Resampling Of An Image By Block-Based Interpolation Or Decimation With Compensation

Jozef Zavacký, Ján Mihalík, Martin Kapinos
Department of Electronics
and Multimedia Telecommunications,
FEI TU, Park Komenského 13, 04021 Košice
Slovak Republic
zavacky@ccsun.tuke.sk, mihalik@ccsun.tuke.sk,
animax3d@animax3d.sk

Abstract

Due to multiple standards on digital coding of image such as JPEG, MPEG, H.261 etc., it is expected that conversion from one picture format to another will be quite necessary for display or recording of different format sources. Conventional approach of 2-D sampling rate conversion by polyphase filters requires relatively large memory and computation power. Therefore, a new efficient method for image resampling (interpolation and decimation) has been presented. The proposed approach performs resampling block by block with overlap. To minimize the overlap special block interpolation kernels are used, with one pixel overlap gives satisfactory result for most of practical images. Proposed method can be efficiently applied to image communication systems where block transforms are used for data compression.

Keywords

Interpolation, decimation, resampling, image communication systems

1. Introduction

In recent years, several standards on digital coding of image and video signals have been proposed such as JPEG, MPEG, H.261, etc. based on discrete cosine transform (DCT). Due to multiple standards, it is expected that conversion from one picture format to another will be quite necessary for display or recording of different format sources. Conventional approach of 2-D sampling rate conversion by polyphase filters requires relatively large memory and computation power [1-2]. As a result, such a - system becomes expensive. With wide applications of digital image and video, it is quite necessary to develop cost effective format conversion systems [3-7] where effective image resampling is essential. An approach for highly efficient image resampling will by introduced. The proposed approach performs resampling block by block with overlap. The amount of overlap can be adjusted depending on the desired quality. To minimize the overlap special block interpolation kernels are used which are developed using a stationary image model.

The conventional approach to resampling for noninteger ratio changes is based on the sampling theorem. A given discrete signal is a complete representation of a continuous signal if the continuous signal is bandlimited and the sampling interval satisfies the Nyquist criterion. Once the original continuous signal is recovered, it is trivial matter to increase the number of samples (i.e. increase the sampling rate) from the continuous signal. This interpolation process with arbitrary ratio can be done pure digitally using the mathematical description of a bandlimited signal by Shannon – Kotelnik's series,

$$f(t)=\sum_{n=-\infty}^{\infty} f(nT)\frac{\sin\omega_B(t-nT)}{\omega_B(t-nT)} \qquad (1)$$

where $\omega_B=\pi/T$ determines the bandwidth of $f(t)$. Increasing the sampling rate does not violate the sampling theorem. Hence, sampling $f(t)$ in (1) with the interval $T_1<T$, we get the resampled (interpolated) signal without any error

$$f(mT_1)=\sum_{n=-\infty}^{\infty} f(nT)\frac{\sin\omega_B(mT_1-nT)}{\omega_B(mT_1-nT)} \qquad (2)$$

For practical purposes, the sampled sinc-function in (2) is approximated by a finite sequence to reduce the number of multiplications within an acceptable error bound. For a decimation with sampling interval $T_2>T$ however, (2) cannot be used because the sampled signal would not properly represent the original signal. It would contain an aliasing error, which has an annoying visual affect when used as a representation of the continuous signal. We can remove the high frequency components before resampling so that aliasing does not occur. Antialiasing low-pass filtered signal is represented as follows using (1)

$$f_L(t) = f(t) * g(t)$$
$$= \sum_{n=-\infty}^{\infty} f(nT) \int_{-\infty}^{\infty} \frac{\sin\omega_B((t-nT)-\tau)}{\omega_B((t-nT)-\tau)} \times \frac{\sin\omega_L\tau}{\omega_L\tau} d\tau \quad (3)$$

where the low-pass filter impulse response is ideally

$$g(t) = \frac{\sin\omega_L t}{\omega_L t} \quad (4)$$

and $\omega_L = \pi/T_2$, $T_2 > T = \pi/\omega_B$. We shall obtain low-pass filtered signal representation

$$f_L(t) = T\sum_{n=-\infty}^{\infty} f(nT) \frac{\sin\omega_L(t-nT)}{\omega_L(t-nT)} \quad (5)$$

Sampling $f_L(t)$ with T_2 gives

$$f_L(mT_2) = T\sum_{n=-\infty}^{\infty} f(nT) \frac{\sin\omega_B(mT_2-nT)}{\omega_B(mT_2-nT)} \quad (6)$$

which does not contain aliasing errors. Notice that interpolation and decimation results have the same form of the computation even though antialias low-pass filtering has been incorporated. Therefore, no distinction will be necessary between interpolation and decimation except for the low-pass filters used for each case. For 2-D signals, the same procedures are applied for row and column operations if separable 2-D filtering is used. In the next section, a block approach is suggested which requires less computation power and memory.

2. Truncated overlap-add resampling of an image

In this section, an efficient approximation method for low-pass type convolutions is given where the contribution from the distant pixels to the current pixel values are decreased as fast as $1/n$, where n is the distance of a pixel form the current resampling point. In the presented approach, interpolation is done block by block with the overlap-add computations limited to a finite distance. Contributions from remote pixels are estimated based on the current image block [8].

First, for the notational simplicity, 1-D development will be discussed with a block size N. 2-D extension is naturally followed. In order to perform the filtering in (2) by overlap-add method input data is passed through and ideal interpolation filter block by block with cut-off frequency $\omega_B = \pi/T$. The output of the filter is sampled with an interval $T_1 < T$. The outputs from the ideal filter (i.e. a sinc-function) are truncated to fit into block size with an overlap γT, where γ is an integer. The errors due to truncation are compensated based on the stationary image model as will be described later. If the overlap γT is sufficiently large, high quality interpolation is obtained at the expense of a relatively large amount of computation. However, γ can be significantly reduced (down to one, corresponding to one pixel distance) for highly efficient computation by employing a proper compensation technique.

Let $g_i(t)$ be the output of the interpolation filter for the *i-th* block input signal $f_i(n) = f_i(nT)$, where $f_i(n) = f(n)R_i(n)$, $i = 0,1,\dots,M-1$, where M is the number of blocks, N is block size, and

$$R_i[n] = \begin{cases} 1, & iN \le n < (i+1)N \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Then the output of the interpolation filter for the *i-th* block is

$$g_i(t) = \sum_{n=-\infty}^{\infty} f_i(nT) \frac{\sin\omega_B(t-nT)}{\omega_B(t-nT)} \quad (8)$$

The overlap-add interpolation is obtained by the sum of the sampled values of $g_i(t)$, $i = 0,1,\dots,M-1$

$$g(m) = g(mT_1) = \sum_{i=0}^{M-1} g_i(mT_1) \quad (9)$$

where $m \in [0,(MNT/T_1 - 1)]$ and M is the total number of blocks.

For the simplicity of the presentation, assume that $t = 0$ is aligned with the first sample of the *i-th* block of the input signal as shown in Fig. 1.

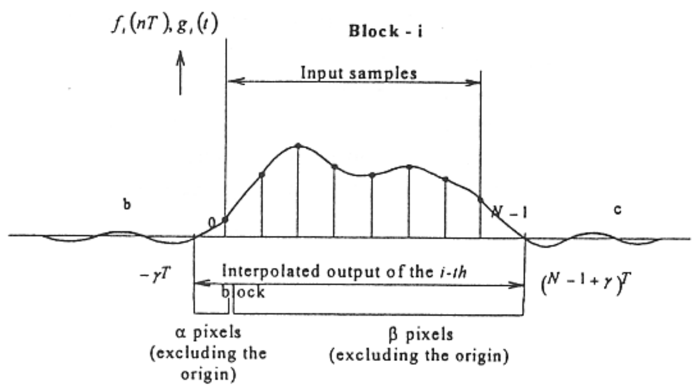

Fig. 1. The i-th block input samples and interpolated output before resampling

The output of the *i-th* block $g_i(t)$ is truncated to $[-\gamma T,(N-1+\gamma)T]$ such that γ pixels are overlapped.

After resampling, the number of pixels at the output of the *i-th* block is B, where B is given by $B = \alpha + \beta + 1$ and

$$\alpha = \left[\frac{\gamma T}{T_1}\right], \ \beta = \left[\frac{(N-1+\gamma)T}{T_1}\right], \qquad (10)$$

where $[x]$ represents the largest integer smaller than or equal to x. These truncated and resampled outputs $g_i(mT_1)$, $m = -\alpha, -\alpha+1, \ldots, \beta$, are added to get the final result. In the presented approach, overlap is limited to a distance γT where overlap-add with the adjacent block is performed as depicted in Fig. 2.

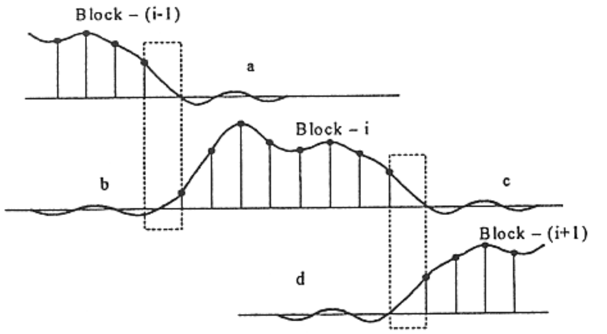

Fig. 2. Block interpolation by overlap-add method

If overlap is sufficiently large, almost perfect resampling is achieved. The amount of computation (and hardware) can be minimized by reducing γ. For practical images, γ can be reduced down to one-pixel distance with proper compensation of errors as described next.

To compensate for the errors due to truncations, truncated overlap portions are estimated using the current block data. In Fig. 2 truncated overlaps a and d are approximated by b and c, respectively and vice versa. Rationale behind this approach is that image data is highly correlated in the homogeneous area and errors near edges are less sensitive to human eyes. Consider the case when the block boundaries are in the homogeneous region. Then b and c are really good estimates of a and d, respectively, due to the homogenity of the blocks. When singularities (edges) are near the boundaries, the estimations have more errors. However, these errors are near the edges or high activity areas, which are less visible due to human visual system characteristics. These facts are demonstrated in the computer simulations in Section 3. Next, efficient compensation method is described by using precompensated block interpolation kernels.

As discussed above, the truncated portions of overlaps from adjacent block are estimated using current block data. In Fig. 2, replacing part $-a$ and $-d$ with part $-b$ and $-c$, respectively is equivalent to using a block symmetric image model depicted in Fig. 3.

Using this image model the interpolated *i-th* block signal is given, in a simple form, as

$$g_i(t) = \sum_{l=0}^{N-1} f_i(l) k_l(t) \qquad (11)$$

where $k_l(t)$ are block-interpolated kernels with a support $[-\gamma T, (N-1+\gamma)T]$ as follows

$$k_l(t) = \left[\sum_{r=-\infty}^{\infty} \frac{\sin \omega_B (t + (2rN - l)T)}{\omega_B (t + (2rN - l)T)}\right] \times R_{(-\gamma T, (N-1+\gamma)T)}(t)$$

$$+ \left[\sum_{s=-\infty}^{\infty} \frac{\sin \omega_B (t + (2sN + 1 + l)T)}{\omega_B (t + (2sN + 1 + l)T)}\right] \times R_{(0, (N-1)T)}(t) \quad (12)$$

where $R_{(a,b)}(t)$ is a rectangular window function as follows

$$R_{(a,b)}(t) = \begin{cases} 1, & a \le t < b \\ 0, & \text{otherwise} \end{cases} \qquad (13)$$

By resampling (11) with desired sampling interval T_1, we get the interpolated *i-th* block signal. Equation (11) has a time-variant filtering form. For the example of $N = 8$, the interpolation kernels are depicted in Fig. 4, where

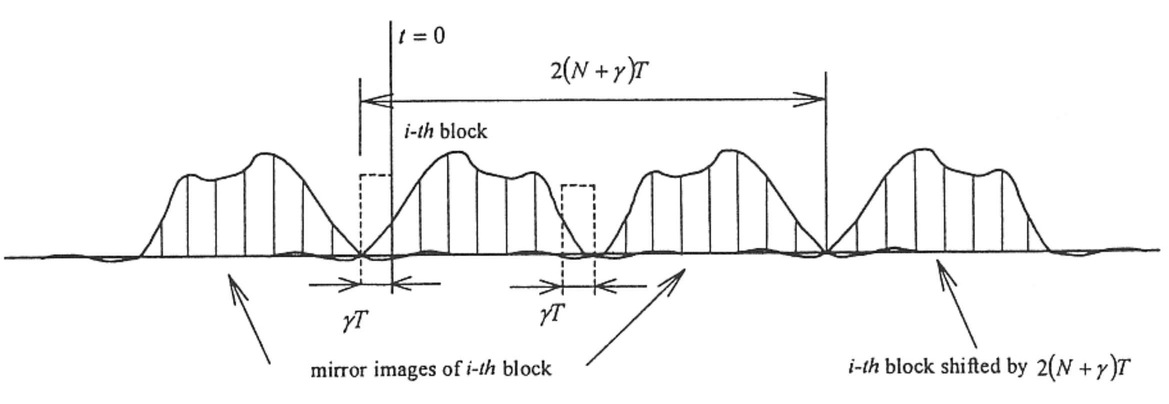

Fig. 3. Block-symmetric image model for compensation of errors due to truncations

$\gamma = 1$. Notice that $k_l(t)$, $l = 0,1,\ldots,N-1$, are defined independent of signals and, as a result, can be precomputed.

The sum of interpolation functions $k_l(t)$, $l = 0,1,\ldots,7$ on the interval $[-\gamma T, (N-1+\gamma)T]$ for $\gamma = 1$, $N = 8$ is shown in Fig. 5.

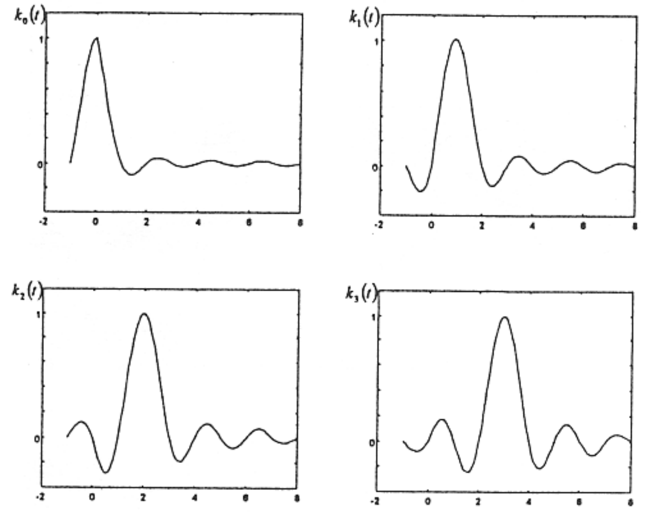

Fig. 4. Interpolation functions $k_l(t)$, $l = 0,1,2,3$ (from top to bottom and then left to right), with $\gamma = 1$, $N = 8$

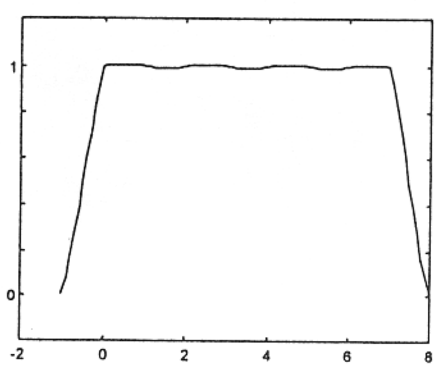

Fig. 5. The sum of interpolation functions $k_l(t)$.

The results of the sampling of the function (11) can be written in a matrix equation form

$$g_i = Kf_i \tag{14}$$

where g_i is vector of sampled values of $g_i(t)$ in the support $[-\gamma T, (N-1+\gamma)T]$ with sampling interval T_1, f_i is an $N \times 1$ vector of input samples and K is an $(\alpha+\beta+1) \times N$ matrix defined from (11) and (12) as follows

$$g_i^T = [g_i(-\alpha) \quad g_i(-\alpha+1) \quad \ldots \quad g_i(0) \quad \ldots \quad g_i(\beta)] \tag{15}$$

$$f_i^T = [f_i(0) \quad f_i(1) \quad \ldots \quad f_i(N-1)] \tag{16}$$

$$K_{p,q} = k_q(pT_1) \quad p = -\alpha,\ldots,\beta \quad q = 0,1,\ldots,N-1 \tag{17}$$

Equation (17) has for arbitrary block size form

$$K_{p,q} = \begin{bmatrix} k_0(-\alpha T_1) & k_1(-\alpha T_1) & \cdots & k_{N-1}(-\alpha T_1) \\ k_0((-\alpha+1)T_1) & k_1((-\alpha+1)T_1) & \cdots & k_{N-1}((-\alpha+1)T_1) \\ \vdots & \vdots & \ddots & \vdots \\ k_0(\beta T_1) & k_1(\beta T_1) & \cdots & k_{N-1}(\beta T_1) \end{bmatrix} \tag{18}$$

The result resampling signal is again obtained by adding of particular output signals $g_i(mT_1)$ (for interpolation) or $g_i(mT_2)$ (for decimation) where $i = 0,1,\ldots,M-1$ a $m \in [0\ldots(M\,NT/T_1 - 1)]$. When the average number of sample points within a block after interpolation or decimation is an integer, the above sampled interpolation kernels $k_q(pT_1)$, $p = -\alpha,\ldots,\beta$, $q = 0,1,\ldots,N-1$, which are elements of matrix K in (17), are sufficient. For example, if an 8-sample block is interpolated to 9 samples, the 9x8 matrix K defined by (18) is sufficient for the resampling of whole image plane. The same is true for the decimation of a block to 7 or 6 sample points with appropriate ω_B 's for the low-pass filter kernel. However, when the average number of sample points from the blocks after resampling are non-integer we should need modified versions of matrix K.

Previous development may be applied to the 2-D case. As in the 1-D case, resampling can be represented as a matrix operation.

Let K_x and K_y be $(\alpha+\beta+1) \times N$ resampling matrices in the x- and y- directions, respectively. The 2-D resampling is done by row and column operations described as multiplications as follows

$$g_{i,j} = K_y f_{i,j} K_x^T \quad , \tag{19}$$

where the matrices $f_{i,j}$ are the $(i,j)th$ $N \times N$ block 2-D signals and $g_{i,j}$ are the $(\alpha+\beta+1) \times (\alpha+\beta+1)$ resampled output block matrix with overlap γT. Overall resampling is obtained by adding all the $(i,j)th$ blocks, $i,j = 0,1,\ldots,M-1$, in (19) with overlap γT.

3. Computer simulation and results

Computer simulations have been done on the image Lena of the size 512x512 pels, which contains high frequency components on the hat pattern as shown in

a)

b)

c)

d)

Fig. 6 The original Image Lena (a) and interpolated images using the block approach with compensation ($\gamma = 1$, block size 8x8) from: b) 64x64 to 512x512 pels, c)128x128 to 512x512 pels, d) 256x256 to 512x512 pels

Fig. 6a. First the image Lena was decimated by the developed method with compensation to three different sizes, namely 64x64, 128x128 and 256x256 pels with one pixel overlap ($\gamma = 1$) and a block size 8x8 pels. Then these decimated images were interpolated by the block method to the original size 512x512 pels with parameters the same as for decimation. Results are shown in Fig. 6b, 6c and 6d, respectively. Table 1 shows signal to noise ratios (SNR) and mean square errors (MSE)

Table 1

Image size after decimation	Interpolation	
	SNR (dB)	MSE
64x64	15,94	460,83
128x128	19,76	187,02
256x256	25,34	51,69

Further experiment was performed such, that image Lena was decimated by the developed method to the size 128x128 pels as previously but it was interpolated by the block method with the same parameters as for

Fig. 7 Interpolated image from 128x128 to 512x512 pels using the block approach without compensation ($\gamma = 1$, block size 8x8)

Fig. 8 Interpolated image from 128x128 to 512x512 pels using the block approach ($\gamma = 1$, block size 8x8) at basic decimation

decimation, however without compensation to the original size 512x512 pels. The result is illustrated in Fig. 7 with SNR = 16,23 dB and MSE = 421,55. Finaly we have carried out the basic decimation by missing out pels in the horizontal and vertical directions of image Lena to size 128x128 pels. Such decimated image was interpolated by the developed method with compensation again with $\gamma = 1$, block size 8x8 pels to the original size 512x512 pels. The interpolated image is shown in Fig. 8, with SNR = 17,0 dB and MSE =353,47.

Comparing the results represented in Fig. 6c and Fig. 7 it is seen that for the same decimation method, however different interpolation one, better result is achieved by interpolation method with compensation. On the other hand comparing the results in Fig. 6c and Fig. 6a, we find out that using different decimation method and the same interpolation one, better result is in the case of the developed decimation method. The given result confirm the high quality of resampling when using the proposed resampling method. Moreover, this method decreases amount of calculations compared to the conventional resampling method [1].

4. Conclusion

A new efficient approach to the resampling image has been presented. The proposed approach gives cost effective, fast and high quality resampling. This method can be efficiently applied to image communication systems where block transforms [9] can be used for data compression [10]. In the decoding stage where inverse block transforms are performed, resampling can be effectively combined. Inverse block transforms are represented as a matrix operation as

$$f_{i,j} = CF_{i,j}C^T, \tag{20}$$

where $F_{i,j}$ is an $N \times N$ 2-D transform coefficient matrix of the $(i,j)th$ block 2-D signal, C is an $N \times N$ 1-D inverse transform matrix and $f_{i,j}$ is an recovered $(i,j)th$ 2-D sample array block. Since the resampling process in (19) has a similar structure as (20), they are readily combined to give

$$\begin{aligned} g_{i,j} &= K_y f_{i,j} K_x^T = K_y C F_{i,j} C^T K_x^T = \\ &(K_y C) F_{i,j} (C K_x)^T = H_y f_{i,j} H_x^T \end{aligned} \tag{21}$$

where matrices H_x and H_y gives a representation similar to (20) which perform inverse transform and resampling at the same time. The proposed approach can be implemented on a single VLSI chip together with an inverse block transform. The proposed resampling scheme will make the image transcoder system affordable for various applications.

References

[1] CROCHIERE, R.E. - RABINER, L.R.: Multirate Digital Signal Procesing. Prentice Hall, Englewood Cliffs, NJ, 1983, pp. 23-42.

[2] MIHALÍK, J. – ZAVACKÝ, J.: Sampling Methods of Signals. J. Electrical Engineering 41, 1990, No. 3, pp. 221-223, (In: Slovak).

[3] KIM, S.P. – WEN-YU SU: Direct Image Resampling Using Block Transform Coefficients. Signal Processing: Image Communication 5, 1993, pp. 259-272

[4] LANZAVECCHIA, S. – BELLON, P.L.: A Moving Window Shannon Reconstruction Algorithm for Image Interpolation. J. of Vizual Communication and Image Representation, Vol. 5, No. 3, 1994, pp. 255-264.

[5] MIHALÍK, J. – ZAVACKÝ, J.: Multichannel Spline Interpolation of Discrete Signals. Radioengineering, Vol. 3, No. 2, 1994, pp. 12-16.

[6] MIHALÍK, J. – ZAVACKÝ, J.: Interpolation of Discrete Signals by Using Spline Functions. Electronic Horizon, Vol. 51, No. 12, 1990, pp. 481-484, (In: Slovak).

[7] MIHALÍK, J. – ZAVACKÝ, J. – Kuba, I.: Spline Interpolation of Image. Radioengineering, Vol. 4, No. 1, 1995, pp. 221-230.

[8] ZAVACKÝ, J. – MIHALÍK, J. – KAPINOS, M.: Resampling of an image by Block-Based Interpolation or Decimation. Proceeding IWSSIP'99 6th International Workshop on Systems, Signals and Image Processing, June 2-4, 1999, Bratislava, Slovakia, pp. 69-72.

[9] MIHALÍK, J.: Adaptive Transform Coding of Image. Electronic Horizon, Vol. 52, No. 11-12, pp. 253-257.

[10] MIHALÍK, J.: Digital Signal Processing. Alfa Bratislava, 1987, (In: Slovak).

ACKNOWLEDGEMENTS

The work presented in this paper was supported by the Grant Agency of Ministry of Education and Academy of Science of Slovak Republic under Grant No. 1/7387/20.

About authors...

Jozef ZAVACKÝ was born on 18th May, 1953 in Slovakia. He received the Ing. Degree from Technical University of Košice, Slovak Republic, in 1977 and the Ph.D. degree in Radioelectronics in 1992. Currently he is Assistent Professor at the Department of Electronics and Multimedia Telecomunications of Technical University, Košice. His present research interests are in signal theory, multichannel systems, sampling and interpolation of signals.

Ján MIHALÍK was born on August 1, 1952 in Slovakia. He graduated from Technical University in Bratislava in 1976. Since 1979 he joined Technical University of Košice, where received his Ph.D. degree in Radioelectronics in 1985. Currently, he is Full Professor at the Department of Electronics and Multimedia Telecomunications of Technical University, Košice. His research interests include information theory, digital image processing and comunication.

Martin KAPINOS was born on 19th July, 1975 in Slovakia. He received the Ing. Degree from Technical University of Košice, Slovak Republic, in 1998. Currently he is Ph.D. student at the Department of Electronics and Multimedia Telecomunications of Technical University, Košice. His present research interests are on sampling and interpolation of signals.