Adaptive Motion Estimation Schemes Using Maximum Mutual Information Criterion

Jing Zhao Dapeng Wu Deniz Erdogmus Yuguang Fang
Zhihai He *

Abstract

We consider the motion estimation problem in video coding. In our previous work [10], we proposed a new motion estimation method where motion estimation is formulated as an optimization problem and an adaptive system under the minimum error entropy criterion is used for motion estimation. In this paper, we develop an adaptive system under the criterion of maximum mutual information to address the motion estimation problem. Our proposed motion estimation algorithms have very low encoding complexity and hence are ideally suited for wireless video sensor networks where limited bandwidth, restricted computational capability, and limited battery power supply impose stringent constraints on the video encoding system.

*Please direct all correspondence to Prof. Dapeng Wu, University of Florida, Dept. of Electrical & Computer Engineering, P.O.Box 116130, Gainesville, FL 32611, USA. Tel. (352) 392-4954, Fax (352) 392-0044, Email: wu@ece.ufl.edu, URL: http://www.wu.ece.ufl.edu. Jing Zhao is with Dept. of Electrical & Computer Engineering, University of Florida, Email: jingzhao@ufl.edu. Deniz Erdogmus is with Department of Computer Science and Electrical Engineering at the OGI School of Science and Engineering, Oregon Health & Science University, Email: derdogmus@ieee.org. Yuguang Fang is with Dept. of Electrical & Computer Engineering, University of Florida, Email: fang@ece.ufl.edu. Zhihai He is with Dept. of Electrical & Computer Engineering, University of Missouri, Columbia, Email: HeZhi@missouri.edu. This work was supported in part by the US National Science Foundation (NSF) under grant DBI-0529012, DBI-0529082, and ECS-0524835.

1 Introduction

The last several years see a surging interest in transmission of video over wireless networks, which leads to considerable increase in the use of mobile communication devices equipped with video cameras. Many of the mobile communication devices are small and battery operated. Therefore they possess very limited power and low computation capability, which pushes the constant need for video compression algorithms of higher computation efficiency and coding efficiency.

To achieve higher efficiency in the video coding system, intra-frame coding and inter-frame coding are employed to reduce spatial redundancy within a single frame and temporal redundancy between adjacent frames respectively. As a key component of most video compression systems, motion estimation exploits the temporal redundancy by predicting the subsequent frames from reference frames. In a typical video system, motion estimation constitutes 70% of the computation load in an encoder [1]. Therefore, it is critical for a motion estimation scheme to achieve low computational complexity for resource-constrained wireless video applications.

Among all motion estimation methods, one important category is a pixel-based approach, where motion vectors are estimated for every pixel. Then each pixel can be predicted from the previously coded reference frame based on the motion vector of each pixel. The prediction error and the motion vectors are transmitted or stored for the reconstruction of the frames.

Another major category is a block-based approach, which is also the most widely used motion estimation technique in various video compression standards. In block-based schemes, each video frame is divided into square blocks of equal size. Within each block, all the pixels are assumed to undergo the same translational motion specified by the motion vector of this block. Therefore each block can be predicted from the previously coded reference frame based on the motion vector of the block. And the resulting prediction error is coded with intra-frame coding techniques. The motion vector is estimated by searching for the best matching block within a search window centered at the corresponding block in the reference frame.

In wireless video applications, an exhaustive block match algorithm (EBMA) might be neither realistic due to its formidable computation complexity, nor cost-effective as the mo-

tion is not completely random. Many algorithms were developed to perform motion estimation with reduced computational complexity, among which are two-dimensional logarithmic (TDL) search [4], block-based gradient descent search [6], three-step search [5], a new three-step search (TSS) [8], the four-step (4SS) search [7], to name a few.

None of the block-based motion estimation algorithms mentioned above effectively utilizes the previous knowledge gained in calculating the motion vectors from one frame to the next. For each step of search, memory of the previous motion vectors is erased and initial conditions reset.

To fully utilize the information gained from the past frames for the estimation of future frames, one solution is to formulate the motion estimation problem as an adaptive filtering problem. In such a video compression system, motion vectors are modeled by an adaptive system, while the traditional approach does not attempt to model the motion vectors.

Based on this system, we present in this paper a new approach to determine the motion vector in an information-theoretic framework. The advantage of our scheme lies mainly in the extremely low computation complexity it achieves. Furthermore, since the motion vector model is to be replicated at the decoder given knowledge about the model and the initial conditions, there is no need to transmit motion vectors. Therefore it provides savings in bandwidth on top of the saving in computation.

The remainder of the paper is organized as follows. In Section 2, we introduce an adaptive motion estimation system and the maximum mutual information criterion. Section 3 presents our schemes based on the system framework and optimization criterion. The pixel-based maximum mutual information scheme (PMaxMI) and block-based maximum mutual information method (BMaxMI) will be described in Section 3.1 and Section 3.2 respectively, which is followed by the computational complexity analysis in Section 3.3. In Section 4, we present the simulation results in terms of root mean squared error (RMSE). Section 5 concludes the paper.

2 Adaptive Motion Estimation System under Maximum Mutual Information Criterion

The organization of this section is as below. We first formulate the motion estimation problem as an adaptive prediction problem in Section 2.1. Then, we introduce the general notion of mutual information and the maximum mutual information criterion in Section 2.2.

2.1 Adaptive Motion Estimation Approach

Consider the block diagram of a general adaptive prediction system in Fig. 1.

At some discrete time $n+1$, the past frame of the video sequence $f(n)$ serves as the filter input, while the present frame of the sequence $f(n+1)$ serves as the desired response. The filter produces an output, $f_p(n+1)$. This output is the best estimate of the current value $f(n+1)$ given $f(n)$ and motion vector $d(n)$. M is the known mapping function of the filter. Our object is to search for the best $d(n)$ based on the values of $f(n)$ and $f(n+1)$, so that the pre-defined cost function J of $f(n)$ and $f(n+1)$ is optimized.

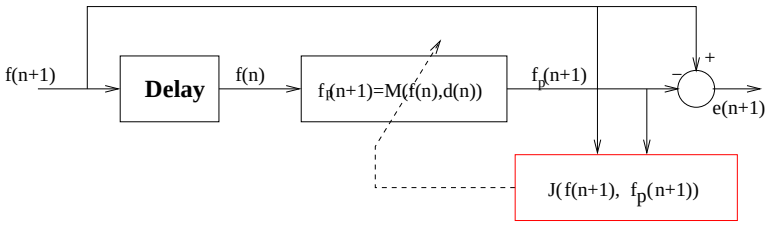

Figure 1: Adaptive Motion Estimation System Diagram

Thus the motion estimation problem can be formulated as follows:

$$f_p(n+1) = M(f(n), d(n)) \tag{1}$$

$$e(n+1) = f(n+1) - f_p(n+1) \tag{2}$$

$$f_r(n+1) = f_p(n+1) + e(n+1) \tag{3}$$

$$d(n+1) = d(n) + \eta \frac{\partial J}{\partial d(n)}, \tag{4}$$

where η is a user-defined fixed parameter (used as a step size).

In this paper, we will focus on motion estimation and assume that lossless transmission is used for error signal so that the image can be reconstructed perfectly. In reality, due to bandwidth restriction, the error signal is quantized and it leads to error accumulation. This problem can be solved by introducing intra frames periodically.

Given the system formulation in Equation (1), if the cost function takes the form of mean square of the error signal e (MSE), we can use the popular least mean square (LMS) algorithm to solve the problem. If the cost function takes the form of the entropy of the error signal e, we can use the Minimum Error Entropy (MEE) method proposed in [10]. Motivated by all these works, we used the maximum mutual information criterion to develop our pixel-wise and block-wise motion estimation schemes. The intuition behind using the maximum mutual information criterion is that mutual information is a good measurement of the discrepancy or dependency between two data sources. In our application scenario, when the mutual information between the original frame and the predicted frame is maximized, the predicted frame $f_p(n+1)$ preserves the most information of the real frame $f(n+1)$.

However, there are no analytical methods to calculate mutual information without presuming knowledge of prior probability density function (pdf). Therefore we use a non-parametric pdf estimator with Parzen Windowing for the estimation of mutual information. This combination yields an estimator simple to compute without imposing any assumptions about the pdf of the data. Thus the method can manipulate mutual information as straightforwardly as the mean square error (MSE) or error entropy. Next, we will introduce the maximum mutual information criterion.

2.2 Maximum Mutual Information Criterion

This section is organized as follows. First, a brief review of non-parametric pdf estimator with Parzen windowing is given in Section 2.2.1. It is followed by the definition of mutual information as the cost function for the motion estimation system in Section 2.2.2, which will facilitate the derivation of motion estimation schemes in Section 3.

2.2.1 Non-parametric pdf estimator

Given N pairs of samples for random variables x and y, the pdf of the 2-D random vector $z = [x, y]^T$ can be approximated by Parzen windowing estimation with a two-dimensional Gaussian kernel of mean zero and variance matrix Σ,

$$p_{X,Y}(x, y) = \frac{1}{N} \sum_{i=1}^{N} \kappa_{\Sigma}(x - x_i, y - y_i) \tag{5}$$

where κ_{Σ} is the Gaussian kernel in Parzen windowing and Σ represents the size of the kernel.

Similarly, the marginal pdf of random variable x, y can be approximated with the sum of a one-dimensional Gaussian kernel located on the samples of x and y as

$$p_X(x) = \frac{1}{N} \sum_{i=1}^{N} \kappa_{\sigma_X^2}(x - x_i) \tag{6}$$

and

$$p_Y(y) = \frac{1}{N} \sum_{i=1}^{N} \kappa_{\sigma_Y^2}(y - y_i) \tag{7}$$

where $\sigma_X^2 = \Sigma_{11}$ and $\sigma_Y^2 = \Sigma_{22}$ are the kernel variance of x and y respectively. We will use $\Sigma' = \Sigma^{-1}$ to represent the inversion of Σ.

The pdf estimators above are solely based on the data without assuming any *a priori* knowledge of the distribution of the data obtained. They are to be used in the development of the mutual information estimator in next section.

2.2.2 Mutual Information and its Non-parametric Estimator

In the signal processing area, mutual information is often used as a measurement of the similarity and dependence between different data sources.

For two random variable x and y, the mutual information is defined as

$$I_s(X; Y) = -\int\int p_{X,Y}(x, y) log \frac{p_X(x) p_Y(y)}{p_{X,Y}(x, y)} dxdy \tag{8}$$

where $p_{X,Y}(x,y)$ is the joint pdf of x, y, and $p_X(x)$ and $p_Y(y)$ are the marginal PDF's of x, y respectively. Equation (8) can be simplified as

$$I_s(X;Y) = -\mathbf{E}(log\frac{p_X(x)p_Y(y)}{p_{X,Y}(x,y)}) \tag{9}$$

where the expectation $\mathbf{E}$ is with respect to $p_{X,Y}(x,y)$.

In [2], a stochastic gradient estimator was developed by employing the standard complexity reduction techniques to Renyi's Entropy by considering only part of the data set. By applying the same technique, when only 1 pair of subsequent samples of $(x,y)^T$ at the instants $n+1$ and n are available, a non-parametric estimator for mutual information is obtained

$$I_s(x,y) \approx \frac{1}{N}\sum_{n=1}^{N} \log \frac{\kappa_\Sigma(x_{n+1}-x_n, y_{n+1}-y_n)}{\kappa_{\Sigma_{11}}(x_{n+1}-x_n)\kappa_{\Sigma_{22}}(y_{n+1}-y_n)} \tag{10}$$

Thus by exploiting the Parzen windowing technique, the estimator for mutual information solely based on the data without assuming any *a prior* knowledge of the distribution of the data is obtained. In the next section, a stochastic gradient estimator is developed by applying the complexity reduction techniques in [2, 3].

3 Schemes Based on Maximum Mutual Information Criterion

Depending on the scale we are looking at, we can apply this technique of simple-to-calculate entropy estimator on the pixel level, thus generating the Pixel-based Maximum Mutual Information(PMaxMI) scheme, or on the block level, thus generating the Block-based Maximum Mutual Information (BMaxMI) scheme, which will be discussed in Section 3.1 and Section 3.2 respectively. The complexity analysis will be presented in Section 3.3.

3.1 Pixel-based Maximum Mutual Information Scheme (PMaxMI)

Considering the motion estimation problem on pixel level, the problem statement can be specified as follows:

Let $f(p,n)$ denote the image intensity at spatio-temporal position (p,n), where $p=[x,y]$ is the pixel location in 2-dimensional space, n is the time index. Given two subsequent frames $f(n)$ and $f(n+1)$, a motion vector (MV) $d(p,n)=[d_x(p,n),d_y(p,n)]$ is defined for each pixel as the 2-D vector field that maps the point in $f(n)$ onto their corresponding location in $f(n+1)$. Our goal is to find an estimate of d for each pixel based on values of $f(n)$ and $f(n+1)$, so that the pre-defined cost function J, the mutual information between the predicted frame and the real frame in this case, is maximized. The modified system diagram is shown below.

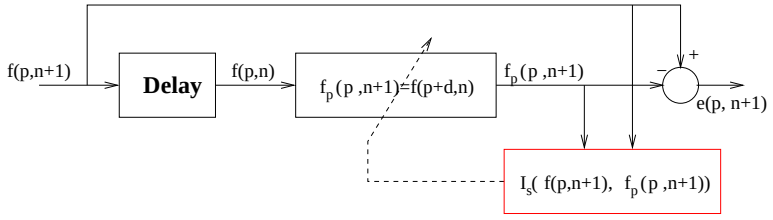

Figure 2: Pixel-based Maximum Mutual Information Motion Estimation System Diagram

The motion estimation system described above can be modified as as follows

$$f_p(p,n+1)=f(p+d(p,n),n) \tag{11}$$

$$e(p,n+1)=f(p,n+1)-f_p(p,n+1) \tag{12}$$

$$e'(p,n+1)=Q[e(p,n+1)] \tag{13}$$

$$f_r(p,n+1)=f_p(p,n+1)+e'(p,n+1) \tag{14}$$

$$d(p,n+1)=d(p,n)+\eta\frac{\partial J}{\partial d(p,n)} \tag{15}$$

where $Q[\cdot]$ in (13) is a quantization function.

As shown in Fig. 2, the filter input is the pixel intensity of the most recent frame, $f(p,n)$. For the adaptive filter, the desired output is the pixel intensity of the present frame

$f(p, n+1)$. At some discrete time, n+1, the output of the filter, $f_p(p, n+1)$, is the estimate of the $f(p, n+1)$ given its most recent values $f(p, n)$. The estimation error, $e(p, n+1)$ is defined as the difference between the filter output $f_p(p, n+1)$, and the desired output $f(p, n+1)$, which will be transmitted and used for the reconstruction of the frame in the decoder side. The cost function $I(f(p, n+1), f_p(p, n+1))$ is the mutual information between the predicted intensity $f_p(p, n+1)$ and real intensity $f(p, n+1)$ for each pixel,

$$J(f_p(p, n), f(p, n)) = I_s(f_p(p, n), f(p, n)) \tag{16}$$

which can be estimated as Equation (10).

Suppose we are given only the current sample $f(p, n+1)$ and the previous sample $f(p, n)$, by applying the same technique as in [2], a non-parametric stochastic estimator for mutual information I_s is obtained in the following step:

For the current sample $n+1$, letting $x(n+1) = f_p(p, n+1)$ and $y(n+1) = f_p(p, n+1)$ in Equation (11), we obtain the cost function in Fig. 2 based on the most recent frames at n and $n+1$ only, i.e.,

$$\begin{aligned} &J(f_p(p, n+1), f(p, n+1)) \\ &= \log \frac{\kappa_\Sigma(f_p(p, n+1) - f_p(p, n), f(p, n+1) - f(p, n))}{\kappa_{\sigma^2_{f_p}}(f_p(p, n+1) - f_p(p, n))\kappa_{\sigma^2_f}(f(p, n+1) - f(p, n))} \end{aligned} \tag{17}$$

where $f_p(p, n+1)$ can be obtained from Equation (1) based on the previous frame $f(p, n)$ and the motion vector $d(p, n)$.

The parameter to determine is $d(p, n)$. So the gradient of this expression with respect to $d(p, n)$ is needed for updating. Note that $d(p, n-1)$ is a known constant at this time, since it is the displacement from the previous frame.

To simplify the representation of the following discussion, let

$$\sigma^2_{f_p} = \Sigma_{11} \tag{18}$$

$$\sigma^2_f = \Sigma_{22} \tag{19}$$

$$K_1 = \kappa_\Sigma(f_p(p, n) - f_p(p, n-1), f(p, n) - f(p, n-1)) \tag{20}$$

$$K_2 = \kappa_{\Sigma_{11}}(f_p(p,n) - f_p(p,n-1)) \tag{21}$$

$$K_3 = \kappa_{\Sigma_{22}}(f(p,n) - f(p,n-1)). \tag{22}$$

Then, Equation (17) is simplified as

$$J(f_p(p,n+1), f(p,n+1)) = \log K_1 - \log K_2 - \log K_3 \tag{23}$$

Among the three terms, only K_1 and K_2 depend on $d(p,n)$. Thus, the partial derivatives of K_1 and K_2 with respect to $d(p,n)$ are obtained as

$$\frac{\partial J}{\partial d(p,n)} = \frac{\frac{\partial K_1}{\partial d(p,n)}}{K_1} - \frac{\frac{\partial K_2}{\partial d(p,n)}}{K_2} \tag{24}$$

where

$$\begin{array}{l} \dfrac{\partial K_1}{\partial d(p,n)} = \dfrac{\partial \kappa_\Sigma(f_p(p,n) - f_p(p,n-1), f(p,n) - f(p,n-1))}{\partial d(p,n)} = \\ -\kappa_\Sigma(f_p(p,n) - f_p(p,n-1), f(p,n) - f(p,n-1))\times \\ ((\Sigma_{12}^{-1} + \Sigma_{21}^{-1})(f(p,n+1) - f(p,n)) + \\ 2\Sigma_{22}^{-1}(f(p + d(p,n), n) - f(p + d(p,n-1), n-1)))\times \\ (f(p + d(p,n), n) - f(p + d(p,n-1), n-1))\times \\ \begin{bmatrix} (f(p+e_1,n) - f(p-e_2,n))/2 \\ (f(p+e_1,n) - f(p-e_2,n))/2 \end{bmatrix} \end{array} \tag{25}$$

and

$$\begin{array}{l} \dfrac{\partial K_2}{\partial d(p,n)} = \dfrac{\partial \kappa_{\Sigma_{11}}(f_p(p,n) - f_p(p,n-1))}{\partial d(p,n)} \\ = -\dfrac{1}{\sigma_{f_p}^2}\kappa_{\sigma_{f_p}}(f_p(p,n) - f_p(p,n-1))\times \\ (f(p + d(p,n), n) - f(p + d(p,n-1), n-1))\times \\ \begin{bmatrix} (f(p+e_1,n) - f(p-e_2,n))/2 \\ (f(p+e_1,n) - f(p-e_2,n))/2 \end{bmatrix} \end{array} \tag{26}$$

which leads to a simple expression of the stochastic gradient:

$$\begin{array}{l} \dfrac{\partial J}{\partial d(p,n)} = \\ (-\Sigma_{11}^{-1}(f(p + d(p,n), n) - f(p + d(p,n-1), n-1)) - \\ ((\Sigma_{12}^{-1} + \Sigma_{21}^{-1})(f(p,n+1) - f(p,n)) + \\ 2\Sigma_{22}^{-1}(f(p + d(p,n), n) - f(p + d(p,n-1), n-1)))\times \\ (f(p + d(p,n)) - f(p + d(p,n-1), n-1)))\times \\ \begin{bmatrix} (f(p+e_1,n) - f(p-e_2,n))/2 \\ (f(p+e_1,n) - f(p-e_2,n))/2 \end{bmatrix} \end{array} \tag{27}$$

where $e_1 = [1, 0]^T$, $e_2 = [0, 1]^T$.

By far we have obtained a stochastic gradient estimator for the mutual information with respect to a motion vector by following a methodology similar to the LMS algorithm and the minimum error entropy algorithm [2] and [10]. By substituting the gradient obtained from Equation (27) in Equations (1) – (4), and imposing a smoothness constraint that the neighboring motion vectors cannot differ by more than a pre-determined threshold, we have defined completely Pixel-based Maximum Mutual Information (PMaxMI) motion estimation scheme.

3.2 Block-based Maximum Mutual Information Scheme (BMaxMI)

Considering the motion estimation problem on block level, the problem statement can be specified as follows:

Given an image which can be divided into B square blocks of S pixels each. The intensity of each pixel in this image is denoted uniquely with $f(p_s^b, n)$. The spatio-temporal position of one pixel is denoted with (p_s^b, n), where b is the index of the block, s is the index of the pixel within the bth block, and n is the time index. Given two successive frames $f(n+1)$ and $f(n)$, a motion vector $d^b(n) = [dx, dy]^T$ is defined for the bth block as the 2-D vector field that maps the blocks in $f(n)$ onto their corresponding location in $f(n+1)$. Our goal is to find an estimate $\hat{d}^b$ for each block based on values of $f(n+1)$ and $f(n)$, so that some pre-defined object function J is optimized.

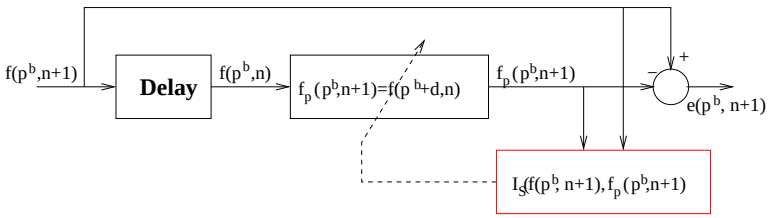

Figure 3: Block-based Maximum Mutual Information Motion Estimation System Diagram

The block diagram of a general adaptive prediction system is shown in Fig. 3. For the bth block, the filter input is the intensity function of the most recent frame, $f(p_s^b, n)$. The desired output of the adaptive filter is the intensity function of the present frame $f(p_s^b, n+1)$. At some discrete time, n+1, the output of the filter, $f_p(p_s^b, n+1)$, is the estimation, or prediction

of the $f(p_s^b, n+1)$ given its most recent values $f(p_s^b, n)$. The estimation error, $e(p_s^b, n+1)$ is defined as the difference between the filter output $f_p(p_s^b, n+1)$, and the desired output $f(p_s^b, n+1)$. $J(f_p(p_s^b, n+1), f(p_s^b, n+1))$ is the optimization criterion, which is a function of $f_p(p_s^b, n+1)$ and $f(p_s^b, n+1)$.

For spatio-temporal position (p_s^b, n), the motion estimation problem can be expressed as follows

$$J(f_p(p_s^b, n+1), f(p_s^b, n+1)) = I_s(f_p(p_s^b, n+1), f(p_s^b, n+1)) \tag{28}$$

$$f_p(p_s^b, n+1) = f(p_s^b + d^b(n), n) \tag{29}$$

$$e(p_s^b, n+1) = f(p_s^b, n+1) - f_p(p_s^b, n+1) \tag{30}$$

$$e'(p_s^b, n+1) = Q[e(p_s^b, n+1)] \tag{31}$$

$$f_r(p_s^b, n+1) = f_p(p_s^b, n+1) + e'(p_s^b, n+1) \tag{32}$$

$$d^b(n+1) = d^b(n) + \eta \frac{\partial J}{\partial d^b(n)} \tag{33}$$

where $Q[\cdot]$ in (31) is a quantization function.

Similarly, a stochastic gradient estimator must be developed in order to apply the maximum mutual information criterion in our problem. By utilizing the same complexity reduction techniques to mutual information entropy of N samples of random variable $[x, y]^T$, a non-parametric stochastic estimator for mutual information is obtained:

Substituting $f_p(p_s^b, n+1)$ and $f(p_s^b, n+1)$ for x and y respectively and the block size S for sample number N in Equation (10), we obtain the cost function for the bth block based on the most recent frames only,

$$I_s(f_p(p_s^b, n+1), f(p_s^b, n+1)) \tag{34}$$

$$\approx \frac{1}{S} \sum_{s=1}^{S} \log \frac{\kappa_\Sigma(f(p_s^b, n) - f(p_s^b, n-1), f_p(p_s^b, n) - f_p(p_s^b, n-1))}{\kappa_{\Sigma_{11}}(f(p_s^b, n) - f(p_s^b, n-1))\kappa_{\Sigma_{22}}(f_p(p_s^b, n) - f_p(p_s^b, n-1))} \tag{35}$$

Thus, we can obtained the derivative of the cost function J with respect to the motion vector d for the bth block following this steps. Let

$$K_{1,s} = \kappa_{\Sigma} \begin{bmatrix} f(p_s^b, n+1) - f(p_s^b, n) \\ f(p_s^b + d^b(n), n) - f(p_s^b + d^b(n-1), n-1) \end{bmatrix} \tag{36}$$

$$K_{2,s} = \kappa_{\Sigma_{22}}(f(p_s^b + d^b(n), n) - f(p_s^b + d^b(n-1), n-1)) \tag{37}$$

$$K_{3,s} = \kappa_{\Sigma_{11}}(f(p_s^b, n) - f(p_s^b, n-1)) \tag{38}$$

Therefore,

$$\begin{aligned} & I_s(f_p(p^b, n+1), f(p^b, n+1)) \\ & = \frac{1}{S}(\sum_{s=1}^{S} \log K_{1,s} - \sum_{s=1}^{S} \log K_{2,s} - \sum_{s=1}^{S} \log K_{3,s}) \end{aligned} \tag{39}$$

Similarly, among the three terms, $K_{3,s}$ does not depend on d.

$$\begin{aligned} \frac{\partial log K_{1,s}}{\partial d} = & \frac{-1}{K_{1,s}}((\sigma_{12}^{-1} + \sigma_{21}^{-1})(f(p_s^b, n+1) - f(p_s^b, n)) + 2\sigma_{22}^{-1} \\ & \times (f(p_s^b + d^b(n), n) - f(p_s^b + d^b(n-1), n-1))) \\ & \times \kappa_{\Sigma}(f(p_s^b + d^b(n), n) - f(p_s^b + d^b(n-1), n-1)) \\ & \times \frac{1}{2} \times \begin{bmatrix} (f(p_s^b, n+1) - f(p_s^b, n)) \\ (f(p_s^b + d^b(n), n) - f(p_s^b + d^b(n-1), n-1)) \end{bmatrix} \end{aligned} \tag{40}$$

and

$$\begin{aligned} \frac{\partial log K_{2,s}}{\partial d} = & \frac{-1}{K_{2,s}} \frac{1}{\Sigma_{22}} \\ & \times (f(p_s^b + d^b(n), n) - f(p_s^b + d^b(n-1), n-1)) \\ & \times \kappa_{\sigma_{22}^2}(f(p_s^b + d^b(n), n) - f(p_s^b + d^b(n-1), n-1)) \\ & \times \frac{1}{2} \times \begin{bmatrix} f(p_s^b + e_1, n) - f(p_s^b - e_1, n) \\ f(p_s^b + e_2, n) - f(p_s^b - e_2, n) \end{bmatrix} \end{aligned} \tag{41}$$

where $e_1 = [1, 0]^T$, $e_2 = [0, 1]^T$.

Therefore, the stochastic gradient to be used for the update of the motion vector is

$$\begin{aligned} &\frac{\partial I_s(f_p(p_s^b, n+1), f(p_s^b, n+1))}{\partial d(p_s^b, n)} \\ &= \frac{1}{S}\sum_{s=1}^{S} \frac{\partial K_{1,s}}{\partial d(p_s^b, n)} \frac{1}{K_{1,s}} - \frac{1}{S}\sum_{s=1}^{S} \frac{\partial K_{2,s}}{\partial d(p_s^b, n)} \frac{1}{K_{2,s}} \end{aligned} \tag{42}$$

The result can be obtained by substituting $\frac{\partial K_{1,s}}{\partial d(p_s^b,n)}$ and $\frac{\partial K_{2,s}}{\partial d(p_s^b,n)}$ with Equations (40) and (41).

So far we have obtained an stochastic gradient estimator for the cost function J with respect to motion vector d^b for each block. By substituting Equation (42) in Equations (28) – (33), and imposing a smoothness constraint that the neighboring motion vectors cannot differ by more than a pre-determined threshold, we obtain the Pixel-based Minimum Error Entropy (BMaxMI) motion estimation scheme.

3.3 Computation Complexity Analysis

Assuming the image size is $M \times M$, with a search range of $R \times R$ and a block size of $B \times B$ we proceed to calculate the number of operations required per pixel.

Here we assume that the above operations can be executed in a single instruction cycle. This is justified because the current DSP and FPGA technology can perform all the operations, except the exponential operation, in a single cycle. The exponential operation can be efficiently implemented using an 8-bit table lookup which also can be executed in a single instruction cycle.

Table 1 summarizes the number of additions, subtractions, exponentials, absolute values, multiplications and conditionals (that is, 'if' statements) required by the algorithm at the encoder [5, 6, 9].

A snapshot of Table 1 is provided in Table 2 for illustration for the case when $R = 16$ and $B = 3$, assuming all the above operations are executed in single instruction cycle.

The results in Table 1 and Table 2 show that compared to the traditional methods, our algorithms have extremely low computational complexity on the encoder side. It is nearly 256 times faster than EBMA, 18 times faster than Gradient descent and nearly 8 times faster than

Table 1: Number of instructions per pixel in encoder.

Operations per pixel	Pixel-based MaxMI	Block-based MaxMI	EBMA Search
Additions	4	4	$(2R+1)^2$
Subtractions	5	5	$(2R+1)^2$
Exponentials	0	0	0
Multiplication	4	4	0
Division	0	0	0
Conditionals	4	$\frac{4}{B^2}$	$(2R+1)^2$
Total	17	$13+\frac{4}{B^2}$	$3(2R+1)^2$

Table 1. Continued

Operations per pixel	Block-Based Gradient Descent	3-Step Search Search
Additions	$B^2+(R-2)(2B-1)$	$8(\log_2(R/2)+1)+1$
Subtractions	$B^2+(R-2)(2B-1)$	$8(\log_2(R/2)+1)+1$
Exponentials	0	0
Multiplication	0	0
Division	0	0
Conditionals	$B^2+(R-2)(2B-1)$	$8(\log_2(R/2)+1)+1$
Total	$3(B^2+(R-2)(2B-1))$	$3(8(\log_2(R/2)+1)+1)$

Table 2: Number of instructions per pixel for $R=16$ and $B=3$ in encoder.

Method	Number of Operations in Encoder
Pixel-based MaxMI	17
Block-based MaxMI	$13+\frac{4}{9}$
EBMA search	4356
Block-based Gradient Descent	316
Three Step Search	132

the 3-step search (TSS). The block-based MaxMI algorithm and the pixel-based algorithm have almost the same complexity. Due to this characteristic, the algorithms have great potential for the application where devices on the encoder side is of limited computational capability, for example, wireless video sensors, and mobile handsets.

Table 3 summarizes and compares the decoding complexity of our schemes and the traditional methods.

Table 3: Number of instructions per pixel in decoder.

Operations per pixel	PMaxMI	BMaxMI	EBMA	Block-Based Gradient Descent	3-Step Search
Additions	4	4	2	2	2
Subtractions	5	5	0	0	0
Exponentials	0	0	0	0	0
Multiplication	4	4	0	0	0
Division	0	0	0	0	0
Conditionals	4	$\frac{4}{B^2}$	0	0	0
Total	17	13.4	2	2	2

A snapshot of Table 3 is provided in Table 4 for illustration for the case when $R = 16$ and $B = 3$, assuming all the above operations are executed in single instruction cycle.

Table 4: Number of instructions per pixel for $R = 16$ and $B = 3$ in decoder.

Method	Number of Operations in Decoder
Pixel-based MaxMI	17
Block-based MaxMI	$13 + \frac{4}{9}$
EBMA search	2
Block-based Gradient Descent	2
Three Step Search	2

The results in Table 3 and Table 4 show that our algorithms also have moderately higher computation complexity on the decoder side, about 8.5 times more complexity than other schemes which have only 2 operations per pixel. Because in our motion estimation system, the encoder and the decoder have the same configuration, and they carried out the same computation process, therefore decoder does not require a separate hardware for the decoder. Due to this reason, it is of the same complexity as its corresponding encoder. However, considering the application scenario, the devices on the decoder side (i.e., sink nodes

in the wireless sensor network or base stations in the ordinary wireless network), generally possesses higher computation capability and more power resource. The increased computation complexity compared to the traditional schemes will not pose a serious burden on the system.

4 Simulation

In this section, we implement our adaptive motion estimation algorithms based on the maximum mutual information criterion as described in Section 3.1 and Section 3.2. We choose the luminance component of several video sequences in QCIF format for the encoding process. For EBMA, a block size of 8×8 is chosen with integer-pel accuracy. The search range is 16×16 pixels. The block-based gradient descent search algorithm is implemented as described in [6] with a block size of 3×3 and a search range of 16×16 pixels with integer-pel accuracy. For the three-step algorithm [7], we use a block size of 8×8 and a search range of 16×16 pixels with integer-pel accuracy. The mean absolute error (MAE) distortion function is used as the block distortion measure for the two algorithms. Since we focus on the study of motion estimation, hence DCT, quantization and entropy coding are excluded in the simulation.

In each algorithm, motion is estimated and compensated using perfectly reconstructed reference frames. The first frame is intra-coded and the rest are inter-coded. The experiment is conducted using frame rates of 10, 5 and 2, respectively. The values of root mean squared error (RMSE) for the four different QCIF sequences are shown in Tables 5, 6, and 7. Note that the mutual-information-based coding algorithm does not necessarily minimize the RMSE, since the optimality criterion for determining the motion vectors is maximum mutual information. Therefore, it is natural that the least-squares based competing methods result in smaller RMSE error levels.

For low bit rate applications, the typical frame rate is usually 10 frames/sec or lower. As frame rate decreases, the temporal correlation between two consecutive video frames decreases. The more the skip rate, the smaller is the probability of finding the true motion vector. From Tables 5, 6, and 7, we notice that there is a 3 dB difference in Y-PSNR values between our algorithm and the three-step search. However, our scheme saves the bit budget for motion vectors, which usually constitutes about 50% of the total budget for

Table 5: Root mean square error for four test video sequences at 10 fps.

Method	Miss America	Coastguard	Suzie	Foreman
EBMA search	2.88	9.29	4.92	8.16
3-Step search	4.06	12.14	9.57	16.24
Block-based Gradient Descent	6.78	14.27	16.28	23.69
Pixel-based MaxMI	6.29	20.36	11.49	20.52
Block-based MaxMI	6.37	22.13	13.28	20.42

Table 6: Root mean square error for four test video sequences at 5 fps.

Method	Miss America	Coastguard	Suzie	Foreman
EBMA search	3.16	11.18	6.35	11.00
3-Step search	5.28	11.94	12.93	21.96
Block-based Gradient Descent	8.51	18.03	19.32	28.55
Pixel-based MaxMI	8.89	23.25	18.28	29.24
Block-based MaxMI	8.99	23.71	20.08	29.48

Table 7: Root mean square error for four test video sequences at 2 fps.

Method	Miss America	Coastguard	Suzie	Foreman
EBMA search	3.63	14.29	8.69	17.91
3-Step search	9.71	23.39	17.99	31.92
Block-based Gradient Descent	12.40	22.18	24.10	37.76
Pixel-based MaxMI	12.06	29.52	25.32	39.39
Block-based MaxMI	17.42	29.78	21.12	37.26

low bit-rate video applications. Therefore, the 3 dB performance loss can be compensated by the bandwidth savings due to not transmitting motion vectors in our scheme and the adaptive motion estimation provides a trade-off between computational complexity and video presentation quality.

5 Conclusion

In this paper, we consider the motion estimation problem in video encoding. Existing motion estimation techniques do not effectively utilize the past knowledge in motion prediction, leading to inefficiency in computation. To address this problem, we proposed adaptive model-based motion estimation algorithms using mutual information. In our schemes, the motion vectors of the current frame are iteratively computed from the previous frame, based on a model. This leads to computational savings because of the knowledge gained in the computation of the previous motion vectors. Our results showed that our scheme significantly reduces the computational complexity, as compared to the existing algorithms.

The salient feature of our adaptive motion estimation algorithm is its very low computational complexity. Hence, our algorithm is ideally suited for wireless video sensor networks, in which computational complexity and energy consumption impose major constraints.

References

[1] M. E. Al-Mualla, C. N. Canagarajah, D. R. Bull, *Video Coding for Mobile Communications: Efficiency, Complexity and Resilience, Signal Processing and Its Applications,* Academic Press, San Diego, USA, 2002.

[2] D. Erdogmus, J. Principe, "An error-entropy minimization algorithm for supervised training of nonlinear adaptive systems," *IEEE Transactions on Signal Processing*, vol. 50, no. 7, pp. 1780–1786, July 2002.

[3] X. Feng, K. A. Loparo and Y. Fang, "Optimal State Estimation with Active Probing for Stochastic Systems: An Information Theoretic Approach," IEEE Transaction on Automatic Control, vol. 42, no. 6, pp. 771–785, 1997.

[4] J. R. Jain, A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Transactions on Commun*, vol. 29, no. 12, pp. 1799–1808, December 1981.

[5] T. Koga, K. Linuma, A. Hirano, Y. Iijima, T. Ishiguro, "Motion-compensated interframe coding for video conferencing," *Proceeding of National Telesystems Conference 81*, pp.G5.3.1–5, New Orleans, USA, December 1981.

[6] L. L. Kuo, E. 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. 4, pp. 419–422, 1996.

[7] M. P. Lai, W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 313–317, 1996.

[8] R. Li, B. Zeng and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 438–442, August 1994.

[9] K. M. Nam, J. S. Kim, R. H. Park, Y. S. Shim, "A fast hierarchical motion vector estimation algorithm using mean pyramid," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 4, pp. 344–351, 1995.

[10] G. Ramachandran, V. Krishnan, D. Wu, Z. He, "A model-based adaptive motion estimation scheme using Renyis entropy for wireless video," *Journal of Visual Communication and Image Representation*, vol. 16, no. 4–5, pp. 432–449, Elsevier, August–October 2005.

[11] Y. Wang, J. Ostermann, Y. Zhang, *Video Processing and Communications*. Prentice Hall Inc., Upper Saddle River, USA, 2001.

[12] J. Yeh, M. Khansari, M. Vetterli, "Motion compensation of motion vectors," *Proceeding of IEEE International Conference on Image Processing*, vol. 1, pp. 574–577, 1995.