Error Concealment Method Based on Motion Vector Prediction Using Particle Filters

Branislav HRUŠOVSKÝ, Ján MOCHNÁČ, Stanislav MARCHEVSKÝ

Dept. of Electronics and Multimedia Communications, Faculty of Electrical Enginnering and Informatics
Technical University of Košice, Letná 9, 040 01 Košice, Slovakia

branislav.hrusovsky@tuke.sk, jan.mochnac@tuke.sk, stanislav.marchevsky@tuke.sk

Abstract. *Video transmitted over unreliable environment, such as wireless channel or in generally any network with unreliable transport protocol, is facing the losses of video packets due to network congestion and different kind of noises. The problem is becoming more important using highly effective video codecs. Visual quality degradation could propagate into subsequent frames due to redundancy elimination in order to obtain high compression ratio. Since the video stream transmission in real time is limited by transmission channel delay, it is not possible to re-transmit all faulty or lost packets. It is therefore inevitable to conceal these defects. To reduce the undesirable effects of information losses, the lost data is usually estimated from the received data, which is generally known as error concealment problem. This paper discusses packet loss modeling in order to simulate losses during video transmission, packet losses analysis and their impacts on the motion vectors losses.*

Keywords

Packet, loss, error concealment, particle filter.

1. Introduction

Packet data transmitted over wireless environment, e.g. WiFi, or in generally any network with unreliable transport protocol, is facing the losses of packets due to network congestion and noises of different kinds. If video signals coded with some of advanced video coding standard are transmitted, these losses have severe impact on resulting video quality due to highly effective redundancy elimination in video coding process. Visual quality degradation could propagate to the subsequent frames due to redundancy elimination in order to gain high compression ratio. Therefore it is necessary to know in which way the packets are lost and one of the possible ways to learn about losses is creation of networks model.

The impact of packet loss can be studied from recorded measurement traces of traffic and loss patterns. To generate error process with similar characteristics as observed in measurements, stochastic model can be modeled [1]. The most popular examples of such models are discrete-time Markov chain models. The use of discrete-time Markov chain models, particularly the 2-state Markov chain model (sometimes called the Gilbert model) has been proposed in [2]. Discrete-time Markov chain models of increasing levels of complexity, including the 2-state Markov chain model have been described in [2], [3].

Obviously, the usage of Gilbert model is quite simple, but its major drawback is inability to correctly model heavily tailed error runs. In such cases, hidden Markov models with up to five states are used to model the distribution of error and error-free burst lengths [4].

Consequently, appropriate error control, recovery or error concealment methods, which have been developed over the times, can be chosen.

On the one hand, traditional error control and recovery methods for data communication are focused on lossless reconstruction of damaged video signal, but they also increase amount of data to be transmitted. However, these techniques introduce some redundancy. On the other hand, signal reconstruction and error concealment have been proposed to obtain close approximation of the original signal or attempt to make the output signal at the decoder less objectionable to human eyes [5].

Error concealment methods can be classified into three categories: 1) spatial, 2) temporal, 3) hybrid. Spatial error concealment techniques use the information from the surrounding correctly received or already concealed blocks to reconstruct damaged area. Typical representative of this class is weighted pixel averaging algorithm. Temporal error concealment techniques use the information of the corresponding blocks from the previous/successive blocks to conceal corrupted block. Typical representative of temporal error concealment methods is boundary matching algorithm and also methods based on Bayesian filtering theory. Hybrid error concealment techniques use the information from the spatial domain as well as information from the temporal domain [5].

This paper is focused on packet loss analysis resulting in creation of packet loss model for fixed network topology, as well as for wireless topology. Consequently, corrupted video sequences were concealed with particle filter based error concealment.

2. Packet Loss Modeling

In order to evaluate quality of transmission, let's have random variable *X*. If packet is not lost, then $X = 0$, otherwise $X = k$ for k lost packets. After that, we can build loss model with infinite number of states (m is infinite value – see Fig. 1). Such model gives us opportunity to model packet loss probabilities in dependence on burst lengths (several consecutively lost packets). For each additional lost packet, which adds to the length of a loss burst, a state transition takes place. If packet is correctly received, then the state returns to $X = 0$ [8].

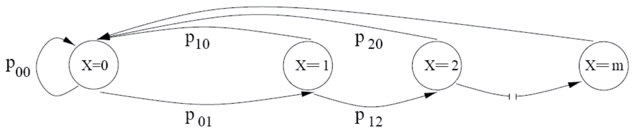

Fig. 1. Loss model with infinite number of states ($m \rightarrow \infty$).

State probability for system with $k > 0$ is $P(X \geq k)$. For finite number of received packets a, state probabilities of the system with $k > 0$ can be approximated with cumulative loss rate [8]:

$$p_{L,cum}(k) = \sum_{n=k}^{\infty} p_{L,n} \;. \tag{1}$$

Cumulative loss rate for $k = 0$, thus for no loss case, can be computed using the following equation:

$$p_{L,cum}(k=0) = 1 - \sum_{n=k}^{\infty} p_{L,cum}(k) =$$

$$= 1 - \sum_{k=1}^{\infty} \frac{\sum_{n=k}^{\infty} o_n}{a} = 1 - \sum_{k=1}^{\infty} \frac{ko_k}{a} = 1 - p_L \tag{2}$$

where $o_k(o_n)$ is occurrence of loss with length k (n).

2.1 Loss Model with Limited Number of States

In order to model packet losses during video or audio transmissions, it is sufficient to use models with limited number of states due to strict requirements, which have to be fulfilled, i.e. it is useless to transmit multimedia through a network with long consecutive packet losses.

Three the most commonly used models with limited number of states are: the *k*-th order Markov chain model, the 2-state Markov chain model and the Bernoulli loss model.

2.2 *K*-th Order Markov Chain Model

K-th order Markov chain model has similar performance measures as model with infinite number of states, however state probability for finite state m is added and also probability for transition from m to m is added. Thus for cumulative loss holds the following term [8]:

$$p_{L,cum}(k) = \sum_{n=k}^{\infty} \frac{o_k}{a} \;. \tag{3}$$

For burst loss $k = m$ holds:

$$p_{L,m}(k) = \sum_{n=m}^{\infty} \frac{(n-m+1)o_n}{a-m+1} \tag{4}$$

and for conditional loss $k = m$ holds:

$$p_{L,m}(k) = \sum_{n=m}^{\infty} \frac{(n-m)o_n}{d-m} \tag{5}$$

where d is the number of dropped packets:

$$d = \sum_{k=1}^{\infty} ko_k \;. \tag{6}$$

K-th order Markov chain model is also known as Extended Gilbert model [9].

2.3 Gilbert Model

Packet loss measurements on the Internet have shown that the probability of loss episodes of length k decreases approximately geometrically with increase of k [10]. Thus it is possible to use simpler packet loss model, e.g. Gilbert model.

A special case of *k*-th order Markov chain model is Gilbert model with $k = 2$, see Fig. 2.

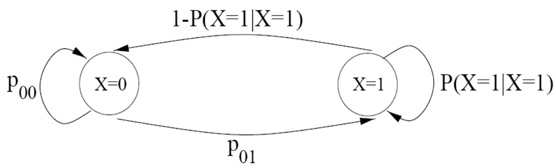

Fig. 2. Gilbert model.

In this model, 0 represents state with no packet loss and on the other hand 1 represents the state of packet being lost.

The matrix for transition probabilities and for state probabilities can be expressed in form:

$$\begin{bmatrix} 1-p_{01} & p_{10} \\ p_{01} & 1-p_{10} \end{bmatrix} \begin{bmatrix} P(X=0) \\ P(X=1) \end{bmatrix} = \begin{bmatrix} P(X=0) \\ P(X=1) \end{bmatrix}. \tag{7}$$

For unconditional probability $P(X = 1)$ holds the following equation:

$$P(X=1) = \frac{p_{01}}{p_{01} + p_{10}} \;. \tag{8}$$

If previous packet is lost, then for conditional probability of having loss holds:

$$P(X=1 \mid X=1) = 1 - p_{10} \;. \tag{9}$$

Gilbert model memorizes only the previous state, thus the probability, that the next packet will be lost is dependent only on the previous state.

Transition probabilities p_{01} and p_{10} can be expressed with the following equations:

$$p_{01} = P(X = 1 \mid X = 0) = \sum_{k=1}^{\infty} \frac{o_k}{a}, \quad (10)$$

$$1 - p_{10} = P(X = 1 \mid X = 1) = \frac{\sum_{k=1}^{\infty} (k-1) o_k}{d-1}. \quad (11)$$

The probability of having a lost episode with length k [10]:

$$p_k = (1 - p_{10})^{k-1} p_{10}. \quad (12)$$

2.4 Bernoulli Model

The simplest way, how to model packet losses, is using Bernoulli model. In this model, the probability, that the packet is lost or correctly received is independent on all other values [2].

Bernoulli model can be characterized by a single parameter r, which describes the probability of packet being lost [2]:

$$\hat{r} = \frac{n_i}{n} \quad (13)$$

where n_i is the number of times, when packet loss has occurred and n is the total number of packets.

3. Bayesian Filtering

Filter theory is the theory of sequentially estimating the underlying state of a system using measurements obtained over time [11]. Bayesian approach to the filtering provides base for the dynamic state estimation problems. Bayesian filters provide a statistical tool for dealing with measurement uncertainty. Bayesian filters estimate a state of dynamic system from noisy observations. These filters represent the state by random variable and in each time step, probability of distribution over random variable represents the uncertainty. [6]

In Bayesian approach, we attempt to construct the posterior PDF of the state given the all measurements. All available information is used to form such PDF. Thus, this PDF represents complete solution [12].

Let x_k, $k \in \mathrm{N}$ be the state sequence [12]:

$$x_k = f_k(x_{k-1}, u_{k-1}, v_{k-1}) \quad (14)$$

where f_k is in generally non-linear function of the previous state $x_{k-1} \in R^{n_x}$, $v_{k-1} \in R^{n_v}$ is state noise, $u_{k-1} \in R^{n_u}$ is known input, n_x, n_v, n_u are dimensions of the state, process and input noise vectors.

Next, let z_k be the measurement [12]:

$$z_k = h_k(x_k, n_k) \quad (15)$$

where $z_k \in R^{n_z}$, h_k is non-linear measurements function, $n_k \in \mathrm{N}^{n_n}$ is measurement noise, n_z, n_n are dimensions of the measurement and measurement noise vectors. We want to find estimate of the x_k based on all available measurements at time k (marked as $z_{1:k}$) by constructing the posterior PDF $p(x_k, z_{1:k})$. It is assumed, that initial PDF $p(x_0 | z_0) \equiv p(x_0)$ is available. Posterior PDF can be obtained recursively in two stages, namely prediction and update. Suppose, that required PDF $p(x_{k-1} | z_{1:k-1})$ at time step k - 1 is available. Then using the system model it is possible to obtain the prior PDF at time step k [12]:

$$p(x_k \mid z_{1:k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid z_{1:k-1}) dx_{k-1}. \quad (16)$$

Prediction step usually deforms, spreads state PDF due to noise.

Measurement z_k is available at time step k, so it can be used to update the prior. Using Bayes rule, we obtain :

$$p(x_k \mid z_{1:k}) = \frac{p(z_k \mid x_k) p(x_k \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})} \quad (17)$$

where the normalizing constant is:

$$p(z_k \mid z_{1:k-1}) = \int p(z_k \mid x_k) p(x_k \mid z_{1:k-1}) dx_k. \quad (18)$$

In the update equation (18), the measurement z_k is used to modify the predicted prior from the previous time step to obtain PDF of the state.

Equations (16) and (17) theoretically allow optimal Bayesian solution. But it is only conceptual solution and integrals in these equations are intractable. Optimal solution exists in some restricted cases such as Kalman Filter and grid-based filters, but in some situations assumptions for these groups of filters do not hold. Then the use of suboptimal solution like particle filters and extended Kalman filters are suitable.

4. Kalman Filter

Kalman filter together with its basic variants are commonly used tools in statistical signal processing, especially in the context of causal, real-time applications.

There are several approaches in the derivation of the Kalman filter. We can assume Gaussian distribution of the deriving process and of the initial state. In the next phase, we derive the posterior distribution of the states given the observations, taking the mean of the resulting distributions as the estimation of the state. The second approach combines a recursive weighted least-squares method with special weighting of the previous estimate of the states in the role of additional measurements [6].

To model state of the internal process, let's assume that posterior density in time k - 1, $p(x_{k-1} | z_{k-1})$, is Gaussian. Hence, $p(x_k | z_k)$ is also Gaussian. Next, random variables v_{k-1} and n_k are independent with normal probability distributions and with covariances labeled as Q_{k-1} and R_k. $f_k(x_{k-1} | v_{k-1})$ and $h_k(x_k | n_k)$ are linear functions. Hence, equations (14) and (15) for derivation of the optimal Bayesian solution can be rewritten to the form:

$$x_k = F_k x_{k-1} + B_k u_k + v_{k-1}, \tag{19}$$

$$z_k = H_k x_k + n_k \tag{20}$$

where F_k and H_k are matrices defining the linear function [6]. In practice, these matrices and covariance matrices Q_{k-1}, R_k might change with each time step or measurement. Since the Kalman filter is recursive estimator, only estimated state from the previous time step and measurement at the current time step are needed to compute current state [7].

Kalman filter is based on Bayesian filtering, and thus it works also in the two phases: prediction and update. Predict stage can be described with the following two equations:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_{k-1} u_{k-1} \tag{21}$$

where $\hat{x}_{k|k}$ is the estimate of the state at time k given observations up to time k and

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_{k-1} \tag{22}$$

where $P_{k|k}$ is the error covariance matrix. Update stage can be described with the following equations :

$$\tilde{y}_k = z_k - H_k \hat{x}_{k|k-1} \tag{23}$$

where $\tilde{y}_k$ is innovation term,

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{24}$$

where S_k is innovation covariance and R_k is covariance of n_k,

$$K_k = P_{k|k-1} H_k^T + S_k^{-1} \tag{25}$$

where K_k is Kalman gain,

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{26}$$

is update state estimate and

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{27}$$

is update estimate covariance.

With using least-square methods we obtain the same results. By using least-square method all the distributions are described by their means and covariances in the derivation procedure.

4.1 Extended Kalman Filter

Kalman filter can be used in estimation of the $x_k \in R^{n_x}$ where posterior PDF is Gaussian in every time step. But in many cases this PDF is not-Gaussian and we need to use different approach such as approximate grid-based method or extended Kalman filter. These methods are also labeled as sub-optimal algorithms [5], [6].

Again, let $x_k \in R^{n_x}$ be the state sequence, but in opposite to the previous case, process is governed by the nonlinear stochastic difference equation:

$$x_k = f(x_k, u_k) + v_{k-1} \tag{28}$$

with measurement $z_k \in R^{n_z}$:

$$x_k = h(x_k) + n_k \tag{29}$$

where v_{k-1} and n_k represent process and measurements noise vectors. In this case, functions f and h are non-linear. Function f can be used to compute state in time step k from the previous estimate and function h can be used to compute the predicted measurement from the predicted state.

Extended Kalman filter is based upon approximation of the Bayes rule using linearization. Again, as well as Kalman filter, its extended version works also in two phases: prediction and update. Predict stage can be described using following equations:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k) \tag{30}$$

where $\hat{x}_{k|k-1}$ is the estimate of the state at time k given observations up to time k-1 and

$$P_{k|k-1} = \hat{F}_k P_{k-1|k-1} \hat{F}_k^T + Q_k \tag{31}$$

where $P_{k|k-1}$ is the error covariance matrix. Update stage can be described with the following equations:

$$\tilde{y}_k = z_k - h(\hat{x}_{k|k-1}) \tag{32}$$

where $\tilde{y}_k$ is innovation term,

$$S_k = \hat{H}_k P_{k|k-1} \hat{H}_k^T + R_k \tag{33}$$

where S_k is innovation covariance and R_k is covariance of n_k,

$$K_k = P_{k|k-1} \hat{H}_k^T + S_k^{-1} \tag{34}$$

where K_k is Kalman gain,

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{35}$$

is update state estimate and

$$P_{k|k} = (I - K_k \hat{H}_k) P_{k|k-1} \tag{36}$$

is update estimate covariance. State transition and observation matrices are defined by following equations:

$$F_k = \left. \frac{\partial f}{\partial x} \right|_{\hat{x}_{k-1}|k-1, u_k}, \tag{37}$$

$$H_k = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_k|k-1} . \quad (38)$$

Equations for the extended Kalman filter shown above utilizes first term in a Taylor expansion of the non-linear function. Utilizing higher order terms is possible, but computational complexity prohibited their use [15].

5. Particle Filter

Particle filter is based on posterior density representation through set of random samples with associated weights. Subsequently, this information provides base for estimate computation [13].

Let the posterior density $p(x_{1:n}, z_{1:n})$ be represented by random measure $\{x_{1:n}^{(m)}, w_n^{(m)}\}_{m=1}^{M}$ in every time step, $x_n^{(m)}$ is the m-th signal particle at the time step n, $x_{1:n}^{(m)}$ is the m-th signal trajectory, $w_n^{(m)}$ is the m-th particle trajectory at the time step n.

The weights are normalized:

$$\sum_i w_k^i = 1 . \quad (39)$$

Posterior density at the time step n can be computed according to the following equation:

$$p(x_{1:n} \mid z_{1:n}) \approx \sum_{m=1}^{M} w_n^{(m)} \delta(x_{1:n} - x_{1:n}^{(m)}) . \quad (40)$$

Particle filters are based on three operations:

- particle generation,
- particle weights computation,
- resampling.

The first two steps are also called sequential importance sampling (SIS). The filter making use of all three steps is called sample importance resampling (SIR) [15].

5.1 Particle Generation

The particles $x_n^{(m)}$ are generated from importance density function $\pi(x_n)$. If importance density function

$$\pi(x_{1:n}) = \pi(x_1 \mid z_1) \prod_{k=1}^{n} \pi(x_k \mid x_{1:k-1}, z_{1:k}) \quad (41)$$

is chosen, then it is possible recursively compute the weights of each particle according the following equation:

$$x_n^{(m)} \approx \pi(x_n \mid x_{n-1}^{(m)}, z_{1:n}) . \quad (42)$$

The importance density function is the important part of particle filter designing, while it generates particles through which a desired probability density function is expressed [13].

5.2 Particle Weights Computation

Particles weights computation and normalization are parts of this step. If the importance density function is described with (41), then the weights are updated according to the following equation [15]:

$$w_n^{*(m)} = w_{n-1}^{m} \frac{p(z_n \mid x_n^{(m)}) p(x_n^{(m)} \mid x_{n-1}^{(m)})}{\pi(x_n^{(m)} \mid x_{1:n-1}^{(m)}, z_{1:n})} \quad (43)$$

and normalization is done by

$$w_n^{(m)} = \frac{w_n^{*(m)}}{\sum_{j=1}^{M} w_n^{*(j)}} . \quad (44)$$

5.3 Resampling

A common problem with particle filters is so-called degeneracy phenomenon. Over time, several weights become high and other become negligible. Consequently, large computational effort is devoted to particles, which have only very small contribution to the approximation of the posterior density function [15]. The main goal of resampling step is to eliminate the particles with small weights and to focus on particles with larger weights. The basic resampling algorithm takes two steps:

- let $\tilde{x}_n^{(i^{(m)})}$ is derived from $x_n^{(m)}$ with probability proportional to $a_n^{(m)}$ ($m = 1,\ldots,M$) and $i^{(m)} = 1,\ldots,M$. New weights for these particles are:

$$\tilde{w}_n^{(i^{(m)})} = \frac{w_n^{(m)}}{a^{i^{(m)}}} \quad (45)$$

- new random measure is $\{\tilde{x}_n^{(i^{(m)})}, \tilde{w}_n^{(i^{(m)})}\}_{i^{(m)}=1}^{M}$

$i^{(m)}$ represents memory indexes where particles are stored as a result of resampling step [13].

6. Packet Loss Analysis

In experimental part of packet loss analysis, we have transmitted video sequence through a fixed network using RTP protocol. Therefore we have modeled network topology in the Network Simulator - ns-2.

The fixed network topology has consisted of 12 end stations, each connected to the separate router with 100Base-TX Ethernet, and 1 video server, as is shown in Fig. 3. The bottlenecks were represented by serial connection between routers. The video server was the source of video sequences and the computer on the other side of the network was a video receiver. Other PCs have introduced some background traffic using FTP and CBR agents. The data rate used by these agents was 0.5 Mbs and it was tried to transmit several flows in each node (1 – 6 flows). The

RTP was chosen as the transport protocol, while it was primarily designed for real time multimedia transmission. The packet size was set to 1052 bytes with header of 28 bytes. Several video sequences with different resolution (from QCIF to PAL) were used to build packet loss model. H.264 video codec was used to convert raw sequences. Prepared sequences were transmitted through the network modeled in ns-2 and packet sequence numbers with delay were logged. These values were used to build Gilbert model.

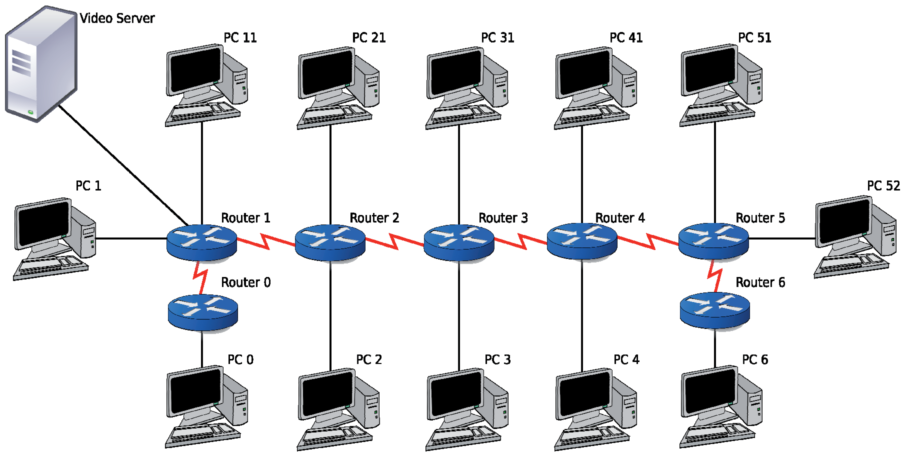

Fig. 3. Fixed network topology for packet loss analysis.

The obtained transition matrix for one flow in each node and PAL video is:

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}. \tag{46}$$

So it is clear, that no errors have appeared during transmission. The situation is changing with increasing the number of flows in nodes. Transition matrices for two to six flows are:

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \tag{47}$$

$$\begin{bmatrix} 0.919397 & 0.83455 \\ 0.080603 & 0.16545 \end{bmatrix}, \tag{48}$$

$$\begin{bmatrix} 0.84843 & 0.82286 \\ 0.15157 & 0.17714 \end{bmatrix}, \tag{49}$$

$$\begin{bmatrix} 0.80217 & 0.82698 \\ 0.19783 & 0.17302 \end{bmatrix}, \tag{50}$$

$$\begin{bmatrix} 0.75977 & 0.81645 \\ 0.24023 & 0.18355 \end{bmatrix}. \tag{51}$$

In the second part of packet loss analysis, we have transmitted video sequence through a wireless network. The principle of obtaining data for the models creation of packets losses remained the same as for fixed network. The basic difference lies in the maximum achieved transfer rate, since standard 802.11g was chosen for wireless network.

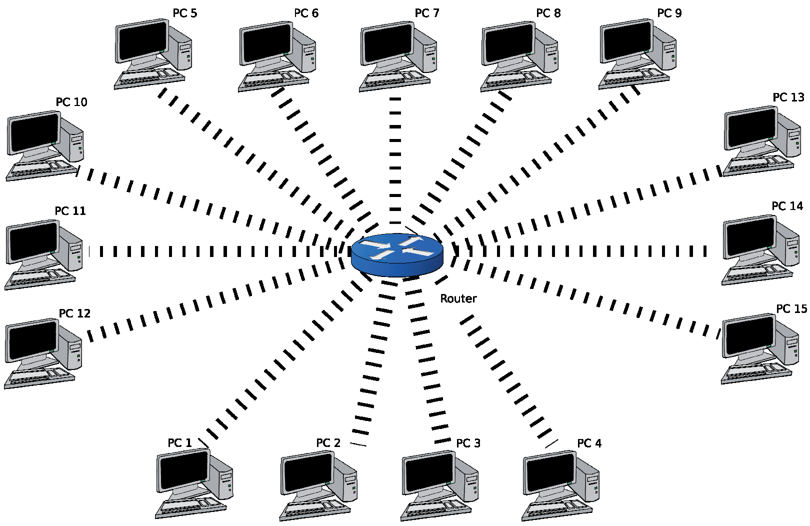

Fig. 4. Wireless network topology for packet loss analysis.

The video stream source was in this case computer “PC1”, and the receiver was represented by “PC9”, see Fig. 4.

The role of other network nodes was equivalent to the role of nodes in the fixed topology – to generate the consistent bit stream in random moments in time using FTP protocol.

Two-state Markov model is particularly suitable for fixed networks, while in wireless networks, due to higher frequency of outages and burst losses, it is preferable to use the N-state Markov model. *N* calculated in the model represents burst losses with the highest number of lost packets. Since the wireless network is more susceptible to failures in comparison with fixed network, packet losses are more common. This has resulted in frequent losses of motion vectors.

7. Autoregressive Model of Motion Vectors

Depending on the amount of movement in the image, space or time-adjacent vectors exhibit some degree of correlation. From this the possibility of lost motion vectors prediction arises based on information from neighboring block.

The proposed system for losses creation and their subsequent concealment is shown in Fig.5.

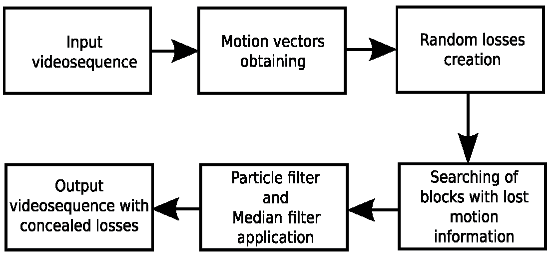

Fig. 5. The scheme of the proposed algorithm employing particle filter.

7.1 Median Filter

A median filter is applied in the proposed method to reduce interference at the edges of the concealed block caused by incomplete replacement of the lost block. Median filtering is a nonlinear method of smoothing, which is capable to remove large differences in brightness values around some point. In this paper, however, it is shown that this is a convenient way how to eliminate so called "block effect".

The essence of this method lies in ordering the luminance values of pixels into the mask according to the size and the new luminance value is determined as the median of the sequence. The mask is designed according to where a concealed block and pixel to which we want to apply the median filter is.

8. Statistical Distribution of Packet Losses

The losses of the motion vectors can be considered to be noise in the motion vector field. Also components of motion vectors can be considered to be statistically independent [14].

It is possible to use Bayesian filtering methods to recover lost motion vectors. If the result of statistical analysis of the losses is occurrence of additive noise (Gaussian noise), then it is suitable to use Kalman filter. Bayesian filters for non-Gaussian environment could offer better performance in other cases.

8.1 Simulation Scenario

Input video sequences were encoded using H.264 video codec and consequently, based on the previous obtained packet loss model, packet losses were applied randomly using Markov model. After that, motion vectors from the correct video sequence and also from the damaged video sequence were extracted. While the damaged video sequence has a lower number of motion vectors, zero motion vectors were added to the adequate place.

The proposed algorithm with the prediction of motion vectors utilizing Kalman filter was verified on four standard video sequences, namely "Foreman", "Mother_Daughter", "Suzie" and "Stefan". All simulations were performed in the MATLAB programming environment. In these sequences frames were randomly selected, in which the losses were generated. The number of blocks generated with a loss of motion vectors within the frame ranged from five to eighty and their position was random. As expected, the proposed method provides the best performance in the sequences "Mother_Daughter" and "Suzie". These two contain the smallest amount of movement. Thus, prediction of motion vectors in this case is considerably simplified. In video sequence “Foreman”, there is a significant amount of movement.

The most difficult movements for the proposed algorithm of error concealment using motion vectors prediction utilizing particle filter followed by edges smoothing using median filter were in the video sequence "Stefan". It contains a tennis match, and thus the greatest amount of movement. There was more than 14 dB difference in average PSNR value between video sequence "Stefan" and video sequence "Mother and Daughter". Simulation results are displayed in Tab.1- 4.

Of course, apart from the resulting reconstructed video quality, the time required to implement concealment of the lost block is also interesting. In our paper, we compare the time required for block prediction using Kalman filter with the proposed method. Moreover, in each table the time required to smooth edges of blocks by using median filter is added to illustrate how much concealing time using particle filter it takes to smooth the edges. In each row of the table there is the average time required for concealment of one lost block. Time labeled as $T_{PARTICLE}$ consists of two values - the time required for the prediction using particle filter and time required for implementation of nonlinear smoothing using median filter.

	PSNR [dB] damaged sequence	PSNR Kalman filter	PSNR Particle filter	T_{KALMAN} [s]	$T_{PARTICLE}$ [s]	T_{MEDIAN} [s]
1	24.228	65.496	65.775	0.000198	0.046933	0.000429
2	23.972	59.654	60.771	0.000341	0.046725	0.000119
3	21.346	46.781	47.522	0.000169	0.046549	0.000216
4	21.673	39.086	41.734	0.000179	0.047791	0.000176
5	19.797	48.342	49.44	0.000170	0.048098	0.000216
6	19.334	48.68	49.328	0.000166	0.046418	0.000178
7	18.682	43.923	44.891	0.000174	0.046602	0.000195
8	18.212	44.577	44.608	0.000257	0.046874	0.000205
9	17.425	43.104	45.014	0.000183	0.046415	0.000199
10	16.574	46.128	49.555	0.000209	0.048696	0.000185
11	17.073	40.818	43.026	0.000240	0.046554	0.000184
12	16.069	44.935	47.843	0.000163	0.046505	0.000178
13	15.993	39.432	43.45	0.000240	0.046908	0.000197
14	15.611	38.059	41.131	0.000172	0.047198	0.000202
15	15.142	41.894	45.123	0.000168	0.046832	0.000181
16	14.799	38.226	40.972	0.000224	0.048239	0.000198

Tab. 1. Simulation results, video sequence Mother_Daughter.

	PSNR damaged sequence	PSNR Kalman filter	PSNR Particle filter	T_{KALMAN} [s]	$T_{PARTICLE}$ [s]	T_{MEDIAN} [s]
1	24.576	36.317	38.006	0.000178	0.046570	0.000444
2	20.853	38.565	40.793	0.000219	0.046035	0.000183
3	19.056	39.826	41.392	0.000194	0.046571	0.000180
4	18.315	36.461	37.784	0.000169	0.046860	0.000187
5	18.04	28.953	33.934	0.000254	0.046799	0.000208
6	16.023	25.055	28.827	0.000247	0.046995	0.000188
7	15.312	36.393	38.058	0.000247	0.046897	0.000198
8	14.59	24.192	27.419	0.000165	0.046193	0.000179
9	15.172	27.898	30.457	0.000210	0.046725	0.000200
10	14.508	27.977	29.165	0.000211	0.046236	0.000188
11	13.851	26.749	29.073	0.000187	0.047834	0.000206
12	13.622	30.073	34.651	0.000170	0.047105	0.000186
13	12.827	29.518	33.792	0.000163	0.047311	0.000189
14	12.725	25.658	27.139	0.000167	0.047182	0.000179
15	12.412	22.421	25.516	0.000200	0.047504	0.000177
16	12.143	31.244	36.164	0.000232	0.047537	0.000191

Tab. 2. Simulation results, video sequence Stefan.

	PSNR [dB] damag edsequ ence	PSNR Kalman filter	PSNR Particle filter	T_{KALMAN} [s]	$T_{PARTICLE}$ [s]	T_{MEDIAN} [s]
1	23.337	32.187	36.962	0.000247	0.046687	0.000747
2	23.008	48.473	48.473	0.000241	0.046797	0.000602
3	18.774	43.209	45.091	0.000241	0.046930	0.000058
4	18.26	45.95	46.018	0.000480	0.046782	0.000192
5	16.119	48.661	51.125	0.000220	0.047477	0.000245
6	14.865	24.028	28.838	0.000232	0.046313	0.000180
7	13.917	36.681	39.471	0.000168	0.047269	0.000190
8	13.412	26.491	31.541	0.000166	0.047153	0.000164
9	12.87	28.019	33.712	0.000210	0.047002	0.000198
10	15.681	35.86	40.454	0.000193	0.046556	0.000190
11	13.764	26.376	30.098	0.000180	0.046968	0.000175
12	11.611	31.778	35.792	0.000166	0.046253	0.000171
13	12.51	37.359	41.402	0.000193	0.046967	0.000202
14	13.092	25.752	27.401	0.000169	0.046637	0.000193
15	10.288	27.532	28.282	0.000210	0.047002	0.000201
16	10.031	36.94	36.633	0.000205	0.051905	0.000179

Tab. 3. Simulation results, video sequence Foreman.

	PSNR damaged sequence	PSNR Kalman filter	PSNR Particle filter	T_{KALMAN} [s]	$T_{PARTICLE}$ [s]	T_{MEDIAN} [s]
1	26.454	45.528	46.893	0.000169	0.046456	0.000703
2	23.145	48.164	49.519	0.000179	0.046311	0.000217
3	20.312	41.418	43.26	0.000191	0.046142	0.000350
4	19.356	47.17	48.612	0.000182	0.046536	0.000192
5	19.392	38.426	40.224	0.000179	0.048590	0.000169
6	18.593	36.105	36.693	0.000189	0.046573	0.000173
7	16.975	45.115	51.141	0.000167	0.046674	0.000175
8	16.458	39.276	42.088	0.000196	0.047547	0.000192
9	16.845	39.137	42.989	0.000206	0.046941	0.000193
10	15.797	40.196	44.049	0.000171	0.046579	0.000207
11	15.884	35.133	37.278	0.000169	0.045869	0.000194
12	14.824	31.895	35.029	0.000198	0.046786	0.000167
13	14.729	43.671	46.569	0.000194	0.046299	0.000180
14	14.632	38.501	41.007	0.000168	0.046324	0.000179
15	14.337	29.188	33.643	0.000193	0.046889	0.000157
16	14.242	25.303	27.489	0.000173	0.048221	0.000194

Tab. 4. Simulation results, video sequence Suzie.

The results show that the mere median filter needs about the same time for smoothing as Kalman filter for prediction the lost motion vectors. Moreover, for the first block this time is always two or three times longer. The total time required for reconstruction of the lost motion information in the proposed method varies in value from 0.045 to 0.051 sec. The time needed for concealment of losses in both cases is independent on the selected video sequence. Fig. 6, 7 and Fig. 8, 9 show an example of error concealment in the video sequences “Mother_Daughter” and “Stefan”. In given frames, 65 lost macroblocks were generated and then concealed. Since video sequence

"Stefan" contains much more movement than video sequence "Mother_Daughter", it is possible to see visible artifacts in the concealed video sequence "Stefan".

Simulation results confirm expectations that improved recovery of lost information is due to computationally demanding procedures. It is therefore necessary to consider in which areas the proposed method can be used.

Fig. 6. 65 corrupted macroblocks, videosequence Mother_Daughter.

Fig. 7. 65 corrupted macroblocks, videosequence Mother_Daughter, concealed frame.

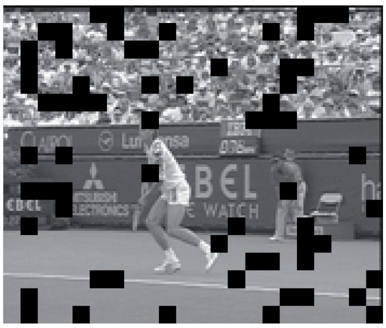

Fig. 8. 65 corrupted macroblocks, video sequence Stefan.

Fig. 9. 65 corrupted macroblocks, video sequence Stefan, concealed frame.

9. Conclusion

The aim of this work was to create models of packet failures transmitted over fixed and wireless networks and their subsequent application to video stream transfer. From obtained damaged video sequences motion vectors were subsequently extracted to determine the statistical distribution of noise arising as a result of motion information loss. Particle filter was used for error concealment. The proposed method surpassed the previous algorithm based on Kalman filtering in image quality, but computational complexity remains its significant disadvantage. One way how to solve this problem is the deployment of a partial filter employing Rao-Blackwell theorem. Such an approach should lead to better results in significantly lower amount of components necessary for correct estimation, and thus to reduction of the time required to predict the lost motion vectors.

Acknowledgements

The work presented in this paper was supported by the Grant of the Ministry of Education and the Academy of Science of the Slovak Republic VEGA under Grant No.1/0045/10 (50%).

This publication is the result of the project implementation of the Centre of Information and Communication Technologies for Knowledge Systems (project number: 26220120030) supported by the Research & Development Operational Programme funded by the ERDF (50%).

References

[1] HAßLINGER, G., HOHLFELD, O. The Gilbert-Elliott model for packet loss in real time services on the internet. In *Proceedings of the 14th GI/ITG Conference on Measurement, Modeling, and Evaluation of Computer and Communication Systems (MMB)*. Dortmund (Germany), 2008, p. 269 – 283.

[2] YAJNIK, M., MOON, S., KUROSE, J., TOWSLEY, J. D. Measurement and modeling of the temporal dependence in packet loss. In *Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '99*. New York (USA), 1999, vol. 1, p. 345 - 352.

[3] YAJNIK, M., KUROSE, J., TOWSLEY, D. Packet loss correlation in the MBone multicast network: Experimental measurements and Markov chain models. *Tech. Rep. UMASS CMPSCI 95-115, University of Massachusetts*, Amherst (Massachusetts, USA), 1995.

[4] PLAZA, J. C., CABRERA, J., JUAREGUIZA, F., GARCIA, N. A New class partitioned discrete model for the characterization of RTP multicast transmission through IEEE 802.11 channels. In *Proceedings of the 66th IEEE VTC2007 (IEEE Vehicular Technology Conference 2007)*. Baltimore (USA), 2007, p. 1907 - 1911.

[5] MOCHNÁČ, J., MARCHEVSKÝ, S., GAMEC, J. Kalman filter based error concealment algorithm. In *Proceedings of Internationales Wissenschaftliches Kolloquium (54. IWK - Information Technology and Electrical Engineering – Device and*

Systems, Materials and Technologies for the Future). Ilmenau (Germany), 2009, p. 1 – 4. ISBN 978-3-938843-44-4.

[6] MOCHNÁČ, J., MARCHEVSKÝ, S., KOCAN, P. Bayesian filtering techniques: Kalman and extended Kalman filter basics. In *Proceedings of 19th international conference Radioelektronika 2009*. Bratislava (Slovak Republic), 2009, p.119-122. ISBN 978-1-4244-6320-6.

[7] MOCHNÁČ, J., MARCHEVSKÝ, S., KOCAN, P. Simulation of packet losses in video transfers using real-time transport protocol. In *Proceedings of 20th international conference Radioelektronika 2010*. Brno (Czech Republic), 2010, p. 1-4, doi: 10.1109/RADIOELEK.2010.5478577. ISBN 978-1-4244-6320-6.

[8] SANNECK, H., CARLE, G, KOODLI, R. A framework model for packet loss metrics based on loss runlengths. In *Proceedings of the SPIE/ACM SIGMM Multimedia Computing and Networking Conference 2000 (MMCN 2000)*. San Jose (California, USA), 2000, p. 177 – 187.

[9] HYUNGKEUN, L., VARSHNEY, P. K. Gap-based modeling of packet losses over the Internet. In *Proceedings of the 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, MASCOTS 2002*. Fort Worth (Texas, USA), p. 507 - 510.

[10] MARKOVSKI, V., XUE, F., TRAJKOVIC, L. Simulation and analysis of packet loss in video transfers using User Datagram Protocol. *Journal of Supercomputing, Kluwer Academic Publishers*, Hingham (Massachusetts,USA), 2001, vol. 20, p. 175 - 196. ISSN 0920-8542.

[11] PUNSKAYA, E. Sequential Monte Carlo methods for digital communications. *PhD. thesis*, Engineering Department, Cambridge University, Cambridge, UK, 2003, p. 30 - 42.

[12] GORDON, N. J., SALMOND, D. J., SMITH, A. F. M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 1993, vol. 140, no. 2, p. 107 - 113.

[13] BOLIĆ, M. Architectures for efficient implementation of particle filters. *Dissertation Thesis*, Stony Brook University, August 2004, p. 1 – 87.

[14] GAMCOVA, M. Nové algoritmy obnovy zašumených vektorov pohybu na báze čislicovej filtrácie. *PhD. thesis*, Department of Electronics and Multimedia Telecommunication, Technical University of Košice, Košice, 2005, p. 20 – 33 (in Slovak).

[15] ARULAMPALAM, M.S., MASKELL, S., GORDON, N., CLAPP, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 2002, vol. 50, no. 2, p. 174 – 188.

About Authors...

Branislav HRUŠOVSKÝ was born on 23-10-1984 in Košice. He received his M.Sc. degree from the Technical University of Košice. He graduated from the Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics in May 2009. He is a PhD. student at the Department of Electronics and Multimedia Communications at the same university since September 2009. His current research interests include error concealment of coded videodata and satellite communications.

Ján MOCHNÁČ was born in 1984 in Stará Ľubovňa (Slovakia). He received his M.Sc. degree in Electronics and Telecommunications in 2007 from the Faculty of Electrical Engineering and Informatics, Technical University of Košice and his Ph.D. degree in September 2010 from the same university. His research interest included error concealment methods for packet.

Stanislav MARCHEVSKÝ received the M.Sc. in Electrical Engineering from the Faculty of Electrical Engineering, Czech Technical University in Prague, in 1976, and PhD. degree in Radioelectronics from the Technical University of Košice in 1985. Currently he is a Professor of Electronics and Multimedia Communications Department, Faculty of Electrical Engineering and Informatics, Technical University of Košice. His teaching interests include switching theory, digital television technology, and satellite communications. His research interests include image nonlinear filtering, neural networks, genetic algorithms, and multiuser detection, spacetime communication, diversity communication over fading channel, and power and bandwidth efficient multiuser communications.