictive process. If an image sequence shows moving objects, then their motion within the scene is represented by the motion vectors. The information from the motion field can be used to predict the content of frames later in the sequence.

Fig. 3 describes the proposed algorithm for motion compensation. The procedure is explained for a video frame at $k$ and $k+1$. We first divide the frame $k$ into blocks of $16 \times 16$ and compare with those of frame $k+1$ using the three step search technique describe in Section 3. After finding the finding the lowest MSE or MAD we determine the motion vector for each block. Since we are using forward difference in the calculation of MSE or MAD, we call this process a forward motion estimation process.
The results of the motion compensation process are shown below:

a) Video Sequence used for the experimentation.

b) Output of displacement vectors using the Three Step Search Method

c) Predicted video frame from forward motion estimation
5. MOTION COMPENSATION USING PREDICTION ERROR

Prediction error is described as the difference between the actual frame \( k+1 \) and the predicted frame \( k+1 \). It can be easily understood from the figure below;

![Fig. 7: Computation of Prediction Error](image)

We can further obtain compression by quantizing the prediction error. The above process is indicated in diagram of the proposed algorithm in fig.3. In our experiments we built a uniform 8 bit quantizer. Since the error information is very small (as seen from fig. 8) we can utilize a lower bit quantizer as well.

![Fig. 8: Prediction Error](image)

The output obtained after adding the prediction error to predicted frame \( k+1 \) is indicated in the figure below,

![Fig. 9: Final Output of frame 61](image)

6. CONCLUSION

The algorithm for motion estimation presented is simple and easy to implement. In our algorithm, we predict only the next frame how-ever, we can extend this to predicting two or more frames. Various compression techniques can be used to compress the original video frame and the predicted frames.

It is observed that the accuracy of the motion estimate is proportional to the computational cost. Using the block matching techniques we can obtain considerable computational saving, however the motion estimates obtained may not be very accurate. Also, the above mentioned method fails to handle rotation and scale variations in the scene. Thus, there is a need for a robust motion estimation technique which can fill the void of the existing motion estimation methods.

7. REFERENCES