ISSN 1330-3651
UDC/UDK 621.397.3:616-073

FRAMEWORK FOR 4D MEDICAL DATA COMPRESSION

Martin Žagar, Mario Kovač, Daniel Hofman

Original scientific paper

This work presents a novel framework for four-dimensional (4D) medical data compression architecture. This framework is based on different procedures and algorithms that detect time and spatial (frequency) redundancy in recorded 4D medical data. Motion in time is analyzed through the motion fields that produce input parameters for the neural network used for motion estimation. Combination of segmentation, block matching and motion field prediction along with expert knowledge are incorporated to achieve better performance. Frequency analysis is done through an extension of one dimensional wavelet transformation to three dimensions. For still volume objects different wavelet packets with different filter banks can be constructed, providing a wide range of frequency analysis. With combination of removing temporal and spatial redundancies, very high compression ratio is achieved.

Keywords: *3D wavelet transformation, 4D medical data, block matching, motion field, temporal and frequency redundancies*

Programski okvir za 4D kompresiju medicinskih podataka

Izvorni znanstveni članak

U ovom radu predložen je novi programski okvir za kompresiju četvero-dimenzionalnih (4D) medicinskih podataka. Arhitektura ovog programskog okvira temelji se na različitim procedurama i algoritmima koji detektiraju vremenske i prostorne zalihosti u ulaznim 4D medicinskim podacima. Pokret kroz vrijeme analizira se pomoću vektora pomaka koji predstavljaju ulazne parametre za neuronske mreže koje se koriste za procjenu pokreta. Kombinacijom segmentacije, pronalaženja odgovarajućih blokova i predikcijom vektora pomaka, zajedno s ekspertnim znanjem moguće je optimirati performanse sustava. Frekvencijska svojstva se analiziraju proširenjem wavelet transformacije na tri dimenzije. Za mirne volumetrijske objekte, moguće je konstruirati različite wavelet pakete s različitim filtrima koji omogućavaju širok raspon analiza frekvencijskih zalihosti. Kombinacijom uklanjanja vremenskih i prostornih zalihosti moguće je postići vrlo visoke omjere kompresije.

Ključne riječi: *3D wavelet transformacija, 4D medicinski podaci, polje pomaka, pronalaženje odgovarajućih blokova, vremenske i prostorne zalihosti*

1 Introduction

Compression enables a more efficient use of transmission and storage resources. Efficient usage of resources may not be an issue with 2D video and audio any more, but 3D video still requires compression due to very large volume of data. Even with constant advances in storage and transmission capacity, compression is likely to be an essential component of multimedia services for many years to come. This work proposes the framework for four-dimensional medical data compression architecture that can be used in various telemedicine applications, mainly in neuroimaging which provides a crucial perspective for basic and clinical human neuroscience.

Given a data, it is important to analyze its mathematical properties and the extent to which it allows forming a conclusion about the shape and motion. To extract the motion information from specific features, the visualization semantics of the system that will synthesize the motion should be known. Deformable models deliver information about the feature in the form of the magnitudes of parameters that control the analysis. These parameters are related to the actions that should be applied to the 3D-model to recreate motion and expressions. This work translates the results of motion models into object visualization parameters.

Motion estimation is used to eliminate large amounts of temporal redundancy that exist in sequences of 3D data. Each object or subvolume of considered scene can have its own freedom of motion and the activity recognition for each part is achieved by using models obtained by the neural network. In this context, an object can be described as a part of the scene that moves with a coherent motion. To detect the objects it is necessary to use one of the segmentation techniques. This work introduces the usage of semi -automated segmentation techniques, based on the modified Canny edge detection as well as surface extraction, since segmentation requires to estimate the surface shape. The strategy of edge-based segmentation is to find object boundaries and segment regions enclosed by the boundaries. When subvolume or object is determined, it can be described by its own motion field.

Once object expressions are modelled, a set of parameters is obtained. The mapping of these parameters onto the corresponding object animation parameters is done by solving for the estimator that relates motion parameters to analysis parameters. To establish the mapping relationship there must be a training process [1]. Therefore, a reliable estimator is constructed based on convolutional neural networks (CNN). This work develops neural networks with applications to the motion estimation of subvolumes. Supervised learning in the context of convolutional neural networks can be associated with the use of optimization-based techniques to adjust the network parameters. The objective is to minimize the cost function that somehow defines the desired behaviour to be achieved.

Different types of time-frequency structures are encountered in complex signals such as 3D video recordings [2]. This motivates the design of bases whose time-frequency properties may be adapted. Wavelet bases are one particular family of bases that represent piecewise smooth signals effectively. In this work, fast one dimensional wavelet transform is extended to three dimensions in order to reduce frequency redundancies. The fast wavelet transform is computed with filters that are separable products of the one-dimensional low-pass and high-pass filters. The wavelet coefficients are calculated with three-dimensional separable convolutions and subsamplings. The decomposition formula is obtained by applying the one-dimensional convolution formula to the separable three-dimensional wavelets and scaling

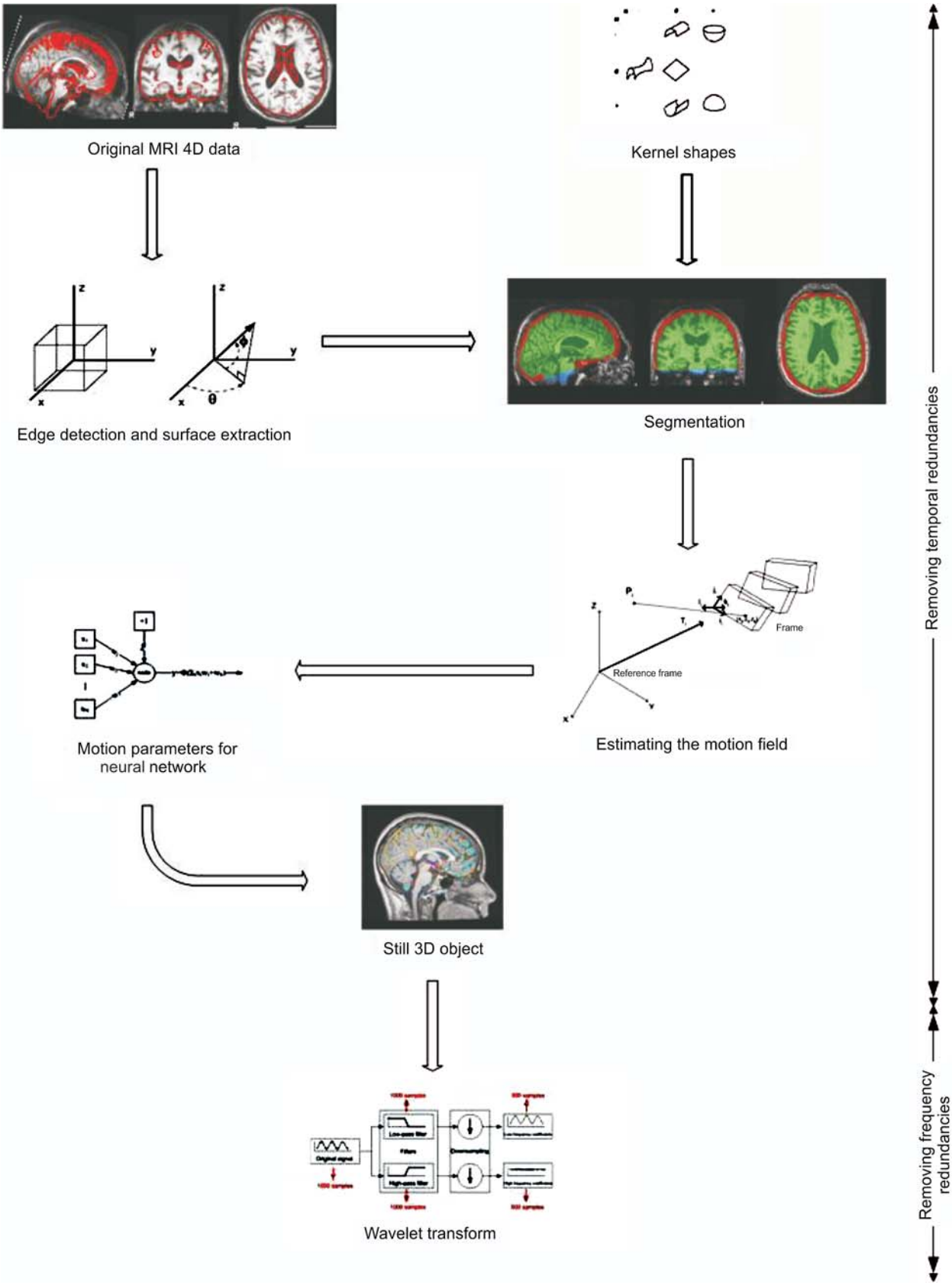

Figure 1 Proposal of 4D medical data compression architecture

functions.

2 System architecture

The proposed new architecture of 4D medical data compression is presented in Fig. 1. It can be divided into two main parts. The first part is exploring temporal redundancies and the second one is exploring frequency redundancies. The temporal analysis begins with a shape analysis of an original 4D dataset based on the edge detection. The dataset is then segmented according to the kernel object that best fits its contours and characteristics. Usage of convolutional neural networks enables that each

segmented subvolume can have its own motion field and its motion is estimated, independently of other subvolumes and their motion fields. After removing the temporal redundancies, frequency redundancies are removed from still volumetric datasets by using the fast 3D wavelet transform. This produces wavelet coefficients which are then coded, ready for transmission or storage.

2.1 Exploring temporal redundancies

Time analysis includes statistical analysis and object registration. During the statistical analysis, 4D data is divided into time frames that are examined. Data segmentation is used for examination of each time frame when trying to match it with a kernel shape. In 3D motion estimation theory, it is usually assumed that the whole system can be described as only one moving object. When dealing with real world multiple moving objects and complex scenes, there are lots of objects moving with different motions, locally in space and in time [3]. In this way the scene is broken down into a number of regions each of which can be well approximated by its own motion from a class of simple models.

2.1.1 3D canny edge detector based segmentation

Detection of edges can be achieved by adopting 3D Canny edge detector [1]. The edge gradient of a 3D volumetric object G can be determined by computing the gradient components $G_x = \partial G/\partial x$, $G_y = \partial G/\partial y$ and $G_z = \partial G/\partial z$ for each voxel $v(x,y,z)$. The second step is to estimate the edge strength with $e_d(x,y,z) = \sqrt{G_x^2 + G_y^2 + G_z^2}$ orientation of the edge normal.

The orientation of edge normal is specified by angles θ and ϕ that can be computed by equations

$$\tan\theta = \frac{G_y}{G_x} \tag{1}$$

$$\tan\phi = \frac{G_z}{\sqrt{G_x^2 + G_y^2}}. \tag{2}$$

2.1.2 Surface extraction

Many 3D objects can be conveniently described in terms of the shape and position of the surfaces they are made of. Surface-based descriptions are used for object classification and motion estimation in the compression process [4]. This section presents a well-known method of finding patches of various shapes which compose the visible surface of an object, adapted to 3D.

The method called HK segmentation [5] partitions a range object into regions of homogeneous shape, called homogeneous surface patches. For a given range object G, the goal is to compute a new object registered with G and with the same size in which each voxel is associated with a local shape class selected from a given dictionary. To solve this problem, two tools are needed: a dictionary of shape classes and an algorithm determining which shape class gives the best approximation of the surface at each voxel.

To estimate the surface shape at each voxel, a local definition of shape is needed. Differential geometry provides a convenient one: the local surface shape can be classified using the sign of the mean curvature H and of the Gaussian curvature K. In Tab. 1, concave and convex are defined with respect to the viewing direction: a hole in the range surface is concave and its principal curvatures negative. At cylindrical points, one of the two principal curvatures vanishes, as, for instance, at any point of a simple cylinder or cone. At elliptic points, both principal curvatures have the same sign, and the surface looks locally like either the inside of a bowl (if concave) or the tip of a nose (if convex). At hyperbolic points, the principal curvatures are nonzero and have different signs; the surface resembles a saddle. This classification is qualitative in the sense that only the sign of the curvatures, not their magnitude, influences the result. This offers some robustness, as sign can often be estimated correctly even when magnitude estimates become noisy.

Table 1 Surface patches classification scheme

K	H	Local shape class
0	0	plane
0	+	concave cylindrical
0	-	convex cylindrical
+	+	concave elliptic
+	-	convex elliptic
-	any	hyperbolic

The expressions of H and K are evaluated at each voxel [6], with signs from Tab. 1. The Gaussian curvature operator K for 3D objects can be computed as

$$K(x,y,z) = \frac{(\nabla G_1^{\perp})^{\mathrm{T}} S_{G1} \nabla G_1^{\perp} + (\nabla G_2^{\perp})^{\mathrm{T}} S_{G2} \nabla G_2^{\perp} + (\nabla G_3^{\perp})^{\mathrm{T}} S_{G3} \nabla G_3^{\perp}}{|\nabla G|^2} \tag{3}$$

using (subscripts x, y, z indicate partial differentiations)

$$G_1^{\perp} = \begin{bmatrix} -G_y \\ G_x \end{bmatrix}, \; G_2^{\perp} = \begin{bmatrix} -G_z \\ G_x \end{bmatrix}, \; G_3^{\perp} = \begin{bmatrix} -G_z \\ G_y \end{bmatrix}, \tag{4}$$

and

$$S_{G1} = \begin{bmatrix} G_{xx} & G_{xy} \\ G_{yx} & G_{yy} \end{bmatrix}, \; S_{G2} = \begin{bmatrix} G_{xx} & G_{xz} \\ G_{zx} & G_{zz} \end{bmatrix}, \; S_{G3} = \begin{bmatrix} G_{yy} & G_{zy} \\ G_{yz} & G_{zz} \end{bmatrix}. \tag{5}$$

Mean curvature H of a 3D object can be extended from a 2D expression as

$$H(x,y,z) = \frac{G_x^2(G_{yy} + G_{zz}) + G_y^2(G_{zz} + G_{xx}) + G_z^2(G_{xx} + G_{yy})}{2\left(G_x^2 + G_y^2 + G_z^2\right)^{3/2}} - \frac{2(G_x G_y G_{xy} + G_x G_z G_{xz} + G_y G_z G_{yz})}{2\left(G_x^2 + G_y^2 + G_z^2\right)^{3/2}}. \tag{6}$$

To summarize, the basic HK segmentation algorithm works as follows: the input is a range volumetric object, G, in an $g_{x,y,z}$ form, and a set of six shape labels, $\{s_1,\ldots,s_6\}$. associated to the classes of Tab. 1. The derivatives G_x, G_y, G_z, G_{xy}, G_{xz}, G_{yz}, G_{xx}, G_{yy} and G_{zz} should be computed first. After computing the H and K, the shape object L can be computed by assigning a shape label s_i to each voxel, according to the rules in Tab. 1.

2.1.3 Estimating the motion field

In estimating the motion field of segmented subvolume the basic assumption is that the motion field is well approximated by a constant vector field $\boldsymbol{v}$ within any small region of the object plane. The optical flow is the approximation of the motion field which can be computed from time-varying volumetric sequences [7]. For each point p_i within a segmented subvolume Q of size $N\times N\times N$, it can be written

$$(\nabla E)^{\mathrm{T}} v + E_t = 0 \tag{7}$$

where $E = E(x,y,z,t)$ is brightness for voxel at point (x,y,z) at time t. Subscript E_t denotes partial differentiation with respect to time. The spatial and temporal derivatives of the voxel brightness are computed at $p_1, p_2, \ldots, p_{N\times N\times N}$. Therefore, the optical flow can be estimated within Q as the constant vector, $\boldsymbol{v}$ that minimizes the functional

$$\psi[v] = \sum_{p_i \in Q} \left[\nabla E^{\mathrm{T}} v + E_t\right]^2 \tag{8}$$

The solution of the least squares problem can be found by solving the linear system $A^{\mathrm{T}}Av = A^{\mathrm{T}}b$ [8]. The i-th row of the $N^3\times 3$ matrix $\mathbf{A}$ is the spatial gradient evaluated at point p_i

$$\boldsymbol{A} = \begin{bmatrix} \nabla E(p_1) \\ \nabla E(p_2) \\ \vdots \\ \nabla E(p_{N\times N\times N}) \end{bmatrix} \tag{9}$$

and $\boldsymbol{b}$ is the N^3 - dimensional vector of partial temporal derivatives of the voxel brightness, evaluated at $p_1, p_2, \ldots, p_{N\times N\times N}$ after a sign change

$$\boldsymbol{b} = -\left[E_t(p_1), \ldots, E_t(p_{N\times N\times N})\right]. \tag{10}$$

This algorithm is as follows. The input is a time-varying sequence of n segmented subvolumes $E_1, E_2, \ldots, E_3$. Let Q be a sub-volume region of $N\times N\times N$ voxels.

- Filter each subvolume of the sequence with a Gaussian filter of standard deviation equal to σ_S (typically 1,5 voxels) along each spatial dimension.
- Filter each subvolume of the sequence along the temporal dimension with a Gaussian filter of standard deviation σ_t (typically 1,5 frames).
- For each voxel of each subvolume of the sequence:
  (a) compute the matrix $\boldsymbol{A}$ and the vector $\boldsymbol{b}$ using (9) and (10)
  (b) compute the optical flow using (8).

The output is the optical flow computed in the last step [9]. The purpose of spatial filtering is to attenuate noise in the estimation of the spatial gradient; temporal filtering prevents aliasing in the time domain. The least squares solution of the constrained system can be obtained as $\boldsymbol{v} = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b$. $\boldsymbol{v}$ is the optical flow (the estimate of the motion field) at the centre of subvolume Q; by repeating this procedure for all segmented subvolumes, a dense optical flow is obtained.

2.1.4 Motion parameters for neural network

To recreate motion, it is necessary to find a mathematical solution to tie analysis to synthesis. To extract motion information from specific features, the visualization semantics of the system that will synthesize the motion should be known. It is also necessary to relate these parameters to the actions that should be applied to the 3D-model in order to recreate motion. These motion models translate the results into object-visualization parameters building motion fields presented in previous section.

Convolutional neural networks can be used to track and recreate motion and expressions from relevant features from an object database by carefully selecting the network parameters. A CNN typically consists of an input layer, one or more hidden layers (convolution layers), and an output layer [10]. The input layer of the CNN contains $M\times M\times M$ input nodes that represent $M\times M\times M$ voxels of the region on interest (ROI) to be recognized. Each convolution layer has local connections with the preceding layer (either the input layer or the previous hidden layer). The nodes in the hidden layer are organized into different groups and the groups between adjacent layers are locally connected by weighting coefficients that are arranged in kernels. The input layer contains $M\times M\times M$ nodes and the kernel size is $K\times K\times K$. The hidden layer is organized in n groups. The number of nodes in each group is $N\times N\times N$ ($N=M-K+1$ for a $K\times K\times K$ kernel). The total number of links between the input layer and the hidden layer is $n\times K^3\times N^3$. Learning is constrained so that the kernel weights connecting two groups in the adjacent hidden layers (or between the input and the hidden layers) are shift-invariant. As a result, the number of independent links is $n\times K^3$. The output layer is fully connected to the previous hidden layer. If the output layer contains N_Z nodes, the total number of links is $N_Z\times n\times N^3$. The signal input into the CNN is convolved with the weight kernels, and the value of the convolution is collected into the corresponding node in the upper layer. The value is further processed by the node through an activation function and produces an output signal. The output signal is forward-propagated in a similar manner.

CNN first extracts the subvolumes in the hidden convolution layers and then classifies these subvolumes in the output stage [10]. The convolution kernels are trained to form a set of feature extractors by supervised learning. Each convolution kernel is expected to function as a subvolume analyzer and perform a specific feature-extracting operation on the considered subvolume. Finally, the output stage is trained to classify and predict the position of the considered subvolume at the output layer.

2.2 Exploring frequency redundancies

Exploring frequency redundancies of still 3D objects is based on the fast wavelet transform. To explore frequency redundancy, the original data is decomposed into low and high frequency coefficients using wavelet transform. To obtain the standard decomposition of a single 3D object, one-dimensional wavelet transform is first applied to each row of voxel values, then to each column and then in depth. This operation gives an average value along with the coefficients.

2.2.1
Fast 3D wavelet transform

The fast wavelet transform is computed with filters that are separable products of one-dimensional low-pass $h[n]$ and high-pass $g[n]$ filters. The separable 3-dimensional low-pass filter is

$$h^0[n] = h[n_1]h[n_2]h[n_3] \tag{11}$$

where $n=(n_1, n_2, n_3)$ represent translating parameters along all 3 dimensions (x,y,z) respectively.

Denote that $u^0[m]=h[m]$ and $u^1[m]=g[m]$. Any integer $\varepsilon=\varepsilon_1,..., \varepsilon_n$ written in a binary form, can be associated to a separable 3-dimensional band-pass filter:

$$g^{\varepsilon}[n] = u^{\varepsilon_1}[n_1]u^{\varepsilon_2}[n_2]u^{\varepsilon_3}[n_3]. \tag{12}$$

The wavelet coefficients at the scale 2^{j+1} are calculated from input segmented subvolume $a_j=Q$ with three-dimensional separable convolutions and sub-samplings. The decomposition formulae are obtained by applying the one-dimensional convolution formula to the separable three-dimensional wavelets and scaling functions for $n=(n_1, n_2, n_3)$. If it is denoted that $\bar{h}^{\varepsilon}[n]=h^{\varepsilon}[-n]$ and $\bar{g}^{\varepsilon}[n]=g^{\varepsilon}[-n]$, then it can be verified that

$$a_{j+1}[n] = a_j * \bar{h}^0[2n] \tag{13}$$

$$d^{\varepsilon}_{j+1}[n] = a_j * \bar{g}^{\varepsilon}[2n]. \tag{14}$$

In an extended way, this can be rewritten as:

$$a_{j+1}[n] = a_j * \bar{h}\bar{h}\bar{h}[2n] \tag{15}$$

$$d^1_{j+1}[n] = a_j * \bar{h}\bar{h}\bar{g}[2n] \tag{16}$$

$$d^2_{j+1}[n] = a_j * \bar{h}\bar{g}\bar{h}[2n] \tag{17}$$

$$d^3_{j+1}[n] = a_j * \bar{h}\bar{g}\bar{g}[2n] \tag{18}$$

$$d^4_{j+1}[n] = a_j * \bar{g}\bar{h}\bar{h}[2n] \tag{19}$$

$$d^5_{j+1}[n] = a_j * \bar{g}\bar{h}\bar{g}[2n] \tag{20}$$

$$d^6_{j+1}[n] = a_j * \bar{g}\bar{g}\bar{h}[2n] \tag{21}$$

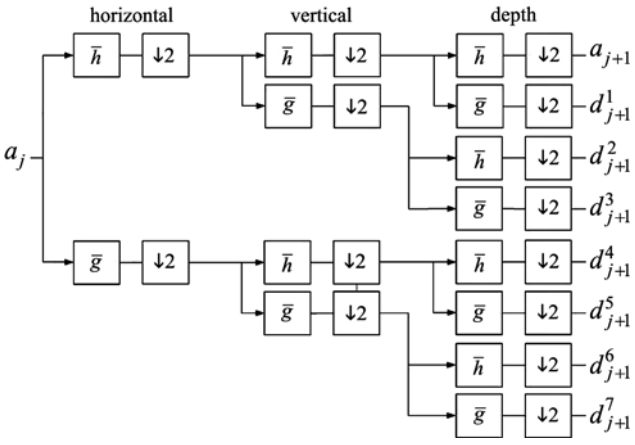

Figure 2 Three-dimensional decomposition and subsampling along rows, columns and depth

$$d^7_{j+1}[n] = a_j * \bar{g}\bar{g}\bar{g}[2n]. \tag{22}$$

Separable three-dimensional convolution can be factored into one-dimensional convolutions along all three dimensions. With the factorization illustrated in Fig. 2, these eight convolution equations are computed with 14 groups of one-dimensional convolutions.

The rows of a_j are first convolved with $h[n]$ and $g[n]$ and subsampled by 2. The columns of these two output volumes are then convolved with $h[n]$ and $g[n]$ respectively and subsampled. Finally, the depth component of these four output volumes is then convolved with $h[n]$ and $g[n]$ respectively and subsampled, which provides eight subsampled subvolumes denoted by formulae (15)-(22).

3
Implementation and conclusion

This chapter presents details of implemented methods and algorithms on four-dimensional data sets (volumes in time). A sequence from 10 and 20 volumes is taken during a typical experiment. There are two different experimental datasets, one representing the brain motion and the other representing human heart beating cycle.

The basic tool used for viewing volumetric data is FSL Viewer [11] with its component tools. It is used for visualization and inspection if output data is processed correctly. The basic tool used for data analysis and processing is MATLAB with its Wavelet Toolbox, Image Processing Toolbox, Neural Networks Toolbox, Fuzzy Logic Toolbox and Filter Design Toolbox.

Brain can be segmented into three main tissue types: grey matter, white matter and cerebro-spinal fluid. An example of segmentation explained in the Section 2.1.1 and 2.1.2 is shown in Fig. 3.

With coding only segmented region of interest (brain dataset) and not the whole head dataset, size of dataset is reduced from 497 kB to 405 kB. The reduction can be even higher if it is segmented into smaller regions of interest.

Figure 3 Segmentation of experimental neuro dataset

A motion estimation process is applied on small segmented regions. In this way, each subvolume that is segmented from the original object can have its own motion field. Motion of the beating heart can be derived heuristically from the subvolumes processing results themselves. Model of the human heart is known and described with expert knowledge. One beat can be approximated with cosine based function. Motion in one cardiac cycle can be approximated by using unscented

Kalman filter that searches the optimal parameter of the convolutional neural network, based on input training data. Results of trained CNN are shown in Fig. 4.

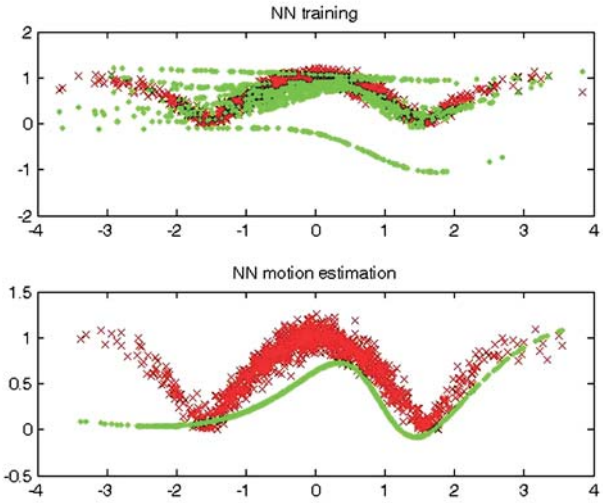

Figure 4 Training of CNN for motion estimation of the human heart cycle

The red dots represent the input motion of one heart cycle. The green dots on the upper part of Fig. 4 represent the training of CNN. After the learning phase, the estimated motion of the heart cycle is represented by the green line on the lower part of Fig. 4. In this way it is necessary to code only the motion field between two consecutive frames, produced with neural network for motion estimation. Original experimental data frame is size of 405 kB and if two consecutive frames are coded without any prediction method, the whole second object must be coded too, so the size is 807 kB (because not all the frames are the same size, e.g. the second frame of experimental data is 402 kB). For coding only motion vectors in motion field it is enough only 31 kB, and for both volumetric frames together 436 kB. With bigger number of frames, compression rate becomes much better, because it is necessary to code only the first frame, and after that only the motion fields of each consecutive frame. In this way for the second frame it is needed only 31 kB instead 402 kB (compression ratio 1:13).

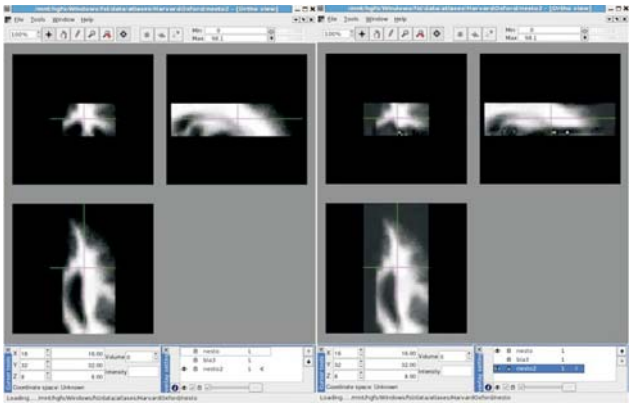

Figure 5 Segment of original experimental 3D neurodata (left) and after 3D FWT and IFWT using the Haar filter (right)

As previously described, frequency analysis is based on fast 3D wavelet transform. During each level of the transform, the scaling function and the seven wavelet functions are applied, respectively, to the original subvolume at that level, forming a total of eight subvolumes, one smooth and seven detailed subvolumes. The wavelet coefficients are the voxel values of eight subvolumes after the transform. With properly chosen wavelet functions, the low-resolution component in the j level is 2^{-3j} of the original subvolume size after the transformation, but contains about 90 % of the total energy in the j level. High-resolution components are spread into different decomposition levels. For those reasons, since the wavelet transform components provide an efficient representation of the original subvolume for compression purposes, different levels of the representation can be encoded differently to achieve a desired compression ratio, up to 1:17. Removing frequency redundancies is done by 3D decomposition with the 3D fast wavelet transform on experimental brain dataset. The results are shown in Fig. 5.

In this example, simple Haar basis [12] was used as a filter basis for wavelet transformation. It can be seen that the transformed data have viewable errors. For example, the side plan of the transformed data has a black area with few white points that does not exist on the original, and the background in noticeable noisy. Instead of the Haar filter, the use of the Farras filter [13] for analysis gives output data with no viewable errors (Fig. 6).

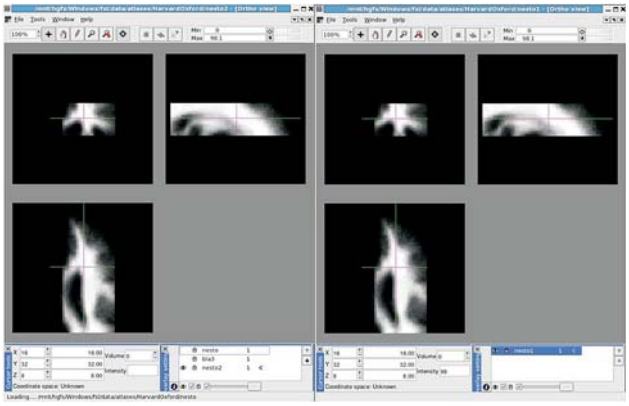

Figure 6 Segment of original experimental 3D neurodata (left) and after 3D FWT and IFWT using the Farras filter (right)

The Farras filter for analysis (F_A) of data whose dimensions are multipliers of number 8 and Farras filter for synthesis (F_S) are defined as

$$F\_A = \begin{bmatrix} 0 & -0{,}0112268 \\ 0 & 0{,}0112268 \\ -0{,}0883883 & 0{,}0883883 \\ 0{,}0883883 & 0{,}0883883 \\ 0{,}6958799 & -0{,}6958799 \\ 0{,}6958799 & 0{,}6958799 \\ 0{,}0883883 & -0{,}0883883 \\ -0{,}0883883 & -0{,}0883883 \\ 0{,}0112268 & 0 \\ 0{,}0112268 & 0 \end{bmatrix}, \quad F\_S = \begin{bmatrix} 0{,}0112268 & 0 \\ 0{,}0112268 & 0 \\ -0{,}0883883 & -0{,}0883883 \\ 0{,}0883883 & -0{,}0883883 \\ 0{,}6958799 & 0{,}6958799 \\ 0{,}6958799 & -0{,}6958799 \\ 0{,}0883883 & 0{,}0883883 \\ -0{,}0883883 & 0{,}0883883 \\ 0 & 0{,}0112268 \\ 0 & -0{,}0112268 \end{bmatrix}. \quad (23)$$

Size of one still uncompressed experimental volumetric data is 405 kB. If this data is compressed with standard WinZip compression, size is 129 kB (compression ratio 1:3). Using of 3D FWT with Farras filter, the original single volumetric data can be compressed on 51 kB with compression ratio 1:8. With Haar basis the results are even better: compressed data uses just 24 kB of memory (compression ratio 1:17). Elapsed time for 3D FWT of experimental data was 1,57 seconds on Pentium IV, 512 MB

Table 2 Results of proposed compression methods applied on experimental neuro-dataset

Size	Original raw dataset	After removing temporal redundancies	After removing spatial redundancies	Overall
Experimental neuro-dataset	24 503 kB	1 366 kB	171 kB	
Compression ratio		1:18	1:8	1:144

RAM. Results after removing temporal and spatial (frequency) redundancies from experimental neurodata are shown in Tab. 2. Overall achieved compression ratio is 1:144.

The suggested data compression architecture can be implemented in telemedicine to improve the quality of service. It incorporates operations for frequency and time analysis of datasets, creating kernel shapes and models, and fitting of medical 4D datasets. Each part of the system architecture is implemented independently and can be used for other approaches such as entertainment applications and other new applications that use 4D data. Building more sophisticated models of regions, allows more direct and interesting questions to be asked at the voxel level, as well as the carrying through of more relevant information to connectivity algorithms. Future research can be directed towards improving performance of the compression architecture through the combination of different wavelet packages, tune up of the neural network for motion estimation, implementation of more complex edge detectors for data segmentation as well as exploration of the different data shapes and models and more featured data (for example, colour dataset).

4 References

[1] Žagar, M. 4D Medical Data Compression Architecture. Doctoral dissertation, Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia, 2009.

[2] Klapan, I.; Kovač, M.; Lončarić, S.; Šimičić, Lj.; Raos, P. Virtual reality and tele-3D-computer assisted surgery in otorhinolaryngology. // Telemedicine-e-Health. 12, 2(2006), pp. 46-52.

[3] Efficient Management of Medical Images. // Telemedicine / Mario Kovač. Zagreb: Telemedicine Association, 2005. pp. 188-191.

[4] Lelescu, D.; Sconfeld, D. Statistical Sequential Analysis for Real-Time Video Scene Change Detection on Compressed Multimedia Bitstream. // IEEE Transactions on Multimedia. 5, 1(2003), pp. 106-118.

[5] Corrochano, E. B. Handbook of Geometric Computing Applications in Pattern Recognition. Computer Vision, Neuralcomputing and Robotics, Springer-Verlag, Berlin, Germany, 2005.

[6] Saito, H.; Baba, S.; Kanade, T. Appearance-Based Virtual View Generation from Multicamera Videos Captured in the 3-D Room. // IEEE Transactions on Multimedia. 5, 3(2003), pp. 303-317.

[7] Knezović, J.; Kovač, M.; Klapan, I.; Mlinarić, H.; Vranješ, Ž.; Lukinović, J.; Rakić, M. Application of novel lossless compression of medical images using prediction and contextual error modeling. // Collegium antropologicum. 31, 4(2007), pp. 1143-1150.

[8] Nascimento, J. C; Marques, J. S. Robust Shape Tracking in the Presence of Cluttered Background. // IEEE Transactions on Multimedia. 6, 6(2004), pp. 852-862.

[9] Hemmendorf, M. Motion Estimation and Compensation in Medical Imaging. Doctoral dissertation, Institute of Technology, Linkoping University, Sweden, 2001.

[10] Tang, H.; Tan, K. C.; Yi, Z. Neural Networks: Computational Models and Applications. Springer-Verlag, Berlin, Germany, 2007.

[11] FSLView, ttp://www.fmrib.ox.ac.uk/fsl/fslview/index.html (18.04.2011.)

[12] Talkuder, K. H.; Harada, K. Haar Wavelet Based Approach for Image Compression and Quality Assessment of Compressed Image. // IAENG International Journal of Applied Mathematics. 36, 1 (2007), pp. 1-9.

[13] Abdelnour, A. F. Tight Frame Various Lengths Filters. // World MultiConference on Systemics Cybernetics and Informatics. 718, (2003), pp. 1-4.

Authors' addresses

Dr. sc. Martin Žagar
Fakultet elektrotehnike i računarstva
Unska 3, 10000 Zagreb, Hrvatska
e-mail: martin.zagar@fer.hr

Prof. dr. sc. Mario Kovač
Fakultet elektrotehnike i računarstva
Unska 3, 10000 Zagreb, Hrvatska
e-mail: mario.kovac@fer.hr

Daniel Hofman, dipl. ing .rač.
Fakultet elektrotehnike i računarstva
Unska 3, 10000 Zagreb, Hrvatska
e-mail: daniel.hofman@fer.hr