ELSEVIER

Image and Vision Computing 17 (1999) 881–895

image AND vision COMPUTING

Genetic algorithm-based interactive segmentation of 3D medical images

S. Cagnoni[a,*], A.B. Dobrzeniecki[b,1], R. Poli[c], J.C. Yanch[d]

[a]*Department of Computer Engineering, University of Parma, Italy*
[b]*Harvard Medical School, Cambridge, USA*
[c]*School of Computer Science, The University of Birmingham, UK*
[d]*Whitaker College of Health Sciences and Technology, MIT, Cambridge, USA*

Received 11 March 1997; received in revised form 23 September 1998; accepted 1 October 1998

Abstract

This article describes a method for evolving adaptive procedures for the contour-based segmentation of anatomical structures in 3D medical data sets. With this method, the user first manually traces one or more 2D contours of an anatomical structure of interest on parallel planes arbitrarily cutting the data set. Such contours are then used as training examples for a genetic algorithm to evolve a contour detector. By applying the detector to the rest of the image sequence it is possible to obtain a full segmentation of the structure. The same detector can then be used to segment other image sequences of the same sort. Segmentation is driven by a contour-tracking strategy that relies on an elastic-contour model whose parameters are also optimized by the genetic algorithm. We report results obtained on a software-generated phantom and on real tomographic images of different sorts. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Genetic algorithm; Elastic contour; Filter

1. Introduction

Segmentation, a central issue of computer vision, is a fundamental pre-processing step in most systems that support medical diagnosis or planning of surgical operations and radiation treatments [1–3].

Despite the amount of research that has been done in the field (see for example [4–7]), medical image segmentation is still tackled on an ad hoc basis, using manually-tunable algorithms with some degree of adaptivity to obtain acceptable results. However, often these algorithms are complex and hard to tune manually, while robust self-adaption schemes are hard to find.

Contour-based segmentation methods are generally computationally efficient, but, when applied to real images, often lack robustness, being quite sensitive to noise and data variability. The reliability of such methods can be improved using techniques for segmentation and data representation based on data-driven elastic models, such as *snakes* [8] and deformable surface models [9], which have been applied successfully also in the field of medical imaging [10–12].

Medical imaging techniques, e.g. computed tomography (CT) and magnetic resonance (MR), often produce sequences of images representing 2D sections of a 3D anatomical structure of interest, which needs to be analyzed as a whole. Temporal image sequences are also acquired, in which the variations in time of a moving anatomical structure (e.g. the heart) can be studied. Segmentation can be used to identify the structure in each section (frame) and produce its full 3D reconstruction or model its deformations (motion) in time. Contour-tracking procedures have been used with some success [13] by constraining the segmentation process with some regularizing assumptions. This allows a contour, detected (or manually traced) on one section (frame), to be used to seed the segmentation of the same structure in a neighboring slice of the sequence.

The most accurate method to produce segmentations of 3D structures is still tracing contours manually. However, owing to its heavy time requirements, manual segmentation is highly impractical. A reasonable compromise is to design segmentation systems based on tools which can self-adapt to new problem classes, or can generate new solutions when none of the available ones produce accurate segmentations. Such systems should also allow clinicians to refine the results manually, when minor inaccuracies are detected. In any case, it is desirable that user's intervention, in terms of time and (manual) workload, be as limited as possible.

* Corresponding author. Present address: Department of Computer Engineering, University of Parma, Viale delle Scienze, 43100 Parma, Italy. Tel.: + 39-0521-905-731; fax: + 39-0521-905-723.

E-mail address: cagnoni@ce.unipr.it (S. Cagnoni)

[1] Danish Technical University, Copenaghen, Denmark.

0262-8856/99/$ - see front matter © 1999 Elsevier Science B.V. All rights reserved.
PII: S0262-8856(98)00166-8

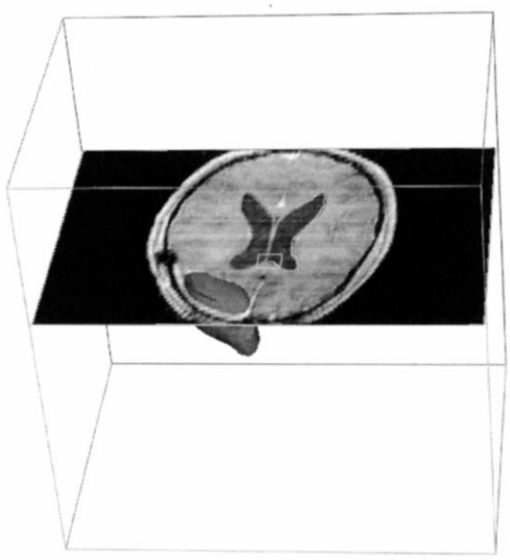

Fig. 1. Example of the SegmentVIEW system used to delineate seed contours. Using a mouse, the user traces contours directly on slicing planes; such a partially-drawn contour is shown near the cursor. Next to this contour is a semi-transparent surface that represents the triangulation of another set of contours (this surface encompasses a low-grade sarcoma visible in this MRI data of a human brain).

Among *learning-from-examples* techniques, in which the problem of image segmentation can be reformulated as an optimization problem, genetic algorithms (GAs) [14,15] have been used for segmentation by Bhandarkar et al. [16], who defined a multi-term cost function which is minimized using a GA-evolved edge configuration. An adaptive approach in which GAs are used to optimize the performances of the Phoenix segmentation algorithm [17] is described in [18]. In [19] Chun and Young mapped a region-based segmentation onto the binary string representing an individual, and evolved a population of segmentations using a fuzzy fitness function. GAs have also been used to find optimal descriptors to represent 3D structures [20]. Poli and Valli [21] use a GA to optimise the parameters of recurrent neural networks to segment echocardiographic images. An approach based on genetic programming [22] is described in [23]. A fairly comprehensive review of other approaches is available in [24].

In the field of signal processing, GAs have been used, for example, to design general-purpose digital filters [25,26], or QRS detectors based on polynomial filters [27], providing solutions characterized by a good trade-off between performance and computation load.

The segmentation method proposed in this article, derived from the one introduced in [28] and further developed in [29] and [30], is aimed at exploiting GA optimization to design contour-based segmenters for specific anatomical structures in 3D medical data sets. The "genetic design" is based on a small set of manually-traced contours of the structure of interest. The method combines the good trade-off between simplicity and versatility offered by polynomial filters with the regularization properties that characterize elastic-contour models.

In the following sections the method is described, along with the results obtained on a software-generated image sequence of a phantom and on sets of real CT and MR images.

2. GA-based design

In the proposed contour-tracking method two cascaded modules are used: a non-linear edge detector and an interpolator based on an elastic-contour model. As the quality of the results depends both on the coefficients of the edge detector and on the parameters of the interpolator, and therefore on the effectiveness of their coupling, we use a GA to optimize all such parameters at the same time.

The GA-based design procedure can be split in two steps:

1. Interactive specification of a set of contours of a structure of interest.
2. GA optimization of the parameters of the edge detector and of the elastic-contour model, based on such contours.

During genetic design, the GA is required to yield optimum performance on the set of sections in which the contour of the structure of interest is manually traced: one *seed contour*, used to initialize the segmentation and one or more *reference contours*, used to evaluate the segmentation quality. This translates into maximizing a fitness function that measures the degree of similarity between the references and the corresponding contours produced by the detector being evolved. The exact definition of the fitness function will be given later, after a more detailed description of the edge detector and of the interpolator.

After genetic design, the extraction of the contours of the structure of interest from the whole data set can be done using the genetically-turned algorithm. The user initializes the process by tracing one seed contour on one section. Then, for each of the following sections in the data set, the edge detector is applied to a neighborhood of the contour extracted on the previous one. The output of the detector (a set of sparse edge points) is then interpolated using the elastic model to recover the complete contour of the structure of interest. The contour-tracking process can be iteratively applied to all sections of the data set in which the structure of interest is present.

The GA-based design is obviously performed only when a new class of problems is tackled. When a previously-obtained contour detector is already available for the problem under consideration, the only operation required by the user is tracing a single seed contour manually.

2.1. Specification of contours

The seed contour and the reference contours are drawn (using special sectioning and visualization tools [31]) on parallel sections, which can arbitrarily cut the 3D data set.

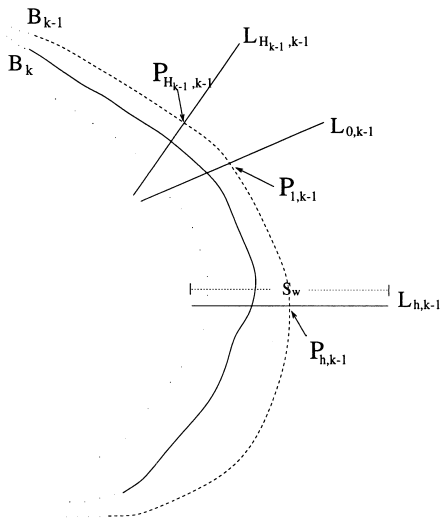

Fig. 2. Definition of the scan lines for the contour-tracking procedure: for each point $P_{h,k-1}$ of the contour B_{k-1}, a scan line $L_{h,k-1}$ centered around $P_{h,k-1}$ is defined, with a length of S_w samples, along the direction of the unit vector orthogonal to B_{k-1} in $P_{h,k-1}$.

If other "correct" contours, lying on sections which are not parallel to the references, are available, the contour points resulting from the intersection between such contours and the slices that still have to be segmented can be pre-set and used to further constrain the elastic contour model.

Fig. 1 shows the 3D visualization tool, which is used to perform this task.

2.2. Edge detector

The edge detector is based on a one-dimensional second-order polynomial filter. This choice was suggested by the encouraging results obtained in previous experiments on ECG signals [27].

As the shape modifications of anatomical structures between subsequent sections Q_k ($k = 0$ for the section on which the seed contour B_0 is traced, $k \leq 1$ for the following sections) are roughly smooth scale changes, the filters are applied to a set of limited-length scan lines on slice Q_k The scan lines are derived from the H_{k-1} points $P_{h,k->1}$, $h = 1$, H_{k-1} belonging to the contour B_{k-1}, extracted or traced on slice Q_{k-1}, as follows (see Fig. 2):

- For each point $P_{h,k-1}$, calculate the unit vector orthogonal to B_{k-1} in the point considered.
- Along the direction of such a vector, resample the input image on a S_w-sample long scan line $L_{h,K-1}$ centered around $P_{h,k-1}$.

Using this strategy, the filter is applied only to a neighborhood of the contour extracted on the last-processed slice. This prevents detection of edges, similar to the ones belonging to the anatomical structure to be segmented that are far from the region of interest and possibly belong to other structures.

The detector is based on a one-dimensional polynomial filter cascaded with a thresholding stage. The filter operates on N_p samples of a scan line, possibly non-consecutive and chosen by the GA, that are located in the neighborhood of the sample x_i for which the filter output $o(x_i)$ is computed. It has the following structure:

$$o(x_i) = c_0 + \sum_{j=1}^{N_p} c'_j \cdot x_{i-d_j} + \sum_{j=1}^{N_p} \sum_{l=j}^{N_p} c''_{j,l} \cdot x_{i-d_j} \cdot x_{i-d_j},$$

where $c_0, c'_j, c''_{j,l} (j, l = 1, \cdots, N_p, l \geq j)$ are the filter coefficients and $d_j (j = 1, N_p)$ are the offsets along the scan line, with respect to $x_i (i = 1, S_w)$, of the N_p samples.

The edge points extracted by the filter have the coordinates of those samples x_i for which the value of $o(x_i)$ is greater than a threshold T. The output of the detector $O(x_i)$ is therefore computed as follows:

$$O(x_i) = \begin{cases} 1 & \text{if } o(x_i) > T \\ 0 & \text{otherwise} \end{cases}.$$

In the tests performed, we have used this edge-detection strategy (see Section 3). In an alternative implementation, thresholding can also be performed according to the following rule:

$$O(x_i) = \begin{cases} 1 & \text{if } (o(x_i) - Z) \qquad (o(x_{i-1}) - Z) < 0 \\ 0 & \text{otherwise} \end{cases}.$$

which implements a threshold-crossing detector, with threshold Z.

2.3. Elastic contour model

To segment a slice, we apply the edge detector to the image, along the scan lines defined in Section 2.2. For each of the H_{k-1} scan lines $L_{h,k-1}$ ($h = 1, H_{k-1}$) considered in the segmentation of slice Q_k, the edge detector extracts a set $\{X_{i,h,k}, i = 1, N(h,k)\}$ of edge points, for which the output of the detector equals 1. Consistently with the notation introduced above, the index h refers to the scan line and the index k to the section where the edge points were detected.

Although the output of the detector already provides an edge-based segmentation of a slice, it is generally relatively noisy. For some scan regions, $N(h,k)$ may be equal to 0 or greater than 1, spurious edges may be present while real ones may be lacking and the estimated position of the detected edge points may be altered by noise. Therefore, to enhance the robustness of the method, a contour-recovery strategy is adopted, which is based on the following hypotheses on the inter-slice and intra-slice variations of the contour of interest

- Inter-slice deformations of the contour are small and smooth, i.e. the shape of the contour does not change abruptly in two consecutive slices, even if the contour may have some high-curvature tract.

- The intensity profile of the edge does not change dramatically along the contour in the same slice.
- The contour points to be detected are close to one another.

These assumptions are incorporated in an elastic contour model whose parameters are set according to an uncertainty function that measures the reliability of each edge point extracted by the detector.

The contour to be detected is represented as a deformable curve attached to all the detected edge points by means of springs of different stiffness. Each spring acts along the direction of the scan line to which its "anchor" edge point belongs.

Let $p_i(h,k)$ denote the coordinate of the edge point $X_{i,h,k}$ extracted by the filter on the scan line $L_{h,k-1}$ and $y(h,k)$ the (unknown) coordinate of the contour point to be detected, in the reference system of such a scan line. The origin of the reference system is coincident with the contour point $P_{h,k-1}$ starting from which the scan line $L_{h,k-1}$ was defined. Let us consider, for the moment, h a continuous variable, which represents the distance, travelled when moving along B_{k-1}. We can define the elastic energy of the model as the functional

$$\varepsilon(y,k) = \int_{B_{k-1}} \left(|\dot{y}(h,k)|^2 + \sum_{i=1}^{N(h,k)} \beta_i(h,k) p_i(h,k) \right) \mathrm{d}h,$$

where $\beta_i(h,k)$ is the stiffness of the spring attached to the edge point $X_{i,h,k}$.

The contour B_k to be recovered is the locus of points $y(h,k)$ minimizing the energy functional $\epsilon(y,k)$. In the actual implementation, the functional is discretized and minimized through a relaxation process, resulting in the (discrete) set $\{y(h,k), h = 1, H_{k-1}\}$ of the coordinates of the pixels, one for each scan line, that represent the contour of the structure of interest. The scan line coordinates $\{y(h,k), h = 1, H_{k-1}\}$ are then translated into (x,y) coordinates, where each pair of coordinates represents a point of B_k. As contour B_{k-1} may have fewer points than B_k, the resulting set of edge points may produce a contour that contains gaps, which are filled through linear interpolation. This choice does not alter the resulting contour shape, because, as long as the hypotheses of small contour variations hold, the distance between two consecutive points in the output set is never greater than few (generally 1–3) pixels.

Although the elastic model used in this work may look similar to classical snakes, there is no direct correspondence between the energy terms used in [8] and those in our energy functional as, in the proposed model, variations with respect to the contour extracted in the previous section are used instead of the actual coordinates of contour points. Thus, our technique does not impose the constraint that the resulting contours be smooth. Smoothness is imposed on the values $y(h,k)$, i.e. on the distance, along the scan lines, between the contour in the current slice and the contour in the previous one. This means that the elasticity of the contour, while preventing the contour B_k from presenting abrupt shape changes with respect to B_{k-1}, does not prevent it from having high curvature tracts. Smooth shapes variations such as scale changes are permitted. This is essential to avoid biases in the shape of the segmented anatomical structures.

The degree of similarity between B_k and B_{k-1} depends on the values of spring stiffness. Large values of β_i (h,k) allow for any shape changes, small β_i (h,k)'s only permit scale changes. Such values are set based on an uncertainty measure attributed to each edge point, so that highly reliable points are given more allowance to deform the contour to be recovered than less reliable points.

We have used the following function as an uncertainty measure for edge point $X_{i,h,k}$ on slice Q_k ($k = 0$ for the manually-traced seed contour) in which lower values are associated with edge points which are more likely to represent actual contour points:

$$\begin{aligned} C_i(h,k) = {} & W_1 d(X_{i,h,k}, P_{h,k-1}) \\ & + W_2 d(X_{i,h,k}, \hat{P}_{h-1,k}) \\ & + W_3 |I(X_{i,h,k}) - \bar{I}_{k-1}|, \end{aligned}$$

where W_1, W_2, W_3 are weighting parameters in the range [0,1], $d(a,b)$ is the Euclidean distance between two points a and b, $\hat{P}_{h-1,k}$ is the best edge point detected in the scan line $L_{h-1,k-1}$ (i.e., the one with the lowest $C_i(h,k)$), $I(X_{i,h,k})$ is the directional derivative (slope), along the scan line, of the edge point $X_{i,h,k}$, and $\bar{I}_{k-1}$ is the average directional derivative, along the scan lines used to extract B_{k-1}, of the points in B_{k-1}.

The first term favours edge points in B_k that are close to the corresponding ones in B_{k-1}, the second term favours consecutive edge points in B_k that are close to each other, the third one favours edge points in B_k whose slope is close to the average slope of the edges in B_{k-1}.

The term $d(X_{i,h,k}, \hat{P}_{h-1,k})$ cannot be calculated for $L_{0,k-1}$: to compensate for this a value proportional to the sum of the other two terms of $C_i(0,k)$ is added. This prevents edges in $L_{0,k-1}$ from having values of $C_i(0,k)$ much lower than the ones in $L_{h,k-1}$, $h > 0$.

The above-defined function $C_i(h,k)$ is used to assign a stiffness coefficient $\beta_i(h,k) = \mathrm{e}^{-C_i(h,k)}$ to the spring connected to each edge point $X_{i,h,k}$ detected along scan line $L_{h,k-1}$. Therefore, the optimization of the controllable parameters of the elastic contour model (the stiffness coefficients $\beta_i(h,k)$), is performed by the GA, indirectly, through the optimization of the weights W_1, W_2 and W_3 of $C_i(h,k)$. This allows for a selective exploitation of the perceptual and geometrical features which are most relevant for the segmentation task being performed.

The use of the uncertainty function $C_i(h,k)$ to set the stiffness parameters has similar effects on the elastic model as the definition of the external forces in the snake proposed by Radeva et al. [32]. In such a model the

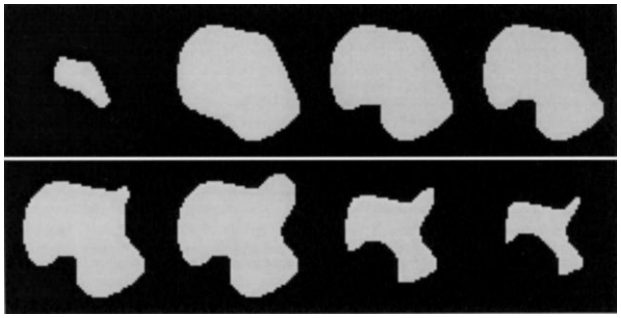

Fig. 3. Eight equally-spaced (6-image interval) images from the software-simulated sequence.

computation of the external forces of the snake also implies a comparison between the directional characteristics of the edges considered with some reference that is either derived from domain knowledge or from other similar cases (as can be, in our method, the corresponding edge points of the previous contour).

2.4. GA details

Using the notation introduced before, we can finally express the fitness function used by the GA as:

$$F = A - \sqrt{\sum_{k \in \kappa} f_k^2},$$

where

$$f_k(y) = M\sqrt{\frac{\sum_{h=1}^{H_{k-1}} (y^r(h,k) - y(h,k))^2}{H_{k-1}}}.$$

κ is the set of reference contours used for the evaluation of each individual, and H_{k-1} the number of scan lines used in the extraction of B_k. The term $y^r(h,k)$ is the actual scan line coordinate, along scan line $L_{h,k-1}$, of the edge point to be extracted. A is a constant large enough to keep $F \geq 0$; M is a constant that is used to keep the fitness values within a desired range.

With the above fitness function, the GA is used to optimize the following parameters, regarding both the edge detector and the elastic-contour model, that drive the segmentation process:

- $N_p + \frac{1+(N_p+1)\cdot N_p}{2}$ filter coefficients, in the range $[-15, 15]$ for c_0 and $c'_i, (i = 1, N_p)$ and in the range $[-1.5, 1.5]$ for $c''_{i,j}(i, j = 1, N_p; j \geq i)$
- $N_p - 1$ offsets[2] of the data processed by the filter, in the range $[-6, 6]$;
- The threshold T of the detector, in the range $[1, e^{20.48}]$
- The weights W_1, W_2, W_3 of the uncertainty function, in the range [0,1], that are used to compute the coefficients β of the elastic contour model.

As, in the tests performed, N_p was set to three, 16 parameters were encoded in the genome used in the GA. The GA implementation adopted in the experiments is GAUCSD-1.4 [33].

All values were encoded in 6-bit strings, resulting in a 96-bit genome. However, as GAUCSD-1.4 provides support for Dynamic Parameter Encoding the representation precision can be considered virtually infinite.

Although the scan line length S_w could also be optimized by the GA, it was pre-set to about one-third of the image side ($S_w = 23$ in the case of 64×64-pixel images). This choice was made based on the following considerations:

- As a rule of thumb, $S_w \approx 2\, d_{\max}$ where $d_{\max}$ is the expected maximum reasonable displacement of a point with respect to the corresponding one in the previous contour (about 10–12 pixels for a 64×64-pixel image, considering that the direction of the displacement is unknown a priori).
- The computation load of the filter is linearly dependent on S_w.

After the edge detector is designed, the edge detection/interpolation process is propagated from the slice where the seed contour was traced to the end of the data set, using the last-calculated contour as a starting point for the next one.

As previously described, if some contours traced on planes that are not parallel to the references are available, further constraints can be introduced to improve the resulting segmentation. This is made by pre-setting the "known" edge points resulting from the intersection of the contours traced on the non-parallel planes with the slices that still have to be segmented. This case is easily accounted for in the elastic model by assigning a priori a large stiffness coefficient to the springs attached to those points.

3. Experimental results

We tested the proposed method both on synthetic and on real medical images. These experiments are described in the following sections.

3.1. Tests on a software-generated image sequence

A software-generated three-dimensional phantom was created. The 3D structure of the phantom is such that its slicing results in a pseudo-tomographic sequence of 64 8-bit greyscale images of size 64×64 pixels which include changes of shape, scale and orientation typical of the contours of anatomical structures, as shown in Fig. 3. The section of the phantom is given grey level 127 on a black (grey level 0) background. The method proposed was applied to the noiseless basic sequence and to two sequences, obtained from the basic one, where zero-mean

[2] One offset is always set equal to 0, as the sample whose corresponding output is being calculated is processed by default.

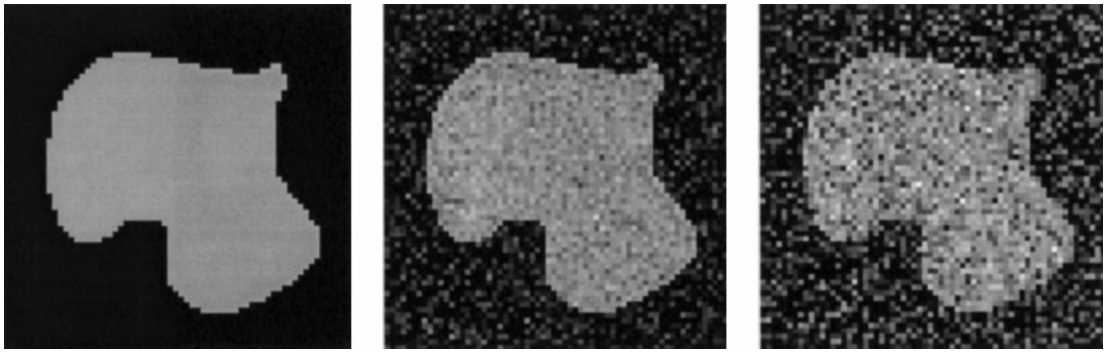

Fig. 4. Slice 32 of the phantom: noiseless (left), with Gaussian noise with $\sigma = 25$ (center), with Gaussian noise with $\sigma = 50$ (right).

Gaussian noise with standard deviation σ of 25 and 50 grey levels was added (see Fig. 4). We will refer to the filters evolved by the GA to segment the phantom with different noise levels as Filt00 (optimized on the noiseless phantom data set $P00$), Filt25 (optimized on the data set $P25$, where noise with $\sigma = 25$ was added to the phantom), and Filt50 (optimized on the data set $P50$, where noise with $\sigma = 50$ was added).

During the application of the GA (training phase), the fitness function was evaluated on three sets of two reference contours extracted from three different levels of the sequence, to ensure good performance on different kinds on inter-slice contour modifications. For each level, a seed contour was traced manually on one image ($k = \{15, 37, 45\}$), along with two contours drawn on the two following images ($k = \{13, 14, 35, 36, 46, 47\}$), which were used as references. Each individual was used to segment the three sets of two reference images, after which its fitness was evaluated by comparing the obtained contours with the references. The total workload for the user was much lower (less than 15%) than the one required by a full manual segmentation.

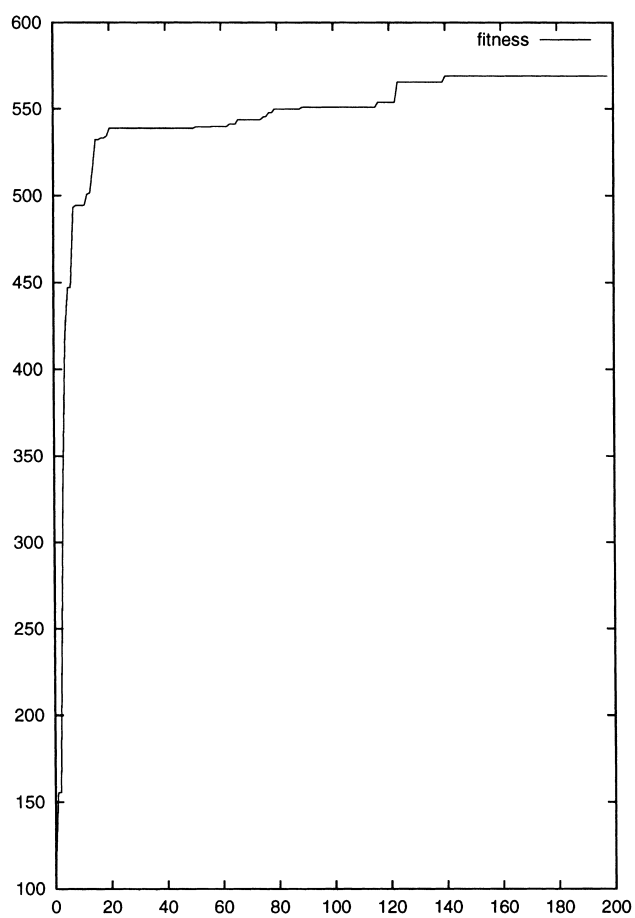

Fig. 5. Plot of best-of-run fitness vs. generations in the experiment in which Filt00 was evolved. A was set to 1500, M was set to 1000.

The GA was run with a population of 200 individuals for 200 generations. One-point crossover (with a probability of 0.6) and mutation operators (with a probability of 0.001) were used. Fitness values (setting $A = 1500$ and $M = 1000$) ranging from around 500 (Filt50) to around 570 (Filt00 and Filt25) were obtained in the experiments. Fig. 5 reports the graph of the best-of-run fitness vs. generations for the experiment in which Filt00 was evolved. In the other two experiments the evolution was quite similar. It is worth noting that, despite the large number of parameters that had to be optimized, and therefore the large number of degrees of freedom of the design, different runs of the GA repeatedly converged to almost the same fitness values.

Table 1 reports the values set by the GA for the parameters of the edge detector in the 3 cases considered (noiseless sequence and noise standard deviation values of 25 and 50), along with the values of the weights for the uncertainty function.

Some observations can be made about the filter coefficients. The constant term and the sum of the coefficients of both the linear and non-linear terms are always negative (except for the linear terms of Filt00[3]). This ensures that, in the presence of a constant input, the output is negative or, at most, null, and therefore lower than the threshold T,

[3] The task to be performed by Filt00 is quite trivial, as it has to detect the same 'clean' edge over and over. Therefore, the huge number of possible solutions for such a problem makes the coefficients found by the GA for Filt00 much less significant for this discussion than the coefficients of the other two filters.

Table 1
Values of the edge detector coefficients set by the GA for the noiseless phantom image sequence and for the same sequence with additive zero-mean Gaussian noise with $\sigma = 25$ and $\sigma = 50$

	$\sigma = 0$	$\sigma = 25$	$\sigma = 50$
c_0	−9.929	−9.817	−5.559
c'_1	−2.374	−0.709	0.625
c'_2	12.187	6.341	−4.771
c'_3	−6.454	−9.691	−3.661
$c''_{1.1}$	0.121	0.427	0.293
$c''_{1.2}$	−0.0993	0.682	0.124
$c''_{1.3}$	−0.103	−0.404	−0.344
$c''_{2.2}$	−0.190	−0.201	−0.127
$c''_{2.3}$	−0.226	−0.928	−0.529
$c''_{3.3}$	−0.119	−0.278	−0.229
d_1	−1	−2	−2
d_2	0	0	0
d_3	4	1	1
T	45.986	2469.818	1316.918
W_1	0.214	0.164	0.213
W_2	0.00187	0.210	0.453
W_3	$3.128 \cdot 10^{-6}$	$2.464 \cdot 10^{-8}$	$5.828 \cdot 10^{-7}$

which is always positive. As the filters are required to produce high outputs in the presence of descending edges (in particular, in the phantom, the grey level of the background is zero), and therefore very low grey-level values are expected to be found after an edge, the coefficients of the linear terms corresponding to "forward" samples (i.e. the samples that follow the one for which the output is computed) are always negative, therefore inhibiting the output as the corresponding input term increases. The fact also that the coefficients of the quadratic terms $c''_{i,j}$ generally tend to become more and more inhibitory (negative), the bigger i and j are, i.e. the more the term refers to forward samples, can be similarly justified. Finally, the absolute values of $c''_{2,3}$ is higher than the other quadratic terms. This means that high input values in the vicinity of the actual sample, i.e., where the edge should be detected (notice that $d_2 = 0$ and $d_3 = 1$), also strongly inhibit the filter output. It should be noticed that the quadratic terms cannot take into consideration the direction of the edge, as they give the same contribution both on the ascending and descending edge. Therefore, the "derivative" behaviour of the filter is mostly given by the linear terms, while the quadratic terms ensure that, in the neighborhood of the edge, low grey level values can be found.

As concerns the weights of the uncertainty function, two observations can be made: the first one is that, with the values set by the GA, the third term of the uncertainty function becomes virtually irrelevant, even if the quantity weighted by W_3 may be two orders of magnitude larger than the ones weighted by W_1 and W_2. The second observation is that W_2 becomes more and more important as noise increases. This can be explained considering that the second term of the uncertainty function somehow represents a regularization term for the contour to be recovered.

The outputs of the edge detectors, obtained by processing an edge of the same slope as the one that is to be detected in the phantom, with and without adding Gaussian noise, are reported in Fig. 6. It can be noticed on the noiseless edge that Filt00 achieves a very precise detection, while Filt25 and Filt50 tend to produce a wider peak. However, in the presence of noise, Filt00 produces a number of spurious peaks, especially with $\sigma = 50$, while Filt25 and Filt50 have virtually the same response, except for a spurious peak produced by Filt25, justified by the presence of an edge-like spike whose slope is quite close (about half as steep) to the one that has to be detected. Such an edge is not detected by Filt50, which shows that the GA correctly evolved filters whose noise-rejection properties increase with noise level in the training images.

A detector having the same structure but based on a standard "Gradient-of-Gaussian" (GoG) filter was also optimized using our genetic design procedure on the $P50$ phantom data set. The same parameters used for the evolution of the polynomial filters were used for the GA. In this case the detector parameters optimized by the GA were six: the three weights of the uncertainty function, a negative threshold T_g (the sign of the GoG was inverted, in order to produce negative peaks on descending edges), the width D of the filter kernel, and the standard deviation σ_g of the Gaussian function. The result of the application of the best GoG filter on the same test signal ($\sigma = 50$) of Fig. 6 is shown in Fig. 7.

It is worth noting that, despite tuning the GoG filter to get a very precise detection of the edge is quite simple, even manually, the filter optimized by the GA tends to select more than one point in the neighborhood of the peak. This behavior, which can also be observed using the polynomial filters, is probably induced by the need of good noise-rejection properties: selecting more than one point in the vicinity of the real edge allows the following elastic-contour interpolation to compensate the effects of false detections more effectively. It also stresses the importance of the combined optimization of the parameters of the filter and of the elastic-contour model. This is probably why the threshold-crossing strategy, that allows only for the detection of one point per crossing, seems to yield worse results (tests not reported).

With the GoG, the optimized values of the weights of the uncertainty function were very different from the ones of the detector based on the polynomial filter. W_1 (0.0048) was about five times smaller, W_2 (0.299) about two times smaller, while W_3 (0.012) was no more negligible. The quantity weighted by W_3 has a much wider range of variability with respect to the other two terms of the uncertainty function. With such values for W_3, it is interesting to observe that edges whose slope is close to the one to be detected make the first two terms of the uncertainty function prevail by an order of magnitude. On the contrary, when the slope of the edge detected is very different from the reference, the third term dominates.

The average fitness values that could be obtained with the

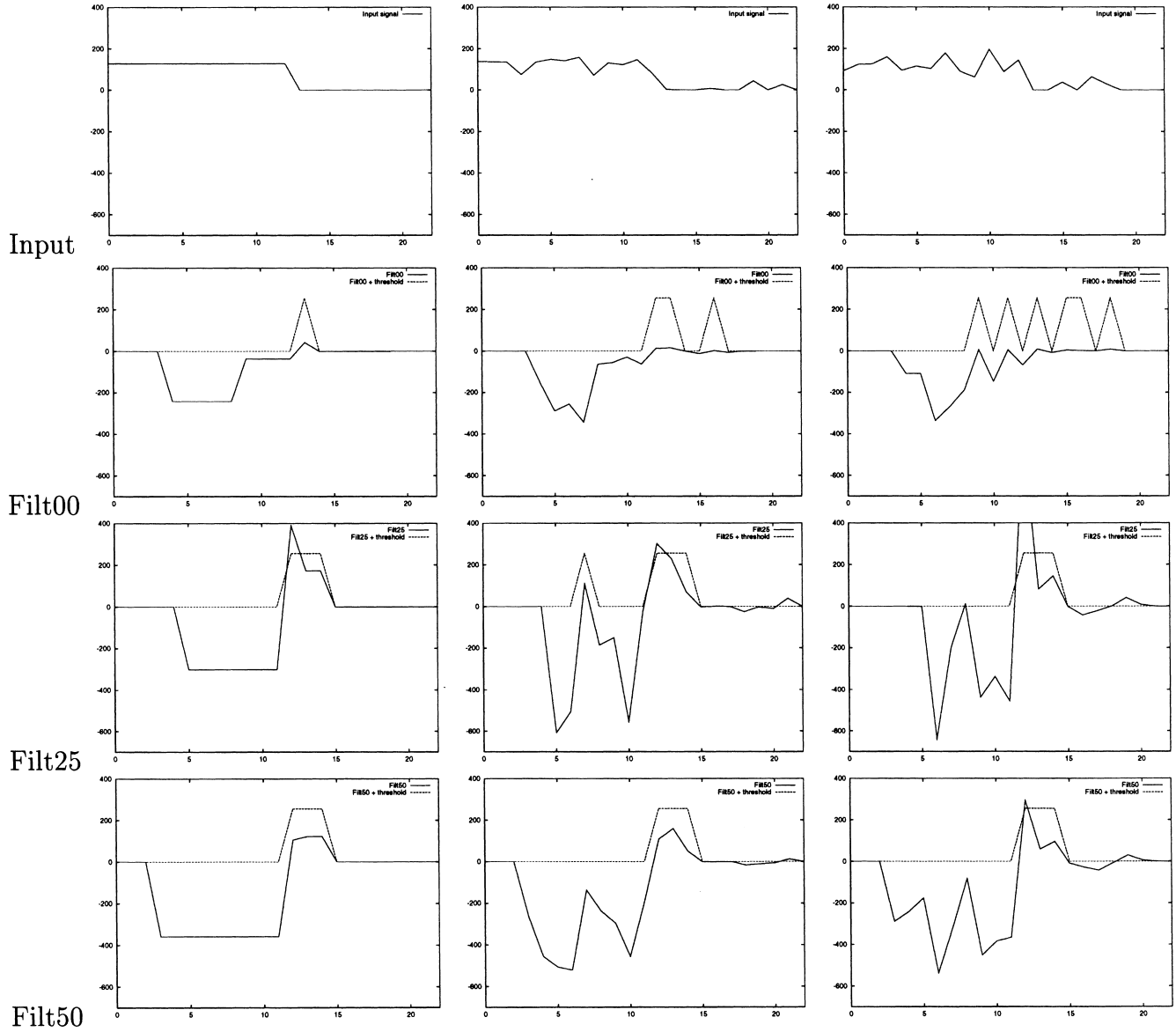

Fig. 6. Results of the application of the filters evolved by the GA for the segmentation of the synthetic phantom. Column 1: noiseless signal; column 2: signal plus noise ($\sigma = 25$); column 3: signal plus noise ($\sigma = 50$). For the sake of readability, filter outputs have been down-scaled by a factor of 40.

GoG were slightly worse than the ones obtained with the polynomial filter. Best fitness values of around 490 were obtained in three runs with respect to values of around 500 consistently obtained in several runs of the GA in designing Filt50. This may indicate that the non-linear part of the polynomial filter actually has a significant role in the detector operation.

This difference was confirmed by the results obtained after the whole sequence was segmented starting from section 45 and proceeding on towards the two sequence ends in two iterations, on which one of the seed contours used in the training phase was traced, as shown by Fig. 8, which reports the RMS error made by the two detectors on each section. It should be observed that the high error values that characterize the first and last sections derive from the fact that, in those sections, the contour "collapses" virtually to one point or one short segment, thus making it difficult, if not impossible, for the elastic model to be defined.

In Fig. 9 the points selected by the polynomial filter and the contour resulting by the elastic-model interpolation are shown for a slice of the $P50$ data set.

These experiments indicate that the method is robust with respect to noise on synthetic structures. In other experiments with synthetic structures (not reported) we noticed that increasing noise level resulted in accuracy decreasing significantly only when the standard deviation of noise is greater than 60–65 grey levels, working with 8-bit greyscale images.

As a further test, we have calculated the volume $\hat{V}_{\text{tot}} = \sum \hat{v}_k$ (where $\hat{v}_k$ is the estimated volume of slice k, i.e., the

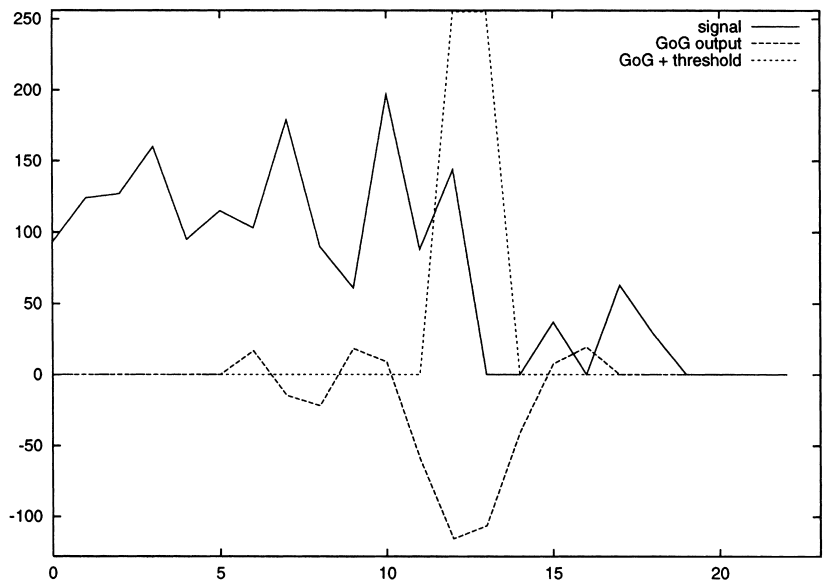

Fig. 7. Results of the application of a Gradient-of-Gaussian filter, optimized on the phantom with Gaussian noise with $\sigma = 50$, to the test signal ($\sigma = 50$). The parameters of the filter are $D = 13$, $T_g = 71$ and $\sigma_g = 1.275$.

number of pixels enclosed within the contour extracted on slice k), and compared it with the actual volume $V_{\text{tot}} = \sum v_k$ of the phantom (v_k is the actual volume of each slice). The following values were also calculated for comparison (N_f is the number of the non-empty slices of the sequence):

- $\Delta V_i = \hat{v}_i - v_i$
- $E = \sum |\Delta V_i|$
- $E\% = 100 \cdot \left(\frac{\sum v_i}{\sum \hat{v}_i} - 1 \right)$
- $\bar{E} = \frac{E}{N_f}$

The results are reported in Table 2, which shows that the volume is slightly overestimated. One reason for this is probably to be found in the assumptions made on contour characteristics, that cause the method to adapt to sudden curvature changes slowly. Another reason for the overestimation may be the choice, adopted while tracing the training contours, of considering the reference contour as the set of pixels including, for each scan line, the first background pixel that is encountered moving outwards from the center of the phantom. Therefore, the GA tended to evolve detectors that produce contours that "wrap" the phantom from outside.

Nonetheless contour detection is very accurate. This can

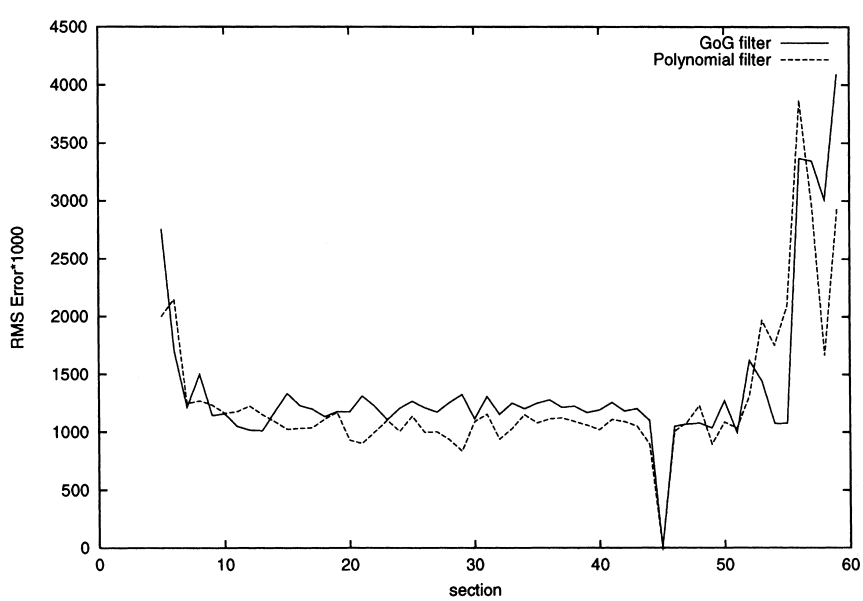

Fig. 8. RMS error vs. section level made by the detectors based on the GoG filter and the polynomial filter in the segmentation of the $P50$ data set.

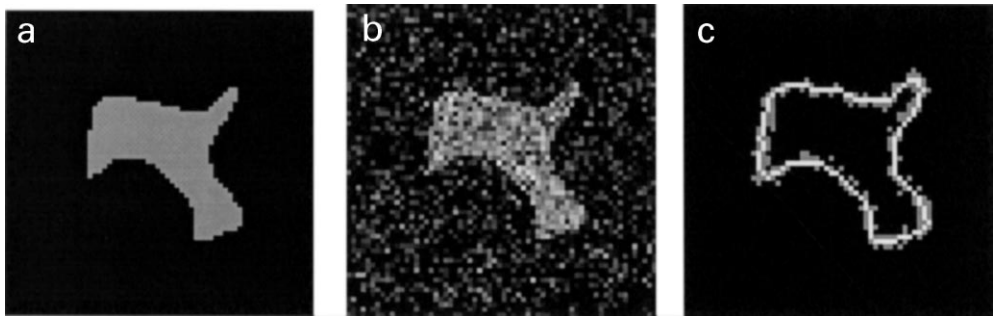

Fig. 9. Segmentation of a slice of the *P*50 dataset: (a) slice 47 of *P*00 (ground truth); (b) slice 47 of *P*50; (c) points selected by the polynomial filter (gray) and extracted contour (white).

be better evidenced with a slice-by-slice analysis. On average, each contour is made up of more than 100 points. As, for each slice, $\bar{E} < 27$ pixels (which is the worst result, obtained with noise standard deviation of 50 grey levels), it can be roughly estimated that, on average, the contour points are dislocated by about one pixel, as can be observed also in Fig. 8.

Finally, we made a 3D reconstruction of the phantom directly from the raw data and compared it to the ones obtained from the contours extracted with the GA-evolved detectors. The 3D reconstructions are shown in Fig. 10. The comparison of the reconstructions obtained from the contours extracted in the presence of different noise levels confirms that the sensitivity of the method to noise is rather small.

3.2. Tests on real images

We have evaluated our method on real images both quantitatively and qualitatively. For quantitative evaluation, the same procedure used for the phantom was applied to a set of real MR images of the brain in which the contours of the brain had to be detected.

This test was inspired by the fact that, besides being essential in radiation treatment-planning to correctly localize brain structures, the accurate detection of the volume occupied by the brain is very important also for automated characterization of brain tissues in MR images, as pointed out in [34]. In fact, grey matter and subcutaneous fat may have similar intensity in images acquired with certain MR imaging techniques.

As the number of images of the real MR sequences was much smaller than in the phantom experiment, the method was applied to one sequence in the training phase, to design the edge detector, and then tested on a similar sequence from a different subject, acquired with the same modalities. The results obtained are also reported in Table 2. These results are even more relevant than the ones obtained on the phantom, as they also demonstrate that the detectors designed using one prototypical case can afterwards be applied to other similar cases, without needing to re-run the design procedure.

As can be seen in the table, contrary to what happened

Table 2
Estimated volume, cumulative absolute error, percent absolute error, average absolute error made in the tests on a noiseless 3D phantom, on the same phantom to which zero-mean gaussian noise with standard deviation of 25 and 50 was added and on a real MR sequence. The actual volume of the phantom is 58708 voxels: the volume of the brain, that had to be segmented in the real MR images, is 9429 voxels

Noise level (σ)	$\sum \hat{v}_i$	E	E%	$\bar{E}$
0	59098	464	0.79	8.44
25	59270	916	1.56	16.65
50	59478	1480	2.52	26.90
MR	9381	238	2.52	23.80

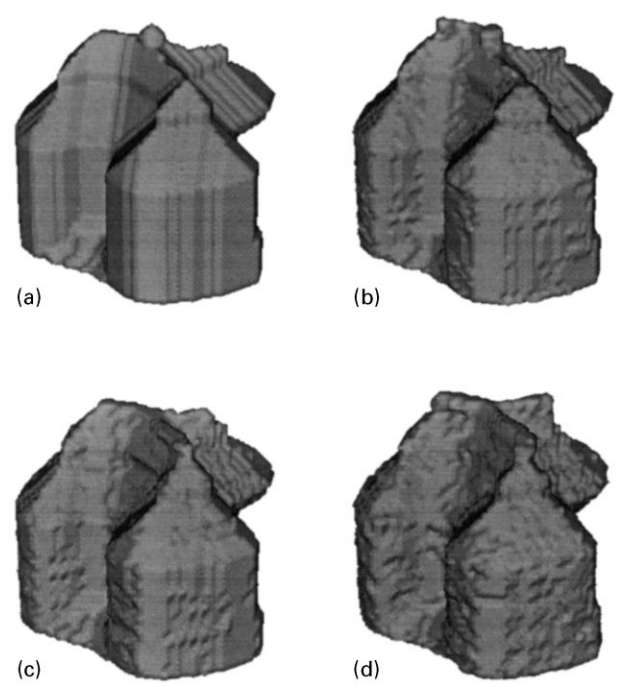

Fig. 10. 3D reconstructions of the software phantom used to test the method from: raw data (a), contours extracted from noiseless sections (b), contours extracted from sections with Gaussian noise with $\sigma = 25$ grey levels (c), contours extracted from sections with Gaussian noise with $\sigma = 50$ grey levels (d).

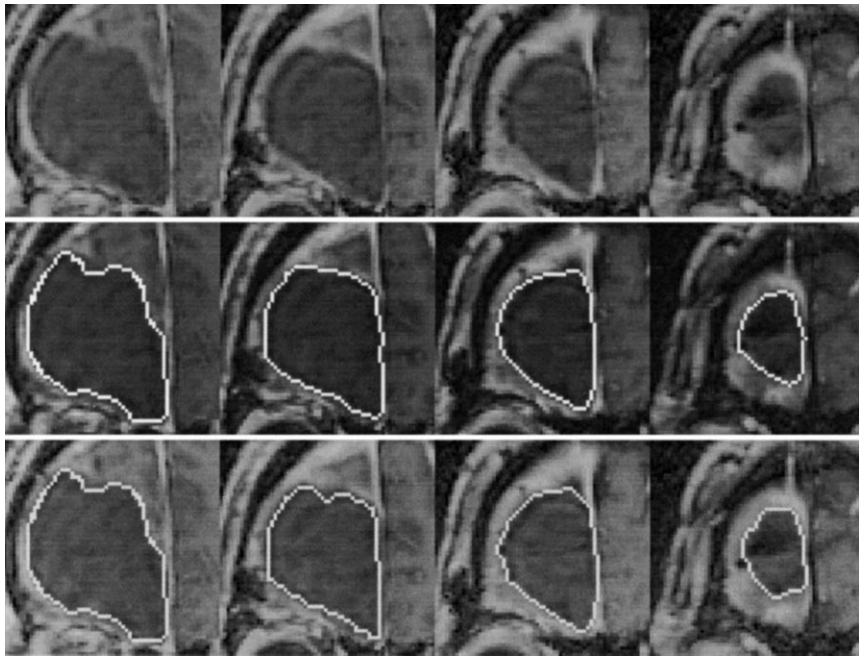

Fig. 11. Four images from an MR sequence of the brain used for our tests (top row). In the mid row, the corresponding segmentations and, in the bottom row, the reference contours traced by the expert were overlaid. Please notice that the image on the left was the starting image, therefore only the corresponding contour traced by the expert is available.

with the phantom, in this test the volume was slightly underestimated by the GA-evolved detector. However, it can be observed that the error made in estimating the volume is significantly smaller than E, which means that an approximately zero-mean distribution of misplacements was obtained for the contour points. This behavior is justified by the fact that the reference contours for the phantom could be precisely traced on the outer border of the phantom, as the boundary between the object and the background could be trivially detected in the noiseless images. On the contrary, the brain contours in the MR images are very fuzzy, and edges much less sharp, caused by both low

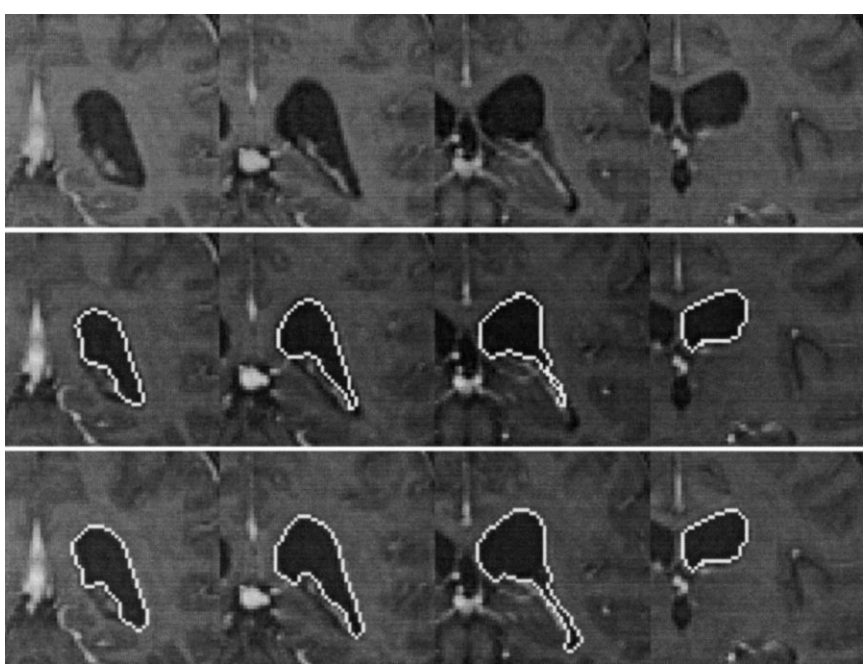

Fig. 12. Four images from an MR sequence of the brain used for our tests (top row). In the mid row, the corresponding segmentations and, in the bottom row, the reference contours traced by the expert were overlaid. Please notice that the image on the right was the starting image, therefore only the corresponding contour traced by the expert is available.

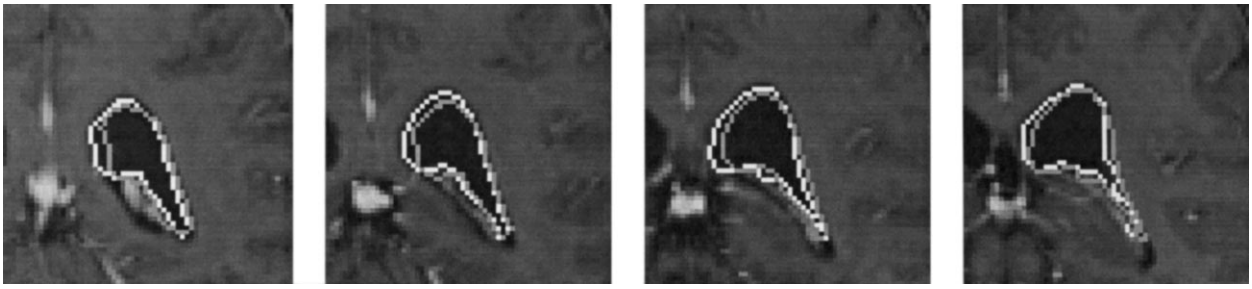

Fig. 13. One of the sequences used in our tests (see also Fig. 12) with the contours extracted overlaid in white and initializations (the contour extracted on the previous slice) overlaid in grey.

resolution and noise, along with other acquisition artifacts such as the so-called "partial-volume effect" [35]. This effect is such that, in tracing the reference contours manually, an inevitable sub-pixel error is committed for each contour point, whose sign can be (almost subliminally) evaluated by the eye of an experienced operator. In this case, the natural tendency of the user is (also almost subliminally) to try and compensate such errors by alternatively over- and under-estimating the location of the contour points.

The good results obtained also on this kind of images, in which the edge profile is much different than in the synthetic phantom, show that the proposed method is quite versatile and can produce detectors with very different behaviors.

Our method has then been qualitatively tested on other sequences of MR images of the brain and on CT images of the chest.

In Figs. 11 and 12 some of the slices belonging to two MR sequences are shown in which a healthy structure (the left brain ventricle) and a pathological one (a brain tumor) were segmented, along with the contours extracted with the proposed method. These results are compared with the contours manually traced by an expert, in order to better

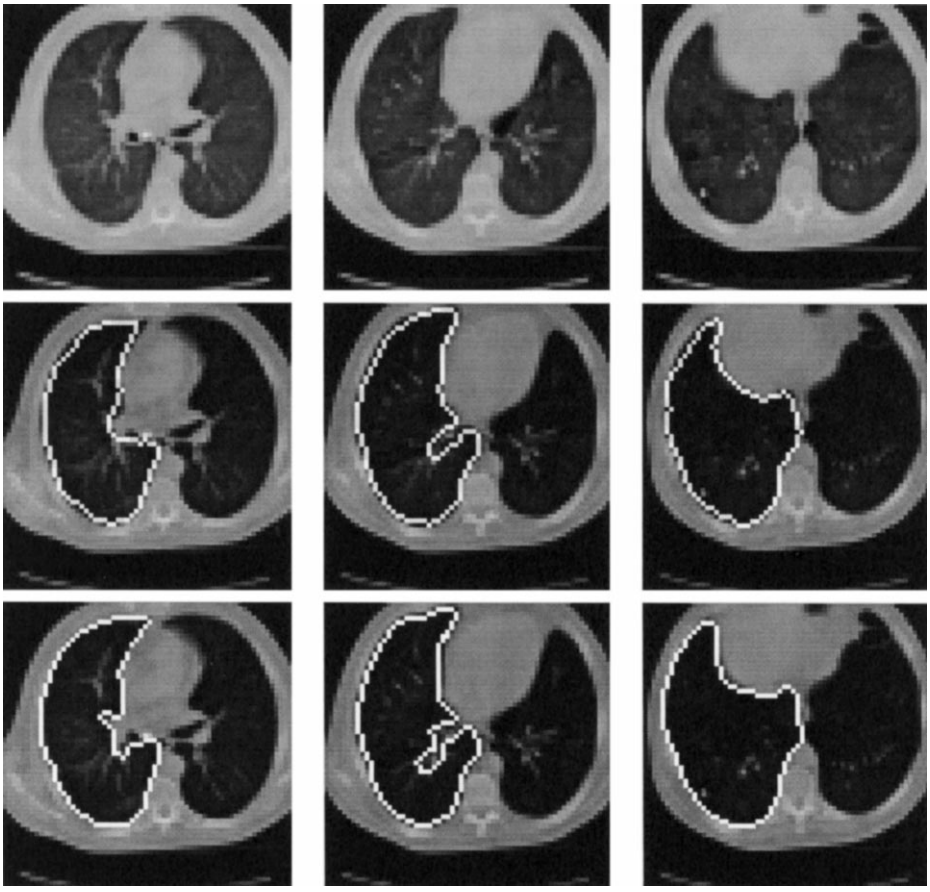

Fig. 14. Results of the segmentation of the right lung (center) performed on a CT sequence of the chest (above), compared with the contours traced by an expert (below).

evaluate their quality. To give an idea of the deformation capabilities of the elastic model, in Fig. 13, we show the same results of Fig. 12 with the initialization of the elastic model overlaid on each image. Similarly, in Fig. 14 we compare the expert's contours with the results obtained on the CT sequence where the structure of interest was the right lung.

Each sequence was derived as a block of consecutive and corresponding regions of interest of size 64 × 64 from a sequence of larger slices (256 × 256) extracted from the 3D data sets with the same tool used to draw the training contours.

The data sets include some of the typical features that usually hamper adaptive-segmentation tasks. Among these are: significant changes of shape across the sequence, the proximity, in the same slice, of edges similar to the ones that have to be detected, and the inter- and intra-slice variability of the characteristics of the boundary. Furthermore, because of the above-mentioned "partial-volume effect" and of the physics of the imaging process, in the case of real images we are no more in the presence of perfect step edges (at most artificially corrupted by Gaussian noise) as in the case of the phantom.

The figures show that our method is quite insensitive to the problems mentioned earlier and perform well also in the presence of rather "smooth" edges. This is because of the selectivity of the GA-designed detector and to the regularization of the contours that is produced by the elastic model, driven by the uncertainty function used to set its parameters. In fact, while limiting the variations between adjacent contours, our method is able to track certain sudden changes of shape between a slice and the following one and to keep away from similar contours that do not belong to the structure of interest.

4. Discussion

The best performance on a given data set can be obtained using the segmenters resulting from the complete filter/detector/elastic-contour design procedure we have described. Unfortunately, the genetic design of the detector may be rather lengthy (a few hours). However, our general architecture can often be simplified to reduce computation time, with very little performance decay. In fact, most computation time is spent to calculate the coefficients of the elastic-contour model. The duration of such a step is roughly proportional to the number of edge points detected by the filter. Such a number can be reduced dramatically if the threshold-crossing selection criterion is adopted. This architecture does not match the performance of the basic one, especially in the presence of noise or of structures with variable edge features. This probably happens because selecting only one point in the vicinity of the real edge does not allow the following elastic-contour interpolation to compensate the effects of a false detection effectively, as mentioned earlier. However, threshold-crossing strategies may reduce computation time by a factor of about four.

As suggested in the introduction, the main goal of the method is to provide users with a versatile and efficient method to support them in tracing contours of a structure of interest interactively, rather than a complex and critical automated segmentation procedure that allows users only to accept or reject its final results. For the sake of evaluation of the intrinsic capabilities of the method, the results reported were obtained directly through automatic segmentation, with no corrective intervention on the part of the user. However, the detectors we use are very simple and demand little in terms of computation load, once the training phase is over (less than 0.5 s per slice, including I/O time, on a PC running Linux and equipped with a 200 Mhz Pentium MMX processor). Thus, even if the results of automatic segmentation are usually correct, the user can be allowed to supervise the process and limit propagation errors by correcting erroneous edge points, possibly re-starting the segmentation from the last-corrected slice, as this operation requires only a few seconds.

One case in which this is a real advantage is when the hypothesis of small and smooth inter-slice changes is incorrect. A possible example is a contour in which a sudden change of curvature, from convex to concave, exists. This situation results in a poor segmentation with the method proposed, if the edges that attract the contour in the first slice move out of the area covered by the corresponding scan line, owing to the convexity change. In that contour segment the elastic model has no forces applied and therefore no local deformation can occur there from one slice to the next one. The results obtained on the TC images appear to be worse than the ones obtained on the MR sequence, for the presence of several of such sudden convexity changes, that are often smoothed out by the elastic contour model. No other relevant problems were encountered in the tests.

5. Conclusions

In this article a method for computer-assisted segmentation of anatomical structures in 3D medical-image sets is described along with some results obtained with the proposed approach.

The method derives segmentation filters and strategies for their application from features extracted from data available along a few contours drawn by the end-user. This makes it suitable for use in computer-aided diagnostic systems that, besides the choice among a library of previously defined or designed segmenters, must offer users the opportunity to design new ones, optimally tuned to specific problems which are not routinely encountered in the clinical practice. Once they have been designed, such new detectors can be added to the library and become of immediate availability in the future.

As the optimization of a simple and efficient filter layout produced by the GA yields computationally-light segmenters, this allows for user supervision of the segementation process. In fact, allowing the user to introduce minor manual corrections does not delay the process significantly. One more feature that makes the proposed method particularly suitable for inclusion in interactive systems is that contours can be specified in 3D space along arbitrary cutting planes, allowing for effective exploration and analysis of 3D data sets.

The rather encouraging results reported may be further improved, as the method constitutes a framework within which a number of variants can be devised: besides the above mentioned modification to reduce computation time, other extensions could consider the use of more powerful 1D and 2D filter layouts and, with minor changes, 3D filters. In this case, elastic 3D surface models like the ones described in [10,36] could be used instead of the 2D elastic-contour model.

Acknowledgements

This work was partially supported by a grant under the British Council-MURST/CRUI agreement. Stefano Cagnoni was supported by Italian National Research Council grants while visiting the Whitaker College Biomedical Imaging and Computation Laboratory at MIT.

References

[1] J.K. Udupa, Display of 3-D information in discrete 3-D scenes produced by computerized tomography, Proceedings of IEEE 71 (1983) 420–431.
[2] R.A. Robb, C. Bariollot, Interactive display and analysis of 3-D medical images, IEEE Transactions on Medical Imaging 8 (1989) 217–226.
[3] J. Ylä-Jääski, F. Klein, O. Kübler, Fast direct display of volume data for medical diagnosis, Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing 53 (1) (1991) 7–18.
[4] S. Liou, R.C. Jain, An approach to three-dimensional image segmentation, Computer Vision, Graphics and Image Processing: Image Understanding 53 (3) (1991) 237–252.
[5] I. Cohen, L.D. Cohen, N. Ayache, Using deformable surfaces to segment 3-D images and infer differential structures, Computer Vision, Graphics, and Image Processing: Image Understanding 56 (2) (1992) 242–263.
[6] R. Deklerck, J. Cornelis, M. Bister, Segmentation of medical images, Image and Vision Computing 11 (8) (1993) 486–503.
[7] M. Joliot, B.M. Mazoyer, Three-dimensional segmentation and interpolation of magnetic resonance brain image, IEEE Transactions on Medical Imaging 12 (2) (1993) 269–277.
[8] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, International Journal of Computer Vision 1 (1987) 321–331.
[9] D. Terzopoulos, Dynamic 3D models with local and global deformations: deformable superquadrics, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (1987) 703–714.
[10] R. Poli, G. Coppini, G. Valli, Recovery of 3D surfaces from sparse data, Computer Vision Graphics and Image Processing: Image Understanding 60 (1) (1994) 1–24.
[11] G. Coppini, R. Poli, G. Valli, Recovery of the 3D shape of the left ventricle from echocardiographic images, IEEE Transactions on Medical Imaging MI-14 (1995) 301–317.
[12] H. Eviatar, R.L. Somorjai, A fast simple active contour algorithm for biomedical images, Pattern Recognition Letters 17 (1996) 969–974.
[13] M. Demi, New approach to automatic contour detection from image sequences: an application to ventriculographic images, Computers and biomedical research 27 (1994) 157–177.
[14] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[15] J. Holland, Adaptation in Natural and Artificial Systems, 2nd ed., MIT Press, Cambridge, Massachusetts, 1992.
[16] S.M. Bhandarkar, Y. Zhang, W.D. Potter, An edge detection technique using genetic algorithm-based optimization, Pattern Recognition 27 (9) (1994) 1159–1180.
[17] S. Shafer, T. Kanade, Recursive region segmentation by analysis of histograms, Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, 1982, pp. 1166–1171.
[18] B. Bhanu, S. Lee, Genetic learning for adaptive image segmentation, Kluwer Academic Publishers, Boston, 1994.
[19] D.N. Chun, H.S. Yang, Robust image segmentation using genetic algorithm with a fuzzy measure, Pattern recognition 29 (7) (1996) 1195–1211.
[20] K. Delibasis, P.E. Undrill, G.G. Cameron, Genetic algorithms applied to fourier descriptor based geometric models for anatomical object recognition in medical images, Computer Vision and Image Understanding 66 (9) (1997) 286–300.
[21] R. Poli, G. Valli, Neural inhabitants of MR and echo images segment cardiac structures, Computers in Cardiology, IEEE Computer Society Press, London, 1993, pp. 193–196.
[22] J.R. Koza, Genetic Programming: On the Programming of Computers by Natural Selection, MIT Press, Cambridge, MA, USA, 1992.
[23] R. Poli, Genetic programming for image analysis, in: John R. Koza, David E. Goldberg, David B. Fogel, Rick L. Riolo (Eds.), Genetic Programming 96, Proc. of the First Annual Conference, Stanford University, MIT press, USA, 1996, pp. 363–368.
[24] C. Bounsaythip, J.T. Alander, Genetic algorithms in image processing – a review, Proc. of the 3rd Nordic Workshop on Genetic Algorithms and their Applications, Metsatalo, University of Helsinki, Helsinki, Finland, 1997, pp. 173–192.
[25] D. Suckley, Genetic algorithm in the design of FIR filters, IEE Proceedings-G 138 (1991) 234–238.
[26] R. Nambiar, C.K.K. Tang, P. Mars, Genetic and learning automata algorithms for adaptive digital filters, Proc. ICASSP-92, IEEE 4 (1992) 41–44.
[27] R. Poli, S. Cagnoni, G. Valli, Genetic design of optimum linear and non-linear QRS detectors, IEEE Trans. On Biomed. Engineering 42 (11) (1995) 1137–1141.
[28] S. Cagnoni, A.B. Dobrzeniecki, J.C. Yanch, R. Poli, Interactive segmentation of multi-dimensional medical data with contour-based application of genetic algorithms, Proc. of First IEEE International Conference on Image Processing, Vol. 3, New York, 1994, pp. 498–502.
[29] S. Cagnoni, A.B. Dobrzeniecki, R. Poli, J.C. Yanch, Optimized contour-based segmentation of 3D medical data using genetic algorithms, Tech. Rep. WCBICL-9501, MIT Whitaker College Biomedical Imaging and Computation Laboratory, 1995.
[30] S. Cagnoni, A.B. Dobrzeniecki, R. Poli, J.C. Yanch, Segmentation of 3D medical images through genetically-optimized contour-tracking algorithms, Tech. Rep. CSRP-97-28, School of Computer Science, The University of Birmingham, 1997.
[31] A.B. Dobrzeniecki, N. Levitt, Interactive and intuitive segmentation of volumetric data: the Segment VIEW system and the Kooshball algorithm, Proc. of 2nd IEEE International Conference on Image Processing, New York, 1995.

[32] P. Radeva, J. Serrat, E. Marti, A snake for model-based segmentation, Proc. of International Conference on Computer Vision (ICCV95), Cambridge, MA, 1995.

[33] N.N. Schraudolph, J.J. Grefenstette, A user's guide to GAUCSD 1.4, Tech. Rep. CS92-249, Computer Science and Engineering Department, University of California, San Diego, La Jolla, CA 92093-0114, 1992.

[34] S. Cagnoni, G. Coppini, M. Rucci, D. Caramella, G. Valli, Neural network segmentation of magnetic resonance spin echo images of the brain, Journal of Biomedical Engineering 15 (9) (1993) 355–362.

[35] S. Webb, The physics of medical imaging, Institute of Physics Publishing, Bristol and Philadelphia, 1991.

[36] R. Poli, G. Valli, Shape from radiological density, Computer Vision and Image Understanding 65 (3) (1997) 361–381.