*Journal of Information Hiding and Multimedia Signal Processing* ©2012 ISSN 2073-4212
**Ubiquitous International** Volume **3**, Number **1**, January **2012**

# A Fast Video Transcoding Algorithm Based on Hybrid Characteristic of Multi-Scale Videos

Zhuo-Yi Lu[1,2] and Ke-Bin Jia[1]

School of Electronic Information and Control Engineering[1]
Beijing University of Technology, Beijing, China
joyv555@gmail.com

Wan-Chi Siu[2]

Department of Electronic and Information Engineering[2]
Hong Kong Polytechnic University, Hong Kong

Received February 2011; revised December 2011

Abstract. *As a key technique in network multimedia signal processing, video transcoding becomes a hot topic in recent years. This paper presents a fast intra mode decision scheme for down-sizing video transcoding in H.264 based on hybrid characteristic of multi-scale videos. In order to reduce the high computational complexity of using conventional intra prediction in the H.264 re-encoder, the proposed scheme firstly utilizes 2D-histogram to extract the spatial characteristic of macro-blocks in the downsized video to choose from intra 16×16 and intra 4×4. Then Support Vector Machine (SVM) is used to exploit the correlation between coding information extracted from the input high-resolution bit-stream and the coding modes of macro-blocks in down-sized video frames. After the SVM classifier, improbable modes in the nine intra 4×4 modes are eliminated and only a small number of candidate modes are carried out using the RDO operations. Hence, remarkable computing time can be saved, up to 74%, while maintaining nearly the same quality of the transcoded pictures.*

**Keywords:** Down-sizing video transcoding, Intra prediction, 2D-histogram, Support vector machine, RDO

1. **Introduction.** The rapid developments of network infrastructures, storage capacity, and computing power, along with advances in video coding technology and standardization are enabling an increasing number of video applications, ranging from multimedia messaging, video telephony, video conferencing over mobile TV, and wireless and wired Internet video streaming, etc. For these applications, a variety of video transmission and storage systems may be employed. Among all the techniques of adapting content to different devices, Scalable Video Coding (SVC) and down-sizing video transcoding are the most efficient ones. In the SVC techniques, the image resolution and bit-rate for each layer are required to be pre-defined [1], which reduces its flexibility. Hence, a video transcoder seems to be a more appealing solution, which can be versatile in both bit-rate and image resolution.

Currently H.264/AVC becomes as a strong candidate for a wide range of applications of the digital market in the near future. It outperforms other video coding standards due to its high coding efficiency at the expense of higher computational complexity and network friendly design [2, 3, 4]. The intra prediction combined with rate-distortion optimization

(RDO) is one of the most compression-efficient and most computation-complex operations of intra coded frames in H.264/AVC. For a straightforward realization of transcoder is to cascade an H.264 decoder and an encoder, it is too time-consuming for real-time applications. In the past few years an intensive eRort has been made to reduce the complexity of H.264/AVC intra prediction, either by reducing the complexity of rate-distortion calculation [5, 6] or by analyzing the pattern direction through edge detection techniques [7, 8, 9]. However, these algorithms are not optimal for transcoding applications since they did not exploit the valuable information from the input bit-stream, which can be re-used to speed up the transocoding process.

In recent years, support vector machines (SVMs) are used in many applications because of its excellent performance in pattern recognition. Inspired by the statistical learning theory, SVMs separate two classes of data by finding an optimal separating hyper-plane. Some researches have been done on video coding and transcoding using SVMs in the last decade. Reference [10] proposed an efficient inter mode decision algorithm for H.263 to H.264 transcoding using SVMs to investigate the relationship between data extracted from H.263 decoding stage and the optimal coding mode in H.264 re-encoding process. Reference [11] detected large and small block modes in H.264/MPEG-4 AVC formed by SVMs using SATD (the sum of absolute transformed diRerences) and CBP (coded block pattern) for feature vectors.

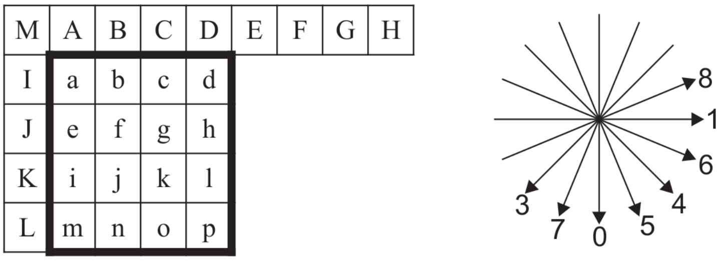
In this paper, we focus on the homogeneous transcoding techniques and they work under the same video coding standard H.264. A hierarchical mode decision scheme using the hybrid characteristic of multi-scale videos for down-sizing video transcoding in H.264 is proposed. Characteristic of macro-blocks (MBs) in the down-sized video is extracted by 2D-histogram to select the optimal mode between intra 16×16 and intra 4×4. Then we make use of the SVMs classifier to exploit the correlation between the coding information of high-resolution video and the coding modes of MBs in down-sized video to conduct an early-termination strategy for intra 4×4 mode decision process. The complexity is reduced significantly compared with the full-search mode decision, with minimum quality loss.

The rest of the paper is organized as follows. Section 2 reviews the principles of intra-frame prediction in H.264/AVC. In Section 3, we introduce the fast intra prediction using 2D-histogram. Section 4 elaborates the proposed algorithm of mode decision for intra 4×4 based on SVMs classifier, and then Section 5 describes the hierarchical mode decision scheme. In Section 6, we conduct a performance evaluation of the proposed algorithm in terms of computational complexity and rate-distortion. Then a comparative study with the recent fast intra-frame prediction methods for H.264 in the literature is provided. Finally, conclusions are drawn in Section 7.

## 2. **Complexity Analysis of H.264 Intra Coding.**

H.264/AVC is a high compression video coding standard due to the contribution of several newly techniques, such as multiple reference frames, integer transform, improved entropy encoding, intra prediction and optimized rate-distortion optimization. The intra prediction mode decision combined with rate-distortion optimization is one of the most compression- efficient and most time-consuming operations of intra coded frames in H.264/AVC.

The intra prediction in H.264/AVC exploits the directional spatial correlation to reduce the spatial redundancy within a single frame, by means of checking the similarity among pixels in the previous coded blocks with the pixels in the current block. Since the pattern of an image can assume diRerent orientation, several directions are tested for the best prediction. The H.264 defines four prediction directions for 16×16 blocks (vertical prediction, horizontal prediction, plane prediction and DC prediction) and nine for 4×4 block [12].



|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| M | A | B | C | D | E | F | G | H |
| I | a | b | c | d |   |   |   |   |
| J | e | f | g | h |   |   |   |   |
| K | i | j | k | l |   |   |   |   |
| L | m | n | o | p |   |   |   |   |

FIGURE 1. Pixels used in intra 4×4 and the prediction directions.

Fig. 1 illustrates the intra prediction encoding procedure and the nine prediction directions in an intra 4×4 block. In the eight prediction modes except for DC mode, which does not represent a pixel prediction direction but a uniform prediction block, modes {1, 4, 6, 8} can be classified into horizontal- direction modes, while modes {0, 3, 5, 7} can be classified into vertical-direction modes. As shown in Fig. 1, intra prediction uses the previously coded upper and left neighboring blocks, lowercase letters (a-p) represent the pixels of the current macro-block to be coded and the uppercase letters (A-M) represent the boundary pixels from the previously coded blocks used for the prediction. Mode 2 (DC) represents a uniform prediction block with intensity equal to the average of [A-D] and [I-L] pixels. In order to select the best prediction direction, rate-distortion optimization is operated, which analyses the bit-rate and distortion produced by each mode and chooses the one with the minimum cost. The cost RDcost is computed by Equ.1 and 2.

$$RD_{cost} = D + \lambda_{mode} \times R \tag{1}$$

$$\lambda_{mode} = 0.85 \times 2^{(Q-12)/3} \tag{2}$$

where $R$ and $D$ represent the distortion and bit-rate for a given prediction direction, respectively. $Q$ and $\lambda_{mode}$ are the quantization parameter (QP) and Lagrange multiplier, respectively. Although good quality and high compression efficiency can be achieved by the mode decision optimization, the computational complexity can be very large, which has a great impact on video transcoding. As a result, a fast intra mode decision algorithm for down-sizing transcoding is very desirable.

In this paper, we propose an innovative approach for intra mode decision based on the combination techniques of 2D-histogram and SVMs, to be used as part of a very low complexity down-sizing video transcoder.

## 3. **Fast Intra Prediction Based on 2D-Histogram.**

Generally speaking, for common video sequences, larger block types (intra 16×16) are more suitable for coding homogenous regions within a video frame such as the background, while smaller block types (intra 4×4) are more often selected for coding non-homogenous regions.

As a result, if we can perform early termination in the brute-force mode search for each MB and only enable a small number of possible modes for RDO evaluation, the computing time can be saved substantially.

3.1. **Feature Extraction Using 2D-Histogram.** Let the current MB be defined as h(x,y) with size of N×N, and L gray-levels. A 3×3 smooth window is used on $h(x,y)$ to get MB $g(x,y)$, with size of N×N and L gray-levels. Then matrix $(s,t)$ is composed of $h(x,y)$ and $g(x,y)$ and its frequency is $f_{st}$. $f_{st}$ counts the pixels in the 2-dimention space where the gray-level of $h(x,y)$ is s and the gray-level of $g(x,y)$ is $t$, subjecting to Eq. 3. Thus, 2D-histogram is drawn with a set of frequency components $\{f_{st}, s, t = 1, 2, ..., L\}$. As shown in Fig. 2 and 3, where the testing MBs were taken from Foreman sequence in QCIF format, X-axis and Y-axis are the gray-levels of $h(x,y)$ and $g(x,y)$ respectively and Z-axis is the corresponding number of pixels.

$$\sum_{s=0}^{L-1}\sum_{t=0}^{L-1} f_{st} = N \times N \tag{3}$$

Histogram illustrates the statistical properties of gray-levels, and can also represent the complexity of texture in an image. If the histogram of an image is mainly centralized on one gray-level, it means that the image is smooth. On the contrary, if the histogram is made up of many gray-levels with nearly the same frequency, the image is with high spatial detail or fast motion. Based on the hypothesis that there is a strong correlation between spatial pixels, 2D-histogram is used to extract the spatial feature of MBs.

Fig. 2 shows the 2D-histogram of a typical MB adopting intra 16×16 mode, where there is only one gray-level with a large number of pixels. Fig. 3 illustrates a MB using intra 4×4 mode, where there are many gray-levels with almost the same number of pixels. As a result, 2D-histogram can reflect the spatial relationship between pixels accurately.

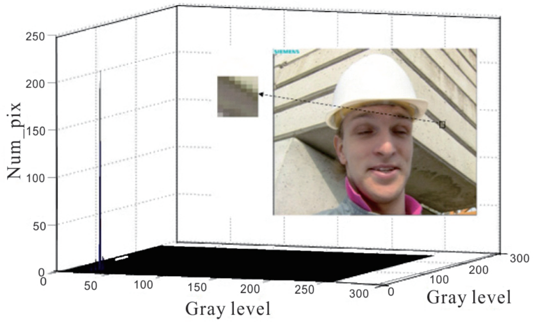


Figure 2. 2D-histogram of intra 16×16 mode

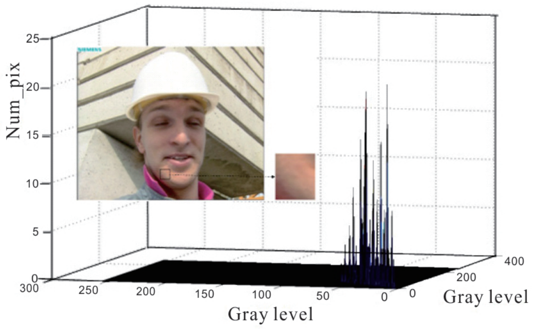


Figure 3. 2D-histogram of intra 4×4 mode

3.2. **Mode Decision by Double-Threshold Method.** Since the spatial feature has been extracted using 2D-histogram, criterion is required to select between intra 4×4 and intra 16×16. In this paper, double-threshold method is adopted. Firstly, two thresholds A and B are set, where A > B. Secondly, 2D-histogram of current MB is drawn and the gray-level with maximum pixels ($MaxValue$) is obtained. Thirdly, $MaxValue$ is compared with A and B. If $MaxValue$ is bigger than A, then the current MB is smooth and adopts intra 16×16 mode. If $MaxValue$ is smaller than B, then the current MB is with more details and adopts intra 4×4 mode. If $MaxValue$ lies between A and B, then both intra 16×16 and intra 4×4 will be predicted.

4. **Mode Decision for Intra 4×4 MB Using Support Vector Machines.** Based on the analysis in Section 2, the high complexity of H.264 video transcoding creates an opportunity for applying machine learning algorithms to reduce the complexity of the transcoder. The main goal of our work is to find regularities of the data (i.e. the coding information of the input bit-stream) and transform them into generalized features to be expressed by the knowledge representation model. We make use of the concept of learning machine to exploit the association or relations between the variables and instances in the dataset, then determine the value of a target (the class) variable (one of the MB modes). From the existing classification models available in the literature, we choose Support Vector Machines (SVMs) due to its excellent performance in pattern recognition. SVMs are applied in our scheme to exploit the correlation between the coding information of original high-resolution video and the coding modes of down-sized video to train SVMs model. Since intra 4×4 mode or intra 16×16 mode has been selected for each MB using 2D-histogram, we will then classify Vertical and Horizontal modes for each intra 4×4 MB. This approach reduces the number of the MB modes without evaluating all the nine possible combination of intra 4×4 modes by the RDO operation, thus can lead to an early termination of mode decision in the H.264 re-encoding process.

In this section we are going to describe the process of using SVMs to build a mode decision classifier. We will highlight the uses of SVM and pin-point its characteristics suitable for this transcoding applications.

4.1. **Support Vector Machine.** Support vector machines (SVMs) are based on Vapnik Chervonenkis (VC) [13] dimension of statistical learning theory and Structural Risk Minimization. The basic principle of SVMs is to find an optimal separating hyper-plane so as to separate two classes of data with the maximal margin. Classification and function approximation are formulated as quadratic programming (QP) problems.
Suppose there is a training set:

$$(x_1, y_1), ..., (xl, yl) \in R^N \times \pm 1 \tag{4}$$

where $l$ is the length of data set.

The task of SVMs is to find the optimal hyper-plane in formulation (5) with the maximum margin

$$margin = \frac{2}{\|w\|}$$

to separate these two classes of data, where $w$ is the weight vector and b is the scale vector.

$$(w \cdot x) - b = 0, w \in R^N, b \in R \tag{5}$$

By introducing Lagrange multiplier, the solution of the optimization problem is as follows:

maximize

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \tag{6}$$

subject to

$$\alpha_i \geq 0, \quad i = 1, ..., l, \quad and \sum_{i=0}^{l} \alpha_i y_i = 0$$

In the case of non-linear separable in the original space, SVMs firstly transform the input space into a high-dimensional feature space through some nonlinear mapping function $\phi = R^N \rightarrow F$ and then construct an optimal separating hyper-plane in the feature space. On the basis of Hilbert-Schnidt theorem [14], any function $K(xi, xj)$ satisfying Mercer's condition can be used in the construction rule which is equivalent to construct an optimal hyper-plane in some feature space. Therefore, the evaluation of decision function requires the inner product of mapping function in explicit form:

$$K(x_i, x_j) = (\phi(x_i) \cdot \phi(x_j)) \tag{7}$$

Substituting the inner product of input vectors for the kernel function, the optimization problem (6) is rewritten as:

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i \cdot x_j) \tag{8}$$

subject to

$$0 \leq \alpha_i \leq C, \sum_{i=1}^{N} y_i \alpha_i = 0$$

SVMs have become one of the most popular methods and constituted a standard choice in classification problems because this approach possesses many useful properties that outperform other methods of classification: First, the optimization problem for constructing an SVM has a unique solution. Second, the learning process for constructing the SVM is rather fast. Third, simultaneous to constructing the decision rule, one can obtain the set of support vectors (SVs). Finally, the implementation of a new set of decision functions can be done by changing only one function (Kernel Function), which defines the inner product in the feature space. In view of the fact that SVMs have achieved excellent performance in many fields of pattern recognition applications, especially for some complex classification problems, it is used in our proposed scheme for creating the mode decision classifier.

4.2. **Feature Vectors Selection.** In this sub-section we will describe the data preparation that can transform video sequences to an appropriate input for the SVMs training process. Recall two key issues in the problem of the SVMs classifier: pattern representation and feature extraction [14]. The feature vector selected should follow the following two important rules [10]: first, the extra computations required to calculate this measure must be as low as possible, second, the feature should have strong correlation with the optimal block types. To develop a systematic block type selection scheme, we have conducted extensive experiments using a set of criteria. After the investigation of the relationship between the coding information and MB coding mode, a set of features are picked.

When intra 4×4 mode is chosen, each of the nine prediction directions has to be evaluated for every 4×4 block within one 16×16 MB. Based on the hypothesis that the edge orientation gives the direction of minimum energy variance within a block and hence can be mapped to an intra prediction mode [15], transform domain coefficients are utilized to determine the edge direction for a block. The edge direction of a spatial domain 4×4 block can be effectively computed using its corresponding few transform domain AC coefficients as in Eq. (9).

$$\tan\theta = \frac{F_{0,1} + F_{0,2} + F_{0,3}}{F_{1,0} + F_{2,0} + F_{3,0}} \tag{9}$$

where $\theta$ is the angle between the edge direction and the horizontal axis as shown in Fig. 4. $F_{u,v}$ are the corresponding transform domain AC coefficients of this particular 4×4 block. For each 4×4 block, the intra prediction mode (shown in Fig. 1), which is the closest to the computed edge direction angle $\theta$ will be chosen as the candidate mode for R-D cost evaluation.

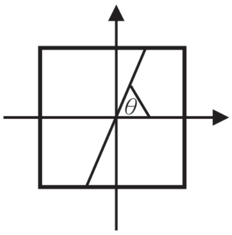


Figure 4. Block edge direction prediction

In addition, in order to guarantee a high classification accuracy, we define the absolute sum of the first row of AC coefficients as *ac_sum_h* (Eq. 10) and the absolute sum of the first column of AC coefficients as *ac_sum_v* (Eq. 11). If *ac_sum_h* is smaller than *ac_sum_v*, the 4×4 block is less likely to be coded in vertical mode, and modes {0, 3, 5, 7} can be removed. On the other hand, if *ac_sum_v* is smaller than *ac_sum_h*, the 4×4 block is less likely to be coded in horizontal mode, and modes {1, 4, 6, 8} can be removed.

$$ac\_sum\_h = \sum_{j=1}^{3} |AC[0, j]| \tag{10}$$

$$ac\_sum\_v = \sum_{j=1}^{3} |AC[i, 0]| \tag{11}$$

Finally, the class labels to be classied or generalized are vertical modes and horizontal modes. In summary, the feature vectors for SVMs classier, represented by FV are shown below.

$$FV = [\tan\theta, ac\_sum\_h, ac\_sum\_v]$$

4.3. **Kernel Function Selection.** As mentioned before, a significant advantage of the SVM is the sparseness representation of the decision function, which allows the SVM to classify new data efficiently. In other words, only the training samples, so-called the support vectors, who lie close to the separating hyper-plane will participate in the specification of the hyper-plane and receive non-zero weights in the quadratic program. The performance of an SVM classifier is determined by selecting appropriate training data and a suitable kernel function. Thus, one of the major tasks of SVM approach is to look for a possible optimal kernel function and its coefficients.

There are four typical kernel functions. 1) Linear kernel function: $(x \cdot x_i)$. 2) Polynomial kernel function: $(x \cdot y + c_n)^p$. 3)Radial basis function: $(\exp(-(\frac{1}{2\delta^2}))\left\|x - y\right\|^2)), \gamma > 0$.4)Two-layer neural function: $\tanh(\gamma \cdot x \cdot y + c_n), c_n \geq \gamma, \|x\| = 1$. These kernel functions can be divided into two categories [13]: the local kernel function (e.g. the RBF) and the global kernel function (e.g. the polynomial kernel function). Fig. 5 illustrates the characteristics of the two types of kernels. For the RBF, the curves denote the shape of the function when $\delta$ equals 0.1, 0.3 and 0.5, respectively. For the polynomial kernel function, the curves represent the shape when p equals 1, 3 and 5, respectively.

Generally speaking, the local kernels are good at extracting the local features rather than the global features of the training samples, or vice versa. Thus it can be stipulated that compared with the global kernels, the local kernels have better learning ability with relative weak prediction ability and universal approximation properties.

Generally speaking, the local kernels are good at extracting the local features rather than the global features of the training samples, or vice versa. Thus it can be stipulated that compared with the global kernels, the local kernels have better learning ability with relative weak prediction ability and universal approximation properties.

For an in-depth study, a measurement J is dened in this paper to evaluate the performance of diRerent kernel functions. For a set of training data $(x_1, x_2, ..., x_l, x_{l+1}, x_{l+2}, ..., x_{2l})$, we assume that $(x_1, x_2, ..., x_l)$ belong to Class 1 and $(x_{l+1}, x_{l+2}, ..., x_{2l})$ belong to Class 2, where $l$ is the length of the data set. First, the centroids C1 and C2 of each class are calculated by the mapping function $\phi$ in equation (12).

$$C_1 = \frac{1}{l}\sum_{i=1}^{l}\phi(x_i), C_2 = \frac{1}{l}\sum_{i=l+1}^{2l}\phi(x_i) \tag{12}$$

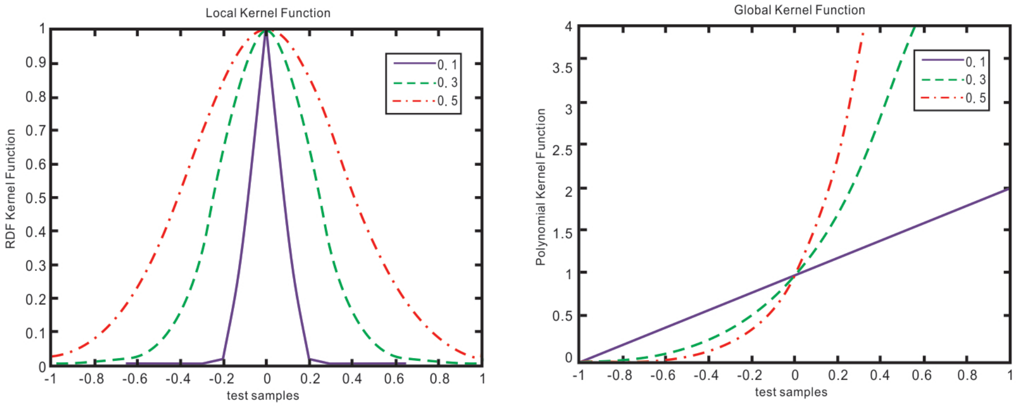


Figure 5. Two types of kernel functions. (a) RBF (b) Polynomial

Second, the 2-norm values of the centroids are calculated. Let us substitute the mapping function $\phi$ by the kernel function $K(x_i, x_j)$ according to equation (7) in the following form.

$$\|C_1\|^2 =< C_1, C_1 >= \frac{1}{l^2}\sum_{i,j=1}^{l} < \phi(x_i), \phi(x_j) >= \frac{1}{l^2}\sum_{i,j=1}^{l} K(x_i, x_j)$$

$$\|C_2\|^2 =< C_2, C_2 >= \frac{1}{l^2}\sum_{i,j=l+1}^{2l} < \phi(x_i), \phi(x_j) >= \frac{1}{l^2}\sum_{i,j=l+1}^{2l} K(x_i, x_j) \tag{13}$$

Third, the cohesion capacity of the homogeneous class $\delta_1^2$ and $\delta_2^2$ can be measured by equation(14).

$$\delta_s^2 = \frac{1}{l}\sum_{m=1}^{l}\|\phi(x_m) - C_s\|^2 = \frac{1}{l}\sum_{m=1}^{l} K(x_m, x_m) + \frac{1}{l^2}\sum_{i,j=1}^{l} K(x_i, x_j), s = 1or2 \tag{14}$$

Meanwhile the capacity of separating heterogeneous classes can be measured by calculating the distance between the centroids using equation (15).

$$\|C_1 - C_2\|^2 =< C_1, C_1 > + < C_2, C_2 > -2 < C_1, C_2 >$$
$$= \frac{1}{l^2}\sum_{i,j=1}^{l} K(x_i, x_j) + \frac{1}{l^2}\sum_{i,j=l+1}^{2l} K(x_i, x_j) - \frac{2}{l}\sum_{i=1}^{l}\sum_{j=l+1}^{2l} K(x_i, x_j) \qquad (15)$$

Finally, we can get the parameter $j$ using equation (16), where the numerator indicates the distance between the centroids of different classes, and the denominator denotes the cohesion capacity of homogeneous class. It can be seen that the larger $J$ is, the better is the distribution of training data in the feature space by using a certain kernel function.

$$J = \frac{\|C_1 - C_2\|^2}{\delta_1^2 + \delta_2^2} \qquad (16)$$

Experiments were done using the selected feature vectors mentioned in the previous subsection as the training data set. The coefficients of each kernel function were obtained by the method of cross validation, which is a common way of validating a model. In a v-fold cross-validation, initially, it is to divide the training sets into v subsets with the same step size. Then, one subset is tested using the classifier trained on the remaining (v-1) subsets. Thus, each instance of the whole training set is predicted once. We set $p = 3$ and $c = 0$ for the polynomial kernel function, $\sigma = 0.333$ for RBF, and $\gamma = 0.333$ and $c = 0$ for the two-layer neural kernel.

From Table 1, we can see that the RBF has the maximum $J$ and the highest prediction accuracy, exhibits a good behavior with respect to the prediction ability. Hence, we choose the RBF in processing the SVMs training and prediction.

## 5. **Hierarchical Intra-Prediction Structure.**

In our proposed intra-frame mode decision architecture,a hierarchical classifier composed of two stages is designed.

1) The first stage is to classify intra 16×16 mode and intra 4×4 mode using 2D-histogram. If it is classified into intra 16×16 modes, intra 4×4 modes are disabled. Otherwise, go to (2).
2) The second stage is to classify vertical modes and horizontal modes for each intra 4×4 MB based on SVMs. If it is classified into vertical modes, modes {1, 4, 6, 8} are disabled. If it is classified into horizontal modes, modes {0, 3, 5, 7} are disabled.

The detailed mode decision scheme is illustrated in Fig. 6. By discarding the improbable modes, only a small number of candidate modes are used and the process of calculating RD cost can be early terminated compared with the full-search mode decision. The incoming video sequence is decoded and the information required by the SVMs classifier is gathered. Since the process of SVMs training and prediction is conducted oR-line, there is no extra computational burden for the transcoder.

## 6. **Experimental Results.**

The training process was done off-line. Hence it does not give any additional computational burden at the time using the SVMs classifier for transcoding. The proposed scheme was implemented in the H.264 reference software JM 12.2. A high quality, easy to use and free libSVM [16] software package developed by Chang et al. was used in our experimental work for SVMs training and prediction.

The process of SVMs training was performed as follows: 1. Convert the data to the input format of libSVM software. 2. Conduct scaling on the individual components of the input data. 3. Use the RBF kernel for mapping data to a higher dimensional space.

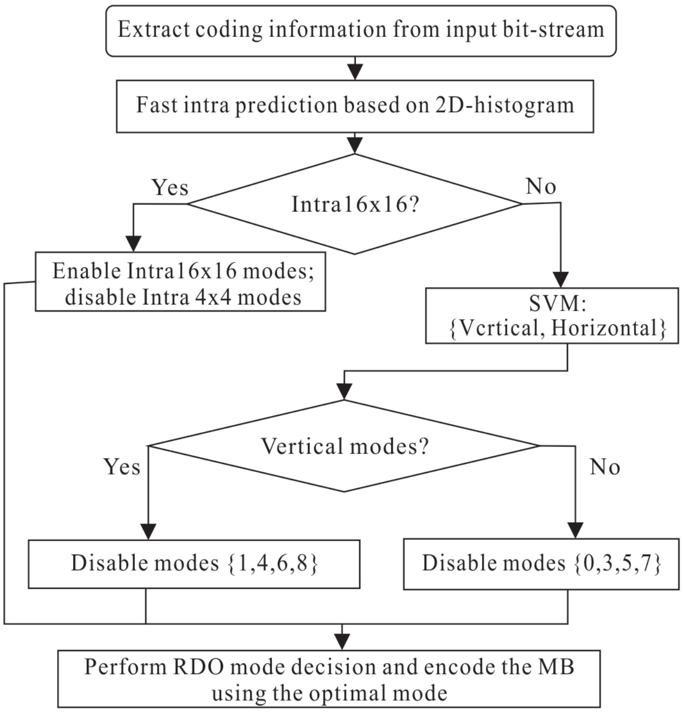

Extract coding information from input bit-stream
Fast intra prediction based on 2D-histogram
Intra16x16?
Yes
No
Enable Intra16x16 modes; disable Intra 4x4 modes
SVM: {Vcrtical, Horizontal}
Vertical modes?
Yes
No
Disable modes {1,4,6,8}
Disable modes {0,3,5,7}
Perform RDO mode decision and encode the MB using the optimal mode

Figure 6. Flowchart of the proposed algorithm

4. Use cross-validation to find the best penalty parameter c and the kernel parameter. 5. Train the whole training set.

Experiments were conducted to evaluate the performance of the proposed algorithm when transcoding videos at commonly used resolution CIF (352G288). We have conducted experiment with typical videos representing a wide range of motion activities, textures and colors. They were firstly encoded and decoded using the H.264 to train the SVM models. Then we down-sizing transcoded the videos by 4. Video sequences Foreman, News, Mobile, Paris, Silent, Stefan, Mother-daughter and Flower in CIF format using quantization parameters (QP) from QP=24 to QP=32. One reference frame and all frames were coded as I-frames.

Table 1. The measurement of $J$ for different kernel functions

| | linear | Polynomial | RBF | Sigmoid |
|---|---|---|---|---|
| $J$ | 0.1870 | 0.2109 | 0.2167 | 0.1847 |
| *Accuracy* | 45.36 | 45.38 | 46.75 | 44.39 |

Three metrics are used to evaluate the comparative performance: 1) degradation of image quality in terms of average Y-PSNR ΔPSNR (dB), 2) increment of bit-rate: $\Delta BR = (BR_{ref} - BR_{prop})/BR_{ref} \times 100\%$ (%), where $BR_{ref}$ and $BR_{prop}$ are the encoding bit-rate of the reference and the proposed method respectively, and 3) encoding time saving: $\Delta T = (T_{ref} - T_{prop})/T_{ref}100\%$ (%), where $T_{ref}$ and $T_{prop}$ are the total encoding times of the reference and the proposed method respectively. We compare the performance of the proposed algorithm ($Theproposed$) with three alternative methods which were fully implemented using the JM12.2 with the conventional full-mode encoder ($JM12.2$), the Majority Mode method with refinement of Horizontal, Vertical and DC modes ($MM + HVDC$) [17], since $MM + HVDC$ is a recent fast intra-mode decision algorithm used in resolution reduction on H.264 transcoders and a fast intra-frame prediction algorithm ($Jia's$) [12].

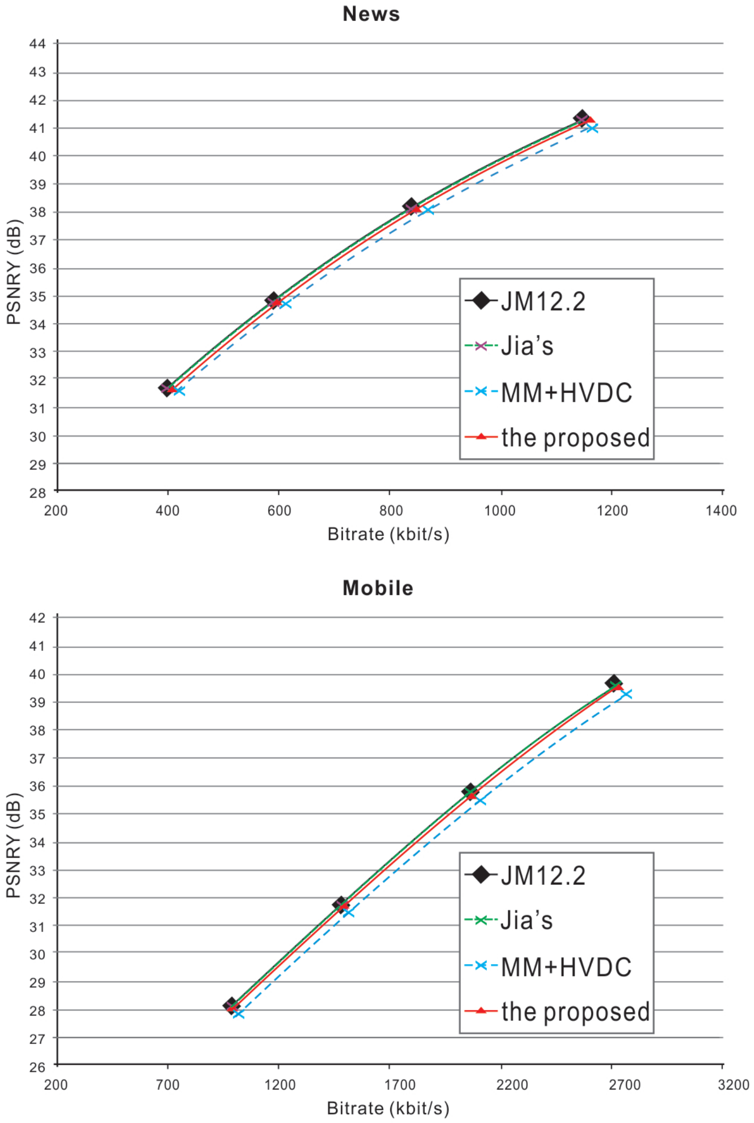


Figure 7. R-D curves of mode decision (a) News (b) Mobile

Table 2 (positive values denote increments whereas negative values denote decrements) shows that the proposed method saves about 67.6% of time with 0.03 dB PSNR degradation and 2.27% extra bits on average, while the $MM + HVDC$ algorithm gives an average time savings of 53.6%, 0.2 dB PSNR degradation and 7.77% extra bits required. Although $Jia's$ method can achieve nearly the same performance with JM12.2, it can only save 13.1% computational complexity on average.

It can be seen from Fig. 7 and Fig. 8 that, the PSNR obtained when applying our proposed algorithm deviates slightly from the result using the complex brute-force H.264. The drop in RD performance becomes negligible as compared to the reduction in computational complexity. In terms of time saving, which is a critical issue in video transcoding applications, the proposed method significantly reduces the computational complexity for re-encoding the video sequences.

7. **Conclusions.** In this paper, we proposed a novel architecture for a low-complexity and high quality H.264 video down-sizing transcoder to solve the problem of mode decision in intra-frame. The low computational complexity is achieved by using the hybrid characteristic of multi-scale videos. Firstly, we extract the spatial characteristic of MBs

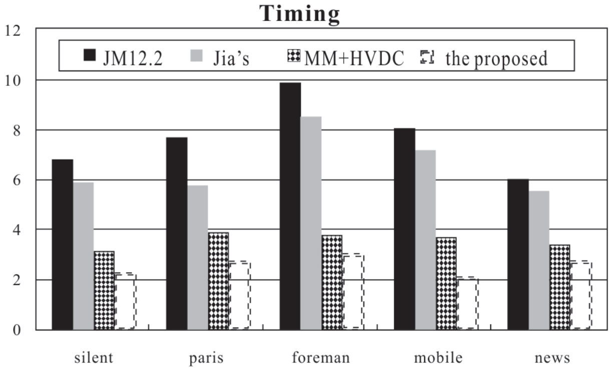


Figure 8. Computational complexity of mode decision

Table 2. Comparison in The Performance Measures of Mode Decision

| Video Sequence | *Jia's* | | | *MM+HVDC* | | | *The proposed* | | |
|---|---|---|---|---|---|---|---|---|---|
| | ΔPSNR | ΔBR | ΔT | ΔPSNR | ΔBR | ΔT | ΔPSNR | ΔBR | ΔT |
| Foreman | -0.005 | +0.347 | -13.19 | -0.128 | +25.72 | -68.30 | +0.043 | +7.851 | -69.72 |
| Paris | -0.005 | +0.141 | -24.41 | -0.158 | +2.36 | -48.90 | -0.033 | +0.853 | -64.68 |
| Mobile | 0 | +0.019 | -10.53 | -0.265 | +2.62 | -53.98 | -0.065 | +0.697 | -74.03 |
| Silent | -0.008 | +0.274 | -13.34 | -0.158 | +5.99 | -55.01 | -0.015 | +2.481 | -67.10 |
| News | -0.038 | +0.255 | -6.93 | -0.118 | +3.86 | -43.39 | -0.055 | +1.270 | -55.13 |
| Stefan | -0.012 | +0.238 | -10.10 | -0.333 | +9.87 | -61.30 | -0.030 | +2.043 | -71.22 |
| Flower | -0.032 | +0.305 | -9.11 | -0.290 | +8.96 | -55.63 | -0.047 | +2.010 | -70.71 |
| Mother-daughter | -0.001 | +0.012 | -16.83 | -0.112 | +2.76 | -42.33 | -0.009 | +0.988 | -67.88 |
| *Average* | -0.01 | +0.20 | -13.1 | -0.20 | +7.77 | -53.6 | -0.03 | +2.27 | -67.6 |

in the down-sized video using 2D-histogram to choose from intra 4×4 and intra 16×16. Secondly, we exploit the correlation between the coding information of the original high-resolution video and the coding modes of the down-sized video to classify the nine modes in intra 4×4 making use of SVMs. Then a hierarchical intra prediction scheme is built for H.264 coding mode decision. This approach reduces the number of candidate modes which are evaluated by RDO, and leads to an early termination of mode decision in the H.264 re-encoding process. Experimental results show that the overall performance of the proposed architecture is good and it considerably reduces the computational complexity by 67.6% on average, with little degradation on image quality.

**Acknowledgment.** This work was supported by the National Natural Science Foundation of China under Grant No. 30970780, Center for Multimedia Signal Processing, Department of Electronic and Information Engineering, Hong Kong Polytechnic University and the Research Grant Council of the Hong Kong SAR Government (PolyU 5268/06E).

## REFERENCES

[1] A. Vetro, Transcoding, Scalable coding and standardized metadata, *Proc. of the 8th International Conference on Visual Content Processing and Representation*, pp. 15-16, 2003.

[2] Pengyu Liu and Kebin Jia, Research and optimization of low-complexity motion estimation method based on visual perception, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, no. 3, pp. 217-226, 2011.

[3] H. Wang, Joyce Liang and C. C. Jay Kuo, Overview of robust video streaming with network coding, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, no. 1, pp. 36-50, 2010.

[4] J. Lou, S. Liu ,Anthony Vetro and M. T. Sun, Trick-play optimization for H. 264 video decoding, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, no. 2, pp. 132-144, 2010.

[5] Chao-Chung Cheng and Tian-Sheuan Chang, Fast three step intra prediction algorithm for 4x4 blocks in H. 264, *Proc. of IEEE International Conference on Circuits and Systems*, vol. 2, pp. 1509-1512, 2005.

[6] Jianfeng Ren, N. Kehtarnavaz and M. Budagavi, Computationally efficient mode selection in H. 264/AVC video coding, *IEEE Trans. Consumer Electronics*, vol. 54, no. 2, pp. 877-886, 2008.

[7] F. Obermeier, M. Durkovic, M. Zwick and K. Diepold, AVC intraprediction mode decision based on 4x4 integer transform coefficients, *Proc. of the 8th International Conference on Image Analysis for Multimedia Interactive Services*, pp. 55-58, 2007.

[8] A. C. Tsai, J. F. Wang, J. F. Yang and W. G. Lin, Effective subblock-based and pixel-based fast direction detections for H. 264 intra prediction, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 18, no. 7, pp. 975-982, 2008.

[9] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. Li, Dajun Wu and Si Wu, Fast mode decision algorithm for intraprediction in H. 264/AVC video coding, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 15, no. 7, pp. 813-822, 2005.

[10] X. Jing, W. C. Siu, L. P. Chau and A. G. Constantinides, Efficient inter mode decision for H. 263 to H. 264 video transcoding using SVMs, *Proc. of IEEE International Conference on Circuits and Systems*, pp. 2349-2352, 2009.

[11] Jaeil Kim, Munchurl Kim, Sangjin Hahm, I. J. Cho and Changsub Park, Block-mode classification using SVMs for early termination of block mode decision in H. 264/MPEG-4 part 10 AVC, *Proc. of the 7th International Conference on Advances in Pattern Recognition*, pp. 83-86, 2009.

[12] K. B. Jia, Xie Jing and Fang Sheng, A fast intra prediction based on autocorrelation, *Chinese Journal of Electronics*, vol. 34, no. 1, pp. 152-154, 2006.

[13] Ying Tan and Jun Wang, Support vector machine with a hybrid kernel and minimal vapnik-chervonenkis dimension, *IEEE Trans. Knowledge and Data Engineering*, vol. 16, no. 4, pp. 385-394, 2004.

[14] V. N. Vapnik, *Statistical Learning Theory*, Wiley, New York, USA, 1998.

[15] B. Shen and I. K. Sethi, Direct feature extraction from compressed images, *Proc. of SPIE Conference on Storage & Retrieval for Image and Video Databases IV*, pp. 404-414, 1996.

[16] C. C. Chang and C. J. Lin, A library for support vector machines, http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[17] Sandro Moiron and Mohammed Ghanbari, Reduced complexity intra mode decision for resolution reduction on H. 264/AVC transcoders, *IEEE Trans. Consumer Electronics*, vol. 55, no. 2, pp. 606-612, 2009.