

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

An Enhanced Markov-chain-modeled Motion Estimation Algorithm by using Support Vector Machine

¹Zhijie Zhao, ^{2,3}Zhimin Cao, ²Maoliu Lin and ¹Xuesong Jin

¹School of Computer and Information Engineering, Harbin University of Commerce, Harbin, China

²School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, China

³School of Electronics Science, North-East Petroleum University, Daqing, China

Abstract: In this study, for improving the performance of Motion Estimation (ME), a new Markov-Chain-Model based motion estimation method is proposed with a multi-class Support Vector Machine (SVM) classification network constructed to get one-step transition probabilities among initial prediction states used in this algorithm. Experimental results show that it is efficient to use the constructed SVM network in the Markov-Chain-Modeled Motion Estimation algorithm to various tested sequences.

Key words: Motion estimation, support vector machine, markov-chain-model

INTRODUCTION

The digital multimedia coding, especially the digital video coding, has been extensively studied. It is well known that there is much of temporal-spatial redundancy existing in video sequence. To exploit the temporal-spatial redundancy, many kinds of block-matching ME algorithms have been widely developed. All of these algorithms are based on the idea that subdividing video frames into square Macro Blocks (MB) and then finding the Motion Vector (MV) for each block within a search range that minimizes the error criteria. The full search algorithm (FS) is a kind of exhaustive block matching method. It searches all of the possible points in the search range. However, this algorithm is terribly time consuming. In order to improve the computational efficiency, lots of fast block matching algorithms have been proposed. These algorithms include: Three step search algorithm (TSS) (Koga *et al.*, 1981), new three step search algorithm (NTSS) (Li *et al.*, 1994), four step search algorithm (FSS) (Po and Ma, 1996), two dimensional logarithmic search algorithm (2D-LOG) (Jain and Jain, 1981), diamond search algorithm (DS) (Zhu and Ma, 2000), Hexagon-Based search algorithm (HEXBS) (Zhu *et al.*, 2002). In addition, algorithms based on spatial or temporal correlations such as MVFAST (Hosur and Ma, 1998, 1999, 2000), PMVFAST (Tourapis *et al.*, 2000, 2001) as well as multi-model based algorithms such as UMHExagonS used in H.264/AVC and method proposed in (Chen *et al.*, 2006).

Among all of the algorithms mentioned above, there are two major deficiencies. Firstly, most of them exploit redundancy by using statistical properties of motion vector without considering the temporal redundancy. Secondly, they can hardly achieve satisfactory results among different video sequences with different motion activities.

In order to deal with these two problems to a certain extent, a fast Markov-Chain-Modeled ME method was proposed in Zhao *et al.* (2010). In that algorithm, the predicting mode of the starting point was modeled as a Markov Chain Model to exploit the temporal redundancy. But the transition probabilities were obtained in a statistical way. Therefore, there will be some increment of the computational cost.

In this study, the fast Markov-chain-modeled ME algorithm (MCMME) is optimized by adopting Support Vector Machine (SVM-MCMME). In the optimized algorithm, a classification network which is composed of many different SVMs is constructed to predict the states transition probabilities of the Markov-chain-model's one-step transition matrix.

MCMME MOTION ESTIMATION METHOD

Early elimination of search: For consideration of the continuity of the video sequence, it is reasonable to believe that a large number of MBs are static (the corresponding MV is zero) or quasi-static (the corresponding MV approximate to zero). Furthermore, statistical results had shown that MVs are mainly

scattered in the region of two pixel distance around the original point (Cheung and Po, 2002), especially the case of video sequence with low-motion activity. Therefore, if the current MB is static or quasi-static, search process should be terminated immediately before doing the model searching. It means that the corresponding criteria value should be calculated first to see whether or not it satisfies the static condition, e.g., $SAD < 512$ in MVFAST. If so, the predicted MV is set to $(0, 0)$. In some other algorithms, such as PMVFAST, the first checked point is the median of three MVs spatially adjacent to the current MB (top, top-right and left) which is called PMV here. If the PMV does not satisfy the stop criteria, a set of other MVs ($(0, 0)$, MV of left MB (MVL), MV of top MB (MVT), MV of top-right MV (MVTR) and *et al.*) are then tested before model searching.

To sum up, before model searching, a set of MVs (PRESET) are tested to decide whether stop or not. If none of the PRESET MVs satisfies the stopping conditions, a starting point correlated to motion activity for model searching should be decided.

Details of MCMME: The existing predicted ME algorithms are mainly based on the zero-centered or median-bias characteristic of motion vectors (Tourapis *et al.*, 2000). A natural question is that whether or not the original point $(0, 0)$ or the PMV can always being the best candidate MV for the first evaluation. To reveal the truth, the result of the subtraction of two continuous motion vector fields (MVF) is introduced here. Two continuous MVF and the corresponding MVDF of foreman.qcif are illustrated in Fig. 1.

From Fig. 1, it can be seen that there are more zero points in MVDF than MVF. Obviously, if the REFMV which is corresponding to the MB at the same spatial position of the current MB in reference frame is chose as the first tested MV, especially for low-activity video sequence, it will be more effective.

Traditionally, all of the PRESET MVs are tested to find whether or not the one that have minimum criteria value satisfy the stopping condition. Some methods (e.g., PMVFAST) test PRESET MVs in a fixed order. It is obvious that the majority for PRESET MVs testing mentioned above can not use the temporal correlation efficiently. In order to exploit temporal correlation existing in the test sequence more deeply, the test order of PRESET MVs should be rearranged according to the distribution feature of MVs or something else.

Base on this consideration, a Markov Chain Model was used to predict the test order. The PRESET are composed of $(0, 0)$, PMV, MVT, MVL, MVTR and REFMV. Let:

$$\Omega = \{(mvx_i, mvy_i) | i = 1, 2, 3, 4, 5, 6\} \quad (1)$$

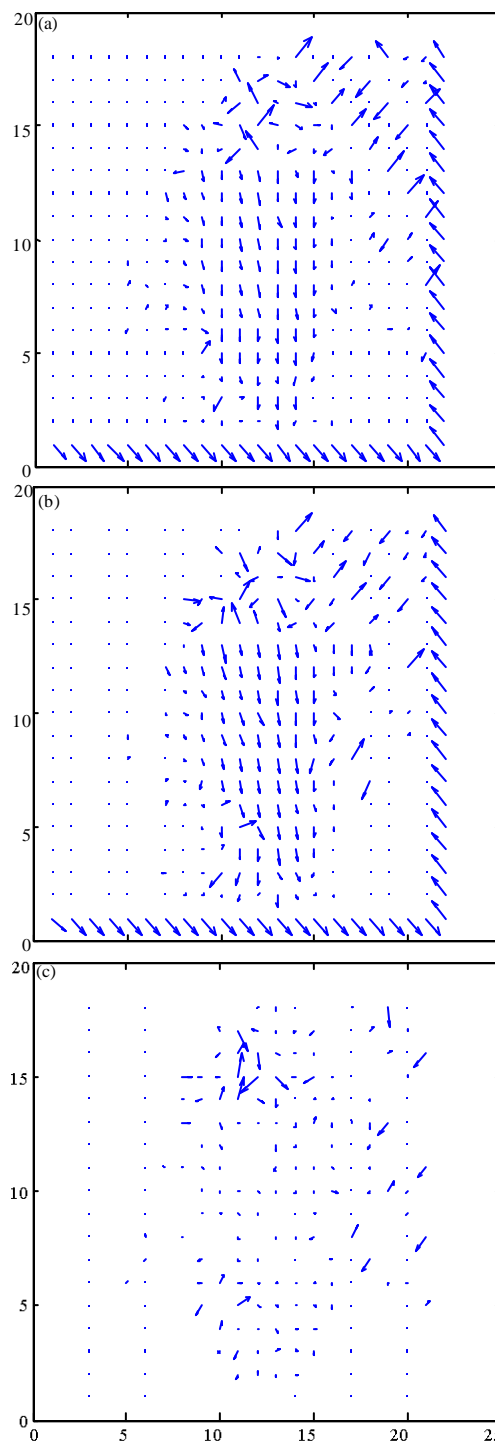


Fig. 1(a-c):(a) and (b) are the two continuous MVFs (MVF12, MVF23) got from foreman.qcif test sequence and (c) The corresponding MVDF

where, mvx_i , mvy_i denote the horizontal and the vertical components of the i th MV in the PRESET,

respectively. These six predicting MVs can be seen as six states. The criteria to determine the current state is:

$$(mvx_{pre}, mvy_{pre}) = \arg \min_i (SAD(mvx_i, mvy_i)) \quad (2)$$

where, mvx_{pre} , mvy_{pre} denote the horizontal and vertical component of the predicted MV, respectively.

To consider the strong correlation among PMV, MVL, MVT and MVTR and to simplify the model, choosing (0, 0), REF MV as well as PMV to compose a state space as below:

$$S_M = \{S1, S2, S3\} \quad (3)$$

where S_M is the state space. S_1, S_2, S_3 denote REF MV predicting mode (temporal mode), original and PMV predicting mode (spatial mode), respectively.

Therefore, the simplified PRESET will be:

$$\Omega = \{(mvx_i, mvy_i) | i = 1, 2, 3\} \quad (4)$$

where, $(mvx_i, mvy_i) | i = 1, 2, 3$ denote REF MV, (0, 0) and PMV.

Until now, the issue for determining the state of the Markov chain model which is for describing the process of predicting tested MV, can be expressed as below:

$$S_i : (mvx_i, mvy_i) = \arg \min_i (SAD(mvx_i, mvy_i)) \quad i = 1, 2, 3 \quad (5)$$

State-transition diagram of the model is shown in Fig. 2.

Let f_n be the predicting mode of MB located at a given position (r, c) in the nth frame of video sequences, then all of the predicting modes $\{f_n : n \geq 0\}$ of continuous MBs at the same location formed a Markov chain. That is, with $\forall n \geq 0$ and any state $\{i, j, i_0, \dots, i_{n-1}; i, j, i_0, \dots, i_{n-1} \in SM\}$, there will be:

$$p(f_{n+1} = j | f_n = i, f_{n-1} = i_{n-1}, \dots, f_0 = i_0) = p(f_{n+1} = j | f_n = i) \quad (6)$$

and then the state transition probability will be defined as:

$$p_{ij}(n, m) = p(f_m = j | f_n = i) \quad m \geq n \quad (7)$$

and the one-step transition probability:

$$p_{ij}(, n+1) = p(f_{n+1} = j | f_n = i) \quad (8)$$

It has to say that the MCMME method is used for video sequence with multi-frames but not for only two frames. And for a video sequence, the transition

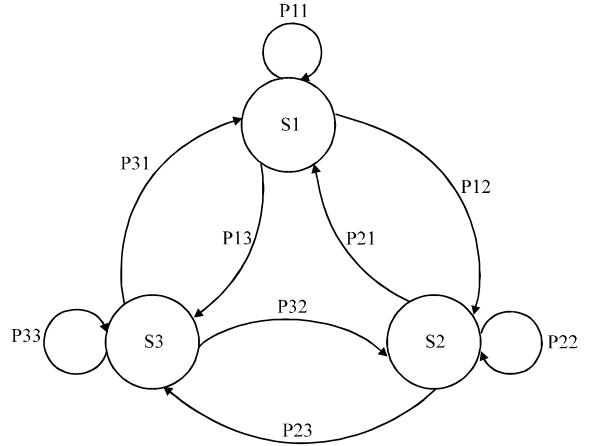


Fig. 2: State-transition diagram of the built Markov chain model

probabilities can be assumed unchanged among a group of frames. Therefore, before MCMME method used, it must do three times of FS algorithm with the first four (1-2, 2-3 and 3-4) or six (1-2, 3-4 and 5-6) frames to calculate the transition probabilities. During the two times of FS algorithms, the corresponding predicting modes of each MB are recorded as $M_{(r,c)}^k$ ($M_{(r,c)}^k \in \{1, 2, 3\}$), where, (r, c) denote the location of MB; $k \in (1, 2)$ denote the index of FS algorithms. And during the second time of FS, the corresponding transition form of each MB is recorded as $T_{F(r,c)}$, where, $TF(r,c) \in \{T_{11}, T_{12}, T_{13}, T_{21}, T_{22}, T_{23}, T_{31}, T_{32}, T_{33}\}$ and $T_{ij} : i, j \in \{1, 2, 3\}$ is denote the transition form of MB is from mode i to mode j. Therefore, at the end of the second FS algorithm, the transition probability can be calculated by Eq. 9:

$$P_{ij} = \frac{N_{T_{ij}}}{N_{MB}} \quad (9)$$

where, $N_{T_{ij}}$ is the number of T_{ij} ; N_{MB} is the total number of MBs in a frame.

After getting the transition probabilities, MCMME algorithm can be used for the rest of motion estimation task. The main steps of MCMME algorithm are shown as below:

- Step 1:** Initialize the corresponding parameter used in the algorithm the same as PMVFAST. Let Found = 0, PredEq = 0. According to the location of current MB, the PMV and the MV of REFMB can be obtained to prepare to test. If MVL = MVT = MVTR, let PredEq = 1
- Step 2:** Get the value of Distance by formula Distance = $|PMV_x| + |PMV_y|$. If PredEq = 1 and the predicted MV is equal to MV of REFMB, let Found = 2

- Step 3:** If Distance > 0 or $T_2 < 1536$ (which is used in PMVFAST) or $\text{PredEq} = 1$, choose the small diamond pattern to do the next search process
- Step 4:** According to the relationship among the transition probabilities, compute the SADs of REFMV, (0,0) and PMV in order. If one of them satisfies the early elimination conditions, then stop search immediately and assign the current predicting mode, go to step 8. If none of them satisfy the conditions, go to step 5
- Step 5:** Compute SADs of the MVL, MVT and MVTR in turn, if one of them satisfies the early elimination conditions, stop search immediately and assign spatial mode to the current predicting mode, go to step 8. Otherwise, get the minimum SAD as MinSAD go to step 6
- Step 6:** Get the minimum SAD of SAD of PMV, SAD of REFMV and the MinSAD as MSAD. If the corresponding MV is equal to REFMV and MSAD are less than the SAD of REFMV, stop search and go to step 8. Otherwise assign the current predicting mode and go to step 8
- Step 7:** Do search with line-small diamond or big diamond pattern. If Found = 2, do simplified (one step) diamond search, go to step 8 and assign the current predicting mode
- Step 8:** Assign the point with the final minimum SAD to the MV of current MB

From the above description, it can be seen that the transition probabilities used are obtained by statistical method with lots of computational burden. Therefore, it is expected to develop some automatic and intellectual way to predict these parameters.

SVM-MCMME MOTION ESTIMATION METHOD

SVM prediction networks: In order to avoid using the first two times of FS algorithms to obtain the state transition matrix, a learning method for Scalable Video Coding system (SVC) is developed in this study. For the excellent generalization ability of SVM, a multi-class SVM classifier is constructed. And then, this classifier is used for predicting the transition probabilities indirectly by classifying the feature vector of the test sequence. Therefore, this SVM classification networks is called SVM prediction networks in this study.

Binary tree structure based SVM classifier

Theory of two-class SVM: For binary problem, the basic idea of SVM classifier is to map the nonlinearly space to feature space in which the optimization hyper-plane is obtained by using nonlinear or linear transformation. And then, the SVM is found to do classification.

The optimal solution of the hyper-plane can be transformed into the following optimization problem:

$$\text{Minimize: } W(\bar{\alpha}) = -\sum_{i=1}^1 \alpha_i + \frac{1}{2} \sum_{i=1}^1 \sum_{j=1}^1 \alpha_i \alpha_j y_i y_j K(\bar{x}_i, \bar{x}_j) \quad (10)$$

$$\text{Subject to: } \sum_{i=1}^1 \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2, \dots, 1$$

where, $K = (K(\bar{x}_i, \bar{x}_j))_{i,j=1}^n$ is a positive semi-definite matrix in that the kernel function $K(\bar{x}_i, \bar{x}_j)$ satisfies the Mercer conditions. This is a convex quadratic program that has exactly one optimal solution.

Theory of multi-class SVM: The traditional SVM can not solve the multi-class classification problem.

For multi-class classification problem, there are lots of methods that are put forward before and comparatively mature. The multi-class classification problem is often decomposed into several standard binary SVM classifiers. There are two types of decomposition: one-against-all and one-against-one.

One-against-all (Bottou et al., 1999): In this case, the multi-class classification functions are constructed between each class and the remaining classes. In this way, the class labels of the training data need to be modified. The label of the current class data is set to +1 and the others are set to -1. Therefore, N binary SVM can be obtained totally. Then, the input testing data is classified by these binary SVM respectively and get the corresponding discriminant functions. By comparing these discriminant functions, the class of the testing data will be the one whose discriminant function is maximal.

One-against-one (Kressel, 1999): In this case, the multi-class classification problem is decomposed into N child classifiers. Every two classes have one child binary SVM classifier. The input testing data is classified by all the child classifiers and calculate the appearing times of every class. The predicting result is chose the one which appear maximal times totally.

Binary tree structure based SVM classifier: If the multi-classification problem is N-class separable, N-1 binary SVM classifiers can be constructed to fulfill the classification task (Cheong et al., 2004).

The first binary SVM is trained by setting the 1st class as +1 class and the other classes as -1. The ith binary SVM is trained by setting the ith class as +1 class and (i+1) th to nth as -1.

Now let us review the Markov Chain Model has been built. There are three states S_1 , S_2 and S_3 . The corresponding prediction modes are M_b , M_o and M_s .

(temporal, original and special modes), respectively. The states transition matrix of this Markov Chain Model is:

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \quad (11)$$

Now, these nine parameters in Eq. 11 can be seen as nine classes. Therefore, a 9-classes binary-tree SVM classification network can be constructed to evaluate the current MB s' feature vector \bar{x} belonging to which class. This problem can be described as follow:

For the given training data:

$$S = \{(\bar{x}_1, y_1), \dots, (\bar{x}_l, y_l); \bar{x}_i \in \mathbb{R}^{n \times 1}, y_i \in Y = \{1, 2, \dots, 9\}\} \quad (12)$$

where, n is the dimension of feature vector; l is the number of training data used; \bar{x}_i is the corresponding feature vector of the i th data and y_i is the corresponding class label.

Constructing a binary-tree SVM classification network to achieve the follows mapping:

$$\bar{x} \xrightarrow{\text{NET-SVM}} Y \quad (13)$$

In order to reflect the essential character of the MB, the feature vector is chosen as:

$$\bar{X}_i = [\mu_i, \sigma, \rho_i^t, \rho_i^s] \quad i \in [1, N_{MB}] \quad (14)$$

where, μ represent the mathematical expectation of the corresponding MB; σ represent the standard derivation of the corresponding MB; ρ is the temporal correlation coefficient between the corresponding MB and its REFMB; ρ_t is the mean value of spatial correlation coefficients of the corresponding MB and its spatial neighbor MBs.

Therefore, a prediction networks is constructed as shown in Fig. 3a.

In Fig. 3a, C-ij, ($i, j = 1, 2, 3$) denote the case that current MB belongs to state transformation from mode i to mode j . For the consideration that a large number of predicting modes is temporal mode and that the number of the other two modes is much smaller, the first SVM (SVM1 in Fig. 3a) is taking T_{ij} described in section 2.2 as the +1 class and the other ones is the -1 class.

Let $\{p_{ij}, i \in \{1, 2, \dots, 8\}, j \in \{1, 2\}\}$ be the probabilities of the positive class and negative class (where, $j = 1$ denote the positive class and $j = 2$ denote the negative one) of the child binary SVMs, respectively.

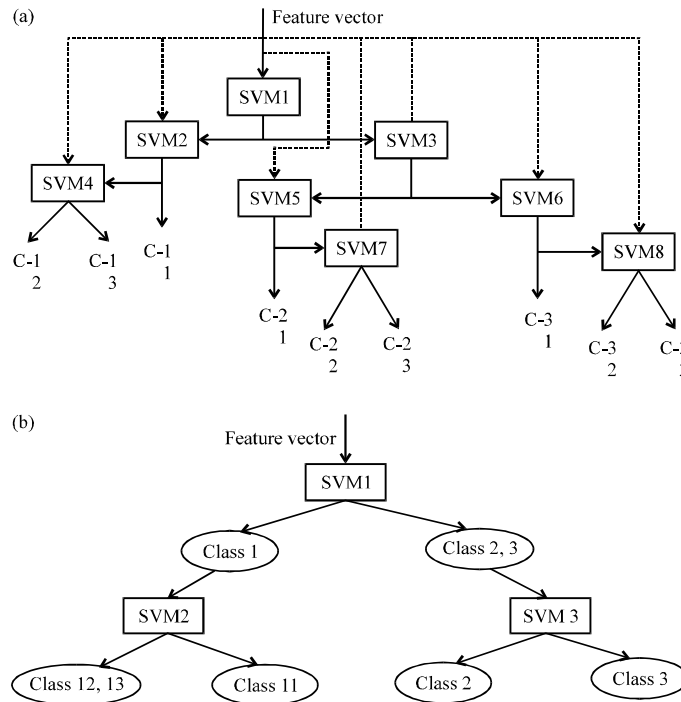


Fig. 3(a-b): Constructed SVM prediction networks, (a) Prediction networks in theory and (b) Prediction networks in practice

For any child binary SVM classifier, if using P_{pos} and P_{neg} denote the probability of +1 and -1 sample data and the correct classification rate is P , then:

$$P_{posout} = P_{pos} \cdot P + P_{neg} \cdot (1-P) \quad (15)$$

$$P_{negout} = P_{neg} \cdot P + P_{pos} \cdot (1-P) \quad (16)$$

where, P_{posout} , P_{negout} are the probabilities of +1 output and -1 output, respectively. Therefore, the nine output of the constructed 9-classes binary tree SVM classification networks are:

$$P_{C-11} = (P_{11} \cdot P_1 + P_{12} \cdot (1-P_1)) \cdot (P_{21} \cdot P_2 + P_{22} \cdot (1-P_2)) \quad (17)$$

$$P_{C-12} = (P_{11} \cdot P_1 + P_{12} \cdot (1-P_1)) \cdot (P_{22} \cdot P_2 + P_{21} \cdot (1-P_2)) \cdot (P_{41} \cdot P_4 + P_{42} \cdot (1-P_4)) \quad (18)$$

$$P_{C-12} = (P_{11} \cdot P_1 + P_{12} \cdot (1-P_1)) \cdot (P_{22} \cdot P_2 + P_{21} \cdot (1-P_2)) \cdot (P_{42} \cdot P_4 + P_{41} \cdot (1-P_4)) \quad (19)$$

$$P_{C-21} = (P_{12} \cdot P_1 + P_{11} \cdot (1-P_1)) \cdot (P_{31} \cdot P_3 + P_{32} \cdot (1-P_3)) \cdot (P_{51} \cdot P_5 + P_{52} \cdot (1-P_5)) \quad (20)$$

$$P_{C-22} = (P_{12} \cdot P_1 + P_{11} \cdot (1-P_1)) \cdot (P_{31} \cdot P_3 + P_{32} \cdot (1-P_3)) \cdot (P_{52} \cdot P_5 + P_{51} \cdot (1-P_5)) \cdot (P_{71} \cdot P_7 + P_{72} \cdot (1-P_7)) \quad (21)$$

$$P_{C-23} = (P_{12} \cdot P_1 + P_{11} \cdot (1-P_1)) \cdot (P_{31} \cdot P_3 + P_{32} \cdot (1-P_3)) \cdot (P_{52} \cdot P_5 + P_{51} \cdot (1-P_5)) \cdot (P_{72} \cdot P_7 + P_{71} \cdot (1-P_7)) \quad (22)$$

$$P_{C-31} = (P_{12} \cdot P_1 + P_{11} \cdot (1-P_1)) \cdot (P_{31} \cdot P_3 + P_{32} \cdot (1-P_3)) \cdot (P_{61} \cdot P_6 + P_{62} \cdot (1-P_6)) \quad (23)$$

$$P_{C-32} = (P_{12} \cdot P_1 + P_{11} \cdot (1-P_1)) \cdot (P_{31} \cdot P_3 + P_{32} \cdot (1-P_3)) \cdot (P_{62} \cdot P_6 + P_{61} \cdot (1-P_6)) \cdot (P_{81} \cdot P_8 + P_{82} \cdot (1-P_8)) \quad (24)$$

$$P_{C-33} = (P_{12} \cdot P_1 + P_{11} \cdot (1-P_1)) \cdot (P_{31} \cdot P_3 + P_{32} \cdot (1-P_3)) \cdot (P_{62} \cdot P_6 + P_{61} \cdot (1-P_6)) \cdot (P_{82} \cdot P_8 + P_{81} \cdot (1-P_8)) \quad (25)$$

By testing various video sequences with different motion activity, it can be found that the ratio of S1 state is more than 50% in most case. This is said that there are lots of test data classified by SVM1 and SVM2. And only the relative information is needed for the MCMME algorithm, a decision was made that, SVM4, SVM5, SVM6, SVM7 and SVM8 can be neglected. Then, the simplified SVM network is as follows in Fig. 3b.

Training data selection: There are lots of test sequences with various motion activities. The selected training data must represent the real distribution of the feature parameters. Obviously, selecting training data from many different sequences randomly or in some other manner is an inefficiency and difficult task. For this consideration,

a certain test sequence should be used to select training data. It assumes that the distribution of amplitude and phases of the MVs should be normal distribution. And then, the corresponding normal distribution fitting process is operated to MVs got from foreman.qcif, bus.qcif, news.qcif, mobile.qcif and football.cif test sequences respectively as well as the total mixed data. The corresponding data is described in Table 1, where Amp_M and Phs_M denote the average mean value of amplitudes and phases of MVs, Amp_D and Phs_D denote the average standard derivations of amplitudes and phases of MVs. The corresponding distribution histograms are shown in Fig. 4.

From the fitting results and the histogram figures, it can be seen that the distribution characteristic of foreman.qcif is almost as same as the total data. Therefore, the training data is selected from foreman.qcif.

The training results of SVM1, SVM2 and SVM3 are shown in Table 2.

Table 3 and 4, where 1, 2, 3, 4, 5, 6 denote training with different training data. For there are little MBs belonging to state S2 and state S3 in test sequence such as news.qcif, in Table 4, there are only three lines of training results.

Table 1: Normal fitting results of amplitude and phase of MVs from different training data

Data	Amp M	Amp D	Phs M	Phs D
Total	1.78	2.9780	-4.2300	40.79
Forman	1.64	2.8269	-14.0500	37.02
News	1.00	2.3031	-0.1369	30.12
Mobile	0.71	2.3031	-0.1369	24.99
Bus	4.42	3.1700	-11.4900	66.72
Football	1.11	2.2800	3.9116	26.82

Table 2: Error rates of SVM1 classifier of six different learned SVM (No. of test points:1791)

Variable	1	2	3	4	5	6	Mean
Carphone	0.13	0.13	0.19	0.19	0.16	0.19	0.17
News	0.13	0.12	0.11	0.12	0.14	0.11	0.12
Md	0.05	0.05	0.03	0.05	0.06	0.07	0.05
Coastguard	0.09	0.11	0.10	0.11	0.12	0.10	0.10
Bus	0.16	0.14	0.24	0.12	0.14	0.13	0.15
Football	0.39	0.37	0.37	0.39	0.40	0.28	0.37

Table 3: Error rates of SVM2 classifier of six different learned SVM (No. of test points:1791)

Variable	1	2	3	4	5	6	Mean
Carphone	0.15	0.12	0.08	0.07	0.13	0.08	0.11
News	0.03	0.03	0.01	0.03	0.03	0.02	0.02
Md	0.08	0.07	0.07	0.09	0.08	0.08	0.08
Coastguard	0.08	0.08	0.11	0.09	0.09	0.11	0.09
Bus	0.22	0.22	0.22	0.23	0.20	0.24	0.22
Football	0.42	0.40	0.36	0.41	0.38	0.43	0.40

Table 4: Error rates of SVM3 classifier of six different learned SVM (No. of test points:681)

Variable	1	2	3	4	5	6	Mean
Carphone	0.25	0.25	0.50	0.25	0.25	0.33	0.31
Bus	0.17	0.33	0.25	0.32	0.17	0.31	0.26
Football	0.29	0.29	0.22	0.19	0.16	0.26	0.24

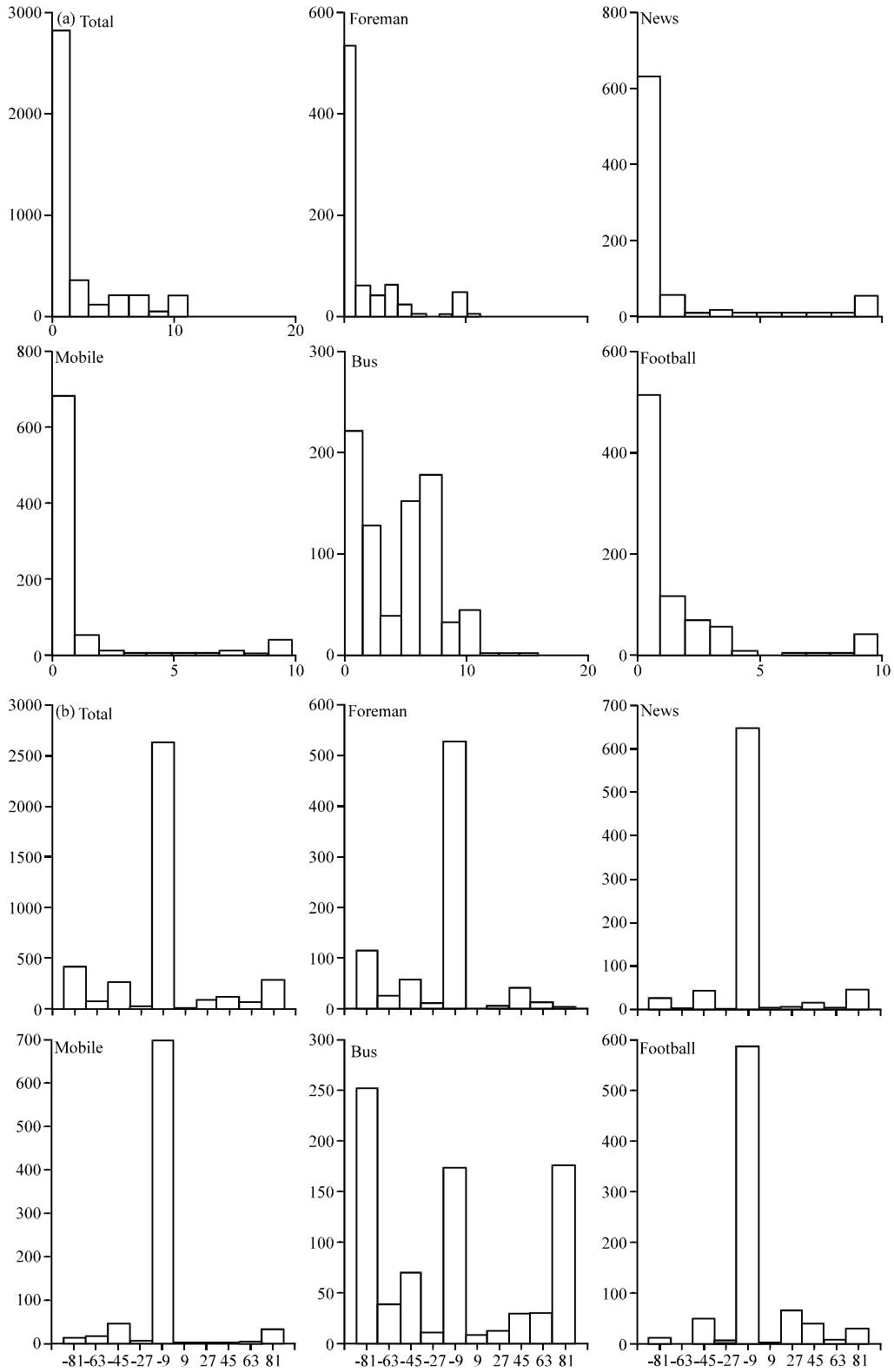


Fig. 4(a-b): Histograms of MVs from six different training sequences, (a) Amplitude histograms and (b) Phase histograms

By using the trained SVM networks, the transition matrix described above can be predicted efficiently while only the relative relation among these transition probabilities is needed. The following matrixes are the predicting results of bus.qcif and football.qcif test sequences which have much dramatic activity:

$$P_{bus} = \begin{bmatrix} 0.446 & 0.090 & 0.090 \\ 0.070 & 0.060 & 0.120 \\ 0.075 & 0.025 & 0.060 \end{bmatrix}$$

$$P_{football} = \begin{bmatrix} 0.20 & 0.09 & 0.14 \\ 0.06 & 0.07 & 0.12 \\ 0.09 & 0.07 & 0.16 \end{bmatrix}$$

Therefore, constructed SVM classification network can be used to predict the transition matrix indirectly and to guide the fast motion estimation procedure.

EXPERIMENT RESULTS

In order to evaluate the performance of the proposed method, the proposed SVM-MCMME method is embedded into a t+2D Scalable Video Coding system, respectively. The corresponding encoder structure is shown in Fig. 5.

The corresponding ME/MCTF structure with the original MCMME algorithm is illustrated in Fig. 6.

The corresponding ME/MCTF structure with the proposed method is just like the structure shown in

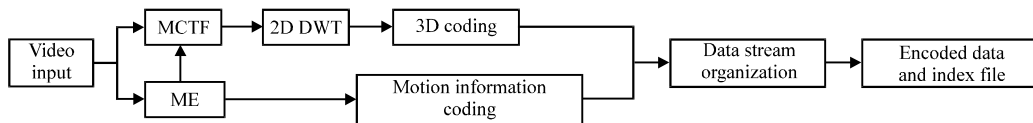


Fig. 5: Encoder structure of the used SVC

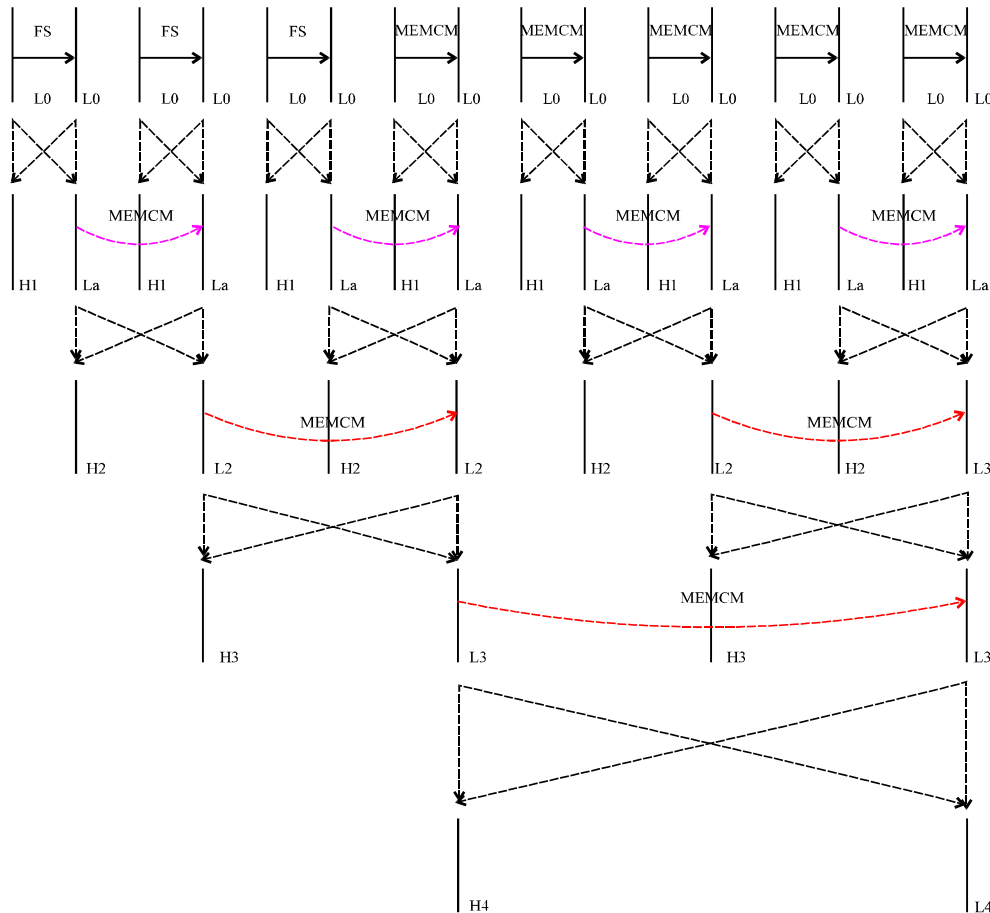


Fig. 6: ME/MCTF structure with the original MCMME algorithm

Fig. 6, the only difference is the first three FS algorithm is changed to MCMME. And the corresponding transition

Table 5: Comparison of the average ME/MCTF time per GOP and the corresponding PSNR

Variable	Bus	Football	Highway	Mobile
Aver time (sec)				
Original	0.9063	0.8944	0.8305	0.9279
proposed	0.4645	0.4507	0.4208	0.4592
Aver PSNR (dB)				
Original	40.7600	42.0300	44.0300	40.3800
proposed	40.8600	42.2200	44.1600	40.4400

probabilities are evaluated by the SVM network constructed off-line.

For obtaining the evaluation data, the average time used for ME/MCTF procedure which account for about 70% of the total encoding time was got for one Group of Pictures (GOP). And highway.qcif, mobile.qcif, bus.qcif and football.qcif were selected as the test sequences with different motion activity. The corresponding data and average PSNR were illustrated in Table 5. The corresponding curves were shown in Fig. 7 and 8.

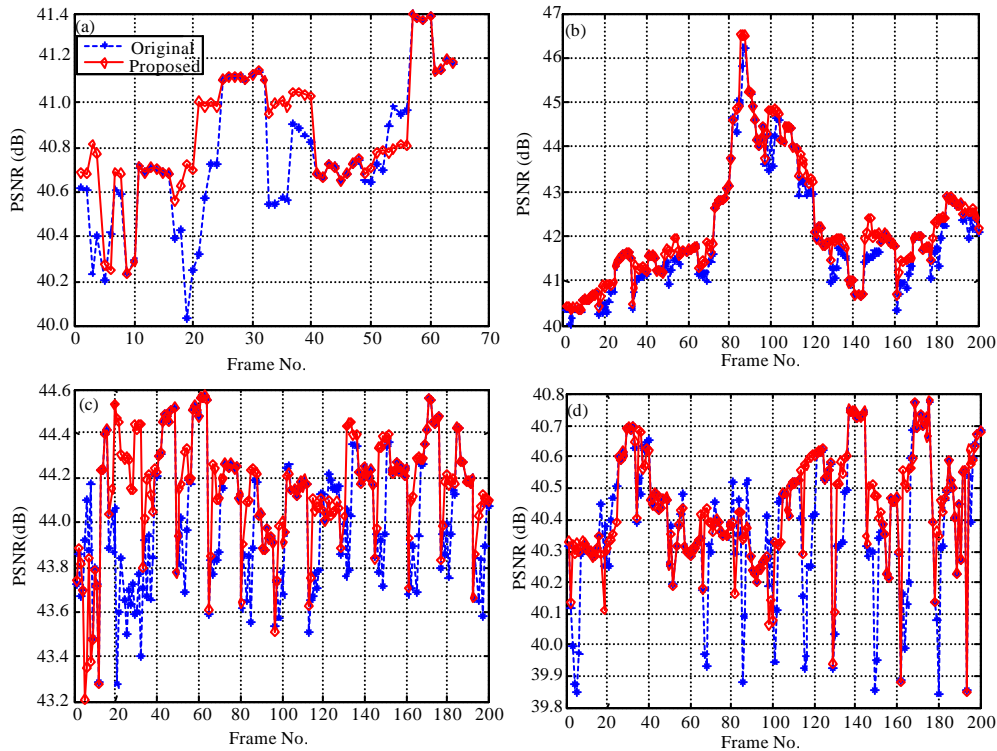


Fig. 7(a-d): PSNR performance curves of four different test sequences with different motion activities, (a) Bus, (b) Football, (c) Highway and (d) Mobile

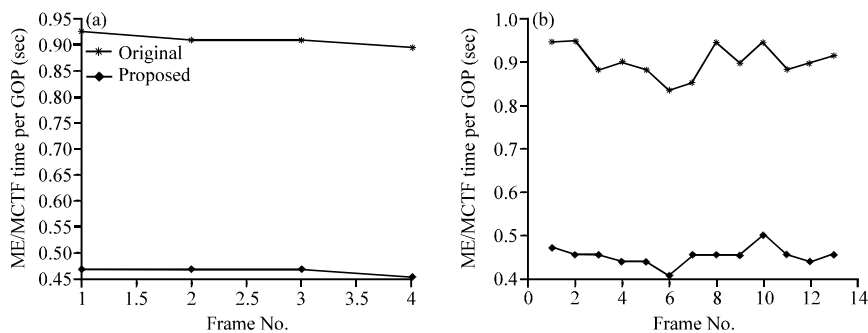


Fig. 8: Continue

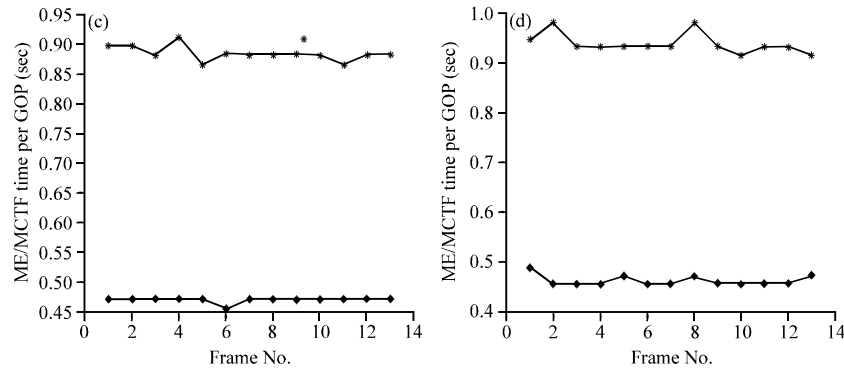


Fig. 8(a-d): Curves of time consumption per GOP of original SVC system and SVC system with SVM-MCMME used of different test sequences with different motion activities, (a) Bus, (b) Football, (c) Highway and (d) Mobile

CONCLUSION

In this study, the predictive motion estimation algorithm-MCMME proposed by us previous and its main drawback is analyzed first. For reducing the computational cost caused by statistical evaluation for using MCMME algorithm. An improved SVM-based method which is called SVM-MCMME is proposed. By using a multi-class SVM classification network, the corresponding transition probabilities of tested sequence is calculated indirectly. For evaluating the performance of the SVM-MCMME method, the performances of SVC system with ME/MCTF structure using the original MCMME algorithm and using the proposed method are tested. The corresponding ME/MCTF time per GOP and PSNR values are obtained. From the experimental results, it can be seen that there is about 100% improved in coding speed even with little PSNR increase.

ACKNOWLEDGMENT

This study is partially supported by the Natural Science Foundation of Heilongjiang Province (F201114, F201245) and by Harbin, a special study of technological innovation fund (2012RFQXG090). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers which have improved the presentation.

REFERENCES

Bottou, L., C. Cortes, J.S. Denker and H. Drucker, 1999. Comparison of classifier methods: A case study in handwritten digit recognition. Proceedings of the 12th IAPR International Conference on Pattern Recognition, October 9-13, 1999, Jerusalem, Israel, pp: 77-87.

Chen, P.H., H.M. Chen, K.J. Hung, W.H. Fang, M.C. Shie and F. Lai, 2006. Markov model fuzzy-reasoning based algorithm for fast block motion estimation. *J. Visual Commun. Image Representation*, 17: 131-142.

Cheong, S., S.H. Oh and S.Y. Lee, 2004. Support vector machines with binary tree architecture for multi-class classification. *Neural Inform. Process. Lett. Rev.*, 2: 47-51.

Cheung, C.H. and L.M. Po, 2002. A novel cross-diamond search algorithm for fast block motion estimation. *IEEE Trans. Circuits Syst. Video Technol.*, 12: 1168-1177.

Hosur, P.I. and K.K. Ma, 1998. Fast block motion estimation based on prediction of initial motion vectors of interleaved blocks using spatial correlation. Proceedings of the IEEE Asia Pacific Conference on Communications and International Conference on Communication Systems, November 23-27, 1998, Singapore, pp: 659-662.

Hosur, P.I. and K.K. Ma, 1999. Motion vector field adaptive fast motion estimation. Proceedings of the 2nd International Conference on Information, Communications and Signal Processing, December 7-10, 1999, Singapore, pp: 1-4.

Hosur, P.I. and K.K. Ma, 2000. Report on performance of fast motion estimation using Motion Vector Field Adaptive Search Technique (MVFAST). ISO/IEC JTC1/SC29/WG11, MPEG2000/M5851, Maui, USA., pp: 1-40.

Jain, J. and A. Jain, 1981. Displacement measurement and its application in interframe image coding. *IEEE Trans. Commun.*, 29: 1799-1808.

Koga, T., K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, 1981. Motion compensated interframe coding for video conferencing. Proceedings of the National Telecommunication Conference, December 1981, New Orleans, LA., pp: G5.3.1-G5.3.5.

- Kressel, U., 1999. Pairwise Classification and Support Vector Machines. In: *Advances in Kernel Methods: Support Vector Learning*, Scholkopf, B., C.J.C. Burges and A.J. Smola (Eds.). MIT Press, USA., ISBN-13: 9780262194167, pp: 255-268.
- Li, R., B. Zeng and M.L. Liou, 1994. A new three-step search algorithm for block motion estimation. *IEEE Trans. Circuits Syst. Video Technol.*, 4: 438-442.
- Po, L.M. and W.C. Ma, 1996. A novel four-step search algorithm for fast block motion estimation. *IEEE Trans. Circ. Syst. Video Technol.*, 6: 313-317.
- Tourapis, A.M., O.C. Au and M.L. Liou, 2000. Fast block-matching motion estimation using Predictive Motion Vector Field Adaptive Search Technique (PMVFAST). ISO/IEC JTC1/SC29/WG11, MPEG99/m5866, Noordwijkerhout, The Netherlands.
- Tourapis, A.M., O.C. Au and M.L. Liou, 2001. Predictive Motion Vector Field Adaptive Search Technique (PMVFAST)-enhancing block based motion estimation. *Proceedings of the SPIE Visual Communications and Image Processing*, January 24-26, 2001, San Jose, CA., USA., pp: 883-892.
- Zhao, Z., Z. Cao, M. Lin and X. Jin, 2010. A motion estimation algorithm based on markov chain model. *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, March 14-19, 2010, Dallas, TX., USA., pp: 1174-1177.
- Zhu, C., X. Lin and L.P. Chau, 2002. Hexagon-based search pattern for fast block motion estimation. *IEEE Trans. Circuits Syst. Video Technol.*, 12: 349-355.
- Zhu, S. and K.K. Ma, 2000. A new diamond search algorithm for fast block-matching motion estimation. *IEEE Trans. Image Process.*, 9: 287-290.