Coll. Antropol. **35** (2011) 2: 487–498
Original scientific paper

3D Surface Analysis and Classification in Neuroimaging Segmentation

Martin Žagar, Hrvoje Mlinarić and Josip Knezović

University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering, Zagreb, Croatia

ABSTRACT

This work emphasizes new algorithms for 3D edge and corner detection used in surface extraction and new concept of image segmentation in neuroimaging based on multidimensional shape analysis and classification. We propose using of NifTI standard for describing input data which enables interoperability and enhancement of existing computing tools used widely in neuroimaging research. In methods section we present our newly developed algorithm for 3D edge and corner detection, together with the algorithm for estimating local 3D shape. Surface of estimated shape is analyzed and segmented according to kernel shapes.

Key words: *NIfTI, shape estimation, Canny detector, surface extraction, neuroimage segmentation*

Introduction

Computing systems are key at all stages of neuroimaging, allowing scientists to control highly sophisticated imaging instruments and analyze vast amounts of complex data those instruments generate. Computing tools integrate an implementation of imaging instruments that capture signals from the brain, guide the behavioral tasks used to probe particular brain systems, reconstruct resulting signals into a three-dimensional representation of the brain, correct and suppress noise, statistically analyze the data, and visualize the results. The collected volumetric data can be segmented on many regions depending on our interest, and they can be stored either in a data warehouse or a database where it is possible to query, compare or update it easily. A variety of neuroimaging technologies allow the structure and function of the intact human brain to be studied with minimal invasion, presenting a tremendous opportunity to better understand it, both in healthy and disordered states. Neuroimaging provides a crucial perspective for basic and clinical human neuroscience.

In functional magnetic resonance imaging (fMRI), many existing tools have been developed piecemeal, by scientists who are interested in answering particular neuroscience questions rather than in producing software products that are optimized for the needs of a broader research community. It is, therefore, not surprising that many of these computing tools are not as robust, universally useful, or easy to use as they could be. And, while a handful of neuroimaging computing tools are suitable for general use, many important tools are not widely available. Furthermore, even those that are in general use make varying assumptions, use different algorithms, or implement similar algorithms in different ways[1]. The rest of paper is organized as follows. Benefits of the NIfTI standard are shortly described at the begging of the Materials and Methods chapter. Following are described edge and corners detection as a base for shape estimation and surface extraction, along with application of multispectral segmentation techniques in neuroimage segmentation. Chapter with results gives examples of segmentation and classification of neuroimages.

Materials and Methods

The NIfTI standard

The NIfTI (Neuroimaging Informatics Technology Initiative) standard aims to design an environment to facilitate convergence on common solutions to widespread problems and maximize the usability of neuroimaging computing tools and provide a common resource for neuroimaging research.

Received for publication February 1, 2009

An example of one typical whole-brain fMRI experiment's time series data dimensionality:

dim[0] = 4	xyzt_units = NIFTI_UNITS_ MM|NIFTI_UNITS_SEC
dim[1] = 63.75	pixdim[1] = 3.75
dim[2] = 63.75	pixdim[2] = 3.75
dim[3] = 20	pixdim[3] = 5.0
dim[4] = 120	pixdim[4] = 2.0

In this example the data has 4 dimensions. The basic dimensions (*x, y, z* axes) are 63.75 mm, 63.75 mm and 20 mm respectively. The voxel dimensions (basic unit by each axis) are 3.75 mm, 3.75 and 5 mm respectively. This means that there are 17*17*4 voxels in total. The time dimension has a basic unit of 2 seconds and the length is 120 seconds. In general, this 4D data is described by 1156 voxels and has 60 frames.

There are 3 different methods by which continuous coordinates can be attached to voxels. The aim of the first is to represent nominal voxel locations as reported by the scanner, or as rotated to some known orientation and location (i.e. displaying data on its original grid). The other two methods display data without any particular spatial orientation or use a standard grid. The continuous coordinates of 3D volumes are referred to as (x, y, z). Voxel (i, j, k) is stored starting at the (i + j*dim[1] + k*dim[1]*dim[2]) *(bitpix/8) location into the dataset array. The (x, y, z) coordinates refer to the center of a voxel and the (x, y, z) axes refer to a subject-based coordinate system, with +x = Right; +y = Anterior; +z = Superior.

The (x, y, z) coordinates are given by the pixdim[] scales, a rotation matrix, and a shift. This is intended to represent scanner-anatomical coordinates, which are often embedded in the data header, and represent the nominal orientation and location of the data. This method can also be used to represent the »aligned« coordinates, which would typically result from some post-acquisition alignment of the volume to a standard orientation. In such a case, the program will need to convert its desired (x, y, z) values into (i, j, k) values in order to extract (or interpolate) the image data. This operation is done by using a qfac factor which is either –1 or 1. The formula for (x, y, z) in terms of header parameters and (i, j, k) is[2]:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} \text{pixdim[1]* i} \\ \text{pixdim[2]* j} \\ \text{gfac* pixdim[3]* i} \end{bmatrix} + \begin{bmatrix} \text{goffset\_ x} \\ \text{goffset\_ y} \\ \text{goffset\_ z} \end{bmatrix} \quad (1)$$

The qoffset_* shifts are in the NIfTI header. The center of the (i, j, k) = (0, 0, 0) voxel (first value in the dataset array) is (x, y, z) = (qoffset_x, qoffset_y, qoffset_z). The rotation matrix R is calculated from the quatern_* parameters.

MRI-specific spatial and temporal information

A few fields of NIfTI data format are provided to store some extra information that is sometimes important when storing the image data from an fMRI time series example. When processing such data into statistical images, the following fields are used:

{freq_dim}
{phase_dim}
{slice_dim}.

These fields encode which spatial dimension (1, 2, or 3) corresponds to which acquisition dimension for MRI data. For example, spiral scan multi-slice EPI freq_dim= phase_dim=0; slice_dim=3; since the concepts of frequency- and phase-encoding directions do not apply to spiral scan. If the value of slice_duration is positive, and if slice_dim is nonzero, it indicates the amount of time used to acquire one slice.

As a conclusion, NIfTI format can store data with different meanings. Imaging data, statistical values and other data (vectors, matrixes, labels) can be saved in.nii or.hdr/.img files. Once a data object is chosen, the use of NIfTI format is unambiguous, since the use of particular required fields for a certain aim is predetermined. In this work, NIfTI format will be used for 3D brain images.

Shape estimation

In segmentation of an image, it is important to define the image features which will be the basis for segmentation. The term image feature refers to two possible entities: a global property of an image or part thereof (e.g. the average grey level, an area in a voxel global feature); or a part of the image with some special properties (e.g. a circle, a line, or a textured region in an intensity image, a planar surface in a range image local feature).

The sequence of operations for most computer vision systems begins with the detection and location of some features in the input images that are local, that are meaningful, and that represent detectable parts of the image[3]. Meaningful means that the features are associated to interesting scene elements via the image formation process. Typical examples of meaningful features are sharp intensity variations created by the contours of the objects in the scene, or image regions with uniform grey levels, for instance images of planar surfaces[4]. Sometimes the image features that are looked for are not observably associated to any part or property of the scene, but reflect particular arrangements of image values with desirable properties. Detectable means that there must exist some algorithm that could detect a particular feature, otherwise this feature is not usable. Edges and corners are most easily detectable features.

Edge detection

Edge points or simply edges, are voxels at or around which the image values undergo a sharp variation. The main problem in edge detection is locating the edges in a given image corrupted by acquisition noise, and which are most likely to be generated by scene elements. Main sharp variations correspond not only to significant contours, but also to image noise results in spurious edges. These edges should be suppressed by the edge detection algorithm[5].

There are various reasons for taking interest in edges. The contours of potentially interesting scene elements such as solid objects, marks on surfaces, and shadows that are often classified as suspicious tissues, all generate intensity edges. Moreover, image contours which are often the basic elements for calibration, motion analysis and recognition, are detected from chains of edge points. Edge detection is typically a three-step process[6]:

1) Noise Smoothing. Suppressing as much of the image noise as possible, without destroying the true edges. In the absence of specific information, assume the noise is white and Gaussian.
2) Edge Enhancement. Designing a filter responding to edges; that is, the filter's output is large at edge voxels and low elsewhere, so that edges can be located as the local maxima in the filter's output.
3) Edge Localization. Deciding which local maxima in the filer's output are edges and which are just caused by noise. This involves thinning wide edges to 1-voxel width (non-maximum suppression) and establishing the minimum value to declare a local maximum an edge (thresholding).

The Canny edge detector[7] is probably the most widely used edge detection algorithm in today's multimedia community. Constructing a Canny detector requires the formulation of a mathematical model of edges and noise and quantitative performance criteria, as well as formalize the desirable properties of the detector (e.g., good immunity to noise) and synthesizing the best filter once models and performance criteria are defined. Edges of intensity images can be modelled according to their intensity profiles. For most practical purposes, a few models are sufficient to cover all interesting edges.

The edge detection operator returns a value for the first derivative in horizontal and vertical direction. The depth dimension (the third dimension) can be achieved by modifying a regular Canny 2D detector[7]. The edge gradient of a 3D image object G can be determined by computing the gradient components, $G_x = \frac{\partial G}{\partial x}$, $G_y = \frac{\partial G}{\partial y}$ and $G_z = \frac{\partial G}{\partial z}$ for each voxel $v(x, y, z)$. The second step is to estimate the edge strength with $e_d(x, y, z) = \sqrt{G_x^2 + G_y^2 + G_z^2}$ as well as the orientation of the edge normal.

As shown in Figure 1, the orientation of edge normal is specified by angles θ and ϕ that can be computed by equations

$$\tan\theta = \frac{G_y}{G_x} \tag{2}$$

and

$$\tan\phi = \frac{G_z}{\sqrt{(G_x)^2 + (G_y)^2}} \tag{3}$$

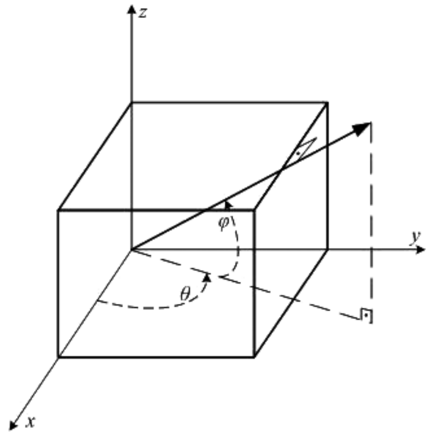

Fig. 1. Orientation representation.

Corners detection

Corners are easier to define in mathematical terms than edges[8]. Corners do not necessarily correspond to any geometric entity of the observed scene. Consider the spatial image gradients $[G_x, G_y, G_z]^T$, and a generic voxel point v, a neighbourhood Q of v, and a matrix C, defined as

$$C = \begin{bmatrix} \Sigma G_x^2 & \Sigma G_x G_y & \Sigma G_x G_z \\ \Sigma G_x G_y & \Sigma G_y^2 & \Sigma G_x G_z \\ \Sigma G_x G_z & \Sigma G_y G_z & \Sigma G_z^2 \end{bmatrix} \tag{4}$$

where the sums are taken over the neighbourhood Q. This matrix characterizes the structure values. The solution is building the eigenvalues of C and their geometric interpretation[9]. Matrix C is symmetric, and can therefore be diagonalized as:

$$C = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{5}$$

The geometric interpretation of eigenvalues λ_1, λ_2 and λ_3 can be explained using a few examples. First, consider a perfectly uniform Q: the image gradient vanishes everywhere, C becomes the null matrix, and eigenvalues are $\lambda_1 = \lambda_2 = \lambda_3 = 0$. Now assume that Q contains the corner of a black cube against a white background: as there are three principal directions in Q, it is expected that $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ and the larger the eigenvalues, the stronger (higher contrast) their corresponding image lines. It is clear that the eigenvectors describe the edge directions, and the eigenvalues the edge strength. A corner is identified by three strong edges, and as $\lambda_1 \geq \lambda_2 \geq \lambda_3$ it is a location where the smallest eigenvalue, λ_3 is large enough. In general terms, at corner voxels the intensity surface has three well-defined, distinctive directions, associated to eigenvalues of C, all of them significantly larger than zero[10].

The procedure for locating the corners is as follows[11]. The input is formed by an image G and two parameters:

the threshold τ on λ_3, and the linear size of a cube window (neighbourhood), for example 2N + 1 voxels. First the image gradient is computed over the entire image *G*, and then for each voxel *v*:

- form the matrix *C* over a $(2N + 1) \times (2N + 1) \times (2N + 1)$ neighbourhood *Q* of *v*;
- compute λ_3, the smallest eigenvalue of *C*;
- if $\lambda_3 > \tau$ save the coordinates of *v* into a list *L*.

The list *L* is then sorted in decreasing order of λ_3. The sorted list is scanned top to bottom: for each current point *v*, all points which belong to the neighbourhood of *v* and appear further on in the list are deleted. The output is a list of feature points for which $\lambda_3 > \tau$ and whose neighbourhoods do not overlap. In the case of corner voxels, the value of *N* is linked to the location of the corner within the neighbourhood[12].

Surface extraction from range images

Once edges and corners are detected, local shape should be estimated. 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 in 3D telemedicine[13].

This section presents a method of finding patches of various shapes which compose the visible surface of an object adapted to 3D. The method called HK segmentation[10] partitions a range image into regions of homogeneous shape, called homogeneous surface patches. For a given range image *G*, the goal is to compute a new image 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.

Shape classes

To estimate 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 the Figure 2 and Table 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.

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

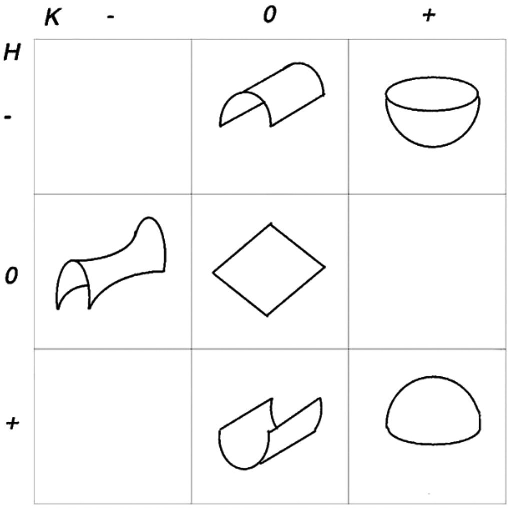

Fig. 2. Shape classes for HK segmentation.

Estimating the local shape

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.

The expressions of *H* and *K* are evaluated at each image point[12], with signs from Table 1. The Gaussian curvature operator *K* for 3D images can be computed as:

$$K(x,y,z) = \frac{(\nabla G_1^{\perp})^T S_{G1} \nabla G_1^{\perp T} + (\nabla G_2^{\perp})^T S_{G2} \nabla G_2^{\perp T} + (\nabla G_3^{\perp})^T S_{G3} \nabla G_3^{\perp T}}{\left|\nabla G\right|^2} \tag{6}$$

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

$$G_1^{\perp} = \begin{bmatrix} -G_y \\ G_x \end{bmatrix}, G_2^{\perp} = \begin{bmatrix} -G_z \\ G_x \end{bmatrix}, G_3^{\perp} = \begin{bmatrix} -G_z \\ G_y \end{bmatrix} \tag{7}$$

and

$$S_{G1} = \begin{bmatrix} G_{xx} & G_{xy} \\ G_{yx} & G_{yy} \end{bmatrix}, S_{G2} = \begin{bmatrix} G_{xx} & G_{xz} \\ G_{zx} & G_{zz} \end{bmatrix}, S_{G1} = \begin{bmatrix} G_{yy} & G_{zy} \\ G_{yz} & G_{zz} \end{bmatrix} \tag{8}$$

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

$$H(x,y,z) = \frac{G_x^2(G_{yy} + G_{zz}) + G_y^2(G_{zz} + G_{xx}) + G_z^2(G_{xx} + G_{yy})}{2(G_x^2 + G_y^2 + G_z^2)^{3/2}} - \frac{2(G_x G_y G_z + G_x G_z G_{xz} + G_y G_z G_{yz})}{2(G_x^2 + G_y^2 + G_z^2)^{3/2}} \tag{9}$$

To summarize, the basic HK segmentation algorithm works as follows: the input is a range image, 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 Table 1. The images of 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 image L can be computed by assigning a shape label l_i to each voxel, according to the rules in Table 1. The output is the shape image L.

In order to be used by subsequent tasks, the output of segmentation algorithm can be converted into a list of symbolic patch descriptors. In each descriptor, a surface patch is associated with a number of attributes which may include a unique identifier position of the patch centre, patch area, information on normals and curvature contour representations, and pointers to neigh bourn patches. Closed form surface models (e.g. quadrics) are fitted to the surface patches extracted by the HK segmentation, and only the model's coefficients and type (e.g. cylinder, cone) are stored in the symbolic descriptors.

Segmentation techniques

Since segmentation requires classification of voxels, it is often treated as a pattern-recognition problem and addressed with related techniques[13–15]. In medical imaging, where variability in the data may be high, pattern recognition techniques that provide flexibility and convenient automation are of special interest. One approach is fuzzy clustering[16], a technique based on fuzzy models and membership function. Another approach is neural network where the classification is based on distributed nonlinear parallel processing[17]. Numerous neural network structures and training algorithms are available and can be applied to medical image segmentation[18].

The strategy of edge-based segmentation algorithms is to find object boundaries and segment regions enclosed by the boundaries[19]. These algorithms usually operate on edge magnitude and phase images produced by an edge operator suited to the expected characteristics of an image. Traditionally, most image segmentation techniques use one type of images, but the performance of these techniques can be improved by combining techniques from several sources (multispectral sources) or by dynamic programming when images are integrated over time[20].

In this work, newly developed segmentation techniques will be described. They are based on semiautomatic, expert-trained, neural network techniques with combination of voxel and region-based multispectral map techniques.

The most intuitive approach for segmentation is global thresholding, when only one threshold based on the image histogram is selected for the entire image. If the threshold depends on local properties of some image regions, it is called local. If local thresholds are selected independently for each voxel (or groups of voxels), thresholding is called dynamic and adaptive.

For images that have biomodal histogram (i.e. grey levels grouped into two dominant sets, object and background), the object can be extracted from the background by a simple operation that compares image values with the threshold value T. Suppose an image $f(x,y,z)$ with a histogram shown on the Figure 3. The threshold image $g(x,y,z)$ is defined as

$$g(x,y,z)=\begin{cases}1, f(x,y,z)>T\\0, f(x,y,z)\leq T\end{cases} \qquad (10)$$

The result of thresholding is a binary image, where voxels with threshold value 1 correspond to objects, while voxels with value 0 correspond to the background. There are a number of selection methods for threshold T based on classification model that minimizes the probability of an error. With the semi-automated version, ventricular volumes from 3D magnetic resonance images can be measured. In this method, an expert (operator) selects two voxels – one inside an object and one from the background. By comparing the distribution of voxel intensities in the circular regions around the selected voxels, the threshold is calculated automatically. It corresponds to the least number of misclassified voxels between two distributions. The result of the thresholding operation is displayed as a contour map and superimposed on the original image. If needed, the operator can manually modify any part of the border. The same technique can also be applied on other input data sets[21,22].

In many applications good segmentation is obtained when the area or perimeter of the objects is minimally sensitive to small variations of the selected threshold level.

If an image contains more than two types of regions, it may still be possible to segment it by applying several individual thresholds. Global thresholding is computationally simple and fast and it works well on images that contain objects with uniform intensity values on a contrasting background. However, it fails if there is a low contrast between the object and the background, if the image is noisy, or if the background intensity varies significantly across the image and is not useful in medical applications[23].

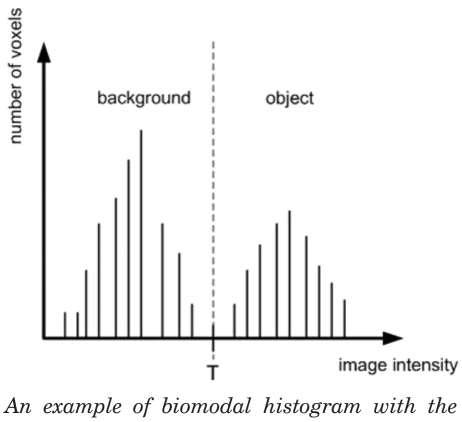

Fig. 3. An example of biomodal histogram with the selected threshold T.

In this case, one solution is to use adaptive thresholding. Adaptive thresholding means locally-oriented, determined by splitting an image into subimages and by calculating thresholds for each subimage, or by examining the image intensities in the neighbourhood of each voxel. The splitting method divides an image into rectangular overlapping subimages and the histograms are calculated for each subimage[24]. The subimages used should be large enough to include both object and background voxels. If a subimage has a biomodal histogram, then the minimum between the histogram peeks should determine a local threshold. If a histogram is unimodal, the threshold can be assigned by interpolation from the local thresholds found for nearby images. In the final step, a second interpolation is necessary to find the correct thresholds for each voxel. Although local thresholding is computationally more expensive, it is generally very useful for segmenting objects from a varying background, as well as for extraction of regions that are very small and sparse.

Image segmentation for medical data

Segmentation can be defined as the identification of meaningful image components[15]. It is a fundamental task in image processing providing the basis for any kind of further high-level image analysis. In medical image processing, a wide range of applications is based on segmentation: volumetric analysis with respect to normal or pathological organ development, temporal monitoring of size and growth in pathological processes, or providing a basis for the applicability of automatic image fusion algorithms when combining the complementary information obtained by different image acquisition modalities.

Still, the simplest way to obtain good segmentation results is segmentation by human (expert knowledge). This yields excellent results, which is due to the fact that human operators do not only use the presented image data information, but also make use of additional model-based knowledge such as anatomical skills as well as complex psychological cognitive abilities, e.g., with respect to orientation in space. However, the segmentation of hundreds of MRI or CT slices by manual contour tracing is a very time-consuming task that requires a considerable amount of human intervention[15]. Therefore, it is desirable to perform computer segmentation. However, this is difficult to achieve, as the complex cognitive abilities can hardly be transferred to computer programs. An efficient strategy to cope with this problem is to present additional image data as an input to automatic image segmentation systems, thus compensating for the lack of high-level image analysis capabilities. A possible realization of this principle is the acquisition and processing of multispectral image data sets, which forms the basis of the segmentation approach.

The main goal of the segmentation process is to divide an image into regions (also called subvolumes) that are homogenous with respect to one or more characteristics or features[8]. Segmentation is an important tool in medical image processing and it has been useful in many applications, but this paper will concentrate on classifying voxels into anatomical regions[16]. There is a wide variety of segmentation techniques and processes depending on the input data (brain, blood vessels, muscles or other parts of human body, and for example tumours from magnetic resonance images). That is why there is no one standard segmentation technique that can produce satisfactory results for all image applications. The definition of the goal of segmentation varies according to the goal of the study and type of data. Different assumptions about the nature of the analyzed images lead to the use of different algorithms. These techniques can be classified into two main categories: edge-based techniques that look for edges between regions with different characteristics, and region segmentation techniques that look for the regions satisfying a given homogeneity criteria [10].

Application of multispectral segmentation techniques in neuroimage segmentation

In this section, segmentation based on the analysis of multispectral neuroimages will be described. Here the image object is examined by more than one different image acquisition techniques, e.g. different MRI sequence protocols. Appropriate preprocessing steps comprise the anatomically correct registration of the data sets and masking a region of interest in which the segmentation should be performed[25]. Finally, each voxel can be characterized as a vector in an n-dimensional feature space composed of the grey level obtained from different image acquisition techniques. Segmentation then becomes the problem of classifying these multidimensional feature vectors to specific components among a given set of alternative meaningful image components.

The general concept of multispectral voxel-based brain segmentation can be explained as follows: n different 3D data sets for each brain are obtained by employing different MRI acquisition parameters. Segmentation aims at classifying each voxel of the multispectral data set as belonging to a specific tissue type, thus obtaining information about the structure and volume of tissue classes[10].

A classical problem with numerous clinical applications is the segmentation of brain imaging data with respect to tissue classes: grey matter, white matter and cerebrospinal fluid. Several other structures such as venous blood may be introduced as additional segmentation classes. However, these additional classes comprise only a small part of the total brain volume. Furthermore, for most of the clinical applications, the focus of interest is reduced to grey and white matter structures. Therefore, minor additional classes are assigned to cerebrospinal fluid.

Although such a threefold classification of brain tissue may be sufficient for numerous clinical applications, it should be emphasized that the presented concept can be extended to an arbitrary number of tissue classes. One may especially think of introducing additional classes for identification of pathological tissue, e.g., multiple sclerosis plaques or malignant brain tumour structures[26].

The concept of voxel based multispectral image segmentation requires anatomically correct alignment of the data sets acquired within different image acquisition procedures. This may already be achieved during the acquisition procedure itself by stabilizing the subject's head position and applying constant field of view in different MRI sequences. In general this will be sufficient in order to obtain an acceptable registration of different MRI data sets. Nevertheless, there may be situations in which motion artefacts cannot be avoided[27].

In these cases additional image registration techniques have to be applied. Registration methods can be classified with regard to the level of human interaction required within the procedure:

- Manual interactive registration by a human observer.
- Semi-automatic procedures that require less amount of human interaction. An example are the so-called »mark-and-link« methods where a human observer identifies anatomical marks that serve as reference points for the registration procedure.
- Fully automatic procedures that do not require any human interaction. These methods are frequently applied for the superposition of data sets obtained in different medical imaging modalities such MRI, PET, SPECT, or CT in order to make use of the complementary diagnostic information provided by different modalities.

After correct anatomical registration, an additional preprocessing step can be performed[8]. All extracerebral structures that are not required for tissue classification task should be excluded from the set. In addition to the brain, various other structures can be identified, such as the skull. By defining a mask, these extracerebral structures are removed. Finally, only the structures belonging to the brain remain in the data set. The restriction to the brain structures by excluding all the voxels in the surrounding tissue structures provides import advantages for the subsequent segmentation task. Those advantages are:

- Vector quantization is restricted only to voxels relevant for the segmentation task. The resulting codebook thus represents the grey level distribution of the brain voxels without the contribution of irrelevant extracerebral voxels.
- The grey level range is restricted. Without presegmentation, some codebook vectors would be oriented on tissue classes outside the brain. This, however, would lead to a lower quality representation of tissue classes within the brain. However, this could be compensated by increasing the total number of codebook vectors applied in the vector quantization procedure.
- Voxels inside and outside the brain with a similar grey-level representation do not cause problems for the brain tissue segmentation task. If presegmentation was omitted, such voxels would be attributed to the same codebook vector, i.e., they could not be separated from each other. This could only be achieved by considerably increasing the number of codebook vectors in order to obtain a more fine-grained resolution of the grey-level feature space, which, in turn, would increase the computational expense for vector quantization.

The last item in particular justifies the additional presegmentation effort[28]. In analogy to image registration, there is a wide scope of methods for presegmentation ranging from manual contour tracing to semiautomatic or fully automatic procedures.

The grey levels of each data set are normalized to the unit interval $[0,1]$. $\tilde{G} \in R^{(m_x, m_y, m_z, l, n)}$ represent the grey levels of a presegmented, multispectral data set, consisting of n data sets with l images each of size $m_x \times m_y \times m_z$. thus. $\tilde{G}_{p,r,s,t,u}$ represents the grey level of a voxel, where u denotes the index of a single data set, t the image, i e., slice number, and p,r,s the x,y and z position within the image, respectively[29].

Let

$$\tilde{g}_{\min}(u) = \min_{1 \le m_x' \le m_x, 1 \le m_y' \le m_y, 1 \le m_z' \le m_z \, 1 \le l' \le l} G_{m_x', m_y', m_z', l', u}, \tag{11}$$
$$u \in [1, \ldots, n]$$

denote the minimal grey level of data set u, and

$$\tilde{g}_{\max}(u) = \max_{1 \le m_x' \le m_x, 1 \le m_y' \le m_y, 1 \le m_z' \le m_z \, 1 \le l' \le l} G_{m_x', m_y', m_z', l', u}, \tag{12}$$
$$u \in [1, \ldots, n]$$

the maximal grey level. Then each data set $u \in [1, \ldots, n]$ is rescaled according to

$$G_{p,r,s,t,u} = \frac{\tilde{G} - \tilde{g}_{\min}(u)}{\tilde{g}_{\max}(u) - \tilde{g}_{\min}(u)}, \quad p \in \{1, \ldots, m_x\}, \tag{13}$$
$$r \in \{1, \ldots, m_y\}, s \in \{1, \ldots, m_z\}, t \in \{1, \ldots, l\}.$$

$G \in R^{(m_x, m_y, m_z, l, n)}$ here denotes the rescaled multispectral data set. An alternative approach would be resealing according to the standard deviation of the grey-level distribution in each of the single data sets. This would reduce the effect of outliers on the rescaling procedure.

After performing the preprocessing steps, multispectral data $G \in R^{(m_x, m_y, m_z, l, n)}$ are obtained consisting of n correctly aligned, normalized data sets. This interprets that each voxel of the multispectral 3D data set represents an n-dimensional feature vector x that is determined by the tissue characteristics of this voxel

$$x = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{pmatrix} = \begin{pmatrix} g_{p,r,s,t,1} \\ g_{p,r,s,t,2} \\ \vdots \\ G_{p,r,s,t,n} \end{pmatrix} \tag{14}$$

The data set $X = \{x\}$ is now presented as the input to a vector quantizer. By unsupervised clustering, a set C of codebook vectors w_j with $C = \{ w_j \in R^n \mid j \in \{1, \ldots N\} \}$ is computed that represent the data set X. Here, the number N of codebook vectors is much smaller than the number of feature vectors[30].

Given a set of feature vectors x in a grey-level feature space G, vector quantization can determine a set of prototypical codebook vectors w_j representing the feature space. This provides the basis for segmentation of the imaging data set in respect to different tissue classes[31].

Each codebook vector w_j is assigned to a tissue class $\lambda \in \{1,\dots m\}$ (for example, 1 – grey matter, 2 – white matter, etc.) that is represented by the codebook. For this reason, for each of the N codebook vectors w_j, all the voxels of the 3D data set belonging to this codebook are labelled automatically. Interactive visual inspection of the images of the 3D data set that contain the maximal number of pixels belonging to a specific codebook vector w_j, usually enables a decision on which tissue class λ is represented by this codebook vector[28]. Thus, it is usually sufficient to analyze N images in order to assign each codebook vector to a tissue class. If a clear decision for a codebook vector cannot be made, additional images with highlighted voxels belonging to a specific codebook vector can be viewed in order to perform a proper tissue-class assignment. As a result, each of the m tissue classes λ is represented by a set of codebook vectors w_j^2.

By assigning each voxel to a codebook vector w_j and each codebook vector to a tissue class λ, all the voxels of the 3D data set can be attributed to a tissue class. This is equivalent to the segmentation of the data set with respect to given tissue classes.

Results

The whole process is explained by the block diagram in Figure 4. Original neuroimage in NIfTI format, which is shown in Figure 5, is segmented according to kernel shapes and based on edge and corner detection. Detection of edges and corners can be achieved by using adaptive thresholding and adopted 3D Canny edge detector. The pseudocode of this procedure is as follows:

```
% 3D Canny edge detection

% compute gradients
drx = gradient component x
dry = gradient component y
drz = gradient component z

% compute orinetation
psi = arctan(drx/dry);
fi = arctan(drz/(sqrt(drx^2+dry^2)));

% compute threshold
drm = make_NIfTI(nonmaximumsuppression(double(drx), double(dry), double(drz)),'sfloat');
tlevel = trianglethreshold(drm);

% if grey level of voxel is higher than threshold, it is in object
drm =
bpropagation(drm>tlevel,drm>(lowerfrac*tlevel),0,2,0);
```

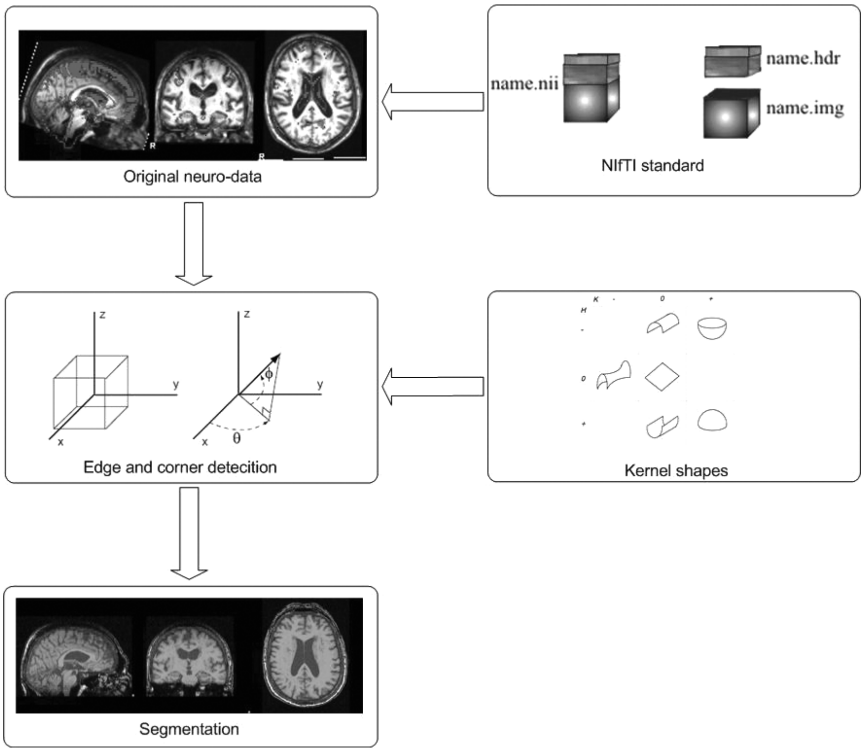

Fig. 4. Schema of complete neuroimaging segmentation process.

```
% compute trianglethreshold(in)
% A smooth histogram
define precision
min_val = min(in);
max_val = max(in);
interval = (max_val-min_val)/ precision;
max_val = max_val+border*interval;
min_val = min_val-border*interval;
[histogram,bins] = diphist(in,[min_val,max_val], precision);
histogram = gaussf(histogram,4);
histogram = double(histogram);

% Find peak
[max_value,max_element] = max(histogram);

% Define: start, peak, stop positions in histogram
sz = length(histogram);
left_bin = [border,histogram(border)];
right_bin = [sz-border,histogram(sz-border)];
top_bin = [max_element,max_value];
```

```
% Find the location of the maximum distance to the
triangle
v1 = top_bin – left_bin;
if any(v1~=0)
v1 = v1/norm(v1);
end
ii = left_bin(1):top_bin(1);
v2 = [ii'-left_bin(1),histogram(ii)'-left_bin(2)];
distance = abs(v1(1).*v2(:,2)-v1(2).*v2(:,1));
v1 = top_bin – right_bin;
if any(v1~=0)
v1 = v1/norm(v1);
end

jj = top_bin(1):right_bin(1);
v2 = [jj'-right_bin(1),histogram(jj)'-right_bin(2)];
distance = [distance;abs(v1(1).*v2(:,2)-v1(2).*v2(:,1))];
[max_distance,bin] = max(distance);
ii = [ii,jj];
```

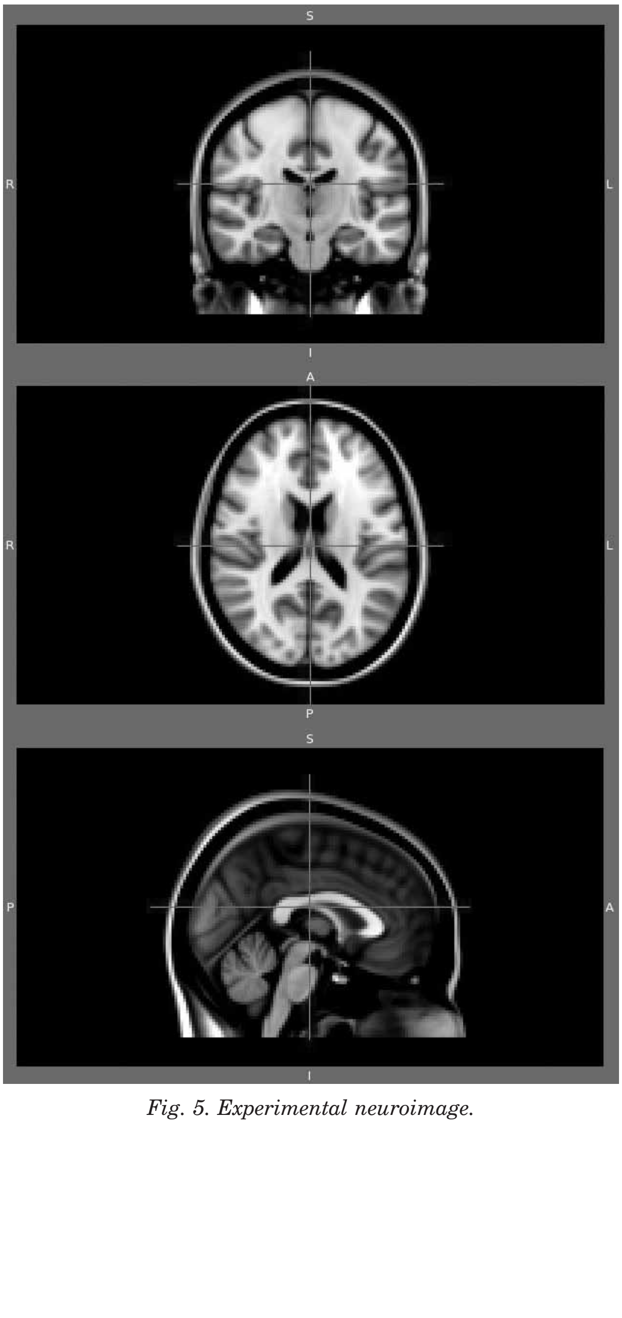

Fig. 5. Experimental neuroimage.

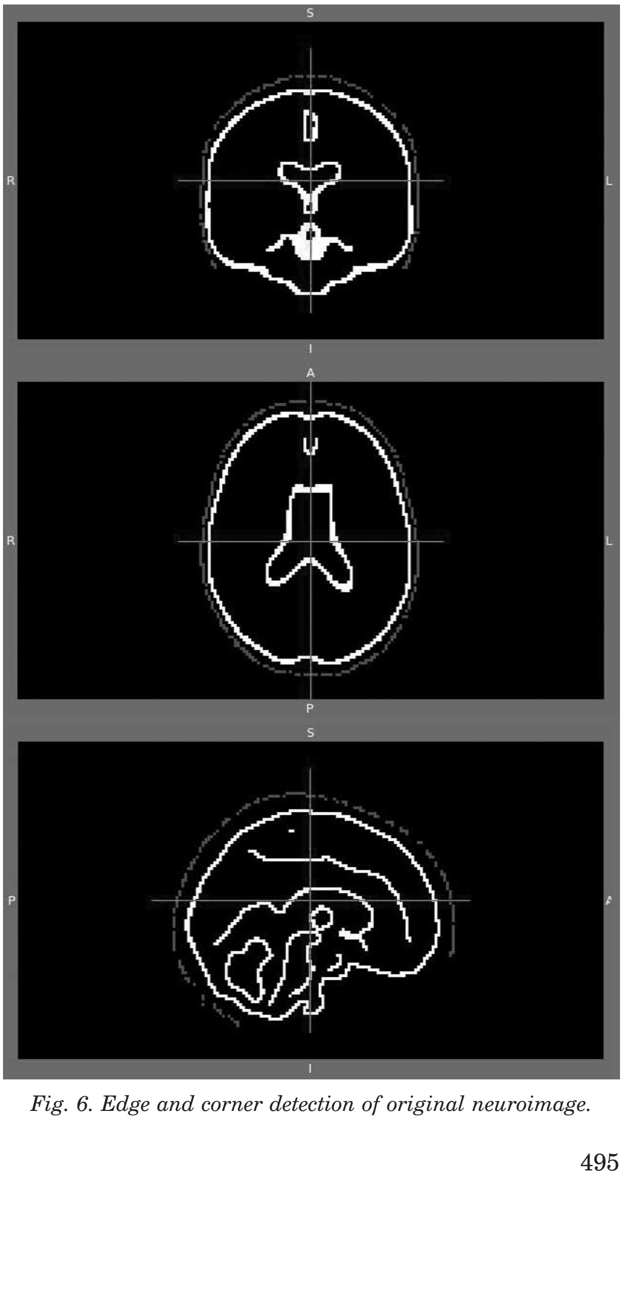

Fig. 6. Edge and corner detection of original neuroimage.

```
bin = ii(bin);
value = bins(bin);
```

Results of edge and corner detection are shown in Figure 6.

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 that can be, for example, used in[32]. Estimated parameter distributions are used to attempt to approach the problem of assessing anatomical connectivity from a statistical angle. According to this, brain can be segmented into three main tissue types:

- grey matter
- white matter
- cerebro-spinal fluid.

An example of segmentation and brain masking is shown in Figure 7.

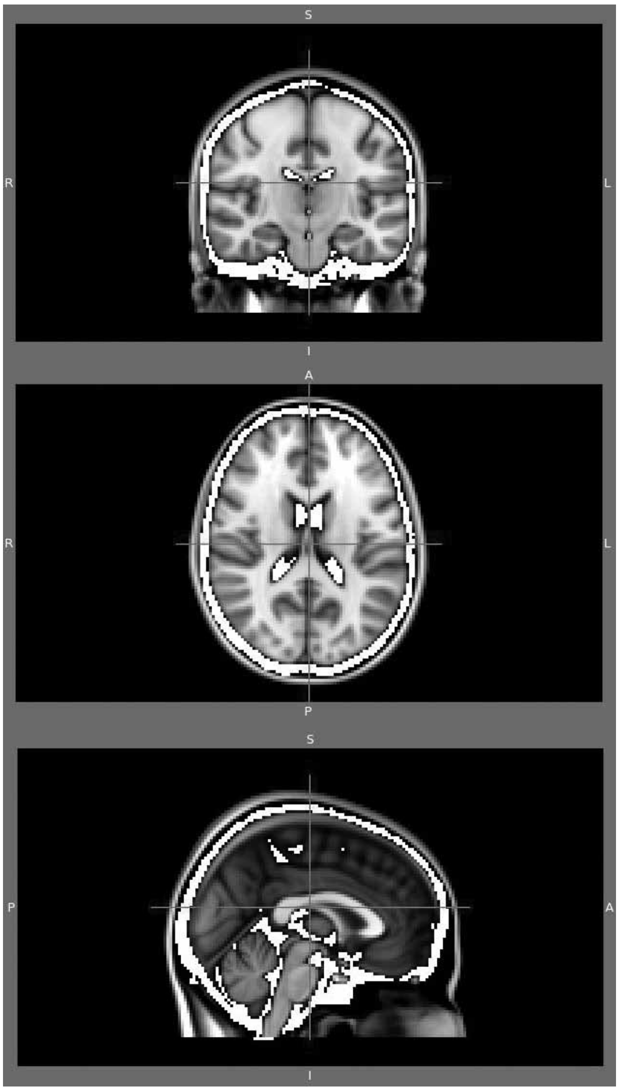

Fig. 7. Example of segmentation and brain masking.

Object registration is based on the anatomical location (voxel location) with accurate intensity values. Registration principles can be divided into spatial transformations (finding and applying transformation), interpolation (speed, accuracy and stability), similarity functions, search and optimisation, and multi-resolution. With coding only segmented region of interest (brain dataset) and not the whole head dataset, size of dataset is reduced from 450kB to 350 kB. The reduction can be even higher if it is segmented smaller region of interest. The pseudocode of segmentation is as follows:

```
% Segmentation

% apply some thresholding rules to ignore certain parts of data
D(D<=40) = 0; %ignore low levels (CSF & air)
D(D>=100) = 0; %ignore high levels (skull & other hard tissues)
D(:,:,1:60) = 0; %ignore spatially low positions (below brain mass)

% isolate brain mass
lev = greythresh(double(im)/max_level) * max_level;
bw = (D>=lev);
L = bwlabeln(bw);

% connected region properties
stats = regionprops(L,'Area')
A = [stats.Area];
biggest = find(A==max(A))

% remove smaller scraps
D(L~=biggest) = 0;
update_sliceomatic(double(D),hSlico2)

% grow back main region (brian mass) – nobkpt
D = imdilate(D,blk);
update_sliceomatic(double(D),hSlico2)

% separate white vs. grey matter
im = imrotate(squeeze(D(30,:,:)),90);
figure(hSlico2)
lev2 = thresh_tool(im,'grey')

% partition brain mass (nobkpt)
L = zeros(size(D)); %0=outside brain (head/air)
L(D<lev2 & D>0) = 2; %2=grey matter
L(D>=lev2) = 3; %3=white matter

% Volumetric Measurements (voxel counting)

%total volume of brain (liters)
brain_voxels = length(find(L(:)>1));
brain_volume = brain_voxels*prod(voxel_size)/1e6

%volume of grey matter (liters) – nobkpt
grey_voxels = length(find(L(:)==2));
grey_volume = grey_voxels*prod(voxel_size)/1e6

%volume of white matter (liters) – nobkpt
white_voxels = length(find(L(:)==3));
white_volume = white_voxels*prod(voxel_size)/1e6

%density calculations (volume ratios) – nobkpt
grey_fraction = grey_volume/brain_volume
white_fraction = white_volume/brain_volume
```

```
% Separate head from background for visualization


% exterior of head (connected inside through ears)
BW = (D1<lev1);

% how many connected regions?
L = bwlabeln(BW);
stats = regionprops(L,'Area')
% which one biggest?
A = [stats.Area];
biggest = find(A==max(A))
BW(L~=biggest) = 0;

% grow layer back
BW = imdilate(BW,blk);

% label head voxels
L = L1;
L(BW<1 & L1<2) = 1; %1=head (0=air, 2=grey,
3=white)
```

Result of segmented neuroimage is shown in Figure 8. From original neuroimage shown in Figure 5 with proposed 3D edge and corner detection and surface analysis we have removed unnecessary data and brain remains as a region of high interest.

Discussion and Conclusion

This work introduces the usage of manual (expert knowledge) and semi-automated segmentation techniques, based on the modified Canny edge and corner detection as well as adaptive thresholding, since segmentation requires the classification of voxels. The strategy of edge-based segmentation is to find object boundaries and segment regions enclosed by the boundaries. By assigning each voxel to a codebook vector and each codebook vector to a tissue class, all the voxels of the 3D data set can be attributed to a tissue class and local shapes can be estimated. Statistical analysis is used for examination of each MRI dataset when trying to match it with a kernel shape. Post-stats processing analysis segmentation is based on adaptive thresholding. Adaptive thresholding is locally oriented, splitting an image into subimages, and calculating thresholds for each subvolume or examining the image intensities in the neighbourhood of each voxel. The splitting method divides an image into rectangular overlapping subvolumes and the histograms are calculated for each subvolume. To detect the subvolumes we propose newly developed 3D edge and corner detector, surface analysis and classification of estimated local shape according to kernel shapes, with goal to obtain fully neuroimaging segmentation.

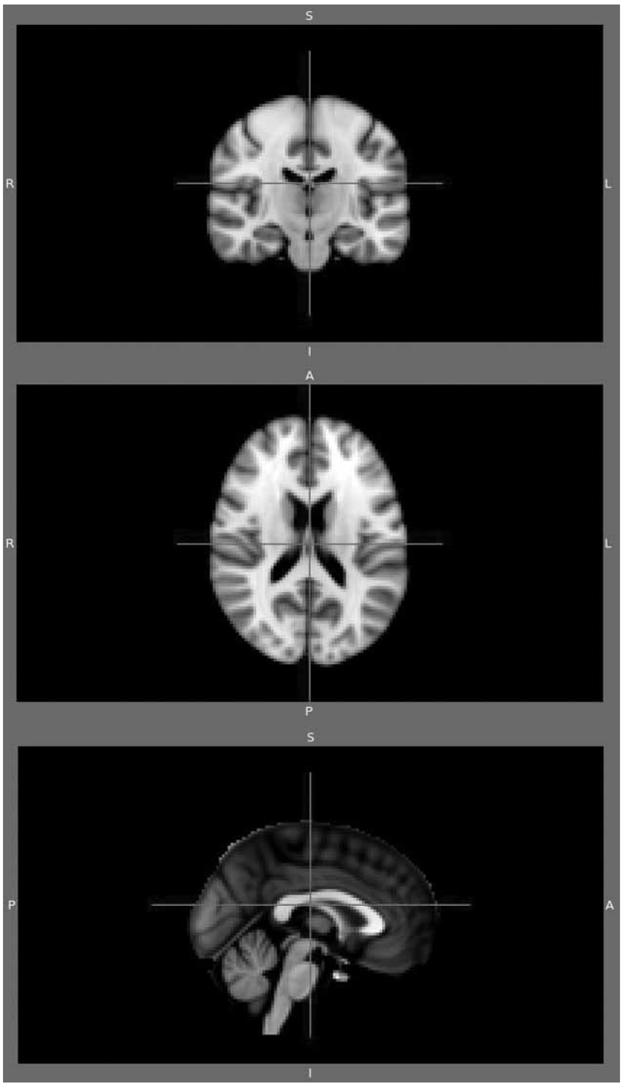

Fig. 8. Result of proposed methods for neuroimage segmentation.

REFERENCES

1. NIfTI-1 Data Format: Source Code, Documentation, Example Files and further information, accessed 08.02.2011. Available from: URL: http://nifti.nimh.nih.gov/nifti-1/. — 2. The NIfTI libraries, accessed 01.04.2011. Available from: URL: http://niftilib.sourceforge.net/. — 3. ZHENHUA W, ZHIJUN Z, An adaptive statistical features modeling tracking algorithm based on locally statistical ROI. In: Proceedings (International Conference on Educational and Information Technology (ICEIT), 2010). — 4. KNEZOVIĆ J, KOVAČ M, ŽAGAR M Lossless Predictive Algorithm for Medical Image Compression. In: Proceedings (Medical Image Understanding and Analysis, British Machine Vision Association, Aberyswyth, UK, 2007). — 5. NASCIMENTO JC, MARQUES JS, Improved Gradient Vector Flow for robust shape estimation in medical imaging. In: (Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2010). — 6. KUO CJ, LIN TG, HUANG RS, ODEH SF, IEEE T, Multimedia, Vol. 5, 1 (2003) 8. — 7. VASAVI KP, LATHA MM, KUMAR NU, A Deterministic Edge Detection Using Statistical Approach. In: Proceedings (International Conference on Computational Intelligence and Multimedia Applications, 2007). — 8. SARRIS N, STRINTZIS MG, 3D Modeling and Animation: Synthesis and Analysis Techniques for the Human Body (IRM Press, Hershey, USA, 2005). — 9. BREJL M, SONKA M, Directional 3D Edge Detection in Anisotropic Data: Detector Design and Performance Assessment. In: Proceedings (Special Issue of Computer Vision and Image Understanding on Analysis of Volumetric Images, University of Iowa, 2006). — 10. CORROCHANO EB, Handbook of Geometric Computing Applications in Pattern Recognition, Computer Vision, Neuralcomputing, and Robotics (Springer, Berlin, Germany, 2005). — 11.

JAKAS JL, AL-OBAIDI A, LIU AY, A comparative study of different corner detection methods, In: Proceedings (IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), 2009). — 12. KNEZOVIĆ J, KOVAČ M, KLAPAN I, MLINARIĆ H, VRANJEŠ Ž, LUKINOVIĆ J, RAKIĆ M, Coll Antropol, 31 (2007) 1143. — 13. KLAPAN I, ŠIMIČIĆ Lj, RIŠAVI R, BEŠENSKI N, PASARIĆ K, GORTAN D, PAVIĆ D VRANJEŠ Ž, Tele-3D-computer assisted functional endoscopic sinus surgery: a new dimension in surgery of the nose and paranasal sinuses. In: KLAPAN I, ČIKEŠ I (Eds) Telemedicine (Telemedicine Association, Zagreb, 2005). — 14. WU X, ZHAI G, YANG X, ZHANG W, IEEE T Image Process, Vol 20, 1 (2011) 36. — 15. BANKMAN IN, Handbook of Medical Imaging Processing and Analysis (Academic Press, San Diego, USA, 2000). — 16. ŽAGAR M, KNEZOVIĆ J, MLINARIĆ, 4D Data Compression Methods for Modeling Virtual Medical Reality. In: Proceedings (18th International Conference on Information and Intelligent Systems, Varaždin, Croatia, 2007). — 17. PRIVITERA CM, STARK LW, IEEE T Pattern Anal, Vol. 22, 9 (2000) 970. — 18. ZHOU J, QI J, IEEE T Med Imaging, Vol 30, 1 (2011) 119. — 19. GEVERS T, IEEE T Multimedia, Vol. 4, 4 (2002) 509. — 20. KLAPAN I, VRANJEŠ Ž, PRGOMET D, LUKINOVIĆ J, Coll Antropol, 32 (2008) 217. — 21. LEE HY, LEE HK, HA YH, IEEE T Multimedia, Vol. 5, 3 (2003) 358. — 22. ROY-CHOWDHURY AK, CHELLAPPA R, KEATON T, IEEE T Multimedia, Vol. 6, 3 (2004) 423. — 23. KLAPAN I, ŠIMIČIĆ Lj, PASARIĆ K, VLAHUŠIĆ A, RIŠAVI R, SRUK V, SCHWARZ D, BARIŠIĆ J, J Telemed Telecare, 8 (2002) 125. — 24. LUO H, ELEFTHERIADIS, IEEE T Multimedia, Vol. 5, 3 (2003) 379. — 25. IZQUIERDO E, GHANBARI M, IEEE T Multimedia, Vol. 4, 1 (2002) 97. — 26. SHIFFMAN S, NG YR, BROSNAN TJ, ELIEZ S, LINKS JM, KELKAR UV, REISS AL, NeuroImage, 20 (2003) 1811. — 27. ŽAGAR M, KOVAČ M, BOSNIĆ I, Lossless and Lossy Compression in 4D Bio-modeling. In: Proceedings (First International Conference in Information and Communication Technology & Accessibility: New Trends in ICT & Accessibility, Tunis, Tunis, 2007). — 28. SANCHEZ V, NASIOPOULOS P, ABUGHARBIEH R, IEEE Acoust Speech, 2, (2006) 1116. —29. NGO CV, PONG TC, ZHANG HJ, IEEE T Multimedia, Vol. 4, 4 (2002) 446. — 30. CEN S, COSMAN PC, IEEE T Multimedia, Vol. 5, 1 (2003) 1. — 31. BAO P, GOURLEY D, IEEE T Multimedia, Vol. 6, 6 (2004) 786. — 32. RADELJAK S, ŽARKOVIĆ-PALIJAN T, KOVAČEVIĆ D, MARINOVIĆ D, DADIĆ HERO E, Coll Antropol, 34 Suppl. 2 (2010) 287.

M. Žagar

University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, 10000 Zagreb, Croatia
e-mail: martin.zagar@fer.hr

ANALIZA I KLASIFIKACIJA OBJEKATA S PRIMJENOM NA 3D NEUROLOŠKE PODATKE

SAŽETAK

Računalni alati integriraju implementaciju instrumenata za prikupljanje slika i moždanih signala, i upravljanje poznatim metodama za rekonstrukciju rezultantnih signala u trodimenzionalnom prikazu mozga, kao i korekciju šuma, statističku analizu podataka i vizualizaciju rješenja. Prikupljeni volumetrijski podaci mogu se segmentirati na više različitih područja, u ovisnosti o interesu prema određenim područjima, te se spremaju u skladišta podataka ili baze podataka gdje se mogu procesirati u svrhu upita, usporedbe ili lakšeg osvježavanja. U ovom radu su razvijeni novi algoritmi za trodimenzionalnu detekciju rubova i kutova koji se koriste u prepoznavanju površina, kao i novi koncept segmentacije neuroloških podataka koja se temelji na multidimenzionalnoj analizi i klasifikaciji oblika. Korištenjem NifTI standarda za opis ulaznih podataka, omogućena je interoperabilnost i proširuje se područje primjene računalnih alata u neurološkim istraživanjima. Prezentirani su novorazvijeni algoritmi za detekciju 3D rubova i uglova kao i algoritmi za procjenu 3D oblika. Površine segmentiranih oblika analizirane su usporedbom s poznatim jezgrenim oblicima.