

A queue-based region growing algorithm for accurate segmentation of multi-dimensional digital images

Jean-Philippe Thiran*, Vincent Warscotte, Benoît Macq

*Université Catholique de Louvain, Laboratoire de Télécommunications et Télédétection, Place du Levant, 2,
B-1348 Louvain-la-Neuve, Belgium*

Received 14 September 1995; revised 2 May 1996 and 13 December 1996

Abstract

An algorithm for automatic and accurate segmentation of multi-dimensional images is presented in this paper. It improves the classical watershed transform whose results are inaccurate when applied on noisy or anisotropic data. This algorithm combines a watershed-like region growing with a very simple marker selection step. It is particularly well suited for accurate segmentation of complex objects, such as the brain in 3D Magnetic Resonance (MR) images of the head since it provides an accurate and fully 3D segmentation in a reasonable computation time. Comparative results of the segmentation obtained by this algorithm and by the classical watershed transform are shown in the case of 3D MR images. Applications of this technique to 3D visualisation and brain sulci identification are also presented. © 1997 Elsevier Science B.V.

Zusammenfassung

Ein Algorithmus zur automatischen und genauen Aufteilung multi-dimensionaler Bilder wird in diesem Papier vorgestellt. Er verbessert die klassische Wasserscheide (watershed)-Transformation, deren Resultate bei Anwendung auf verrauschte oder anisotrope Daten ungenau werden. Dieser Algorithmus kombiniert das Wachstum der Region basierend auf dem watershed-Prinzip mit einem sehr einfachen Merkmals-Selektionsschritt. Er ist besonders gut zur genauen Segmentierung komplexer Objekte geeignet, wie das Gehirn in 3D-Magnetik-Resonanz (MR) Bildern des Kopfes und erlaubt eine genaue und vollständige 3D-Segmentierung in angemessener Rechenzeit. Vergleichende Resultate der Segmentierung mit Hilfe dieser Methode und der klassischen watershed Transformation werden für 3D-MR Bilder vorgestellt. Anwendungen dieser Technik im Bereich 3D-Visualisierung und Gehirn-(sulci)-Identifikation werden ebenso präsentiert. © 1997 Elsevier Science B.V.

Résumé

Dans cet article, nous présentons un algorithme de segmentation automatique et très précise d'images multi-dimensionnelles. Il améliore l'algorithme des Lignes de Partage des Eaux (Watershed) classique, dont les résultats sont imprécis lorsqu'on traite des images bruitées ou anisotropiques. Cet algorithme combine une croissance de région basée sur le principe des Lignes de Partage des Eaux avec une procédure très simple de sélection des marqueurs. Il est particulièrement bien adapté à la segmentation précise d'objets complexes, comme le cerveau dans une image de Résonance Magnétique (IRM) 3D, car il fournit une segmentation précise et complètement tridimensionnelle en un temps raisonnable. Des résultats comparatifs de segmentations obtenues par cet algorithme et par les Lignes de Partage des Eaux classiques sont présentés, ainsi que des exemples d'applications en visualisation 3D et en repérage de sulci dans le cerveau. © 1997 Elsevier Science B.V.

Keywords: 3D segmentation; MRI; Watersheds; Region growing; 3D images

* Corresponding author. Tel.: +32 10 47 80 74; fax: +32 10 47 20 89; e-mail: thiran@tele.ucl.ac.be.

1. Introduction

In the field of pattern recognition and computer vision, one of the main steps is the segmentation of the objects or regions of interest. Segmentation techniques of 2D images have been studied for many years. In 3D computer vision, dynamic scene analysis has attracted researchers interested in transforming a sequence of noisy two-dimensional inputs into a description of a scene in terms of objects, their three-dimensional shape, their motion through space. Only recently has the so-called spatio-temporal approach [16, 17, 25, 26] started using long sequences. In this approach, pictures are considered as time-varying stimuli occupying a three-dimensional space in which x and y are two spatial dimensions and t is the temporal dimension. In other applications, such as medical imaging, a three-dimensional image forms an image solid which is represented as a three-dimensional matrix of grey levels, $f(i, j, k)$. Each grey level represents a certain relevant property associated with the location (i, j, k) in the three-dimensional world. For instance, in Computed Tomography (CT) images, the grey levels represent the average of the attenuation coefficient of X-rays at point (i, j) in the k th cross-section image. In Magnetic Resonance (MR) images, every voxel (volume element) pictures a physical parameter, like the concentration of hydrogen at this location in the volume.

Nowadays, medical image analysis often requires accurate segmentation of morphological data from either Magnetic Resonance Imaging or X-ray Computed Tomography. Most segmentation research in computer vision has addressed the problem for 2D images, and very little work has been done for the 3D case. The most widely used technique in attempts to address this 3D segmentation problem is 3D edge detection [18, 33, 36, 2]. Similar to two-dimensional edge detector [20, 32], the three-dimensional edge detection method tries to locate edges along the boundaries of volumes. These edges are then linked to form groups of edge segments. These detected contours are not guaranteed to form closed volume boundaries in many cases. Therefore, the edge detection technique does not always offer the suitable level of abstraction for analysis of 3D images. Moreover, in many applications, a fast and accurate segmentation is needed, specially in the case of noisy images. This is typically

the case in medical imaging. In many studies, fast 3D imaging sequences are acquired to prevent spatial resolution losses due to patient motion but at the price of an increased noise level. Applications such as morphological shape analysis, automatic registration of different imaging modalities or visualisation of volumetric data require accurate 3D segmentation as an initial step.

Because of the great interest of an automatic segmentation of the brain in MR images, several authors have already proposed segmentation algorithms. Many segmentation procedures rest on the use of edge detection algorithms [2, 15, 3]. In these methods, the main problem is related to the choice of the resolution of the detected contours. A low resolution segmentation is not accurate enough for our applications, while with a high resolution segmentation, too many non-relevant details are detected. On another hand, procedures based on connectivity algorithms have also been proposed [6, 13, 14] but in those cases, different objects often remain connected by small links. Other algorithms require operator intervention during the segmentation process [15, 12, 31]. Neural networks have also been used for MRI segmentation [24, 11, 35] but in many cases the computation time becomes prohibitive.

In this paper, a complete image segmentation procedure is proposed. This algorithm will be illustrated in the case of the segmentation of the brain in 3D MR images of the head. This kind of segmentation has to be bounded by strict constraints. First, a very good accuracy is needed in this field of medicine where the smallest error can lead to major consequences. Secondly, the computation time for the complete processing must be short enough to be acceptable for the physician in a clinical routine procedure. And finally, the segmentation process has to be fully executed in 3D, and not slice by slice, since the brain is a complex organ whose surface presents many fissures corresponding to the lobe folds. Some voxels of a region could therefore be disconnected with the others into a given slice but could though be connected through adjacent slices. A fully 3D algorithm is thus needed.

We propose an algorithm called Queue-based Region Growing, whose mechanism is close to that of the watersheds. The next section will present this concept of watersheds as well as segmentation results

obtained by means of a classical watershed transform (WST). We will show that these results do not totally answer the above-mentioned constraints, mainly because of the inaccuracy of the detected contours. We will explain the origin of this inaccuracy. Section 3 will show how the Queue-based Region Growing algorithm overcomes those issues. As we will see, the major characteristic of this algorithm, compared with the classical watershed implementations, is that the processing is made in the pixel domain and not on a gradient image. We will present the principle of our algorithm, describe its two main steps and present segmentation results. In Section 4 we will discuss these results and introduce some interesting applications of this technique. Finally, conclusions will be drawn in Section 5.

2. The classical watershed transform

The watershed transform (WST) is one of the most efficient segmentation tools provided by Mathematical Morphology. Coming from topography, the concept of watersheds has been adapted to the field of image processing [7, 5, 4, 19]. Like in other approaches, the contours of an image are defined as lines where the grey level varies quickly in comparison with the other pixels of the neighbourhood. As the gradient of the image represents this variation, edge detection can be done by searching the local maxima of the positive gradient image. The classical watershed transform realises a partition of the pixels of the original image into a set of regions (called catchment basins) separated by lines which are the crest lines of the gradient image.

Thanks to recent works, efficient algorithms have been developed, based on a flooding definition of watersheds, allowing a general use of the watershed transform for image segmentation. The Vincent and Soille algorithm [34] is one of the oldest of the flooding methods. This algorithm examines all the pixels in the image at successive greyscale values, and operates in two stages, a sorting step followed by a flooding step. The sorting step produces a list of the pixels sorted in order of ascending greyscale value. Once the pixels have been sorted it is possible to operate on them at successive colour levels as if a flood were progressing up through the levels of the image. Each minimum

is given a unique label. The following procedure is applied to the pixels of each colour level in turn. All pixels at the current level that have a neighbour at a lower level are put in a first-in-first-out (FIFO) queue and are labelled with their neighbour's minimum identifier and distance to that minimum plus 1. The queue is then processed. Each pixel, a , is taken from the queue and its unmarked neighbours at the current height level are put at end of the queue and labelled with a 's identifier. This allows the minima to spread or flood within the confines of the height level. Extra logic identifies pixels on the boundaries equidistant from two flooding minima and then labels them as watershed pixels. A final scan of the grey level identifies and labels new minima [10]. More recently, optimum processing speed in the computation of this watersheds has been achieved by the use of hierarchical waiting queues [21, 22]. Meyer presented two watershed algorithms in [8]. These algorithms are based on a flooding definition of watersheds and a two level ordering relation provided by an ordered queue.

These watershed transforms fulfil two constraints imposed by the problem of the brain segmentation in MR images: they are quite fast and easily applicable to 3D images. But as it is well-known, the result of the rough application of the WST on the original gradient image suffers from oversegmentation, due to the noise affecting the original image. If we consider a grey level profile (for instance an horizontal line in a Magnetic Resonance Image of a human head), and if we compute the (1D) positive gradient of this profile, we can see that the noise causes many local minima, as illustrated in Fig. 1. Computed in 2D or in 3D, each local minimum will be the starting point of a basin and thus will cause oversegmentation.

Various solutions can be used to improve this situation. The most efficient one is the use of markers, as firstly proposed in [23]. The first 3D watershed algorithm using markers and hierarchical queues was proposed by Gratin [9]. This leads to efficient 3D segmentation of medical images. But the use of markers often requires various pre-processing steps, such as filtering, contrast enhancement or definition of inner and outer markers. Those steps can be quite complex, increasing the total computation time.

In this paper, we will introduce an accurate queue-based watershed-like region growing algorithm. In this two-step algorithm, the marker selection phase will be

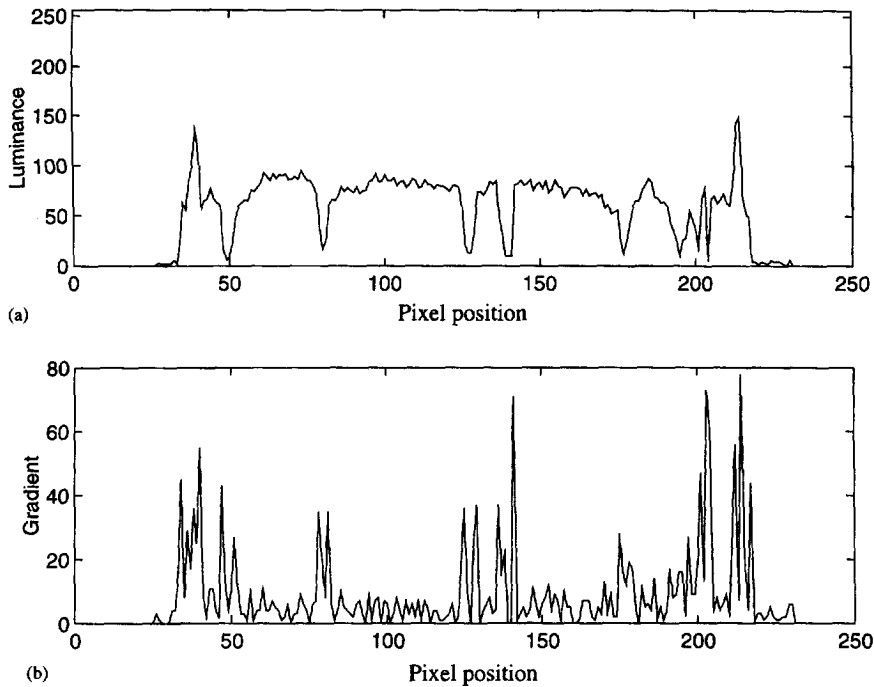


Fig. 1. (a) Grey level profile of a single line of a MR Image, (b) its 1D gradient.

very simple, and embedded into the first step of the algorithm, and not in a pre-processing step. In order to introduce this method, we will first analyse the results obtained by a classical watershed algorithm.

When working with the classical watershed (without markers), important pre-processing tasks have to be applied in order to achieve acceptable results. Pre-processing can be applied to the image before the computation of its gradient. The gradient image can be filtered or thresholded to reduce the number of catchment basins. The merging of basins after the computation of the watersheds is also possible. In fact, several processing have to be combined to achieve an acceptable segmentation. Fig. 2 shows the result obtained after application of the classical WST, as described by Vincent and Soile [34], combined with the following treatments. First, we threshold the original image: every voxel with a grey level lower than a given value (in this case: 8% of the maximum intensity of the image) receives a value equal to zero. By this way we get rid of a part of the noise which affects the background of the image. After this, the image is filtered by a $3 \times 3 \times 3$ median filter [30]

and finally the gradient image is computed from this filtered image. The WST is then computed on this gradient image. The complete processing ends by a region merging operation based on the region mean grey levels, and similar to the one presented in [28].

The results obtained by this way are not very satisfactory, mainly because of the inaccuracy on the location of the edges of the brain. Typically, the fissures on the surface of the brain are not well detected. As they determine the position of the lobe folds (called *sulcii*), their detection is very important in many applications.

This problem is due to the fact that the WST is applied on a gradient image where each voxel is computed as the *mean value* of the variation of the luminance in the 6 different directions (4 directions in the case of a 2D image). This leads to mistakes on edge position when the grey level varies quickly from one direction to another on a small number of pixels (or voxels). Actually, there is a lack of definition in the location of the contours which is a consequence of the use of the gradient [27]. Other kinds of gradient could be used, which should be more appropriate for

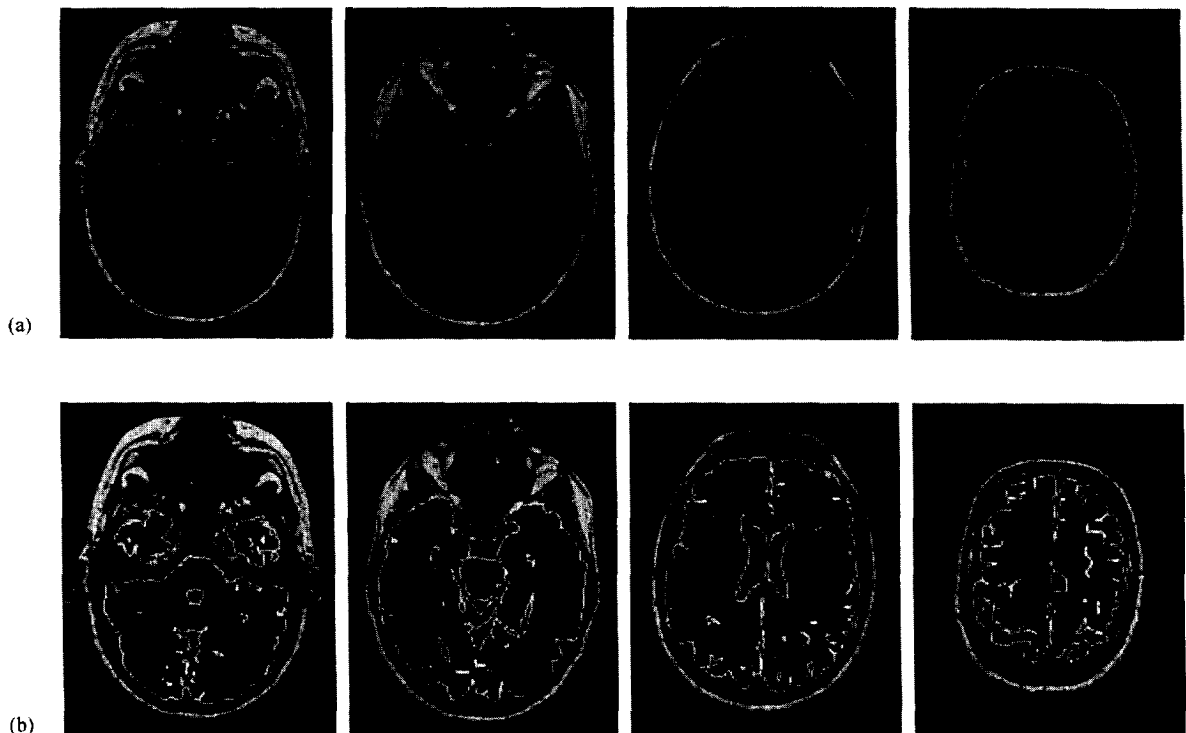


Fig. 2. Segmentation of a 3D MR Image by the classical watershed transform combined with pre- and post-processing of the original image: four slices of the original image (a) with the detected contours superimposed (b).

this situation, such as the maximal variation in the 6 directions, the difference between the maximal and the minimal variation in the 6 directions, or the pixelwise difference between a unit-size morphological dilation of the original image and the original image itself. But still, they may fail in detecting small details with the gradient. In the case of a segmentation of a binary image made of alternative black and white horizontal lines of thickness 1, the image is so full of details that there is no space left for representing the gradient. This can also be the case around the sulci. As the exact location of these sulci is required, we will define a modified version of the watershed algorithm which will work directly on the original image and not on its gradient.

This idea was proposed by Meyer in [22] to deal with colour images. In [1], Adams and Bischof proposed an algorithm called Seeded Region Growing, build on a quite similar approach. In this algorithm, the segmentation is controlled by choosing a set of

starting zones, called seeds. The region growing process evolves inductively from the seeds; each step of the algorithm involves the addition of one pixel to one of the above sets until it meets another region. This growing procedure is similar to the one of the watersheds, but in this case, the pixels are added to a region according to the difference between their grey level and the mean grey level of the region, while in the watershed approach, the region growing is mainly governed by the local gradient level of the pixels.

In the next section, we will introduce our algorithm, the Queue-based Region Growing.

3. The queue-based region growing algorithm

3.1. Principle

In a metaphoric way, comparing with the flooding definition of the watersheds, we can summarise the

principles of the Queue-based Region Growing by the following statement: “*The water should flood at the same time to a voxel P and to its neighbour P' if these voxels have the same grey level*”.

For the implementation, let us notice two important points: first we will apply a watershed-like region growing directly on the original image. Second it is now impossible to sort the different voxels of the image according to their gradient value, since the gradient is not computed any more: the height of a voxel in the topographic relief depends on the path followed by the water that floods to it.

The proposed algorithm rests on a set of *FIFO* queues which will contain pointers to voxels and can be decomposed into two main steps: the boring step and the flooding step. The first step allows to determine the different catchment basins called *starting zones*, or *seeds*, which are the different sets of connected voxels with the same luminance, corresponding to the local minima of the gradient image in the classical WST. All the basins receive different labels and the voxels belonging to these basins are labelled. The second step deals with the voxels not labelled after the first step, and simulates the immersion.

3.1.1. The boring step

In order to define the seeds, we examine all the successive voxels of the image in a given order. Once we find a non-labelled voxel which has the same luminance as at least one of its neighbours, we give to these voxels a new label and we extent this new basin with the following recursive procedure. We compare the luminance of a voxel of the basin with the luminance of each of its non-labelled neighbours. If one of these neighbours has the same luminance, we label it and apply the same procedure to it. Else the neighbour is put into a queue whose identification number corresponds to the absolute difference between the luminances of these points. The neighbour receives then a temporary label equal to the opposite value of the identification number of the queue. So if a voxel is put into a given queue, it means that it has an already positively labelled neighbour and that the grey level difference of these two voxels is to the identification number of the queue. This number also represents the value of the directional gradient of the voxel and thus the step of the inflow during which the voxel will be reached.

Now we have to be careful with the voxels that have several neighbours with different positive labels. For such a voxel, the above-mentioned recursive procedure will first introduce it in a queue of identification number equal to the grey level difference between this voxel and the first encountered neighbour. The voxel will be (negatively) labelled in consequence. Later in the progression of the recursive procedure, this point will be added in another queue only if the identification number of this queue is lower than the opposite of the label of the voxel. The label of this voxel will then be changed.

This step results in a set of positively labelled voxels surrounded by negative-labelled voxels. The untreated voxels keep an initial label called *init*. An example of the result of this boring step on a 7×7 matrix is presented in Fig. 3.

Let us notice that we can choose other criteria for the definition of the starting zones. Considering every flat zone will result in an oversegmentation. We will discuss this point in Section 3.3.

3.1.2. The flooding step

This second step realises the extension of the catchment basins by simulating the flooding through a recursive procedure which progressively empties the queues and labels the voxels of the queues, starting from the non-empty queue with the lowest identification number. Each voxel extracted from a queue is correctly labelled, taking into account that:

- (1) Some voxels receive the label called *WSHED*. For instance this is the case if the smallest difference between the luminance of a voxel and the luminance of one of his positively labelled neighbour is obtained for more than one direction.
- (2) Some already positively labelled voxels can still remain into a queue. This is the case of those voxels with several positively labelled neighbours, as we have seen in the boring step. They have already been extracted from a queue and labelled according to the corresponding identification number. But they remain in one or several other queues, corresponding to their other neighbours. In this case, those already positively labelled voxels must be ignored.

After the extraction of a voxel, we have to examine its neighbours to determine when they will

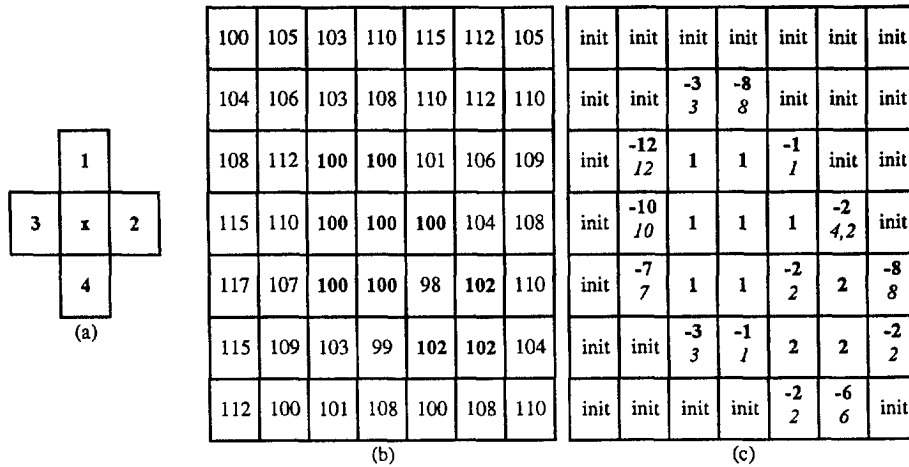


Fig. 3. Typical result of the boring step of the queue-based region growing: (a) example of the order used to examine the neighbours of a pixel, (b) original grey level values, (c) label of the pixels (bold) an identification number of the corresponding queue (italic).

be reached by the water (if they are not yet). This procedure is similar to the one used for the boring step. In fact, when starting to empty a queue of identification number n , we are sure that the queues of identification number m ($m < n$) are empty. Progressively, when treating voxels of queue n , it is probable that voxels will be added to queues of identification number lower or equal to n . Naturally, the voxels of the queues of lower identification number have to be examined before voxels of queue $n + 1$ and before the lastly added voxels of queue n . To distinguish between both types of voxels of queue n , we insert a fictitious voxel (*fictv*). All the voxels of this queue are treated until we encounter a *fictv* and we continue to empty this queue only when all the queues of lower identification number are empty again.

After the application of this algorithm the result is an image whose voxels are labelled according to the catchment basin they belong to. The voxels with the label *WSHED* can also be integrated to the catchment basins as explained in [34].

3.2. Performances

The algorithm described in this paper runs in linear time with respect to the number N of voxels in the original image. For the boring step, only one scan of the N voxels is necessary in order to label the voxels

with at least one neighbour with the same luminance. During the flooding step we label all the voxels of the queues. That means at most one scan of the image. But for each voxel of the queue, we treat a part of its neighbours. For a V -connectivity underlying grid, we would need V scans if all the neighbours were put in the queues. Now, a positively labelled voxel whose neighbours are being treated, has at least one positively labelled voxel in its neighbourhood. Therefore the maximum number of scans is V . In fact the real number of scans necessary for the flooding step is lower than V because more and more voxels are labelled.

With regard to the memory requirements, we need:

- (1) an output image of the same size as the initial image.
- (2) a set of queues of pointers to voxels. The number of queues depends on the number of grey levels in the image (typically 256).

A solution to reduce the memory requirements consists in limiting the number of objects of the resulting image. In many practical cases and principally in medical imaging, the goal of the segmentation is to isolate one object from the background and the number of labels can then be strongly reduced. Reduction of memory requirements can also be achieved by limiting the number of queues. In fact, the queues with high identification numbers are rarely used except for strongly noisy images.

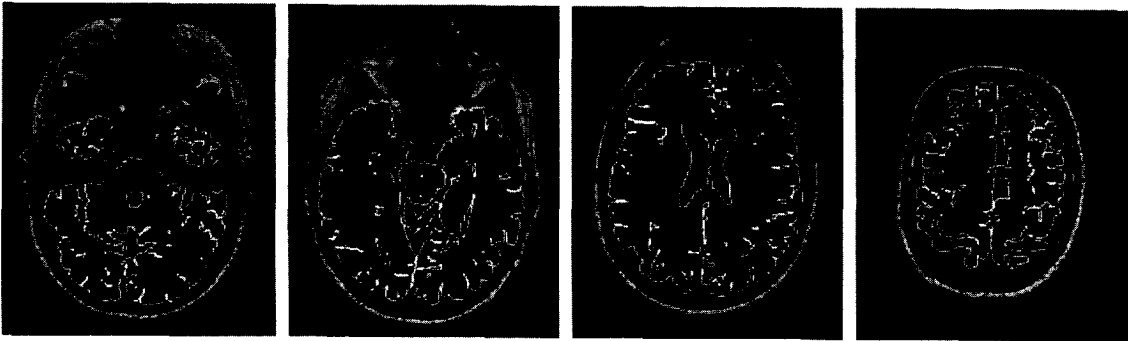


Fig. 4. Segmentation of a 3D MR image by the queue-based region growing: four slices of the original image with the detected contours superimposed.

The total computation time of the segmentation of a MR Images ($256 \times 256 \times 90$ voxels) is about 3 min on a low-cost workstation.

3.3. Results

In this algorithm, the selection of the starting zones, that can be called *seeds* or *markers*, is operated by finding homogeneous zones in the original image. This can be done by selecting every region with two or more pixels of the same grey level, as explained above. But this will of course lead to oversegmentation again. In our implementation, we consider that a starting zone is made of points whose grey level difference is lower than a given threshold. Referring to the classical watershed transform, this is equivalent to a thresholding of the gradient image. For Magnetic

Resonance images of a head, where the brain is quite homogeneous, this means that the *boring step* will operate like a simple region growing procedure resulting in a large starting zone, that will extend during the *flooding step*, by a watershed-like procedure, to achieve very accurate contours.

Fig. 4 presents four slices of the data set with the contours of the segmented brain (white coloured) superimposed. This result was obtained by the computation of the Queue-based Region Growing with a grey level difference threshold of 15.

4. Discussion and applications

Comparing Figs. 2 and 4 one can see that the accuracy of the detected contours is much better on the

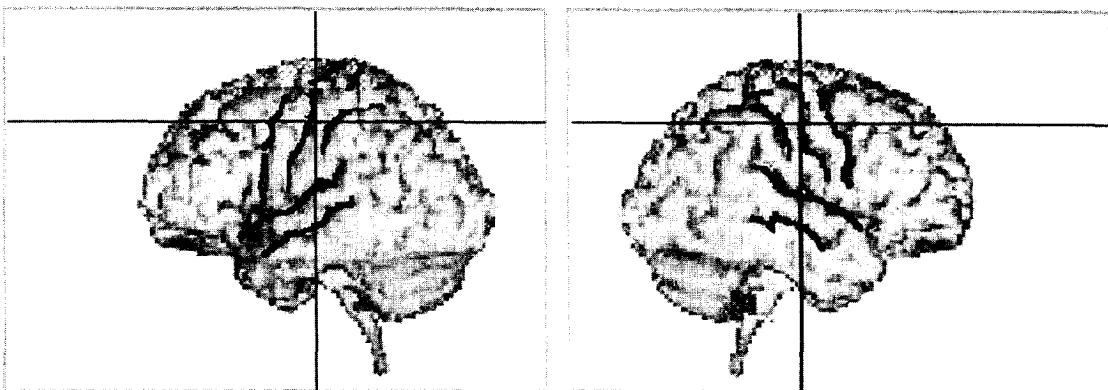


Fig. 5. Application of the 3D accurate segmentation by queuebased region growing: 3D surface rendering and interactive edition of brain sulcii.

latter, especially around the sulcii. These high quality results can be used in morphometric applications (e.g. localisation and characterisation of brain tumors), 3-D rendering of the brain surface, or automatic labelling of the brain regions (gyrii and sulcii). One of our most successful application merges precise anatomy from MRI with Positron Emission Tomography (PET) data. PET is a functional imaging technique whose spatial resolution is relatively poor as compared to MRI. One may consequently merge the information from these two modalities provided they are accurately registered [29]. The precise description of the brain surface obtained by the Queue-based Region Growing is used to define (and label) sulcii as 3D polylines, using multiple views in a surface viewer (stereo definition) as shown by Fig. 5. Rendering is operated by integral shading [2]. The surface viewer is spatially connected to a volume viewer containing the registered PET data and the sulcii are merged into the functional PET volume.

5. Conclusions

In this paper, we have shown how to solve a difficult segmentation problem by means of a watershed-like transform, called Queue-based Region Growing. The results provided by the classical watershed algorithm suffers from oversegmentation and inaccuracy. By working on the original image and not on a gradient image, we have gone beyond these limits when still preserving all its advantages (i.e. fully 3D-segmentation, short computation time, etc.).

The result obtained can be used for visualisation but can also serve as preliminary step of a more complex analysis of brains such as assisted diagnostic of different brain illnesses, localisation and characterisation of brain tumors or automatic identification of the various regions of the brain by the detection of specific *sulcii*. All these applications need an accurate segmentation of the brain as an initial step.

Acknowledgements

The authors wish to thank Dr. Ch. Michel (Positron Emission Laboratory, Université Catholique de Louvain) and Ir. P. Fourez for their major contribu-

tions to this research. The works of J.-Ph. Thiran are supported by the Belgian F.R.I.A.

References

- [1] R. Adams, L. Bischof, Seeded region growing, *IEEE Trans. Pattern Anal. Machine Intell.* 16 (June 1994) 641–647.
- [2] M. Bomans, K.-H. Höhne, U. Tiede, M. Riemer, 3-D segmentation of MR Images of the head for 3-D display, *IEEE Trans. Med. Imag.* 9 (2) (June 1990) 177–183.
- [3] M.E. Brummer, R.M. Mesereau, R.L. Eisner, R.R.J. Lewine, Automatic detection of brain contours in MRI data sets, *IEEE Trans. Med. Imag.* 12 (2) (June 1993) 153–166.
- [4] S. Beucher, Watersheds of functions and picture segmentation, in: *Proc. IEEE Internat. Conf. Acoust. Speech Signal Process.*, Paris, France, May 1982, pp. 1928–1931.
- [5] S. Beucher, C. Lantuéjoul, Use of Watersheds in contour detection, in: *Proc. Internat. Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation*, Rennes, France, September 1979, pp. 17–21.
- [6] H.E. Cline, C.L. Dumoulin, H.R. Hart, W.E. Lorensen, S. Ludke, 3D reconstruction of the brain from magnetic resonance images using a connectivity algorithm, *Magnetic Resonance Imag.* 5 (1987) 345–352.
- [7] H. Digabel, C. Lantuéjoul, Iterative algorithms, in: J.L. Chermant (Ed.), *Proc. 2nd European Symp. on Quantitative Anal. Microstructures in Material Science, Biology and Medicine*, Stuttgart, West Germany, October 1977, Riederer, pp. 85–99.
- [8] E.R. Dougherty, *Mathematical Morphology in Image Processing*, Marcel Dekker, New York, 1993, pp. 433–481.
- [9] C. Gratin, F. Meyer, *Mathematical Morphology in three dimensions*, in: *Proc. 8th Internat. Congress of Stereology, Acta Stereologica*, Irvine (CA), 11 (Supp 1), 1992, pp. 551–559.
- [10] D. Haggard, M. Razaz, P. Atkin, Analysis of watershed algorithms for greyscale images, in: *Proc. 1996 IEEE Internat. Conf. on Image Process. ICIP'96*, Vol. 3, 1996, pp. 41–44.
- [11] L.O. Hall, A.M. Bensaïd, L.P. Clarke, R.P. Velthuisen, M.S. Silbiger, J.C. Bezdek, A comparison of neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain, *IEEE Trans. Neural Networks* 3 (5) (September 1992) 672–682.
- [12] K.-H. Höhne, W.A. Hanson, Interactive 3D segmentation of MRI and CT volumes using morphological operators, *J. Comput. Assisted Tomography* 16 (2) (March–April 1992) 285–294.
- [13] T.L. Jernigan, G.A. Press, J.R. Hesselink, Methods for measuring brain morphologic features on magnetic resonance images using a connectivity algorithm, *Arch. Neurol.* 47 (1990) 27–32.
- [14] M. Joliot, B.M. Mazoyer, Three-dimensional segmentation and interpolation of magnetic resonance brain images, *IEEE Trans. Med. Imag.* 12 (2) (June 1993) 269–277.

- [15] D.N. Kennedy, P.A. Filipek, V.S. Caviness, Anatomic segmentation and volumetric calculations in nuclear magnetic resonance imaging, *IEEE Trans. Med. Imag.* 8 (1) (March 1989) 1–7.
- [16] S.P. Liou, R. Jain, Motion detection in spacio-temporal space, *Comput. Vision Graphics Image Process.* 45 (1989) 227–250.
- [17] S.P. Liou, R.C. Jain, An approach to three-dimensional segmentation, *CVGIP: Image Understanding* 53 (3) (1991) 237–252.
- [18] H.K. Lui, Two- and three-dimensional boundary detection, *Comput. Graphics Image Process.* 6 (1977) 123–134.
- [19] F. Maisonneuve, Sur le partage des eaux, Internal rep. CMM, School of Mines, Paris, France, 1982 (in French).
- [20] D. Marr, E. Hildreth, Theory of edge detection, *Proc. Roy. Soc. (London)* 207 (1980) 187–217.
- [21] F. Meyer, Un algorithme optimal de ligne de partage des eaux, Actes du 8eme Congres AFCET, Lyon-Villeurbanne, France, 1991, pp. 847–859.
- [22] F. Meyer, Color image segmentation, in: *Proc. 4th Internat. Conf. Image Process. Appl. Maastricht*, 1992, pp. 523–548.
- [23] F. Meyer, S. Beucher, Morphological segmentation, *J. Comm. Image Represent.* 1 (1) (September 1990) 21–46.
- [24] M. Oskan, H.G. Sprenkels, B.M. Dawant, Multi-spectral magnetic resonance image segmentation using neural networks, in: *Proc. Internat. Joint Conf. on Neural Network*, Vol. 1, San Diego, June 1990, pp. 429–434.
- [25] M. Pardàs, P. Salembier, Time recursive 3D segmentation for image sequences, in: *Proc. EUSIPCO-94, VII European Signal Process. Conf.*, Edinburgh, 1994, pp. 18–21.
- [26] M. Pardàs, P. Salembier, 3D morphological segmentation and motion estimation for image sequences, *Signal Processing* 38 (1) (1994) 31–43.
- [27] M. Pardàs, P. Salembier, L. Torres, 3D morphological segmentation for image sequence processing, in: *Proc. IEEE Winter Workshop on Nonlinear Signal Processing*, Tampere, Finland, 1993, pp. 6.1–6.3.
- [28] T. Pavlidis, Y.T. Liow, Integrating region growing and edge detection, *IEEE Trans. Pattern Anal. Machine Intell. PAMI-12* (3) (March 1990) 225–233.
- [29] U. Pietrzyk, K. Herholtz, G. Fink, A. Jacobs, R. Mielke, I. Slanzky, M. Wurker, W.D. Heiss, An interactive technique for 3-D image registration: validation for PET, SPECT, MRI and CT brain studies, *J. Nucl. Med.* 35 (1994) 2011–2018.
- [30] A. Rosenfeld, A. Kak, *Digital Picture Processing*, 2nd ed., Academic Press, New York, 1982.
- [31] T. Schiemann, M. Bomans, U. Tiede, K.-H. Höhne, Interactive 3D segmentation, in: *SPIE, Proc. Visual. Biomed. Comput.*, Vol. 1808, 1992.
- [32] V. Torre, T.A. Poggio, On edge detection, *IEEE Trans. Pattern Anal. Machine Intell. PAMI-8* (2) (March 1986) 147–163.
- [33] J.K. Udupa, S.N. Srihari, G.T. Herman, Boundary detection in multidimensions, *IEEE Trans. Pattern Anal. Machine Intell. PAMI-4* (1982) 41–50.
- [34] L. Vincent, P. Soile, Watersheds in digital spaces: an efficient algorithm based on immersion simulations, *IEEE Trans. Pattern Anal. Machine Intell. PAMI-13* (June 1991) 583–597.
- [35] A.J. Worth, D.N. Kennedy, Segmentation of magnetic resonance brain images using analogue constraint satisfaction neural networks, *Image Vision Comput.* 12 (6) (July 1994) 345–354.
- [36] S.W. Zucker, R.A. Hummel, A three-dimensional edge operator, *IEEE Trans. Pattern Anal. Machine Intell. PAMI-3* (1981) 324–331.