Adaptable Neural Networks for Content-based Video Adaptation in Low/Variable Bandwidth Communication Networks

Anastasios Doulamis and Georgios Tziritas
Computer Science Department, University of Crete
Heraklion, Crete, Greece
Tel + 30 2810 393523, +30 2810 393517
Email: adoulam@cs.ntua.gr, tziritas@csd.uoc.gr

Abstract – In this paper, an adaptable neural network model is used for real time video delivery over communication networks of low and variable bandwidth, such as the wireless ones. The scheme performs video delivery in content domain in contrast to the previous approaches in which only temporal frame skipping is adopted. The proposed method requires no buffering of video frames and thus imposing no frame delay. In particular, in case of low bandwidth conditions, the proposed scheme estimates the number of frames that best represent the sequence within a time segment and transmit this number for delivery instead of a temporal frame skipping. Multiple key frames are considered by optimally approximating the real bandwidth availability with a rational fraction. Key frame estimation is accomplished using a neural network model capable of predicting the indices of the most appropriate key frames that are to be delivered without being available the video information. The model takes into account the previous information as it has been evaluated by the already delivered information. The proposed scheme is based on an efficient recursive estimation algorithm since the network weights cannot be considered constant throughout video transmission. This is due to the fact that content as well as bandwidth characteristics vary from time to time.

I INTRODUCTION

Efficient video delivery and transmission over low and time variable networks, (e.g., the wireless networks), require to tailor video information with a minimal reduction of its quality. Conventional time sampling algorithms discard audiovisual information, with respect to bandwidth constraints, without taking into account content characteristics. As a result, useful information maybe discarded since humans perceive video quality as a function of video content fluctuation rather than video temporal variation.

Adaptation algorithms for video delivery have attracted many researchers in the past. Examples include the Fine Granularity Scalability (FGS) [2] scheme of the MPEG-4 standard [1]. The FGS scheme decomposes video into a base layer (BL) and one or several enhancement layers (EL's). In the base layer, the basic video quality is delivered whereas video quality is improved by the information carried on the enhancement layers.

Other works for video adaptation include spatial and temporal scalable algorithms, such as the ones presented in [3]-[6]. In particular, [3] proposes the so called FC-DCT method while [4] the FM-DCT one. Both techniques are applied on the MPEG compressed domain. In [5] the temporal scalability is achieved through an efficient management of B frames, while in [6] a new frame is introduced in levels based on the FSG approach and the temporal variation of the bandwidth.

The above mentioned methods tailor video quality either by reducing the spatial or the temporal information that cannot be delivered. However, in the recent algorithms of video abstraction and summarization, video information are organized audiovisual data with respect to the content characteristics. Algorithms for shot detection can be considered as the first attempts towards a content-based video adaptation [7], [8]. Other more complicated approaches are based on the extraction of multiple key-frames from a shot, able to effectively describe the shot content [9]. In [10], video is decomposed into a sequence of "sub-shots" and a motion intensity index is computed for each of them. Then, all indices are quantized into predefined bins, with each bin assigned a different sampling rate and key-frames are sampled from each sub-shot based on the assigned rate.

The main drawback, however, of all the above mentioned methods, is that content organization is performed in a static way preventing content adaptation in terms of bandwidth variations. This means that the amount of the delivered audiovisual content is not restricted by the network channel characteristics, and the terminal devices requirements. Bandwidth adaptability has been reported in many works such as the [11], [12] and [13]. In particular, in [11] when the bandwidth is lower than the required one, the first video frame is delivered, while the remaining are skipped until bandwidth requirements are satisfied. Such an approach, however, performs only *linear adaptation*, since content information is not taken into account. Another policy, considering the variation of motion activity between the sequence where a frame is transcoded and the sequence where that frame is skipped, has been proposed in [12]. A dynamic method for frame skipping has been proposed in [13]. However, despite its dynamic nature, the aforementioned methods do not exploit content information.

A content-based video adaptation scheme has been proposed in one of our earlier works [14] which adaptively discards the audiovisual content that cannot be afforded by the network, (e.g., due to the bandwidth variations), at a *minimum content cost*. In particular, the algorithm optimally estimates the amount of information that best represents the content of a given time segment by extracting a single key frame which is then delivered. The algorithm assumes that the frames which are to be delivered are available so that at any time the best representative frame can be calculated. This implies either a buffering delay or that the total video information is accessible since it is stored in a file. Thus, the above mentioned scheme cannot be applied to a real time video capturing procedure. In addition, only a single key frame is extracted quantizing bandwidth variation into rational fractions of 1/*N*.

In this paper, we extend the work of [14] by eliminating the above mentioned constraints. In particular, we consider a) a real time capturing process so that frames that are to be delivered cannot be transmitted before their capturing, b) multiple key frames can be delivered within a time interval by approximating the real bandwidth availability with any rational fraction that is closer to it.

In particular, key-frame prediction is performed by the use of an adaptable neural-network model. The model takes into account content fluctuation in previous times and the current content conditions and recursively estimates the new network weights used in the prediction process. This is due to the fact that content as well as bandwidth characteristics vary from time to time.

The proposed network weight updating is performed in an optimal way so that a) the network response is approximately equal to current conditions (after the adaptation, the network should correctly predict the actual key frames as they are estimated right after the capturing of the respective time segment) and b) a minimal degradation over the previous network knowledge is accomplished. The proposed adaptive neural network architecture actually simulates a recursive implementation of a non-linear autoregressive model (RNAR), which is suitable for complex and non-stationary processes, such as the key frame indices. In contrast to conventional neural network training algorithms, where generally require long training periods, the computational complexity of the proposed scheme is very small and can be applied to real time applications.

This paper is organized as follows. Section II formulates the key frame prediction problem. Section III presents the adaptable model used for key frame prediction, while section IV the optimal weight updating algorithm. In section V, the algorithm adopted for the actual key frame selection is described while in section VI we illustrate the proposed real time video content adaptation. Experimental results and comparisons with other approaches are shown in section VII. Finally, section VIII concludes the paper.

II Key Frame Prediction

Let us assume that at a time *t*=*n*, the encoder should deliver a number of video frames lower than the captured ones since the current network bandwidth, say $B(n)$, is lower than the minimum required for video transmission B_o. The ratio $L(n)= B(n)/B_0$ expresses the proportion of the bandwidth reduction compared to the minimum required. Thus, $L(n)$ expresses the number of frames that should be skipped. Since $L(n)$ is a real number but the number of frames that should be skipped is integer, we bound $L(n)$ by a rational fraction, the numerator of which expresses the number of frames that should be delivered, while the denominator the frame (time) segment within which the frame delivery should be completed. Let us denote this rational fraction as N_o / M_o, where $N_o < M_o$ since only lower bandwidth characteristics are of interest. In order to avoid long time segment which significantly increases the computational complexity of the proposed scheme, we constraint the denominator to be smaller than $M_{\max}$ i.e., $M_o < M_{\max}$.

Thus, N_o is the number of frames that can be delivered within a time segment of M_o frames bounded by the $M_{\max}$. In the framework of this paper, the N_o frames that should be delivered are selected with respect to the content domain. In particular, the most representative frames within the time segment M_o are selected and these frames are transmitted through the low and variable bandwidth network. Thus, in the proposed scheme video delivery is performed using a *content-based sampling* by discarding frames the content of which can be represented by other frames.

Since, however, we are referring to a real time capturing system, the key frames cannot be selected unless capturing of the respective time segment is performed. For this reason, a prediction model is adopted in this paper, able to provide a reliable estimate of the key frame indices. In particular, a neural network model is used as key frame predictor approximating a Non-linear AutoRegressive model (NAR). The network receives input information derived from the previous time segment as well as the evaluation of the current key frame prediction.

It should be mentioned that in such dynamically changed environments, the neural network parameters cannot be considered constant throughout video transmission. This is due to the fact that the content characteristics may vary from time to time and thus the corresponding network weights. For this reason, following network prediction, an evaluation model is activated exploiting the actual results so that in next prediction steps better exploitation of the current content fluctuation is accomplished.

Let us denote as $\mathbf{y}(n) = [y_1(n) \cdots y_{M_{\max}}(n)]^T$ the neural network output that predicts the indices of key frames at time $t=n$ within a time segment of M_o frames. Since $M_{\max}$ is the maximum possible time segment, the neural network outputs are bounded by $M_{\max}$.

Let us assume, without loss of generality, that $0 \le y_i(n) \le 1$. Values of $y_i(n)$ close to one indicate that the respective frame presents high probability of being selected as representative. On the contrary, values close to zero corresponds to frames whose the probability of being selected as key frames are low.

By examining the number $L(n)$ the model can estimate the closest rational fraction N_o / M_o that approximates the real number $L(n)$ as much as possible under the constraint that $M_o < M_{\max}$. This can be mathematically expressed as

$$\{N_o, M_o\} = \arg \min_{\{N<M \& M<M_{\max}\}} \{L(n) - \frac{N}{M}\} \qquad (1)$$

Having estimated the numbers N_o, M_o, we can calculate the number of key frames that should be delivered. Thus, among all values of $y_i(n)$, the N_o elements with the highest values are selected as key frames since these indices represent the highest probability of being these frames the content representatives.

The neural-network model predicts the key frame indices based on the knowledge already obtained in the network by considering the fluctuations of the previously delivered content. In particular, let us denote as $\mathbf{f}_i$ the feature vector of the ith frame of the sequence. Vectors $\mathbf{f}_i$ are calculated in our case as in [14] so that the processing is applied directly on the MPEG compressed domain. More specifically, the color histogram in the Y Cr Cb color space and the histogram of the MPEG motion vectors are used as appropriate features. These descriptors can be directly calculated from the MPEG compressed stream without requiring video decoding, a process that is tedious and time consuming. In particular, to estimate the color histograms in P and B frames with a minimal decoding, the method of [8] is used. More specifically, the method of [8] exploits the motion information of P and B and the known DCT coefficient of the reference I frame to calculate the color characteristics of the P and B frames.

In addition, motion vectors are only available in P and B frames and not in I frames. For the I frames, the same vectors as the ones estimated in the exactly previous P or B frame are taken into account. This assumption is based on the fact that, almost the same motion activity will be encountered within successive frame unless of a shot change.

As a result, real time estimation of the feature elements can be obtained. Figure 1 illustrates a graphical description of the proposed content-based sampling scheme. In particular, at time $t=n$, the current bandwidth availability is calculated and the numbers N_o, M_o are estimated as in (1). Then, the neural network model is used to estimate the key frame indices as will be described in the following and *before capturing* the referring frames. After the frames' capturing, evaluation of the prediction results is accomplished and new network weights are estimated to improve prediction accuracy for the following frames.

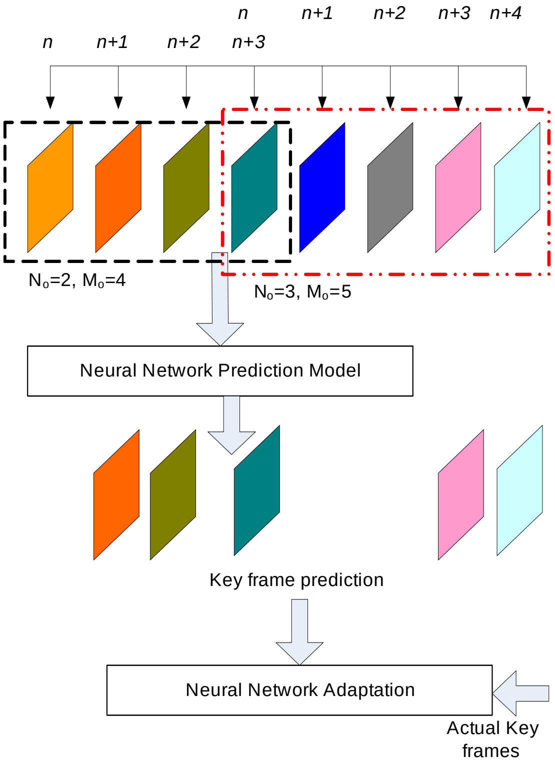


Figure 1. A graphical description of the proposed scheme.

III NON-LINEAR AUTOREGRESSIVE MODEL FOR OPTIMAL KEY FRAME PREDICTION

Let us assume in the following that the key frames selection depends on the previous content characteristics within a time interval p. That is,

$$\mathbf{y}(n) = g(\mathbf{z}(n-1), \mathbf{z}(n-2), ..., \mathbf{z}(n-p)) + e(n) \qquad (2)$$

where $g(\cdot)$ is a non-linear function, and $e(n)$ an independent and identically distributed (i.i.d) error. The variable p denotes the order of the model, i.e., the size of the previous time window that should be used so as to provide a reliable key frame prediction. Vectors $\mathbf{z}(i)$

expresses the magnitude of the respective feature vector $\mathbf{f}_i$ i.e., $\mathbf{z}(i) = \|\mathbf{f}_i\|$.

The main difficulty in implementing the non-linear model of (1) is that function $g(\cdot)$ is actually unknown. However, in [15], it has been shown that a feedforward neural network, is able to implement such a model, within any acceptable accuracy.

Let us denote as $\mathbf{w}_i = [w_{i,1} \cdots w_{i,p+1}]^T$, $i = 1,2,\cdots l$ the $(p+1)\times 1$ vectors containing all weights $w_{i,k}$, $k = 1,\cdots,p$ which connect the *i*th hidden neuron to the *k*th input element and $w_{i,p+1}$ the biases of the *i*th neuron. In this notation, we assume a network of *l* hidden neurons. Let us also define as $\mathbf{v}^{(i)} = [v_1^{(i)} v_2^{(i)} \cdots v_l^{(i)}]^T$, with *i*=1,2,…,$M_{\max}$, an $l \times 1$ vector, which contains the network weights, say $v_j^{(i)}$, connecting the jth hidden neuron to the ith output neuron and as $\theta^{(i)}$ the respective bias. Then, vector $\mathbf{w} = [\mathbf{w}_1^T\ \mathbf{w}_2^T \cdots \mathbf{w}_l^T\ \mathbf{v}^{(1)} \cdots \mathbf{v}^{(M_{\max})}\theta]^T$ represents all network weights and biases among all the $M_{\max}$ outputs. In this case, the network output ith network output, i.e., the ith key frame index, say $y^{(i)}$ is predicted as

$$y(i) = (\mathbf{v}^{(i)})^T \cdot \mathbf{u}(\mathbf{g}) + \theta^{(i)} \tag{3}$$

With

$$\mathbf{u}(\mathbf{g}) = \begin{bmatrix} u_1(\mathbf{g}) \\ \vdots \\ u_l(\mathbf{g}) \end{bmatrix} = \begin{bmatrix} h(\mathbf{w}_1^T \cdot \mathbf{g}) \\ \vdots \\ h(\mathbf{w}_l^T \cdot \mathbf{g}) \end{bmatrix} = \mathbf{h}(\mathbf{W}^T \cdot \mathbf{g}) \tag{4}$$

where $\mathbf{W}$ is a $(p+1)\times l$ matrix, the columns of which correspond to the weight vector $\mathbf{w}_i$, that is $\mathbf{W} = [\mathbf{w}_1\ \mathbf{w}_2 \cdots \mathbf{w}_l]$ and $\mathbf{h}(\cdot)$ a vector-valued function, the elements of which correspond to the activation functions, say $h(\cdot)$, of hidden neurons. In our case, the sigmoid function is used as $h(\cdot)$. Vector $\mathbf{g}$ denotes the neural network input and in our case it is given as the magnitude of the feature vectors of the p previous frames, i.e.,

$$\mathbf{g} = [\mathbf{z}(n-1)\mathbf{z}(n-2)\cdots\mathbf{z}(n-p)\ 1]^T \tag{5}$$

The $\mathbf{g}$ is a $(p+1)\times 1$ input vector containing the *p*-previous samples $\mathbf{z}(n-1)\ \mathbf{z}(n-2)\cdots\mathbf{z}(n-p)$ plus a unity to accommodate the bias effect and the number of activated neurons.

IV OPTIMAL WEIGHT ADAPTATION

In the previous implementation, the model parameters, i.e., the network weights, are considered constant throughout video transmission. However, in dynamic environments, where the system characteristics change through time, this assumption deteriorates the prediction accuracy, since the model response cannot be adapted to current conditions [16]. After capturing, the neural network output can be evaluated and thus the prediction results. In this case, the network weights can be updated so that the prediction is adjusted to the current content fluctuations.

Let us denote as S^c a set, which contains the *actual* indices of the key frames in the M_o time segment after the capturing of all M_o frames, the extraction of the feature vectors $\mathbf{f}_i$ and thus the $\mathbf{z}(i) = \|\mathbf{f}_i\|$ and the application of the key frame selection algorithm which is described in the following section. Key frame selection are defined similarly to the prediction index $\mathbf{y}$(*n*). That is,

$$S^c = \{\cdots,(\mathbf{z}(i),d_i),\cdots\} \tag{6}$$

where $\mathbf{z}(i)$ is the magnitude of the feature vector of the ith frame of the M_o time segment refers to the query feature vector, and d_i the respective probability of this frame of being key frame or not. High values of d_i indicates that the probability of the ith frame to be a key frame is high. On the contrary, as the value of d_i decreases, the probability of selecting the ith frame as key frame is also decreases.

Let us denote by $\mathbf{w}(n)$ the network weights *before* the adaptation at time *t*=*n*. Similarly, let $\mathbf{w}(n+1)$ denote the network weights *after* the adaptation. This means that the following key frames sample will be estimated using the new weights $\mathbf{w}(n+1)$, while the previous ones have been predicted based on the previous weights $\mathbf{w}(n)$. At time *t*=*n*, the bandwidth is measured and the number of key frames required to be transmitted is estimated based on the rational approximation of the real number *L*(*n*). Then, the new weights $\mathbf{w}(n+1)$ are estimated by minimizing the following equation

$$y_i^{\mathbf{w}(n+1)} \approx d_i, \text{ for all } i \in S^c \tag{7}$$

Equation (7) expresses the fact that after capturing, we wish the new network weights to *perfectly* predict the actual key frames as they have been estimated by the algorithm described in section V. We have added the superscript $\mathbf{w}(n+1)$ in the network output y_i to indicate its dependence on the new network weights $\mathbf{w}(n+1)$.

Usually, the number of samples of set S^c is much smaller than the number of coefficients $\mathbf{w}(n+1)$ that should be estimated. Therefore, equation (7) is not sufficient to uniquely identify the parameters $\mathbf{w}(n+1)$ To achieve uniqueness in the solution, an additional requirement is imposed, which takes into consideration the variation of the similarity measure. In particular, among all possible solutions, the one that satisfies (7) and simultaneously causes a minimal modification of the network weights is chosen. That is

$$\hat{\mathbf{w}}(n+1) : y_i^{\mathbf{w}(n+1)} = z(i) \tag{8a}$$

$$s.t.\ \min_{\mathbf{w}} \left\| \mathbf{w}(n+1) \right\| \qquad (8b)$$

In equation (8), we denote as $\hat{\mathbf{w}}(n+1)$ the optimal network weights. The constraint term of equation (8a) indicates that, the proposed on-line learning strategy modifies the similarity measure so that, after the adaptation, the current content fluctuation is satisfied as much as possible. On the contrary, the term of (8b) expresses that the adaptation is accomplished with a minimal modification of the already estimated network knowledge.

IV.1 *Recursive Estimation of the Network Weights*

In this section, a recursive algorithm is presented to perform the constraint minimization of (8). Therefore, the scheme yields to a *Recursive Weight Estimation* algorithm.

Let us consider that at time *t=n* the network is represented by the weights $\mathbf{w}(n)$.

Let us now assume, that the model parameters at the (*n*+1)th iteration, i.e., the $\mathbf{w}(n+1)$, are related to the model parameters $\mathbf{w}(n)$ at the nth iteration as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \Delta\mathbf{w} \qquad (9)$$

where $\Delta\mathbf{w}$ refers to a small increment of the model coefficients. Equation (9) indicates that a small modification of the coefficients is adequate to satisfy the current content fluctuation as expressed by (8a).

In the following, we deal with the analysis of equation (8a), i.e., the constraint of the minimization. In particular, based on equation (9), linearization of the non-linear activation functions $h(\cdot)$ is permitted using a first order Taylor series expansion. Then, equation (8a) can be decomposed in a system of linear equations, as indicated by the following theorem

Theorem 1: The constraint expressed by equation (8a) under the assumption of (9) is decomposed to a system of linear equations of the form $\mathbf{c}(n) = \mathbf{A}(n) \cdot \Delta\mathbf{w}$, where vector $\mathbf{c}(n)$ and matrix $\mathbf{A}(n)$ depends only on the neural network eights at the following iteration.

The proof of Theorem 1 is given in [17].

□

Vector $\mathbf{c}(n)$ expresses the difference between the desired probability of a frame to be selected as key frame and the one provided by the system before frame capturing, i.e., using the weights $\mathbf{w}(n)$. In particular, vector $\mathbf{c}(n)$ is given as

$$\mathbf{c}(n) = [\cdots c_i(n) \cdots]^T \qquad (10)$$

with

$$c_i(n) = d_i - y_i^{\mathbf{w}(n)}(n) \qquad (11)$$

Furthermore, matrix $\mathbf{A}(n)$ is given as

$$\mathbf{A}(n)^T = \left[\ \cdots \mathbf{a}_i(n) \cdots \right] \qquad (12)$$

where the columns $\mathbf{a}_i(n)$ are appropriately defoned with respect to the previous network weights $\mathbf{w}(n)$,

$$\mathbf{a}_i(n) = [\mathrm{vec}\{(d_i - y_i^{\mathbf{w}(n)}(n)) \cdot (\mathbf{g}^{(r)})^T\}^T \quad \mathbf{u}(r)^T]^{\ T} \quad (13)$$

where

$$\mathbf{u}(n) = \mathbf{h}(\mathbf{W}^T(n) \cdot (d_i - y_i^{\mathbf{w}(n)}(n))) \qquad (14)$$

with $\mathbf{h}(\cdot) = \underbrace{[h(\cdot) \ldots h(\cdot)]}_{L\ \text{times}}^T$ a vector containing the activation functions $h(\cdot)$. Vector $\mathbf{g}(n)$ is given as follows

$$\mathbf{g}(n) = \mathbf{D}(n) \cdot \mathbf{v}(n) \qquad (15)$$

with matrix $\mathbf{D}(n)$ expresses the derivatives of the elements of vector $\mathbf{u}(n)$, i.e.,

$$\mathbf{D}(n) = diag\{\delta_1(n), \cdots, \delta_L(n)\} \qquad (16)$$

In (16) $diag\{\}$ refers to a diagonal matrix.

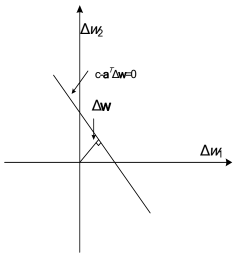
Based on the previous equations, it can be seen that, vector $\mathbf{c}(n)$ and matrix $\mathbf{A}(n)$ are *only* related with the coefficients $\mathbf{w}(n)$ at the *t=n* time segment.

The second constraint as expressed by (8b) is analyzed using Lagrange multipliers. In this case, the aforementioned minimization problem is written as

$$\Delta\mathbf{w} = \underset{\Delta\mathbf{w}}{\mathrm{argmin}}\left((\Delta\mathbf{w})^T \cdot \Delta\mathbf{w} + \boldsymbol{\lambda}^T \cdot (\mathbf{c}(n) - \mathbf{A}(n) \cdot \Delta\mathbf{w})\right) \quad (17)$$

where the elements of vector $\boldsymbol{\lambda}$ corresponds to the Lagrange multipliers. Differentiating equation (17) with respect $\Delta\mathbf{w}$ and $\boldsymbol{\lambda}$ and setting the results equal to zero, we obtain

$$\Delta\mathbf{w} = \mathbf{A}^T(n) \cdot (\mathbf{A}(n) \cdot \mathbf{A}^T(n))^{-1} \cdot \mathbf{c}(n) \qquad (18)$$



A two-dimensional graphical representation of the proposed approach is shown in Figure 2.

Figure 2. A graphical representation of the proposed optimal small weight perturbation.

V INITIAL NEURAL NETWORK TRAINING-ACTUAL KEY FRAME ESTIMATION

The desire actual key frames are estimated in our case as follows. Let us also denote as *J*(*k*) the energy of shape coefficients to the *k*th previous frame among the *p* available. Thus, index *k*=1,…,*p*. To calculate the desire values, initially, the first derivative of signal *J*(*k*), say $J'(k)$, is evaluated with respect to time index *k*. Since, however, variable *k* takes values in discrete time, the first derivative is approximated as the difference of feature vectors between two successive frames $Jk) = J(k+1) - J(k)$.However, the previous operator is rather sensitive to noise since differentiation of a signal stresses the high pass components. For this reason, a weighted average of the first derivative, say J'_w, over a window, is used to eliminate the noise influence. Particularly, the weighted first derivative is given as

$$J'_w(k) = \sum_{l=\alpha_1(k)}^{l=\beta_1(k)} w_{l-k}\big(J(l+1) - J(l)\big), k\text{=}0,\ldots,p\text{-}2 \qquad (19)$$

where $\alpha_1(k) = \max(0, k - N_w)$, and $\beta_1(k) = \min(p-2, k+N_w)$ and 2*N_w+1 is the length of the window, centered at frame *k*. It can be seen from (19) that the window length linearly reduces previous segment limits used for key frame prediction. The weights w_l are defined for $l \in \{-N_w, N_w\}$; in the simple case, all weights w_l are considered equal to each other, meaning that the derivatives of all frame feature vectors within the window interval present the same importance,

$$w_l = \frac{1}{(2N_w + 1)}, \; l\text{=-}N_w,\ldots,N_w \qquad (20)$$

Similarly the second weighted derivative, $J''_w(k)$, for the *k*-th frame is defined as

$$J''_w(k) = \sum_{l=\alpha_2(k)}^{l=\beta_2(k)} w_{l-k} J''(k) \qquad (21)$$

where $J''(k) = J'(k+1) - J'(k)$, k=0,…,p-3 and $\alpha_2(k) = \min(0, k - N_w)$, $\beta_2(k) = \min(p-3, k+N_w)$

As explained previously, the local maxima and minima of J'' are considered as appropriate curve points, i.e., as time instances for the selected key-frames. Note that J'' is a discrete time sequence. Hence, as the value of the weighted second derivative reaches zero, the highest (closest the one) the probability of selecting the respective frame as key frame. On the contrary, as the weighted derivative increases the probability of the frame to be selected as representative decreases. Using such a notation, the probabilities are examined and the set of probable actual key frames is defined to estimate the desire vector *di*.

VI REAL TIME VIDEO CONTENT ADAPTATION

In this section, we describe the concept behind the algorithm for transmitting selectively frames over a network of low and time-variable bandwidth.

Without loss of generality, let us assume that the first key frame among the N_o in the time segment M_o is the one with index *J*, where *J* is an integer 0≤*J*<M_o equal to the time segment interval. It is clear that, upon a key frame selection (i.e., selection of the most representative frame among the candidate for skipping), all the previous frames should be discarded since a later transmission of a previous frame cannot be allowed. However, during the transmission of the *J* frame, it is probable for the network bandwidth to change. Two different cases can be discriminated.

The first concerns an increase in the network bandwidth, whereas the second a decrease. If the network bandwidth increases, the transmission of the representative frame will be completed at a time $t<M_o$. On the contrary, if the network bandwidth decreases, the representative will be delivered at a time $t \geq M_o$. In the first case, however, frames whose indices are smaller than *J* cannot be considered, though the bandwidth increase may allow such as delivery. Let us denote as *c* the completion time of the representative frame. Then, the new time *t*=*n* is updated as follows

$$n_{new} = \max(J, c) \qquad (22)$$

where $c \geq M_o$ for a bandwidth decrease, while $c<M_o\text{-}1$ for a bandwidth increase.

At times t=n_{new} a new network adaptation is activated and the recursive algorithm proposed in the following sections is activated.

VII EXPERIMENTAL RESULTS

In this section, we evaluate the proposed scheme by comparing it with the standard frame - skipping method and the method proposed in [11] (the Single Key Frame Method) and [14]. It should be mentioned that the proposed technique can be applied in case of a real-time video capturing instead of the compared approaches which require frame buffering (i.e., frame delay). In the experiment conducted, we use a concatenation of the well known sequences "foreman", "container", "mobile" and "tempete", which include color variation and motion giving. Thus, a sequence of 1,155 frames duration is created.

We modeled *B*(*n*) as a normal distribution with mean value μ varying from 0.1 μ to 0.5 μ. The bandwidth was changing value every Y frames, where Y was getting values from a normal distribution with mean value equal to 30 and standard deviation 10. No buffer is used instead of the method of [14], yielding a real-time video capturing. For each frame, the features described in section II are extracted without requiring decoding of the video sequence.

An objective evaluation criterion is adopted which compares the actually delivered video sequences, by skipping frames that cannot be afforded due to network bandwidth variations with the real ground truth sequences. The comparison is performed on the feature–based space since this better represents video content. Let us denote as $\mathbf{f}_i^a$ the feature vectors of the *i*th frame of the ground sequence and as $\mathbf{f}_i^d$ the feature vector for the *i*th frame of the actually delivered sequence, using some video adaptation scheme, (for example the proposed method or the technique of [14]). Then, as evaluation criterion, say E, we define

$$E = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{f}_i^a - \mathbf{f}_i^d\right\| \tag{23}$$

It should be mentioned that, in the delivered sequence, there are frames that have not been transmitted due to bandwidth constraints (frame loss). For these frame, vector $\mathbf{f}_i^d$ does not exist. We assume that the feature vectors of the lost frames are the same as the vectors of the exactly previously transmitted frames, i.e., the video freezes.

Having defined the criterion E, we can define next the improvement ratio I as the "gain" of our method compared to the other approached. More specifically, I is given as

$$I = (E_{comp}\text{-}E_{net})/E_a \tag{24}$$

where E_{net} is the error of (23) achieved using our technique, while E_{comp} the error of (23) obtained using another compared method.

A feedforward neural network is used to predict the key frames. The network is trained using the proposed adaptable algorithm as described in sections III-V, whereas the network structure, i.e., the number of neurons, remains the same. In the specific experiments, a neural network of 30 hidden neurons is chosen. For the initial training of the network, content characteristics of 20 shots, different than that used in the experimental sequence and with an average duration of 202 frames, are taken into account. For each shot, the color and motion features of the MPEG sequence are exploited.

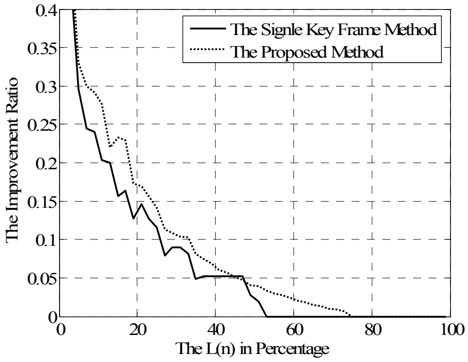


Figure 3. The improvement ratio I versus $L(n)$ for a bandwidth variation of 0.1 μ. *Solid line*: The improvement ratio of the method of [14] compared to the method of [11]. *Dotted line*: The improvement ratio of the proposed method compared to the method of [11].

Figure 3 illustrated the improvement ratio I with respect to the current bandwidth condition, as expressed in percentage compared to the B_o. More specifically, the horizontal axis is the number $L(n)$. The results have been derived for a bandwidth variation of 0.1 μ standard deviation. As is observed, there is an increase of the performance for all $L(n)$ values, whereas the proposed method outperforms both the compared ones. For high values of the bandwidth, i.e., closer to B_o the improvement decreases while for values close to B_o it reaches zeros, i.e., the proposed scheme presents the same performance as of [14 and [11] (the Single Key Frame Method). The solid line of Figure 3 indicates the improvement ratio I as derived by the method of [14] compared to the method of [11] (the Single Key Frame Method). It is clear that for all values of $L(n)$ the ratio I is lower than that derived from our method, indicating that the proposed neural network model correctly predicts the sequence key frames. The slight deterioration of I is due to the fact that in [14], only a single frame is extracted and thus content fluctuation cannot be efficiently described. It should be mentioned that in this case, a buffer of 30 frames length is adopted since real time video capturing is not compatible with the algorithm of [14].

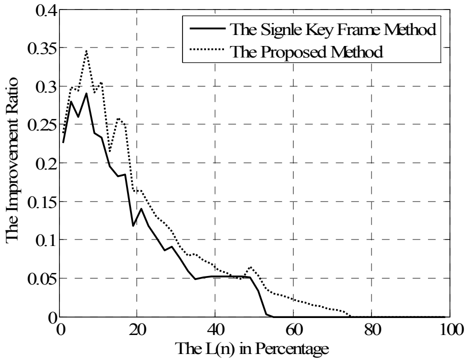


Figure 4. The improvement ratio I versus $L(n)$ for a bandwidth variation of 0.2 μ. *Solid line*: The improvement ratio of the method of [14] compared to the method of [11]. *Dotted line*: The improvement ratio of the proposed method compared to the method of [11].

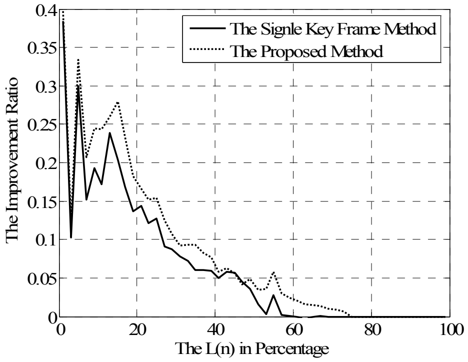


Figure 5. The improvement ratio I versus $L(n)$ for a bandwidth variation of 0.5 μ. *Solid line*: The improvement ratio of the method of [14] compared to the method of [11]. *Dotted line*: The improvement ratio of the proposed method compared to the method of [11].

The same conclusions are drawn from the Figures 4 and 5 which show the improvement ratio I with respect to the values of $L(n)$ but for different values of bandwidth variation. In particular, in Figure 4 the results have been obtained using a bandwidth variation of 0.2 μ, while in

Figure 5 of 0.5 μ. In all cases, the proposed neural network-based scheme better exploits bandwidth resources by delivering a sequence the content follows the actual content variations and characteristics.

VIII Conclusions

In this paper, a new content-based sampling algorithm is proposed appropriate for video delivery of low and variable communication networks, such as the wireless ones. The algorithm requires no buffer and thus it can be applied for real-time video capturing instead of the most of the previous approaches which introduce buffer delays. An adaptable neural network model is used for key frame prediction based on the previous content variation. The network weights are updated each time the frame capturing is completed so that the network trusts the current content fluctuation as much as possible while on the other hand a minimum variation of the already network knowledge is accomplished.

Experimental results indicate the out-performance of the proposed scheme compared to the algorithm of [11] and our earlier approach of [14]. The improvement ratio is more evident in cases of low bandwidth conditions and thus of high congestion.

IX References

[1] ISO/IEC JTC1/SC29/WG11 N3156, "MPEG-4 Overview," Doc. N3156, Maui, Hawaii, December 1999.

[2] W. Li, “Overview of Fine Granularity Scalability in MPEG-4 Video Standard,” *IEEE Trans. CSVT*, vol. 11, no. 3, pp. 301–317, Mar. 2001

[3] MPEG Video Group, “MPEG-4 Video Verification Model-Version 2.1,” ISO/IEC JTCI/SC29/WG11, May 1996.

[4] Qingwen Hu and S. Panchanathan, “Image/video spatial scalability in compressed domain,” *IEEE Trans. on Industrial Electronics,* Vol. 45 , No. 1 pp. 23-31, Feb. 1998.

[5] M. Domanski, A. Luczak, and S. Mackowiak, “Spatio-temporal scalability for MPEG video coding”, *IEEE Trans. CSVT*, Vol.10, No. 7, pp.1088-1093, Oct. 2000.

[6] M. van der Schaar and H. Radha,“A hybrid temporal-SNR fine-granular scalability for Internet video” *IEEE Trans. on CSVT*, Vol. 11, No.3, pp. 318-331, March 2001

[7] N. V. Patel and I. K. Sethi, “Video shot detection and characterization for video databases,” *Pattern Recognition*, Vol. 30 No. 4, pp. 583-592, April 1997.

[8] B. L. Yeo and B. Liu, “Rapid scene analysis on compressed videos,” *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 5, pp. 533- 544, Dec. 1995.

[9] A. Doulamis, N. Doulamis, and S. Kollias, "A fuzzy video content representation for video summarization and content-based retrieval," *Signal Processing*, Vol. 80, pp. 1049-1067, June 2000.

[10] J. Nam and A. H. Tewfik, “Video Abstract of Video,” *Proc. of the IEEE Inter. Workshop on Multimedia Signal Processing*, pp. 117-122, Copenhagen, Denmark, Sept. 2000.

[11] M.A. Bonuccelli, F. Lonetti, and F. Martelli, “Temporal Transcoding for Mobile Video Communication,” *Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous 2005)* July 17-21, 2005, San Diego, California.

[12] H. Shu and L.-P. Chau. Frame-skipping transcoding with motion change consideration. In *Proc. of IEEE ISCAS 2004*, Vol. 3, pp. 773–776, Vancouver, Canada, May 2004.

[13] J.-N. Hwang, T.-D. Wu, and C.-W. Lin. Dynamic frameskipping in video transcoding. In *Proc. of IEEE Workshop on Multimedia Signal Processing*, pp. 616–621, Redondo Beach, CA, USA, Dec. 1998.

[14] A. Doulamis, D. Kosmopoulos and N. Doulamis, “Content based Frame Sampling for Efficient Video Delivery over Low/Varying Bandwidth Networks,” *submitted to the IEEE International Conference on Multimedia and Expo (ICME)*, Toronto, Canada, 2006.

[15] J. Connor, D. Martin and L. Altas, “Recurrent Neural Networks and Robust Time Series Prediction,” *IEEE Trans. on Neural Networks,* vol. 5, no. 2, pp. 240-254.

[16] A. Doulamis, N. Doulamis and S. Kollias, “On Line Retrainable Neural Networks: Improving the Performance of Neural Networks in Image Analysis Problems,” *IEEE Trans. on Neural Networks*, vol. 11, no.1, January 2000.

[17] A. D. Doulamis, N. D. Doulamis and S. D. Kollias, "An Adaptable Neural Network Model for Recursive Non-Linear Traffic Prediction and Modeling of MPEG Video Sources," *IEEE Trans. on Neural Networks*, Vol.14, No. 1, pp. 150-166, January 2003.