LOSSLESS ENCODING OF 3D HUMAN HEAD MODEL TEXTURES

Ondrej KOVÁČ, Ján MIHALÍK
Department of Technologies in Electronics, Department of Electronics and Multimedia Communications,
Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic,
tel. +421 55 602 3013, e-mail: {ondrej.kovac,jan.mihalik} @tuke.sk

ABSTRACT

The paper describes technique of lossless compression of 3D human head model textures. The technique of texture generation based on two orthogonal views is described. The texture of whole head has to be a single image, it is necessary to transform a 3D model to the 2D texture plane. This is achieved by cubic transformation. Coordinates of pixels inside the polygons of triangulation in source images are given by computing of barycentric coordinates. Results of the generation based on two orthogonal views are two partial textures which have to be joined without visible edges. This is done by merge of the Laplace pyramids of partial textures multiplied by the Gaussian pyramids of the binary mask. The final smoothed texture is decomposed by lifting integer discrete wavelet transform. Sub-images are decomposed into bit planes which are coded by binary state arithmetic coding.

Keywords: *texture, 3D model, human head, lossless compression, bit planes, DWT*

1. INTRODUCTION

Today, social networks and even virtual reality [1, 2] are popular for mutual communication of people. Therefore it is necessary to solve issues of generation and coding of virtual human characters. These characters are represented by 3D models and appropriate textures. The article deals only with 3D human head model. Although it is well known that the transmission speed of access networks has increasing tendency we can not forget importance of source coder compression. This can be achieved by one of most modern video-compression technique called the model coding [3]. In this case of the compression only changes of vertices coordinates are transmitted. These changes are defined as animation and movement parameters. We will not deal with the coding of these parameters. The article is focused on generating, smoothing and lossless coding of textures. In the second chapter a method of the 3D model to 2D texture plane transformation is described. After this transformation we propose a method of texture generation and smoothing. For this is used the method of image merging by the Laplace pyramid. The smoothed textures have to be lossless encoded. We suggest a combination of discrete wavelet transformation (DWT) and binary state arithmetic coding. The smoothed textures will be decomposed into space of the DWT coefficients. The well-known classical DWT uses convolution between impulse responses of filters and an input signal [4]. This approach causes signal extension and the design of DWT processor is difficult. In addition this concept does not provide integer output. Hence, it is better to use the lifting implementation of DWT [5]. This technique does not result in a signal extension and the design consist only of prediction, summation and rounding. Thus the obtained wavelet sub-images are decomposed to bit planes. The bit planes are subsequently encoded by lossless binary state code [6]. Also it will be shown, it is possible to code the bit planes of not transformed image. In conclusion, compression efficiency of the proposed method is analyzed.

2. TEXTURES GENERATION

Texturing, in human head images coding [7], gives a 3D model final and natural form. The texture represents values of pixels which are given by merging of two or more source images [8]. Front and profile view are the most common. The texturing means better value of similarity between a generated clone and a real human head. The 3D human head model texturing itself is a process of transferring texture value on the surface of a 3D shaped model. Benefit of the texture in the model coding is that the texture and a real human head are not changing but only conditions of environment are changing. Therefore it is sufficient sending the texture only once, e.g. in the first frame of a video transmission. This is main benefit of the model coding based system [9]. The shape of human head is different for each human. Therefore we need 3D models fitted on a real human with different shape. In Fig. 1 a 3D model and its projection into two orthogonal images of human head is shown.

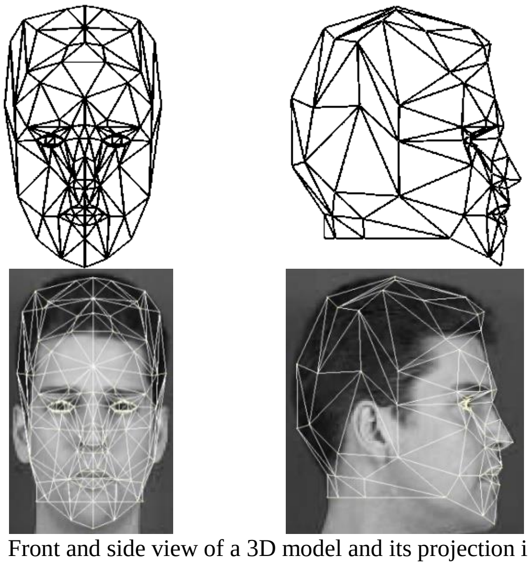

Fig. 1 Front and side view of a 3D model and its projection into orthogonal source images

2.1. Cubic transformation

Each model is defined by vertices and their mutual connections. An appropriate transformation of vertices coordinates is needed for texturing of surfaces of the closed model triangulations. The transformation must be performed, so the contributions of the polygons projected into the 2D horizontal-vertical (hv) plane do not overlap. The transformation can be categorized into closed and open transformations. It is necessary to uncouple the closed 3D model before its transformation in the case of the open transformation. The closed one does not need this treatment what simplifies the implementation. On the other hand this transformation introduces the excessive deformation of the model which causes errors in the texture. Hence we use the opened cubic transformation. As it was mentioned above, 3D model has to be uncoupled in back side which is the best for human head model. The transformation of 3D model into 2D texture plane is given by equations

$$h_{2D} = \begin{cases} h_{3D} + m_1 \left[m_2 \left(1 - \dfrac{r - r_{min}}{|r_{max} - r_{min}|} \right) \right]^3 \\ h_{3D} - m_1 \left[m_2 \left(1 - \dfrac{r - r_{min}}{|r_{max} - r_{min}|} \right) \right]^3 \end{cases} \quad (1)$$

$$v_{2D} = m_3 v_{3D} \quad (2)$$

where constants m$_1$, m$_2$, m$_3$ are used for set parameters of the transformation, r$_{min}$, r$_{max}$ are minimal and maximal values of r coordinate of whole 3D triangulation. A 3D shaped model transformed into 2D texture plane is shown in Fig. 2.

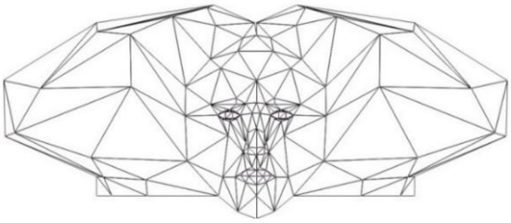

Fig. 2 3D shaped model transformed by the cubic transformation

2.2. Textures generation with smoothing

It is necessary to define which pixels of source images will be used in the generation. This identification is done by projection of a 3D model into source textures. This is shown in Fig. 1. The spatial transformation causes deformation of polygons. In general, shift and scaling of the polygons can occur. This leads to change of coordinates of point inside each polygon. Even after deformation there is a dependence between inside points and vertices of the polygon. This dependence is best described by barycentric coordinates [10, 7] which define coordinates of point on the base of three points of the polygon. In principle, texturing of a 3D model directly from source images is possible but it leads to processing of two textures and it also need sending information about which polygons use front and side view. On the other hand in this case the transformation is not needed. Hence it is better to use one texture. In case of generation one texture arises two partial textures. The first one is made from front source image pixels and the second one is made of side source image pixels. These partial textures are shown in Fig. 3.

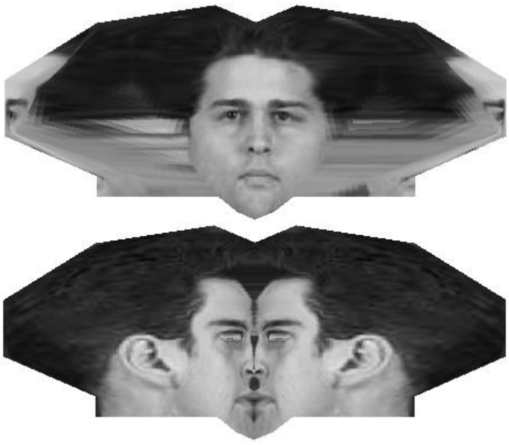

Fig. 3 Front and side partial texture

It is clear that these textures cannot be used in the texturing of 3D model. They have to merge into the final texture. This merge must be without visible edges in region of merging of partial textures. This is possible to be done by merging using the Laplace pyramid [11, 12]. Oval shaped binary mask is used for defining which pixels from the both partial textures will be used. This mask together with the partial textures are input of algorithm of merging image by using the Laplace pyramid. A final smoothed texture is shown in Fig. 4.

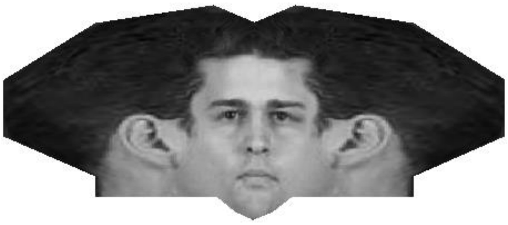

Fig. 4 Final smoothed texture

3. LOSSLESS TEXTURE CODING

As we mentioned above, in the model based video coding, a texture is send only once in the first frame. Hence, arising requirements on quality of the texture cause that the texture has to be lossless coded. The Model coding is designated for slow transmission channels therefore the texture has to be coded by high compression. The way how to satisfy these requirements will be shown in the next chapters.

3.1. Fast integer wavelet transformation

Drawback of classical DWT is that coefficients of the decomposition are real numbers. In digital signal processing it is better to work with integers. In computation of classical DWT is used the convolution between an input discrete signal and impulse responses of used filters. As it is known the convolution introduces extension of signals (sequences). Such extended signals have to be post-processed. This DWT implementation is also computationally demanding. These drawbacks are eliminated by the integer DWT (IDWT) [13, 14]. This implementation is not causing extension thus input sequences can be directly replaced by the DWT decomposition. Backward transformation is done directly from forward transformation by reordering of operations. IDWT has simple realization and can be used in lossless compression. By this method it is possible to achieve DWT with arbitrary wavelets. Let as assume biorthogonal bank of filters (BF) BF(5,3) [15, 16]. One decomposition stage with this BF can be replaced by lifting implementation in Fig. 5.

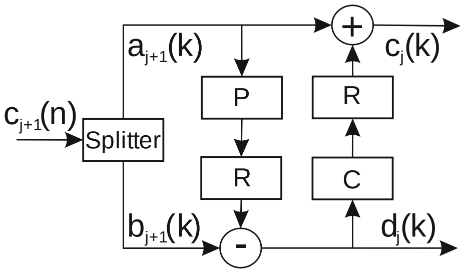

Fig. 5 IDWT decomposition stage

Let as assume an input sequence $c_{j+1}(n)$. This sequence is split into two branches. The bottom branch contains odd samples $b_{j+1}(k)$ and the top branch contains even samples $a_{j+1}(k)$ of the input sequence. Basis of this method is prediction of odd samples from even samples through predictor P. The predicted samples are deducted from odd samples. The resulting sequence are coefficients of details $d_j(k)$ of the DWT representation. The coefficients of approximation are results of a summation of even samples and samples of correction from corrector C. This correction is important, because that sequence of approximation $c_j(k)$ has to reflect the properties of input sequence $c_{j+1}(n)$. The correction and prediction are followed by rounding. This operation provides integer output of DWT. In general, properties of corrector and predictor of lifting implementation depend on used BF. Approximation and detail coefficients are computed by equations.

$$d_j(k) = b_{j+1}(k) - \left\lfloor \frac{1}{2}\left[a_{j+1}(k) + a_{j+1}(k+1)\right]\right\rfloor \qquad (3)$$

$$c_j(k) = a_{j+1}(k) + \left\lfloor \frac{1}{4}\left[d_j(k-1) + d_j(k)\right]\right\rfloor \qquad (4)$$

where operation $\lfloor * \rfloor$ is rounding to nearest integer.

3.2. Bit planes decomposition

In general, each image [17] can be expressed by two dimension raster of pixels and each pixel can be described by N bits as follows

$$pixel = p_0 2^0 + p_1 2^1 + \cdots + p_{N-1} 2^{N-1} \qquad (5)$$

Individual bits p_i correspond to the respective weights 2^i for i=0, 1... N-1. If we pick up only p_0 in such way defined pixel from entire polytonal image, we get its bit plane of zero weight (BP_0). Using the same process all other bit-planes (BP) until BP_{N-1} are created. If we use some transformation of image we cannot talk about pixels. This image consist of spectral or decomposition coefficients. From the theory we know that these coefficients can have also negative values. Hence, the decomposition algorithm has to be supplemented by a solution of this problem [7]. The problem of negative coefficients can be solved by two ways. The first one is decomposition on absolute bit planes (ABP) and the second one is decomposition on real bit planes (RBP). In case of ABP is created the bit plane which determine if the coefficient is positive or not. This plane is called the sign plane (BP_s). When BP_s is created others bit planes are created from absolute values of the transformed image. By the RBP algorithm we get positive and negative bit planes. The decomposition is applied on absolute values of the transformed image coefficients. Sign on coordinates of the particular coefficient determines if positive or negative bit plane will be created. It is clear that bit planes are binary images.

3.3. Binary state arithmetic encoding

Common used entropy codes like Huffman or Shanno-Fano codes have high degree of redundancy suppression but still use integer code length of each source alphabet symbol. Thus, if 3,1b is sufficient for encoding, one symbol code with length 4b has to be used. This disadvantage is eliminated by using the arithmetic coding [17, 18, 6].

3.3.1. Pixels probability approximation

Probability will be approximated by a value of 2^{-Q}. This means to find such linear refracted function L(p) from domain of definition $D_f(L(p))$, which is the best approximation of the entropy function H(p). Each value 2^{-Q} lies in another interval given by linear part of the function L(p). The function L(p) is given by equation (6) and domain of definition is given by (7).

$$L(p) = \begin{cases} f_1(p), & p \in (\, c_1, \infty > \\ f_2(p), & p \in (\, c_2, c_1 > \\ f_2(p), & p \in (\, c_3, c_2 > \\ & \vdots \\ f_k(p), & p \in (\, 0, c_k > \end{cases} \qquad (6)$$

$$D_f(L(p)) = \bigcup_{i=1}^{k} D_f(f_i(p))\ ;\ p \in (\, c_{i+1}, c_i > \qquad (7)$$

where f is linear refracted function, operand U is joining of domains and k is number of subintervals. Linear parts of the function L(p) are given by intersections of functions f(p). This is shown in Fig. 6.

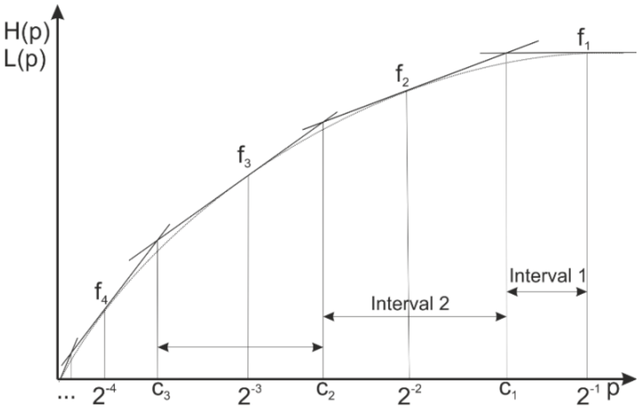

Fig. 6 Approximation of entropy H(p) by linear refracted function L(p)

3.3.2. Image model building

Arithmetic coder replaces input stream of symbols by decimal number from probability interval <0, 1). Arithmetic coder and decoder use input sequence of symbols also model of image which provides information about probabilities. For binary images the model is given by template which determines distribution of image points to the current point. An example of 10 points template for JBIG algorithm is in Fig. 7. Current point is marked as cross and points of template are marked as black spots.

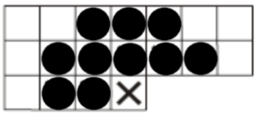

Fig. 7 Ten points template for JBIG algorithm

Model of the binary image is achieved by making all of combination "0" and "1" which the template can acquire. Each state of the model is given by equation (8).

$$S_i = \sum_{n=1}^{k} c_n 2^{n-1} \tag{8}$$

where i is index of state, k is number of points in template and c_n is binary value of point.
When we have defined model like this, it is possible to determine how frequently individual states of the model appear in a binary image and what value has the last point of the template. From this analysis it is possible by equation (9) to determine conditional probability between actual state and actual value of the binary image point (bp).

$$p(c = v|S_k) = p(S_{v,k}) = \frac{n_{v,k}}{n_k} \tag{9}$$

where v={0,1} is actual point value, S_k is actual state, $S_{v,k}$ is value of state with regard to actual point value and n_k is number of occurrences of S_k. It is clear that value of n_k in (9) cannot be 0. Hence in case when $n_k = 0$ we define $p(S_{v,k})=0$. In case where part of template lies out of image these points of template are assumed as 0.

3.3.3. Encoding

In the previous chapters statistical analysis of the binary image was shown. The final arithmetic code of image points sequence is recursively assembled by equations (10-13).

$$A(sk) = A(s)2^{-Q} \tag{10}$$

$$A(sm) = \lfloor A(s) - A(sk) \rfloor \tag{11}$$

$$C(sm) = C(s) \tag{12}$$

$$C(sk) = C(s) + A(sm) \tag{13}$$

The operation 2^{-Q} is shifting of Q bits to the right and it simultaneously presents the probability approximation of less probable bp. A(s) is size of the probability subinterval, A(sm) is size of this subinterval for more probably bp and A(sk) for less probably bp. Operation $\lfloor * \rfloor$ is removing of low significant bits. C(s) is lower limit of the probability subinterval. C(sm) and C(sk) are lower limits of the subinterval for more and less probably bp. By the recursively encoding values A(s) and C(s) are achieved and the resultant code will be binary variable form interval <C(s),C(s)+A(s)).

4. RESULTS

The resulting smoothed texture with bit depth 8b and raster size 512x512 pixels was encoded by our proposed method of the lossless texture coding. In the first experiment the texture is decomposed directly into its 8 bit planes. After that the bit planes were encoded by the binary state arithmetic code (BSAC). In Tab. 1 middle length of code words (CW) for each BP are listed. Abbreviation BIN represents binary code. In the second experiment texture was decomposed into its wavelet representation. The resultant wavelet sub-images were also decomposed into BP. The results for BSAC applied on ABP are listed in Tab.2 and for RBP in Tab. 3. In Tab.4 middle lengths of code words and level of the compression for all three experiments are compared.

Table 1 Middle length of CW after application of BSAC on BP of the texture in spatial domain.

	Middle length of CW [bit/bp]	
BP	BIN	BSAC
7	1	0.0116
6		0.0200
5		0.0730
4		0.1096
3		0.1617
2		0.2115
1		0.2302
0		0.2352
Sum	**8**	**1.0529**

From Tab. 1 it is obvious that middle length of CW after application of BSAC on the bit planes is significantly reduced. In comparison with 8 bit of binary code middle length is only 1.053 what mean compression about 86.84%.

Table 2 Middle length of CW after application of BSAC on ABP of the texture in IDWT domain.

	Middle length of CW [bit/bp]	
BP	BIN	BSAC
S	1	0.2388
7		0.0011
6		0.0048
5		0.0106
4		0.0295
3		0.0658
2		0.1238
1		0.2162
0		0.2667
Sum	**9**	**0.9572**

Table 3 Middle length of CW after application of BSAC on RBP of the texture in IDWT domain.

	Middle length of CW [bit/bp]	
BP	BIN	BSAC
7	1	0.0011
6		0.0036
5		0.0079
4		0.0199
3		0.0462
2		0.0876
1		0.1574
0		0.2142
-0		0.2102
-1		0.1564
-2		0.0833
-3		0.0422
-4		0.0176
-5		0.0052
-6		0.0020
Sum	**15**	**1.0548**

Table 4 Evaluation of efficiency of BSAC.

	Domain	No. of BP	Middle length of CW [bit/bp]	Reduced redundancy [bit/ bp]	Compression [%]
-	spatial	-	8	-	-
BP	spatial	8	1.0529	6.9471	86.84
RBP	IDWT	15	1.0548	6.9452	86.82
ABP	IDWT	9	0.9572	7.0428	88.04

From Tab.4 it is clear, that higher level of compression is achieved by decomposition of IDWT into RBP compared to ABP and BP. From the obtained results it is also clear that with BSAC quite high compression in all cases is achieved. The best compression is achieved by encoding ABPs of the texture transformed into the IDWT domain. The difference between the compression by encoding BPs in spatial domain and the best compression in IDWT domain is only 0.097 bit/BP what means 1.2 % increase of the compression. Hence, it is necessary to consider profitability of IDWT in regards to raster size in this application. In the compression images with large raster increasing compression of about 1.2 % can be favorable, although at the expense of increasing complexity of the system implementation. With the compression texture by JPEG in lossless mode the compression of 86.32 % was achieved. By application of the proposed compression technique on the selected texture slightly better results than with JPEG was achieved.

CONCLUSION

In this paper we dealt with problematic of lossless compression of 3D human head textures. In short, we described the procedure of texture generation on the base of two orthogonal views on human head and the 3D model transformed into 2D texture plane. We suggested a solution of the partial textures blending without transition on edges problem. The textures were transformed by lifting implementation of the integer DWT. These transformed textures were decomposed into bit planes. Wavelets coefficients are real number. Hence, there are two possible methods of decomposition of the transformed images into BP. The result of the first method (ABP) is creation of the absolute bit planes and sign bit plane. The second method (RBP) produces positive and negative bit planes. Bit planes can be encoded by the arithmetic binary state code. In addition the texture was decomposed into BP also in the space domain. From the experimental results it is clear that the best compression was achieved by encoding of ABP in the IDWT domain. Also we figured out that with our proposed method comparable results as with JPEG are achieved.

REFERENCES

[1] ZYDA, M.:From visual simulation to virtual reality to games, *Computer 38.9*, pp. 25-32, 2005.

[2] MISLOVE, A. – MARCON, M. – GUMMADI, K.P. – DRUSHEL, P. – BATTACHARJEE, B.: Measurement and analysis of online social networks, in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007.

[3] AIZAWA, K. – HUANG, T.S.: Model-based image coding advanced video coding techniques for very low bit-rate applications, *Proceedings of the IEEE*, vol. 83, no. 2, pp. 259-271, 1995.

[4] ANTONINI – MARC: Image coding using wavelet transform, *Image Processing, IEEE Transactions on 1.2*, pp. 205-220, 1992.

[5] LIAN, C. –CHEN, K. F. –CHEN, H. H. –CHEN, L. G.: Lifting based discrete wavelet transform architecture for JPEG2000, *In Circuits and Systems*, no. 2, 2001.

[6] MRAK, M. – MARPE, D. – GIRGIC, S.: Comparison of Context-Based Adaptive Binary Arithmetic Coders in Video Compression, in *Proc. 4th EURASIP Conference focused on Video/Image Processing and Multimedia Communications*, Zagreb, 2003.

[7] MIHALÍK, J. – KOVÁČ, O.: Číslicové spracovanie textúr ľudskej hlavy, Košice: LDIPV, FEI TU Košice, 2014.

[8] LEE, W. S. – MAGNENAT-THALMANN, N.: Fast head modeling for animation, *Image and Vision Computing*, vol. 18, no. 4, pp. 355-364, 2000.

[9] DOENGES, P. K. – CAPIN, T. K. – LAVAGETTO, F. – OSTERMANN, J. – PANDZIC, I. S. – PETAJAN, E. D.: MPEG-4: Audio/video and synthetic graphics/audio for mixed media, *Signal Processing: Image Communication*, vol. 9, no. 4, pp. 433-463, 1997.

[10] MEYER, M. – BARR, A. – LEE, H. – DESBRUN, M.: Generalized barycentric coordinates on irregular polygons, *Journal of graphics tools*, 2002.

[11] BURT, P. – ADELSON, E.: The Laplacian Pyramid as a Compact Image Code, *Communications, IEEE Transactions*, vol. 31, no. 4, pp. 532-540, 1983.

[12] KOVÁČ, O. – MIHALÍK, J.: Generovanie a vyhladzovanie textúr ľudskej hlavy, *Elektrorevue*, vol. 16, no. 1, 2014.

[13] PUTTARAJU, M. – ASWATHA, D. A.: FPGA Implementation of 5/3 Integer DWT for Image Compression, *International Journal of Advanced*

Computer Science and Applications, vol. 3, no. 10, 2012.

[14] YEH, P. S. – ARMBRUSTER, P. – KIELY, A. – MASSCHELEIN, B. – MOURY, G. – SCHAEFER, C. – THIEBAUT, C.: The new CCSDS image compression recommendation., in *Aerospace Conference, 2005 IEEE* .

[15] SWELDENS, W.: The Lifting Schneme: A New Philosophy in Biorthogonal Wavelet Constructions, in *Proc.SPIE 2569*, 1995.

[16] DILLEN, G. – GEORIS, B. – LEGAT, J. D. – CANTINEAU, O.: Combined line-based architecture for the 5-3 and 9-7 wavelet transform of JPEG2000., *Circuits and Systems for Video Technology,* vol. 13, no. 9, 2003.

[17] GLADIŠOVÁ, I. – MIHALÍK, J.: Stavové aritmetické kódovanie binárnych obrazov, *Acta Electrotechnica et Informatica,* vol. 3, no. 3, pp. 36-44, 2003.

[18] HOWARD, P. – VITTER, J.: Analysis of Arithmetic Coding for Data Compression, *Communication of the ACM,* vol. 28, no. 6, 1992.

Received October 10, 2015, accepted January 10, 2015

BIOGRAPHIES

Ondrej Kováč was born on 1986. In 2011 he received the Ing. degree in Multimedia telecommunications from the Faculty of Electrical Engineering, Technical University of Košice. His PhD. thesis topic was focused to texture generating, 3D modeling and coding of human head. Since June 2015, he has been working as assistant professor at the Department of Technologies in Electronics, Technical University of Kosice.

Ján Mihalík graduated from the Technical University of Bratislava in 1976. In 1979, he joined the Faculty of Electrical Engineering and Informatics of Technical University of Košice, where he received his PhD degree of radioelectronics in 1985. Currently, he is a full professor of electronics and telecommunications and the head of Laboratory of Digital Image Processing and Videocommunications at Department of Electronics and Multimedia Telecommunications. His research interests include information theory, image and video coding, digital image and video processing and multimedia videocommunications.