Hybrid Image Segmentation Using Watersheds and Fast Region Merging

Kostas Haris, Serafim N. Efstratiadis, *Member, IEEE*, Nicos Maglaveras, *Member, IEEE*,
and Aggelos K. Katsaggelos, *Fellow, IEEE*

数据结构

Abstract—A hybrid multidimensional image segmentation algorithm is proposed, which combines edge and region-based techniques through the morphological algorithm of watersheds. An edge-preserving statistical noise reduction approach is used as a preprocessing stage in order to compute an accurate estimate of the image gradient. Then, an initial partitioning of the image into primitive regions is produced by applying the watershed transform on the image gradient magnitude. This initial segmentation is the input to a computationally efficient hierarchical (bottom-up) region merging process that produces the final segmentation. The latter process uses the region adjacency graph (RAG) representation of the image regions. At each step, the most similar pair of regions is determined (minimum cost RAG edge), the regions are merged and the RAG is updated. Traditionally, the above is implemented by storing all RAG edges in a priority queue. We propose a significantly faster algorithm, which additionally maintains the so-called nearest neighbor graph, due to which the priority queue size and processing time are drastically reduced. The final segmentation provides, due to the RAG, one-pixel wide, closed, and accurately localized contours/surfaces. Experimental results obtained with two-dimensional/three-dimensional (2-D/3-D) magnetic resonance images are presented.

Index Terms— Image segmentation, nearest neighbor region merging, noise reduction, watershed transform.

I. INTRODUCTION

IMAGE segmentation is an essential process for most subsequent image analysis tasks. In particular, many of the existing techniques for image description and recognition [1], [2], image visualization [3], [4], and object based image compression [5]–[7] highly depend on the segmentation results. The general segmentation problem involves the partitioning of a given image into a number of homogeneous segments (spatially connected groups of pixels), such that the union of any two neighboring segments yields a heterogeneous

异构

segment. Alternatively, segmentation can be considered as a pixel labeling process in the sense that all pixels that belong to the same homogeneous region are assigned the same label. There are several ways to define homogeneity of a region based on the particular objective of the segmentation process. However, independently of the homogeneity criteria, the noise corrupting almost all acquired images is likely to prohibit the generation of error-free image partitions [8].

Many techniques have been proposed to deal with the image segmentation problem [9], [10]. They can be broadly grouped into the following categories.

Histogram-Based Techniques: The image is assumed to be composed of a number of constant intensity objects in a well-separated background. The image histogram is usually considered as being the sample probability density function (pdf) of a Gaussian mixture and, thus, the segmentation problem is reformulated as one of parameter estimation followed by pixel classification [10]. However, these methods work well only under very strict conditions, such as small noise variance or few and nearly equal size regions. Another problem is the determination of the number of classes, which is usually assumed to be known. Better results have been obtained by the application of spatial smoothness constraints [11].

Edge-Based Techniques: The image edges are detected and then grouped (linked) into contours/surfaces that represent the boundaries of image objects [12], [13]. Most techniques use a differentiation filter in order to approximate the first-order image gradient or the image Laplacian [14], [15]. Then, candidate edges are extracted by thresholding the gradient or Laplacian magnitude. During the edge grouping stage, the detected edge pixels are grouped in order to form continuous, one-pixel wide contours as expected [16]. A very successful method was proposed by Canny [15] according to which the image is first convolved by the Gaussian derivatives, the candidate edge pixels are isolated by the method of nonmaximum suppression and then they are grouped by hysteresis thresholding. The method has been accelerated by the use of recursive filtering [17] and extended successfully to 3D images [18]. However, the edge grouping process presents serious difficulties in producing connected, one-pixel wide contours/surfaces.

Region-Based Techniques: The goal is the detection of regions (connected sets of pixels) that satisfy certain predefined homogeneity criteria. In region-growing or merging techniques, the input image is first tessellated into a set of homogeneous primitive regions. Then, using an iterative merging

process, similar neighboring regions are merged according to a certain decision rule [12], [19]–[21]. In splitting techniques, the entire image is initially considered as one rectangular region. In each step, each heterogeneous image region of the image is divided into four rectangular segments and the process is terminated when all regions are homogeneous. In split-and-merge techniques, after the splitting stage a merging process is applied for unifying the resulting similar neighboring regions [22], [23]. However, the splitting technique tends to produce boundaries consisting of long horizontal and vertical segments (i.e., distorted boundaries). The heart of the above techniques is the region homogeneity test, usually formulated as a hypothesis testing problem [23], [24].

Markov Random Field-Based Techniques: The true image is assumed to be a realization of a Markov or Gibbs random field with a distribution that captures the spatial context of the scene [25]. Given the prior distribution of the true image and the observed noisy one, the segmentation problem is formulated as an optimization problem. The commonly used estimation principles are *maximum a posteriori* (MAP) estimation, maximization of the marginal probabilities (ICM) [26] and maximization of the posterior marginals [27]. However, these methods require fairly accurate knowledge of the prior true image distribution and most of them are quite computationally expensive.

混合技术 *Hybrid Techniques:* The aim here is offering an improved solution to the segmentation problem by combining techniques of the previous categories. Most of them are based on the integration of edge- and region-based methods. In [20], the image is initially partitioned into regions using surface curvature-sign and, then, a variable-order surface fitting iterative region merging process is initiated. In [28], the image is initially segmented using the region-based split-and-merge technique and, then, the detected contours are refined using edge information. In [29], an initial image partition is obtained by detecting ridges and troughs in the gradient magnitude image through maximum gradient paths connecting singular points. Then, region merging is applied through the elimination of ridges and troughs via similarity/dissimilarity measures.

The algorithm proposed in this paper belongs to the category of hybrid techniques, since it results from the integration of edge- and region-based techniques through the morphological watershed transform. Many morphological segmentation approaches using the watershed transform have been proposed in the literature [30], [31]. Watersheds have also been used in multiresolution methods for producing resolution hierarchies of image ridges and valleys [3], [32]. Although these methods were successful in segmenting certain classes of images, they require significant interactive user guidance or accurate prior knowledge on the image structure. By improving and extending earlier work on this problem [8], [33], [34], the proposed algorithm delivers accurately localized, one pixel wide and closed object contours/surfaces while it requires a small number of input parameters (semiautomatic segmentation). Initially, the noise corrupting the image is reduced by a novel noise reduction technique that is based on local homogeneity testing followed by local classification [35]. This technique is applied to the original image and preserves edges remarkably

well, while reducing the noise quite effectively. At the second stage, this noise suppression allows a more accurate calculation of the image gradient and reduction of the number of the detected false edges. Then, the gradient magnitude is input to the watershed detection algorithm, which produces an initial image tessellation into a large number of primitive regions [31]. This initial oversegmentation is due to the high sensitivity of the watershed algorithm to the gradient image intensity variations, and, consequently, depends on the performance of the noise reduction algorithm. Oversegmentation is further reduced by thresholding the gradient magnitude prior to the application of the watershed transform. The output of the watershed transform is the starting point of a bottom-up hierarchical merging approach, where at each step the most similar pair of adjacent regions is detected and merged. Here, the region adjacency graph (RAG) is used to represent the image partitions and is combined with a newly introduced nearest neighbor graph (NNG), in order to accelerate the region merging process. Our experimental results indicate a remarkable acceleration of the merging process in comparison to the RAG based merging. Finally, a merging stopping rule may be adopted for unsupervised segmentation.

In Section II, the segmentation problem is formulated and the algorithm outline is presented. In Section III, a novel edge-preserving noise reduction technique is presented as a preprocessing step, followed by the proposed gradient approximation method. In Section IV, the watershed algorithm used and an oversegmentation reduction technique are briefly described. In Section V, the proposed accelerated bottom-up hierarchical merging process is presented and analyzed. Results are presented in Section VI on two-dimensional/three-dimensional (2-D/3-D) synthetic and real magnetic resonance (MR) images. Finally, conclusions and possible extensions of the algorithm are discussed in Section VII.

II. PROBLEM FORMULATION AND ALGORITHM OUTLINE

Let $L = \{0, 1, \cdots, L_m\}$ be the set of intensities and $S = \{(x, y) : 1 \leq x \leq N_c, 1 \leq y \leq N_r\}$ be the spatial coordinates of a pixel in a N_r-row by N_c-column image. The $m \times n$ neighborhood of pixel $p = (x, y)$ over S is defined as follows:

$$\mathcal{N}_{m \times n}(p) = \{q = (z, w) \in S : |x - z| \leq \lfloor m/2 \rfloor, |y - w| \leq \lfloor n/2 \rfloor\}$$

where m, n are odd and $\lfloor \cdot \rfloor$ denotes the largest integer not greater than its argument. In the 3-D case, the neighborhood of point $p = (x, y, z)$ is defined in a similar way. In our formulation, it is assumed that the true image X is corrupted by additive independent identically distributed Gaussian noise. Hence, the observed image Y is defined as follows

$$Y(p) = X(p) + n(p), \quad p \in S \tag{1}$$

where $n(p) \sim N(0, \sigma^2)$ is the zero-mean Gaussian distribution with standard deviation σ. It is also assumed that the true image is piecewise constant. More specifically, there is a partition of S, namely $\Delta_{K^*}(S) = \{R_1, R_2, \cdots, R_{K^*}\}$, for

some natural number K^*, such that

$$S = \bigcup_{k=1}^{K^*} R_k,$$

$$R_i \cap R_j = \emptyset, \quad \forall i,j \in \{1,2,\ldots,K^*\}, \quad \text{for } i \neq j,$$
$$R_i, \forall i \in \{1,2,\ldots,K^*\}, \text{ is connected,} \qquad (2)$$
$$X(p) = C_m, \quad \text{if } p \in R_m, \quad \forall m \in \{1,2,\cdots,K^*\}$$

where C_m is a constant, and

$$C_m \neq C_n, \quad \text{if } R_m \text{ and } R_n \text{ are adjacent.} \qquad (3)$$

It is reminded that two regions are adjacent if they share a common boundary, that is, if there is at least one pixel in one region, such that, its 3×3 neighborhood contains at least one pixel belonging to the other region. According to the above formulation, the output of the image segmentation algorithm should be the image partition $\Delta_{K^*}(S)$. In addition, it is assumed that small pixel neighborhoods contain either one (homogeneous) or two (heterogeneous) regions [8], [35].

Fig. 1 shows the stages of the proposed segmentation algorithm. The aim of the first stage is the reduction of the noise corrupting the image while preserving its structure, based on the above homogeneity/heterogeneity assumption for the image regions. The proposed noise reduction technique is applied locally by processing the neighborhood of each pixel separately. The underlying idea is estimating the true pixel intensity by detecting the presence (or absence) of image structure (homogeneity versus heterogeneity) and by applying the appropriate estimation technique, as explained in Section III. At the second stage, the gradient of the smoothed image is calculated using the Gaussian filter derivatives with a small scale since the noise has already been substantially reduced at the first stage. Then, the gradient magnitude is calculated and thresholded appropriately. At the next stage, the resulting gradient magnitude is passed on to the watershed detection algorithm, which produces an initial image partition, namely $\Delta_{K_0}(S)$. It is assumed that $K_0 \geq K^*$ and that there exists a sequence of region merges that transforms $\Delta_{K_0}(S)$ to the true partition $\Delta_{K^*}(S)$. In other words, each region $R_m^{K^*}$ of Δ_{K^*} is assumed to be the union of a number of Δ_{K_0} regions. At the final stage, a novel fast region-merging process is applied to the most similar pair of adjacent regions at each step. The merging process may be terminated either interactively or with the use of a given stopping rule based on hypothesis testing.

III. NOISE REDUCTION AND GRADIENT COMPUTATION

At the first stage of the segmentation algorithm, the following smoothing technique is proposed. For a given pixel p of the observed image Y, its square (or cubic in 3-D imaging) neighborhood $\mathcal{N}_{n \times n}(p)$, for odd n, is considered the support based on which a binary decision about the presence or absence of homogeneity must be reached. A homogeneous $\mathcal{N}_{n \times n}(p)$ is considered to be a sample of size $N = n \times n$ of a Gaussian random variable with mean μ and variance σ^2. A heterogeneous $\mathcal{N}_{n \times n}(p)$ is considered to be a sample of size N of a random variable following the distribution of a mixture

TABLE I
VALUES OF PARAMETER C FOR VARIOUS NEIGHBORHOOD
SIZES AND VALUES OF THE SIGNIFICANCE LEVEL α

NEIGHBORHOOD SIZE	PARAMETER C		
$n \times n$	$\alpha = 0.05$	$\alpha = 0.025$	$\alpha = 0.001$
3×3	0.7230	0.9483	1.2322
5×5	0.4566	0.5746	0.7192
7×7	0.3265	0.4286	0.5102
9×9	0.2577	0.3164	0.3868
11×11	0.2181	0.2654	0.3553
13×13	0.1837	0.2228	0.2974
15×15	0.1787	0.1920	0.2557
17×17	0.1397	0.1688	0.2243
19×19	0.1247	0.1505	0.2000
$3 \times 3 \times 3$	0.4402	0.5527	0.6905
$5 \times 5 \times 5$	0.2154	0.2609	0.3492
$7 \times 7 \times 7$	0.1282	0.1546	0.2051

of two Gaussian distributions with prior probabilities P_i, mean values μ_i and common variance $\sigma_i^2 = \sigma^2$, for $i = 0, 1$. The probability density function of the mixture is given by

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{i=0}^{1} P_i \exp\left\{-\frac{(x-\mu_i)^2}{2\sigma^2}\right\}. \qquad (4)$$

According to the above formulation, the maximum likelihood (ML) ratio test gives

$$\mathcal{N}_{n \times n}(p) \text{ is homogeneous,} \quad \text{if } \hat{S}^2 \leq (1+C)\sigma^2 \qquad (5)$$

where $\hat{S}^2$ is the sample variance of $\mathcal{N}_{n \times n}(p)$, that is,

$$\hat{S}^2 = \frac{1}{N} \sum_{q \in \mathcal{N}_{n \times n}(p)} \left(Y(q) - \frac{1}{N}\sum_{r \in \mathcal{N}_{n \times n}(p)} Y(r)\right)^2.$$

Parameter C is determined by the significance level of the test (i.e., the probability of wrongly accepting homogeneity), based on the fact that the random variable $N\hat{S}^2/\sigma^2$ is distributed according to χ^2_{N-1}, under the homogeneity hypothesis. Table I gives the values of parameter C for various neighborhood sizes and significance levels. If $\mathcal{N}_{n \times n}(p)$ is decided to be homogeneous, then the true value of pixel p is estimated by the sample mean of $\mathcal{N}_{n \times n}(p)$, which is the best estimator (unbiased and minimum variance) in the case of Gaussian noise, namely,

$$\hat{X}(p) = \frac{1}{N} \sum_{q \in \mathcal{N}_{n \times n}(p)} Y(q). \qquad (6)$$

If $\mathcal{N}_{n \times n}(p)$ is decided to be heterogeneous, then the unknown mixture parameters P_i, μ_i in (4) are estimated and used in calculating the threshold T_c. Then, the central pixel neighborhood is classified into one of the two mixture components and, therefore, the intensity estimation is given by

$$\hat{X}(p) = \begin{cases} \hat{\mu}_1, & \text{if } Y(p) > T_c \\ \hat{\mu}_0, & \text{otherwise} \end{cases} \qquad (7)$$

where $\hat{\mu}_i$ is an estimate of μ_i. The value of T_c is calculated according to the following formula:

$$T_c = \frac{\hat{\mu}_0 + \hat{\mu}_1}{2} + \frac{\sigma^2}{\hat{\mu}_1 - \hat{\mu}_0} \ln\frac{\hat{P}_0}{\hat{P}_1} \qquad (8)$$

where $\hat{P}_i$ is an estimate of P_i. In [8], [35] it was shown that when the signal-to-noise ratio (SNR) $\rho = \frac{|\mu_0 - \mu_1|}{\sigma}$ is greater than two, the algorithm reduces noise quite efficiently while preserving the image structure. The mixture parameters are estimated by the method of moments using the three first sample moments [8], [35]. The closed-form estimators are

$$\hat{\mu}_i = \frac{1}{2}[\beta - (-1)^i \sqrt{\beta^2 - 4\gamma}], \tag{9}$$

$$\hat{P}_i = (-1)^i \frac{\hat{\mu}_{\bar{i}} - c_1}{\hat{\mu}_{\bar{i}} - \hat{\mu}_i}, \quad \text{for } i = 0, 1 \tag{10}$$

where $\bar{i} = \text{mod}(i + 1, 2)$,

$$\beta = \frac{c_3 - c_1 c_2}{c_2 - c_1^2}, \quad \gamma = \frac{c_1 c_3 - c_2^2}{c_2 - c_1^2},$$

$$c_1 = m_1, \quad c_2 = m_2 - \sigma^2, \quad c_3 = m_3 - 3m_1\sigma^2,$$

and m_j is the jth order sample moment of $\mathcal{N}_{n \times n}(p)$, for $j = 1, 2, 3$. Experimental comparisons of the moment estimator with the ML estimator have shown that, when the classes are well-separated, the estimators yield nearly identical estimates [8]. Provided that the original image follows the adopted piecewise constant model and the noise is above a certain level, the performance of the proposed noise reduction method is superior to that of other methods, such as linear filtering, median filtering and anisotropic diffusion [36]. The performance of the noise reduction stage depends on the accurate estimation of the noise variance σ^2 in the observed image. Several noise variance estimation methods have been proposed in the literature [37]. Also, the noise reduction stage depends on the value of parameter N which may be user defined (rule of thumb) or can be evaluated based on the estimated noise variance.

At the second stage of the segmentation algorithm, the gradient field of the smoothed image $\nabla \hat{X}(p)$ is computed. Among the known gradient operators, namely, classical (Sobel, Prewitt), Gaussian or morphological, the Gaussian derivatives have been extensively studied in the literature [12]. Provided that the original noise level is not high or the noise has been effectively reduced in the first stage, then all the above operators may perform well. However, if the original noise level is high or the noise has not been effectively reduced in the first stage, the use of small scale Gaussian derivative filters may further reduce noise. Finally, the gradient magnitude image $\|\nabla \hat{X}(p)\|$ is calculated and used as described in the next section.

IV. INITIAL SEGMENTATION USING WATERSHEDS

In this stage, an initial image partition into primitive regions is obtained using the image gradient magnitude. As mentioned in Section I, edge-based segmentation algorithms operate by detecting edges using the intensity gradient and, then, by grouping them in order to form contours/surfaces. Edge detection is typically based on image gradient processing, e.g., nonmaximum suppression and magnitude thresholding. Then, image pixels are labeled as either edge or nonedge ones [15], [18]. However, since the latter labeling decisions are local,

for each pixel there is a probability of either 1) accepting a nonedge pixel as an edge one, or 2) rejecting an edge pixel as nonedge one. The first type of error corresponds to the detection of false edges due to noise, whereas, the second one to the partial grouping of edges which results in contours being broken into small edge groups (edgels) separated by gaps. The above errors do not allow the generation of the contour representations required by higher level analysis, namely, one-pixel wide closed curves.

A different approach to image segmentation, which overcomes the problem of disconnected contours and false edges, is the application of the morphological watershed transform to the gradient magnitude image [30], [31]. This approach allows the generation of an initial image partition into regions and, consequently, region-based techniques can be used in order to produce closed, one pixel-wide contours/surfaces. In the following, we briefly describe the fast watershed detection algorithm proposed by Vincent and Soille [31].

Let X be a greyscale digital image. Watersheds are defined as the lines separating the so-called catchment basins, which belong to different minima. More specifically, a minimum M at intensity level (altitude) h in X is a connected set of pixels with intensity h, such that it is impossible to reach a pixel of intensity h' without having to pass from a pixel of intensity h'', where $h' < h < h''$. The catchment basin $C(M)$ associated with the minimum M is a set of pixels, such that, if a drop of water falls at any pixel in $C(M)$, then it will flow down to the minimum M. The watersheds computation algorithm used here is based on immersion simulations [31], that is, on the recursive detection and fast labeling of the different catchment basins using queues. The algorithm consists of two steps: sorting and flooding. At the first step, the image pixels are sorted in increasing order according to their intensities. Using the image intensity histogram, a hash table is allocated in memory, where the i-th entry points to a list containing the image locations of intensity i. Then, this hash table is filled by scanning the image. Therefore, sorting requires scanning the image twice using only constant memory. At the flooding step, the pixels are quickly accessed in increasing intensity order (immersion) using the sorted image and labels are assigned to catchment basins. The label propagation is based on queues constructed using neighborhoods [31].

The output of the watersheds algorithm is a tessellation of the input image into its different catchment basins, each one characterized by a unique label. Among the image watershed points, only these located exactly half-way between two catchment basins are given a special label [31]. In order to obtain the final image tessellation, the watersheds are removed by assigning their corresponding points to the neighboring catchment basins.

The input to the watersheds algorithm is the gradient magnitude image $G(p) = \|\nabla \hat{X}(p)\|$ computed at the previous stage, as described in Section III. The high sensitivity of the watersheds algorithm to noise yields a very large number of catchment basins, leading to oversegmentation. Fig. 9(b) illustrates an oversegmentation case resulting from applying the watershed detection algorithm to the MR image shown in Fig. 7 (right). Earlier attempts in oversegmentation reduction

一种不同的图像分割方法，克服了轮廓断开和假边的问题，是将形态分水岭变换应用于梯度幅度图像[30]，[31]。这种方法允许将初始图像分区生成到区域，因此，可以使用基于区域的技术来产生封闭的、像素范围的轮廓/曲面。
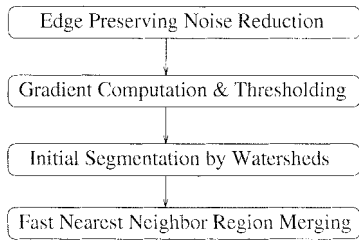
Fig. 1. Flow diagram of the proposed segmentation algorithm.

[30], [31], do not use all the regional minima of the input image in the flooding step but only a small number of them. These selected regional minima are referred to as *markers*. Prior to the application of the watershed transform, the intensity image can be modified so that its regional minima are identical to a predetermined set of markers by the homotopy modification method [30]. This method achieves the suppression of the minima not related to the markers by applying geodesic reconstruction techniques and can be implemented efficiently using queues of pixels [38]. Although markers have been successfully used in segmenting many types of images, their selection requires either careful user intervention or explicit prior knowledge on the image structure.

In our approach, image oversegmentation is regarded as an initial image partition to which a fast region-merging procedure is applied (see Section V). As explained in Section V, the larger the initial oversegmentation, the higher the probability of false region merges during merging. In addition, the computational overhead of region merging clearly depends on the size of this initial partition, and consequently the smallest possible oversegmentation size is sought. One way to limit the size of the initial image partition is to prevent oversegmentation in homogeneous (flat) regions, where the gradient magnitude is low since it is generated by the residual noise of the first stage (see Fig. 1). The watershed transform is applied to the thresholded gradient magnitude image G_T, where the pixels of G having value smaller than a given threshold T are set to zero. Due to thresholding, many of the regional minima of G located in homogeneous regions are replaced by fewer zero-valued regional minima in G_T. These regional minima may contain isolated groups of high gradient values that are not segmented by the watershed transform, since each pixel in any of the isolated groups is associated with the same regional minimum. However, high threshold values T may cause merging of regional minima in G that correspond to different neighboring objects, if the gradient magnitude along their common boundary is not sufficiently high. In this case, only one catchment basin is created, since the two objects share a common regional minimum and, consequently, no watersheds will be constructed between them. Merging of two neighboring regional minima may also be encountered even when the thresholding operation creates a one-pixel wide path of zero values connecting the two regional minima. This behavior of the watershed transform is expected since the decision to create watersheds it is not based on local processing but on the global topographic image shape.

In order to lower the probability of merging minima corresponding to different objects, we propose the use of G_S in place of G in the thresholding process, that is,

$$G_T(p) = \begin{cases} G(p), & \text{if } G_S(p) > T \\ 0, & \text{otherwise} \end{cases} \qquad (11)$$

where G_S is produced by smoothing only the noncandidate edge pixels in G. A candidate edge pixel is defined as having intensity value in G that is a local maximum along the direction of the gradient vector at that pixel. Note that, in G_S the noncandidate edge-pixels being in the neighborhood of candidate ones are relatively enhanced. Finally, the value of T in (11) may be determined directly based on the estimated noise variance $\hat{\sigma}^2$. For example, typical values of T which produce satisfactory initial oversegmentation reduction in almost all experimental cases considered are less than $\hat{\sigma}/2$. Fig. 9(d) shows the initial segmentation result for the image of Fig. 7 (right) after oversegmentation reduction using gradient thresholding. 快速最近邻区域合并

V. FAST NEAREST NEIGHBOR REGION MERGING
逐步优化层次区域合并

A. Stepwise Optimal Hierarchical Region Merging

In addition to the above oversegmentation reduction method, there still remain neighboring regions that could by merging yield a meaningful segmentation, on the principal that each region is homogeneous and sufficiently different from its neighbors. More specifically, let Δ_{K_0} be the initial image partition produced by the watershed detection algorithm. It is assumed that Δ_{K_0} satisfies (2) while there are many region pairs not satisfying (3). The goal of the hierarchical region merging process presented here is to transform Δ_{K_0} to the sought image partition Δ_{K^*} (see Section II) using a sequence of region pair merges. Therefore, the problem is finding the optimal sequence of merges in the sense that its application to Δ_{K_0} produces Δ_{K^*}. It is also assumed that Δ_{K^*} minimizes an objective function F defined over the space of K-partitions and, therefore, the sequence of merges that produces an image partition which minimizes F is sought. The problem belongs to the class of combinatorial optimization problems and, hence, finding its global solutions is a very difficult task [39]. The global solutions may be found using exhaustive search in which case all possible sequences of merges are applied to Δ_{K_0} and then evaluated using the cost function. However, even for small sequences, the search space is extremely large. The solution adopted here is based on the stepwise optimization of F. In other words, the sought sequence of merges is constructed step-by-step, where at each step the region pair merge that produces the partition with the minimum value of F is selected. Such a sequence of merges does not guarantee the construction of the optimal K^*-partition [40], [41]. The latter suggests that the sequence of region merges may contain false ones, that is, merges of dissimilar regions. The probability of a false merge depends both on the sizes of the regions to be merged and on the noise variance. Therefore, the larger the initial partition, the higher the expected number of false merges during merging. In this respect, the worst possible initial partition would be the one where each image pixel is a separate region. In our approach,

通过保持边缘降噪和梯度幅度阈值来避免严重的过度分割

severe oversegmentation is avoided through edge preserving noise reduction and gradient magnitude thresholding (Section III).

B. Region Dissimilarity Functions 区域差异函数

The objective cost function used in this work is the square error of the piecewise constant approximation of the observed image Y, which yields a measure of the approximation accuracy and is defined over the space of partitions. Let $\mathcal{R}_M = \{R_M^1, R_M^2, \cdots, R_M^M\}$ be a M-partition of image Y and $R_M^k = \{p_{k,1}, p_{k,2}, \ldots, p_{k,\|R_M^k\|}\}$ be the set of pixels belonging to region R_M^k. In the piecewise constant approximation of Y, the image intensity in each region $R_M^k, k = 1, 2, \cdots, M$, of partition $\mathcal{R}_M$ is approximated by one parameter, which minimizes the square error with respect to the data Y and is equal to the mean value of Y in R_M^k, namely

$$\mu\big(R_M^k\big) = \frac{1}{\|R_M^k\|} \sum_{i=1}^{\|R_M^k\|} Y(p_{k,i})$$

where $\|R\|$ denotes the cardinality of set R. The corresponding square error is

$$E\big(R_M^k\big) = \sum_{i=1}^{\|R_M^k\|} \big[Y(p_{k,i}) - \mu\big(R_M^k\big)\big]^2.$$

Therefore, the total square error is

$$E(\mathcal{R}_M) = \sum_{k=1}^{M} E\big(R_M^k\big).$$

It is clear that, if R_M^* is the optimal M-partition with respect to the squared error, then the optimal $(M-1)$-partition is generated by merging the pair of regions of R_M^*, which minimizes the following dissimilarity function [41], [42]:

$$\delta\big(R_M^{*i}, R_M^{*j}\big)$$
$$= \frac{\|R_M^{*i}\| \cdot \|R_M^{*j}\|}{\|R_M^{*i}\| + \|R_M^{*j}\|} \big[\mu\big(R_M^{*i}\big) - \mu\big(R_M^{*j}\big)\big]^2 \mathcal{I}(i,j) \quad (12)$$

where

$$\mathcal{I}(i,j) = \begin{cases} 1, & \text{if regions } R_M^{*i}, R_M^{*j} \text{ are adjacent} \\ +\infty, & \text{otherwise.} \end{cases}$$

设置阈值

According to the above formulation, the most similar pair of regions is the one minimizing (12).

The determination of the optimal number of segments K^* is performed by checking the value of $\delta(\cdot, \cdot)$. If δ is greater than a certain threshold, then the merging process is terminated. This threshold can be determined by using the knowledge on the noise distribution (hypothesis testing) [8].

使用区域邻接图进行合并

C. Merging Using the Region Adjacency Graph

The data structure for representing partitions is the region adjacency graph (RAG) [43], [44]. The RAG of a K-partition is defined as an undirected graph, $G = (V, E)$, where $V = \{1, 2 \cdots, K\}$ is the set of nodes and $E \subset V \times V$ is the set of
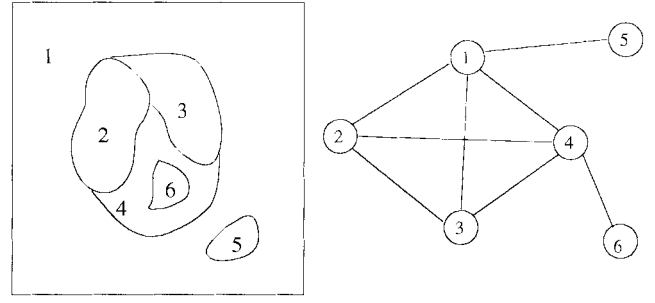
无权无向图

将成本分配给每个图边，表示两个相邻区域之间的差异

Fig. 2.   Six-partition of an image (*left*), andthe corresponding RAG (*right*).

将成本分配给每个图边，表示两个相邻区域之间的差异
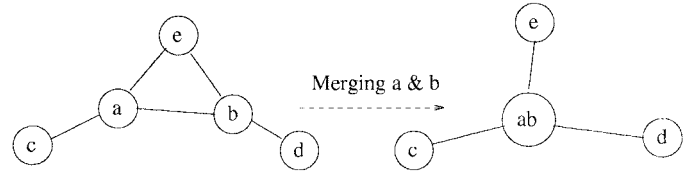
Fig. 3.   Merging of two RAG nodes.

由于合并，节点a与节点b的边缘消失

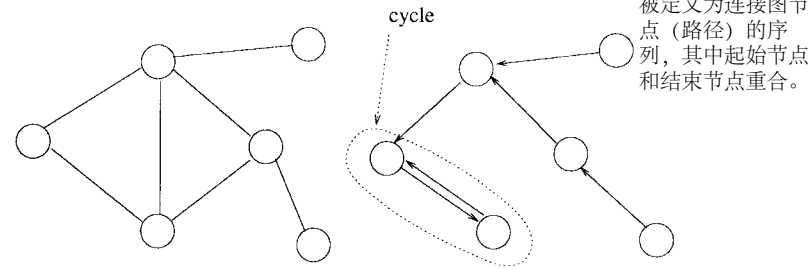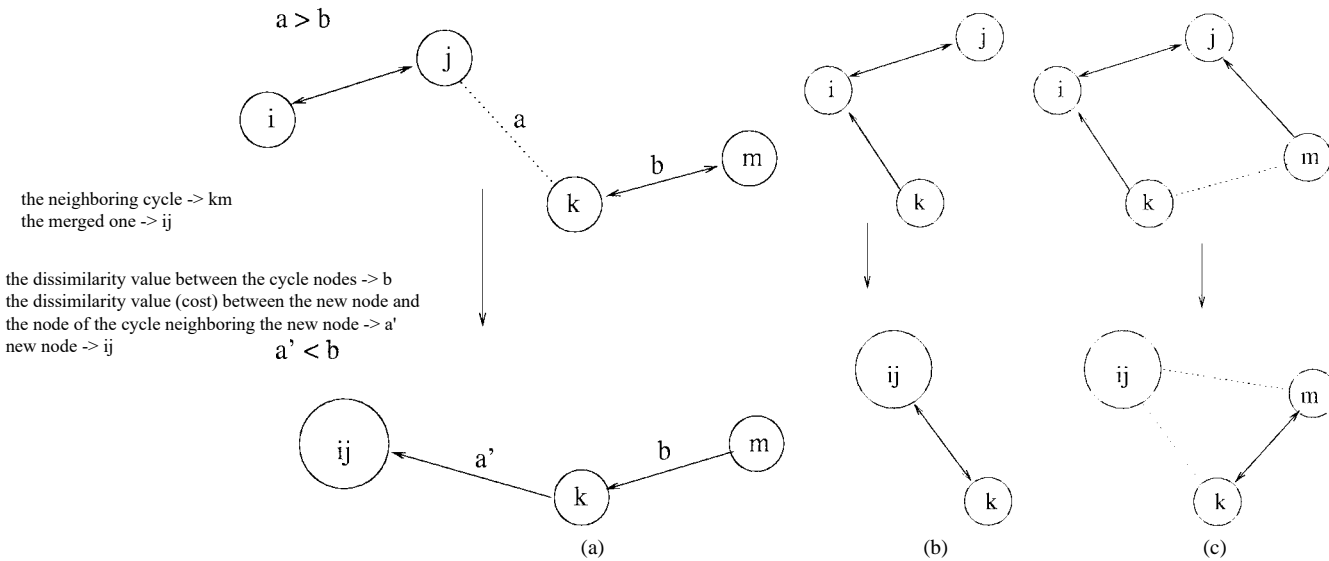在NNG中的循环被定义为连接图节点（路径）的序列，其中起始节点和结束节点重合。

Fig. 4.   RAG (*left*) and one of its possible NNG's (*right*).

参考：最相似的一对相邻区域对应于具有最小成本的边缘。由于在每个合并步骤都需要具有最小成本的边缘，因此适当的数据结构是一个优先队列。

edges. Each region is represented by a graph node and between two regions (nodes) $i, j \in V$ exists the edge (i, j) if the regions are adjacent. An example of a six-partition of an image and the corresponding RAG are shown in Fig. 2. A cost is assigned to each graph edge expressing the dissimilarity between the two adjacent regions. The most similar pair of adjacent regions corresponds to the edge with the minimum cost. Since, at each merging step the edge with the minimum cost is required, the appropriate data structure is a priority queue which can be implemented efficiently by a heap [44], [45]. All RAG edges are stored in a heap according to their costs.

After the application of the watershed transform, the produced K-partition image is used for the construction of the initial RAG (K-RAG) that will be input to the region merging procedure. This requires one raster scan of the partition image during which the neighboring relations (RAG edges) between image regions (RAG nodes) are identified. Then, the size and the intensity sum of each RAG region are computed and stored in the corresponding RAG node since they are used by the subsequent region merging process. Constructing the heap of edges requires the calculation of edge costs (12) and takes $O(\|E\|)$ time using the bottom-up heap construction method [45], [46].

参考

a > b

the neighboring cycle -> km
the merged one -> ij

the dissimilarity value between the cycle nodes -> b
the dissimilarity value (cost) between the new node and
the node of the cycle neighboring the new node -> a'
new node -> ij

a' < b

(a)   (b)   (c)

Fig. 5. Examples of the three possible NNG-cycle modification types due to merging. (a) NNG-cycle cancellation. (b), (c) NNG-cycle creation with (b) and without (c) the participation of the node resulting from merging. Notation: RAG edge (···), NNG edge (→) and NNG cycle (↔).

a < b

a

b

a' > b

a'   b

a > b

a

b

a' < b

a'   b
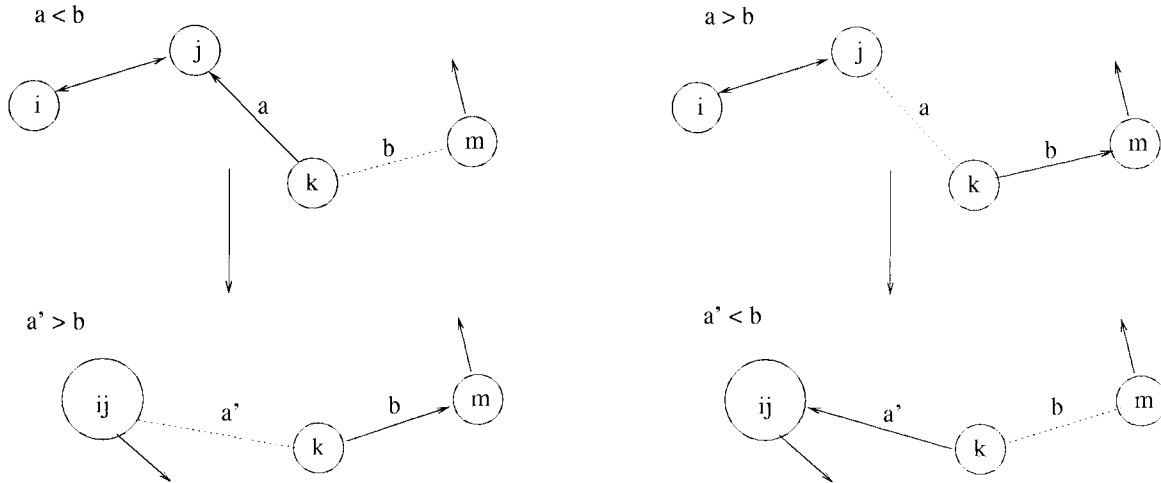
Fig. 6. Two examples of NNG-edge modification due to merging.

Given the RAG of the initial K-partition (K-RAG) and the heap of its edges, the RAG of the suboptimal $(K-n)$-partition ($(K-n)$-RAG) is constructed by the following algorithm, which implements the stepwise optimization procedure described above.

Input: RAG of the K-partition (K-RAG).
Iteration: For $i = 0$ to $n - 1$
  Find the minimum cost edge in the $(K-i)$-RAG using the edge heap.
  Merge the corresponding pair of regions to get the $(K - i - 1)$-RAG.
  Update the edge heap.
Output: RAG of the $(K-n)$-partition ($(K-n)$-RAG).

At each merging step, the edge with the minimum cost is removed from the heap in $O(\log_2(\|E\|))$ time and the corresponding nodes are merged. The merging operation causes changes in both the RAG and the heap. All RAG nodes that neighbored a node of the merged node pair must restructure

their neighbor lists. Also, the dissimilarity values (costs) of the neighboring nodes with the node resulting from the merging stage change and must be recalculated using (12).

The positions of the changed-cost edges in the heap must be updated, requiring $O(\log_2(\|E\|))$ time for each update. In addition, a few edges must be removed since they are canceled due to merging. This is illustrated in Fig. 3, where a merging example of two RAG nodes is given. Before the merging of nodes a and b, node e is a common neighbor to a and b. After their merging, one of the edges (a, e), (b, e) must be removed from the RAG and the heap. Then, the positions of the changed-cost edges in the heap must be updated (edges (ab, c), (ab, d), (ab, e) in Fig. 3). However, since these positions are unknown, a linear search operation requiring $O(\|E\|)$ time results in $O(d_n \|E\| \cdot \log_2(\|E\|))$ time for each merge, where d_n denotes the degree of the node produced by merging the most similar region pair. Therefore, due to the usual large heap size, the total computation time is
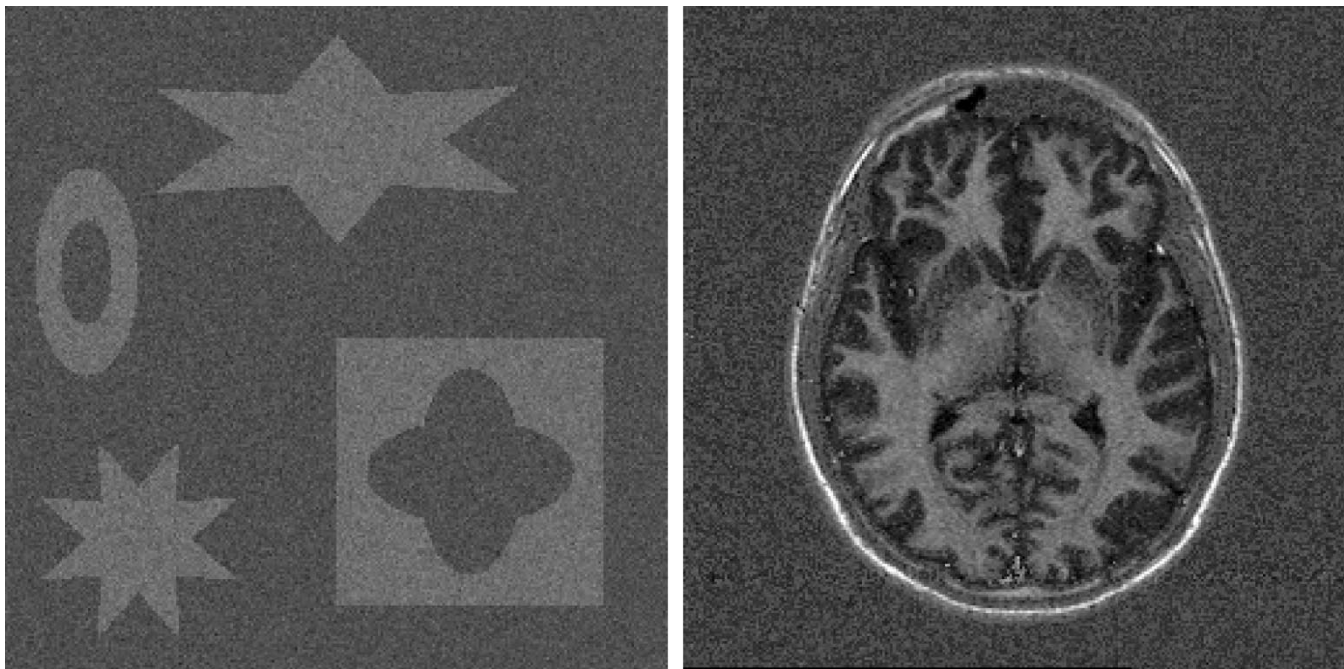
边缘参与合并

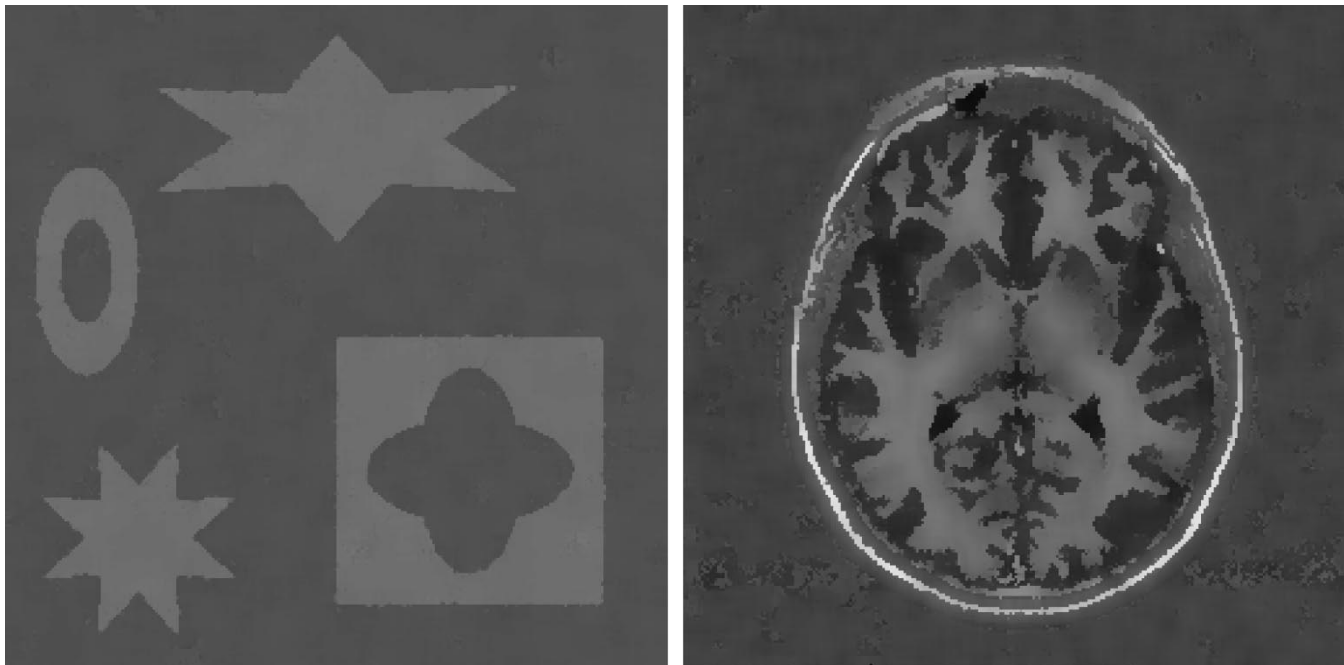Fig. 7. Synthetic image (left) and real medical MR image (right).

合成

Fig. 8. Result of the noise reduction stage on the images of Fig. 7.

considerably increased. This is particularly true in 3-D images where the initial partition usually contains a very large number of regions.

D. Fast Nearest Neighbor Merging

The proposed solution to accelerate region merging is based on the observation that it is not necessary to keep all RAG edges in the heap but only a small portion of them [8]. Specifically, we introduce the NNG, which is defined as follows. For a given RAG, $G = (V, E)$ and a symmetric dissimilarity function $S : V \times V \to \Re$, the NNG, namely,

对称

$G_m = (V_m, E_m)$, is a directed graph with $V_m = V$ and the directed edge $(i, j), i, j \in V$ belongs to E_m, if $S(i, j) = \min\{S(i, k) : (i, k) \in E\}$. An example of a RAG and one of its possible NNG's is shown in Fig. 4. When there are more than one nodes minimizing S, the edge is directed toward the node with the minimum label. The above definition implies that the out-degree of each node is equal to one. The edge starting at a node is directed toward its most similar neighbor.

A cycle in the NNG is defined as a sequence of connected graph nodes (path) in which the starting and ending nodes coincide (see Fig. 4). By definition, the NNG contains $\|V\|$

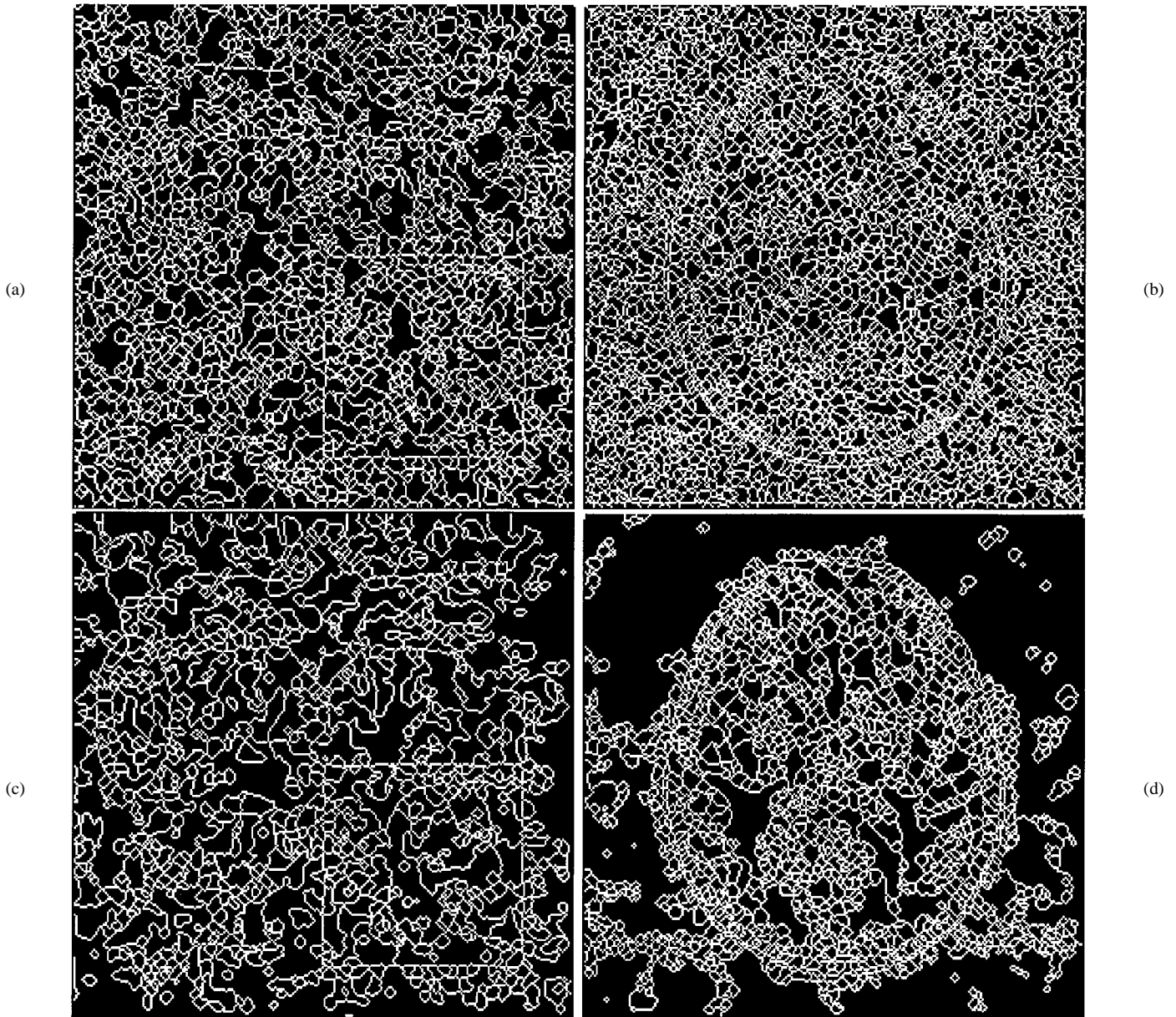当有多个节点最小化S时，边缘指向具有最小标签的节点。上述定义意味着每个节点的out-degree和等于一。从节点开始的边缘指向其最相似的邻居。

Fig. 9.   Initial segmentation results of the images in Fig. 8 after applying the Gaussian filter ($\sigma = 0.7$) and thresholding. (a) $T = 0$ (2672 regions). (b) $T = 0$ (3782 regions). (c) $T = 5$ (1376 regions). (d) $T = 5$ (1997 regions).

edges and has the following properties [8].

Property 1: The NNG contains at least one cycle.

Property 2: The maximum length of a cycle is two.

Property 3: The regions of the most similar pair are connected by a cycle. 最相似对的区域通过一个循环连接。

Property 4: A node can participate at most one cycle.

Property 5: The maximum number of cycles is $\lfloor \|V\|/2 \rfloor$.

A direct consequence of Property 3 is that the heap must keep only the NNG cycles and not all the RAG edges. In this case, the worst-case size of the heap is equal to $\lfloor \|V\|/2 \rfloor$. Property 4 implies that each cycle is uniquely determined by one of the regions it connects. This fact allows the significant reduction of the linear search to constant time search during the merging stage. In implementing the search, an auxiliary array with size equal to the heap size is used to store the positions of the

graph cycles in the heap. The additional computational effort required is one swap operation in the auxiliary array following each swap operation in the heap.

After the RAG construction, the NNG is formed by searching for the most similar among the neighbors of each RAG node. Then, the NNG cycles are identified by a scan of the NNG. The heap of cycles is filled using the bottom-up method and the merging proceeds as follows.

Input: RAG and NNG of the K-partition.

Iteration: For $i = 0$ to $n - 1$
           Find the minimum cost edge in the $(K - i)$-RAG using the cycle heap.
           Merge the corresponding pair of regions to get the $(K - i - 1)$-RAG.
           Update the NNG and the cycle heap.

Output: RAG and NNG of the $(K - n)$-partition.

在RAG构建之后，NNG是通过在每个RAG节点的邻居中搜索最相似的节点来形成的。 然后，通过对NNG的扫描来识别NNG循环。 循环堆使用自下而上的方法填充，合并进行如下：
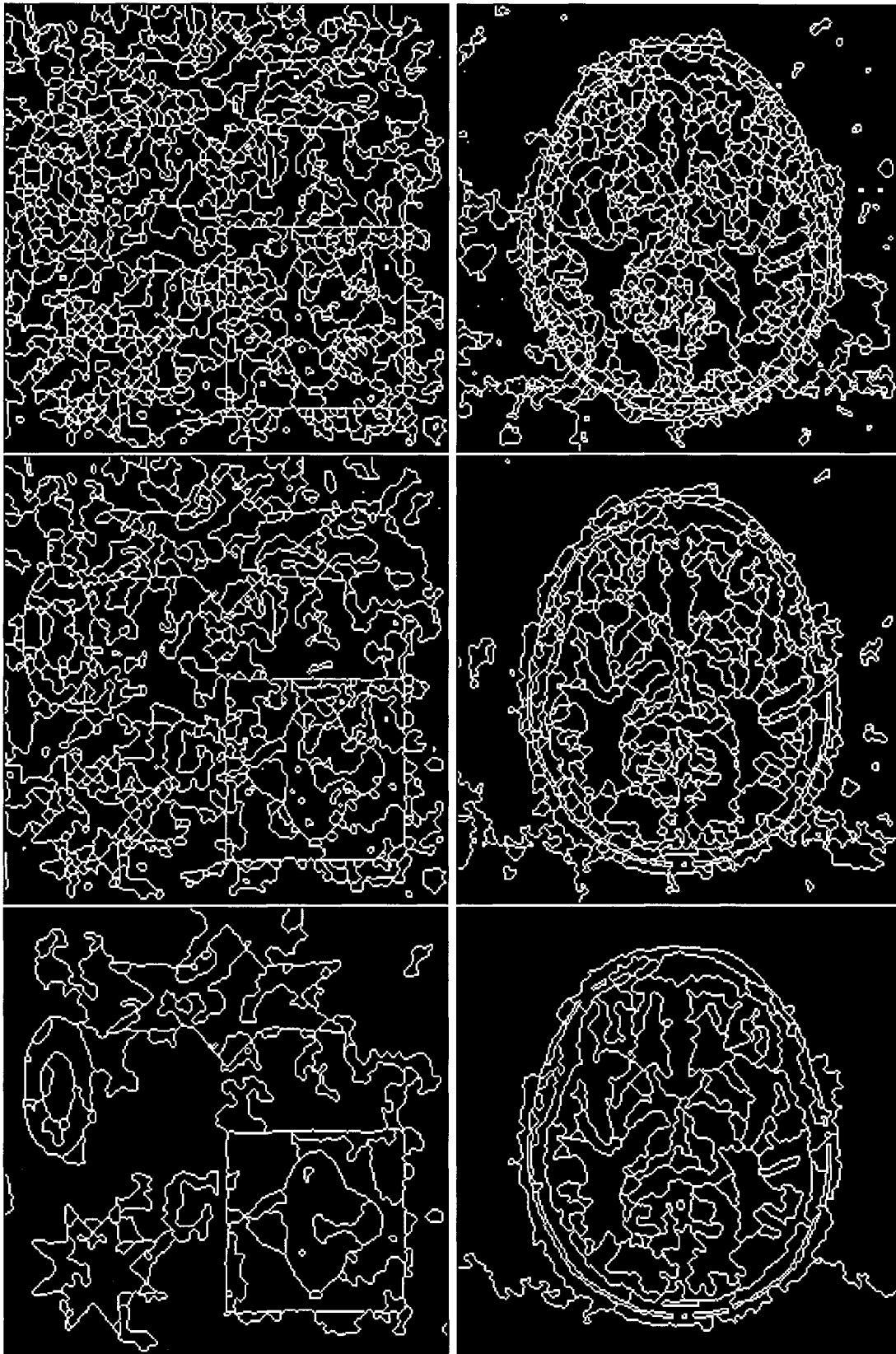
Fig. 10.   Intermediate segmentation results. Top: 1000 regions. Middle: 500 regions. Bottom: 100 regions.
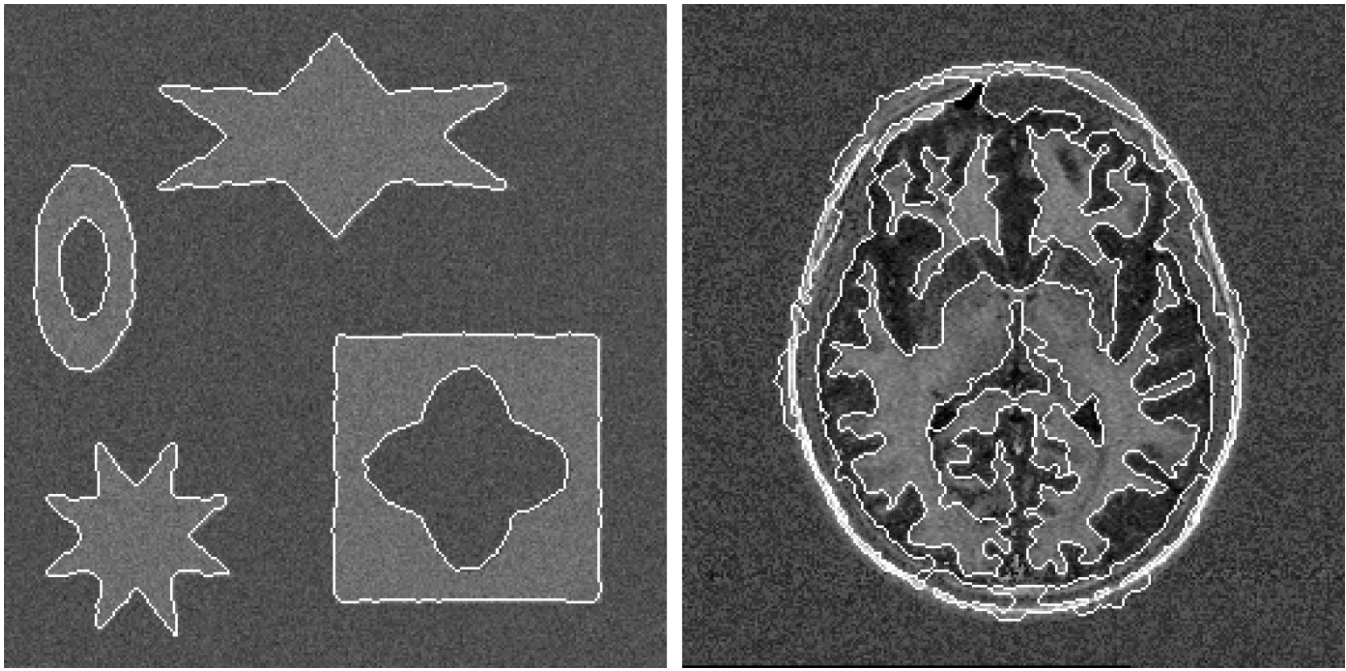
Fig. 11.   Final segmentation results overlaid on the original images. Left: 7 regions. Right: 25 regions.

检测新节点的二阶邻域大小

During the merging operation, the NNG is updated as follows. When the nodes of a cycle are merged, the costs of the neighboring RAG edges and, consequently, the structure of the NNG are modified. Two NNG cycles are defined as neighbors if there is at least one RAG edge connecting two of their nodes. For example, in Fig. 5(a), the cycles $(i, j), (k, m)$ are neighbors because nodes i and k are connected through a RAG edge (dotted line). A neighboring cycle to the merged one is canceled when the dissimilarity value (cost) between the new node and the node of the cycle neighboring the new node is less than the dissimilarity value between the cycle nodes. For example, in Fig. 5(a) before merging nodes i and j, the cost a of RAG edge (j, k) is greater than the cost b of cycle (k, m), where $a = \delta(j, k)$ and $b = \delta(k, m)$. After merging, the cost a' of RAG edge (ij, k) happens to be smaller than cost b $(a' < b)$, resulting in the cancellation of cycle (k, m). When a cycle is canceled, then it must be removed from the heap. The canceled cycle position in the heap is determined directly without any search, using the auxiliary array. Merging may also cause the creation of new cycles [see Fig. 5(b) and (c)] that must be inserted in the heap. In Fig. 5(b), the merging of nodes i and j creates the NNG edge (ij, k) which together with the NNG edge (k, ij) creates the new NNG cycle (ij, k). According to this scenario, the node resulting from the above merging (i.e., node ij) participates in the new NNG cycle. Another possible scenario is the creation of new cycles without the participation of the node resulting from merging. For example, in Fig. 5(c) merging modifies the dissimilarity values in such a way that causes the cancellation of NNG edges $(k, i), (m, j)$ and the creation of NNG edges $(k, m), (m, k)$, thus resulting in the NNG cycle (k, m).

The NNG edges not forming cycles may also be modified. These modifications must be taken into account for maintaining NNG consistency during merging and they do not affect the cycle heap. Two examples illustrating the NNG edge modifications are shown in Fig. 6. In order to detect the modification type of NNG edges (Figs. 5 and 6) the second order neighborhood of the new node must be examined resulting in $O(d_n^{(2)} + \beta \cdot \log_2(\|B\|))$ time for each merge, where $d_n^{(2)}$ denotes the second-order neighborhood size of the new node, β is the number of NNG cycles modified by the merge, and B is the set of the NNG cycles stored in the heap. Hence, the proposed method is expected to be particularly fast for graphs with small mean node degrees. The latter is true for the RAG's as the experiments demonstrate below.

VI. Experimental Results and Discussion

The two 2-D (256×256, 8 b/pixel) images shown in Fig. 7 were used in order to illustrate the stages of the segmentation algorithm and visually assess the quality of the segmentation results. The synthetic image [Fig. 7(a)] is piecewise constant, the background intensity level is 80, the object intensity level is 110 and contains simulated additive white Gaussian noise with standard deviation $\sigma = 10$. Fig. 7(b) shows a noisy T1-weighted MR brain image, where the noise was statistically tested and found to be approximately additive Gaussian with estimated standard deviation $\sigma = 13$. Concerning the noise corrupting MR images, it is safe to assume that its distribution is Gaussian at least within tissues where the signal has large values [47], [48].

Fig. 8 shows the result of the proposed edge-preserving noise reduction stage. In our work, the required estimate of the noise standard deviation was evaluated by the maximum value of the cumulative histogram of all local variances, each computed at the corresponding 13×13 neighborhood. In the above MR image the estimated noise standard deviation was $\hat{\sigma} = 13.5$. The window size was set to 11×11 for the synthetic image and 9×9 for the MR image, and it affects

不形成循环的NNG边缘也可以被修改

在我们的工作中，噪声标准差的所需估计值是由所有局部方差的累积直方图的
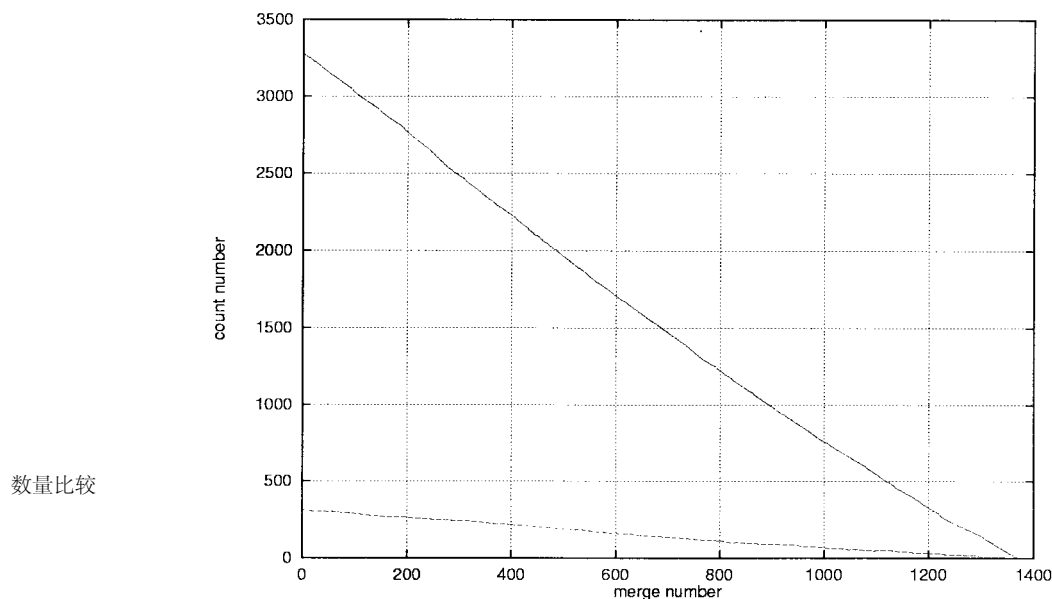最大值来评估的，每个直方图都是在相应的13x13邻域计算的.

数量比较

Fig. 12. Number of RAG edges (solid line) and NNG cycles (dotted line) as a function of the merge number for the image in Fig. 8 (right).
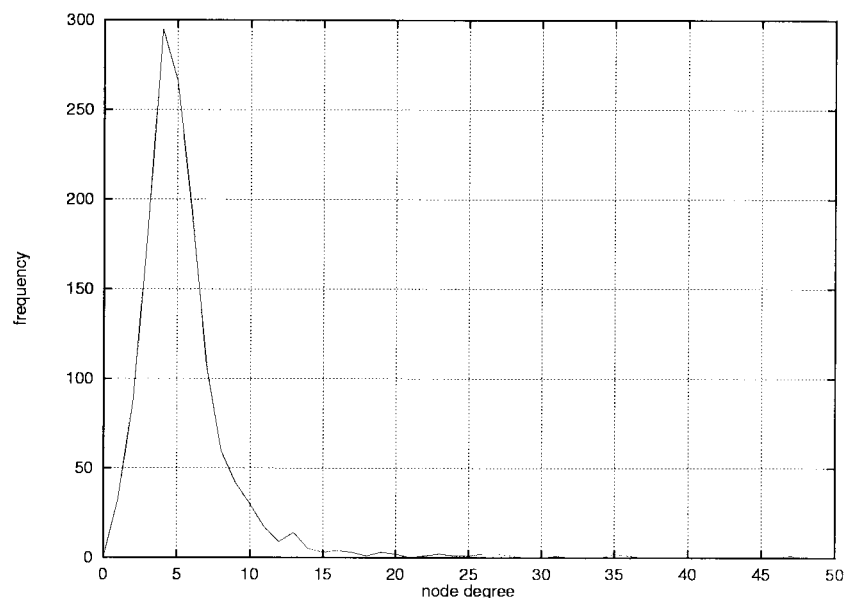

Fig. 13. Histogram of the RAG node degree for the image in Fig. 8 (right).

the performance of the noise reduction algorithm as follows. For large window sizes, the power of the homogeneity test (i.e., the probability of correctly accepting heterogeneity) is large in the case of step edges, while it is relatively small in the case of bar edges. Therefore, the thin features of the image (lines, corners) are oversmoothed. For small window sizes, the power of the homogeneity test is small and the variance of the mixture parameter estimates is large. Therefore, the resulting noise reduction is small. However, the above phenomena occur for very noisy images. In Fig. 8, it is clear that the noise is sufficiently reduced while the image context is preserved and enhanced. Note that the proposed noise reduction algorithm does not impose any smoothness constraints and, therefore, when the noise level is not high, the image structure is preserved remarkably well. However, we believe that the lack of smoothness constraints is the source of

在保留和加强图像背景的同时，噪声得到了充分的减少。

the nonrobust behavior of the algorithm on very noisy images. In addition, the adopted image model does not handle more complex structures such as smooth intensity transitions (ramp edges) and junctions. 利用具有尺度0.7的高斯滤波器导数计算平滑图像的梯度幅度

At the second stage, the gradient magnitude of the smoothed image is calculated using the Gaussian filter derivatives with scale $\sigma = 0.7$. Then, the gradient magnitude was thresholded using (11), where the smoothed gradient magnitude G_S was produced by 3×3 neighborhood averaging of noncandidate edge pixels. At the third stage, the watershed detection algorithm was applied to the thresholded image gradient magnitude. Fig. 9 shows the initial tessellations of the images produced by the application of the watershed detection algorithm on the image gradient magnitude for various thresholds. It is clear that the larger the threshold the smaller the number of the regions produced by the watershed

其中平滑的梯度幅度Gs是由非随机边缘像素的3X3邻域的均值组成

镶嵌
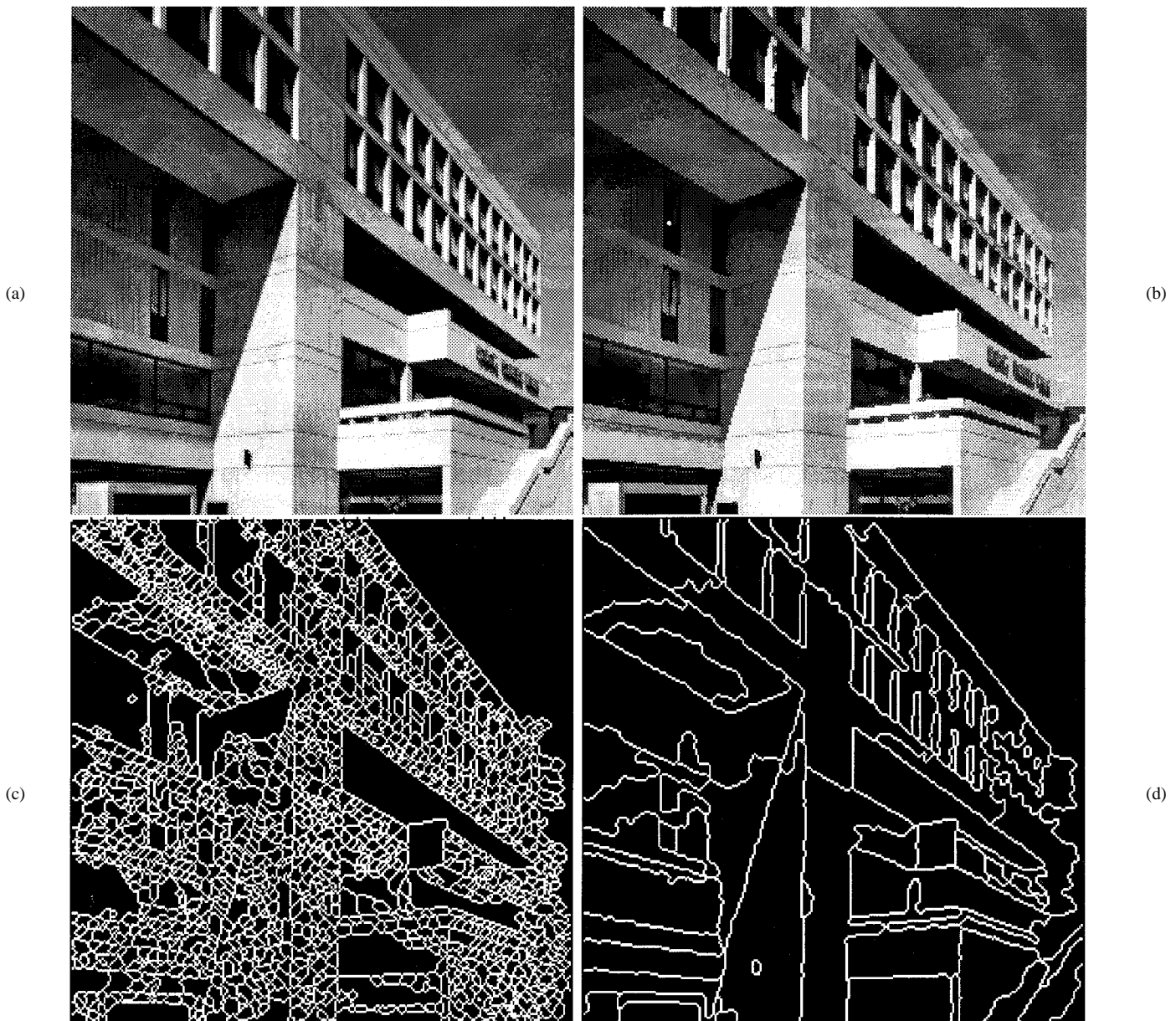
Fig. 14. Segmentation of a natural image. (a) Original image ("MIT"). (b) Result of the noise reduction stage. (c) Initial segmentation after gradient thresholding($T = 2$, 2347 regions). (d) Final segmentation (80 regions).

detection algorithm. However, the use of high thresholds may destroy part of the image contours, which cannot be recovered at the merging stage of the segmentation algorithm. More specifically, it was observed that when the noise is not high, the choice for the threshold value close to the noise standard deviation is safe. However, when noise is high, small threshold values should be used. This is justified from the fact that when noise is high, the noise reduction algorithm may oversmooth part of the image intensity discontinuities resulting in low gradient magnitudes. Therefore, the use of high threshold values in (11) may destroy part of the object boundaries.

The initial tessellations are used at the last stage of the algorithm for the construction of the RAG's and NNG's, and then the merging process begins. Fig. 10 shows several intermediate results of the merging process using the corresponding initial segmentation results shown in Fig. 9(c) and (d). The

final segmentation results are given in Fig. 11 with seven and 25 regions, respectively. The number of regions of the initial image tessellation determines the computational and memory requirements for the construction and processing (merging) of the RAG and NNG. The number of the RAG-edges and the number of NNG-cycles are shown in Fig. 12 as a function of the number of merges. The size of the cycle heap is nearly one order of magnitude smaller than the size of the heap of RAG edges. As explained in Section V, the additional computational effort for manipulating the NNG at each merge of region pair depends on the distribution of the second order neighborhood size in the RAG. In Fig. 13, a typical histogram of the RAG degree at an intermediate stage of merging is shown. As expected, the RAG is a graph with low mean degree and this explains the low additional computational effort for the NNG maintenance.

TABLE II
TYPICAL EXECUTION TIMES OF THE PROPOSED SEGMENTATION ALGORITHM AND ITS STAGES WITH AND WITHOUT THE USE OF THE NNG

ALGORITHM STAGE	EXECUTION TIMES (percentage of total time)			
	2D Image (256×256)		3D Image ($256 \times 145 \times 16$)	
	RAG	RAG & NNG	RAG	RAG & NNG
Noise Reduction	4.9 sec (21.7%)	4.9 sec (48.1%)	48 sec (13.4%)	48 sec (37.2%)
Gradient Computation	2.3 sec (10.2%)	2.3 sec (22.5%)	15 sec (4.2%)	15 sec (11.6%)
Initial Segmentation	0.8 sec (3.5%)	0.8 sec (7.8%)	14 sec (3.9%)	14 sec (10.8%)
Region Merging	14.6 sec (64.6%)	2.2 sec (21.6%)	280 sec (78.4%)	52 sec (40.3%)
Total time	**22.6 sec**	**10.2 sec**	**357 sec**	**129 sec**

The proposed segmentation algorithm was also applied to natural images, such as, the standard "MIT" image (256×256, 8 b/pixel) shown in Fig. 14(a). This image contains sharp intensity transitions and can be considered as approximately piecewise constant in most of its areas. The image was assumed to contain Gaussian noise with estimated standard deviation $\sigma = 1.5$. The result of the noise reduction stage using a 5×5 window is shown in Fig. 14(b). Due to the fine detail in the image the window size can not be large ($n > 7$) because the two Gaussian mixture assumption in heterogeneous pixel neighborhoods may not hold. The initial segmentation result using gradient thresholding ($T = 2$, 2347 regions) and the final segmentation result (80 regions) are given in Fig. 14(c) and (d), respectively. Note that, despite the simplicity of both the underlying image model and the dissimilarity function used, the majority of important image regions were successfully extracted.

The 3-D version of the algorithm was applied to a $16 \times 145 \times 256$ MR cardiac image, a slice of which is shown in Fig. 15(a). Fig. 15(b) shows the result of the noise reduction stage, where a $3 \times 5 \times 5$ window was used. Fig. 15(c) shows the initial segmentation which resulted from the watershed detection stage on the thresholded gradient magnitude image, where the scale of the Gaussian filter was 0.7 and the threshold $T = 2$. Lastly, Fig. 15(d) shows the final segmentation result containing 40 3-D regions.

Based on our experiments we concluded, that the smaller the number of the initial (correct) partition segments, the better the final segmentation results. On the other hand, the use of thresholds producing initial partitions with small number of segments may cause the disappearance of a few significant contours. The 2-D and 3-D version of the proposed image segmentation algorithm were implemented in the C programming language on a Silicon Graphics (R4000) computer. Table II shows typical execution times and percentages with respect to the total time for each stage of the proposed algorithm with and without the use of NNG. Note that the noise reduction stage requires a great percentage of the total execution time. This is due to the current implementation in which the required parameters are computed at each window position separately. The noise reduction algorithm may be accelerated by considering a faster implementation, namely, using the separability property in order to compute the sample moments [12].

Finally, regarding the memory requirements of the proposed algorithm, they are high due primarily to the watershed algorithm [31]. At the merging step, the memory required for
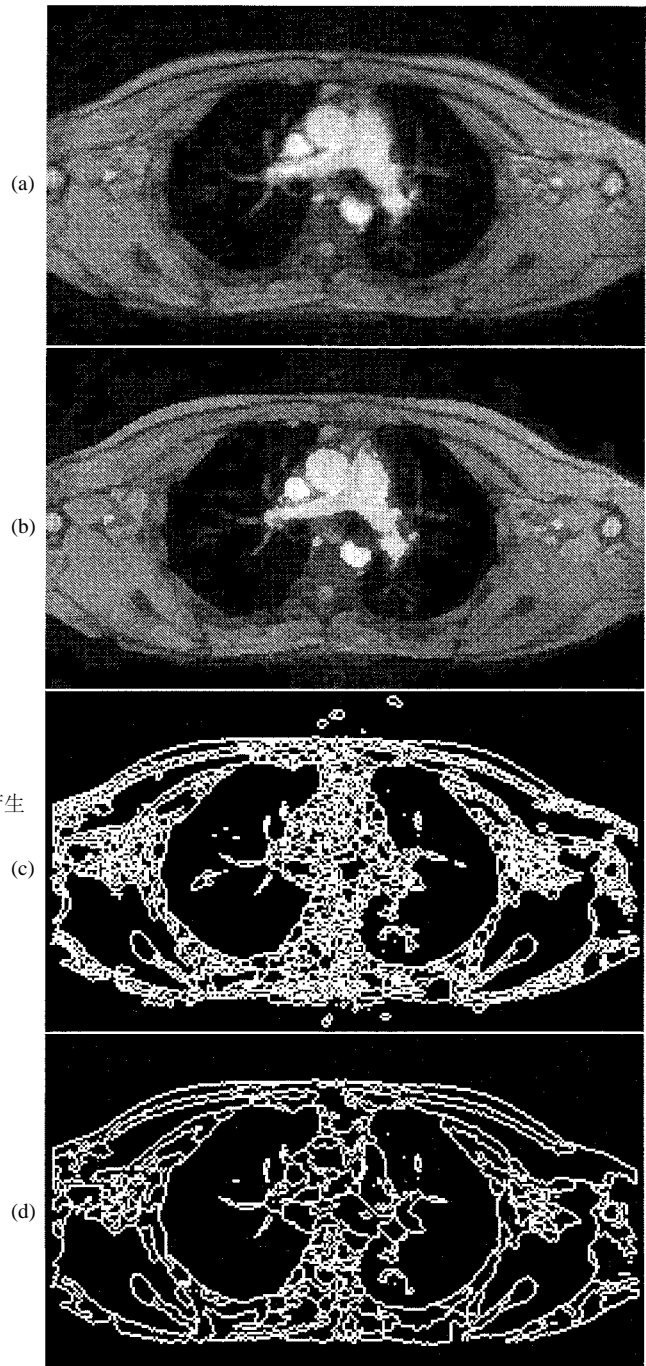

Fig. 15. Three-dimensional image segmentation results. (a) Raw 3-D MR image (slice 5). (b) Smoothed image. (c) Initial oversegmentation (3058 regions). (d) Final segmentation (40 regions).

the representation of the RAG, NNG and heap of cycles is $O(\|V\|+\|E\|)$ which is quite small compared to that required by the watershed algorithm alone.

VII. CONCLUSION

A fast hybrid segmentation algorithm was presented which integrates edge and region-based techniques via the watershed detection algorithm. The output of the algorithm is the RAG of the final segmentation based on which closed, one-pixel wide object contours/surfaces may readily be extracted. In addition, the RAG provides information about the spatial relationships between objects and can drive knowledge-based higher level processes as a means of description and recognition. The overall approach provides a general framework in which gradient and region-based techniques are combined. Furthermore, the proposed algorithm allows the interactive control of the stopping point by storing intermediate partitions. In other words, the user may select the iteration at which the resulting segmentation is acceptable.

The proposed segmentation algorithm was implemented for the 2-D and 3-D cases and produced very satisfactory results both with respect to segmentation performance and execution times. However, the memory requirements are relatively high due to the watershed detection algorithm [31]. Also, when the SNR is low, it is expected that the proposed noise reduction stage may not perform well as far as edge preservation is concerned. Finally, although the proposed region dissimilarity function was proven quite suitable for near piecewise constant images, the use of more complex functions may give better results on the expense of computational complexity during the merging process. For instance, the zero-order polynomial approximation can be used as a first step in a sequence of increasing order piecewise polynomial approximations (variable order fitting [20]) in order to segment images of high structural complexity, e.g., range images. However, the computationally efficient extension of the proposed technique to this direction is an open research topic.

Future research is directed toward the improvement of the 3-D version of the algorithm and its extension to the segmentation of moving 3-D images [49].

ACKNOWLEDGMENT

REFERENCES

[1] P. Suetens, P. Fua, and A. J. Hanson, "Computational strategies for object recognition," *ACM Comput. Surv.*, vol. 24, pp. 5–61, Mar. 1992.
[2] P. Besl and R. Jain, "Three-dimensional object recognition," *ACM Comput. Surv.*, vol. 17, pp. 75–145, Mar. 1985.
[3] K. Hohne, H. Fuchs, and S. Pizer, *3D Imaging in Medicine: Algorithms, Systems, Applications*. Berlin, Germany: Springer-Verlag, 1990.
[4] M. Bomans, K. Hohne, U. Tiede, and M. Riemer, "3-D segmentation of MR images of the head for 3-D display," *IEEE Trans. Med. Imag.*, vol. 9, pp. 253–277, June 1990.
[5] M. Kunt, M. Benard, and R. Leonardi, "Recent results in high-compression image coding," *IEEE Trans. Circuits Syst.*, vol. 34, pp. 1306–1336, Nov. 1987.
[6] P. Willemin, T. Reed, and M. Kunt, "Image sequence coding by split and merge," *IEEE Trans. Commun.*, vol. 39, pp. 1845–1855, Dec. 1991.
[7] F. D. Natale, G. Desoli, D. Giusto, and G. Vernazza, "Polynomial approximation and vector quantization: A region-based integration," *IEEE Trans. Commun.*, vol. 43, 1995.
[8] K. Haris, "A hybrid algorithm for the segmentation of 2D and 3D images," Master's thesis, Univ. Crete, 1994.
[9] N. Pal and S. Pal, "A review on image segmentation techniques," *Pattern Recognit.*, vol. 26, pp. 1277–1294, 1993.
[10] R. Haralick and L. Shapiro, "Image segmentation techniques," *CVGIP*, vol. 29, pp. 100–132, 1985.
[11] K. Mardia and T. Hainsworth, "A spatial thresholding method for image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 919–927, Nov. 1988.
[12] A. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ; Prentice-Hall, 1989.
[13] V. Nalwa, *A Guided Tour of Computer Vision*. Reading, MA: Addison-Wesley, 1993.
[14] D. Marr and E. Hildreth, "Theory of edge detection," in *Proc. R. Soc. Lond. B*, 1980, no. 207, pp. 187–217.
[15] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine. Intell.*, vol. PAMI-8, pp. 679–698, Nov. 1986.
[16] V. Torre and T. Poggio, "On edge detection," *IEEE Trans. Pattern Anal. Machine. Intell.*, vol. PAMI-8, pp. 147–163, Mar. 1986.
[17] R. Deriche, "Using Canny's criteria to derive a recursively implemented optimal edge detector," *Int. J. Comput. Vis.*, pp. 167–187, 1987.
[18] O. Monga, R. Deriche, G. Malandain, and J. P. Cocquerez, "Recursive filtering and edge tracking: Two primary tools for 3D edge detection," *Image Vis. Comput.*, vol. 9, pp. 203–214, Aug. 1991.
[19] A. Rosenfeld and A. Kak, *Digital Picture Processing: vol. II*. New York: Academic, 1982.
[20] P. Besl and R. Jain, "Segmentation through variable-order surface fitting," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 167–192, Mar. 1988.
[21] R. Beveridge *et al.*, "Segmenting images using localized histograms and region merging," *Comput. Vis., Graph., Image Process.*, vol. 2, pp. 311–347, 1989.
[22] S. Horowitz and T. Pavlidis, "Picture segmentation by a tree traversal algorithm," *J. Assoc. Comput. Mach.*, vol. 23, pp. 368–388, Apr. 1976.
[23] S. Chen, W. Lin, and C. Chen, "Split-and-merge image segmentation based on localized feature analysis and statistical tests," *CVGIP: Graph. Models Image Process.*, vol. 53, pp. 457–475, Sept. 1991.
[24] Z. Wu, "Homogeneity testing for unlabeled data: A performance evaluation," *CVGIP: Graph. Models Image Process.*, vol. 55, pp. 370–380, Sept. 1993.
[25] R. Dubes and A. Jain, "Random field models in image analysis," *J. Appl. Stat.*, vol. 16, pp. 131–164, 1989.
[26] J. Besag, "On the statistical analysis of dirty images," *J. R. Stat. Soc. B*, vol. 48, pp. 259–302, 1986.
[27] J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," *J. Amer. Stat. Assoc.*, vol. 82, pp. 76–89, Mar. 1987.
[28] T. Pavlidis and Y. Liow, "Integrating region growing and edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 225–233, Mar. 1990.
[29] L. D. Griffin, A. C. F. Colchester, and G. P. Robinson, "Scale and segmentation of grey-level images using maximum gradient paths," *Image Vis. Comput.*, vol. 10, pp. 389–402, July/Aug. 1992.
[30] F. Meyer and S. Beucher, "Morphological segmentation," *J. Vis. Commun. Image Represent.*, vol. 1, pp. 21–46, Sept. 1990.
[31] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 583–598, June 1991.
[32] J. M. Gauch and S. M. Pizer, "Multiresolution analysis of ridges and valleys in gray-scale images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 635–646, June 1993.
[33] S. Orphanoudakis, G. Tziritas, and K. Haris, "A hybrid algorithm for the segmentation of 2D/3D images," in *Proc. IPMI'95*, June 1995.
[34] K. Haris, S. Efstratiadis, N. Maglaveras, and C. Pappas, "Hybrid image segmentation using watersheds," in *SPIE Proc. Vis. Commun. Image Process.*, Orlando, FL, Apr. 1996, vol. 2727, pp. 1140–1151.
[35] K. Haris, G. Tziritas, and S. Orphanoudakis, "Smoothing 2-D or 3-D images using local classification," in *Proc. EUSIPCO*, Edinburgh, U.K., Sept. 1994.
[36] K. Haris, S. N. Efstratiadis, N. Maglaveras, and C. Pappas, "Image noise reduction based on local classification and iterated conditional modes," in *Proc. IWISP*, Manchester, U.K,. Nov. 4–7, 1996.
[37] S. Olsen, "Estimation of noise in images: An evaluation," *CVGIP: Graph. Models Image Process.*, vol. 55, pp. 319–323, July 1993.
[38] L. Vincent, "Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms," *IEEE Trans. Image Process.*, vol. 2, pp. 176–201, Aug. 1993.
[39] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, May 1983.

[40] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[41] J. Beaulieu and M. Goldberg, "Hierarchy in picture segmentation: A stepwise optimization approach," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 150–163, Feb. 1989.

[42] J. H. Ward, "Hierarchical grouping to optimize an objective function," *J. American Stat. Assoc.*, vol. 58, pp. 236–245, 1963.

[43] D. Ballard and C. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[44] X. Wu, "Adaptive split-and-merge segmentation based on piecewise least-square approximation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 808–815, Aug. 1993.

[45] C. van Wyk, *Data Structures and C Programs*. Reading, MA: Addison-Wesley, 1988.

[46] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1990.

[47] P. Santago and H. Gage, "Quantification of MR brain images by mixture density and partial volume modeling," *IEEE Trans. Med. Imag.*, vol. 12, pp. 566–574, Sept. 1993.

[48] K. Ottenberg, "Model-based extraction of geometric structure from digital images," Ph.D. dissertation, Univ. Hamburg, Hamburg, Germany, 1994.

[49] K. Haris and M. Strintzis, "Segmentation of 3D MR image sequences," in *Proc. Int. Workshop on Stereoscopic and Three Dimensional Imaging*, Santorini, Greece, Sept. 1995, pp. 270–274.

Kostas Haris was born in Florina, Greece, in 1971. He received the B.Sc. and M.Sc. degrees in computer science, both from the University of Crete, Herakleion, Greece, in 1992 and 1995, respectively. He is currently a candidate for the Ph.D. degree in medical informatics at the Aristotle University of Thessaloniki.

From 1992 to 1995, he was a Research Assistant at the Institute of Computer Science, Foundation of Research and Technology (FORTH), Herakleion, Greece. Since 1995, he has been a Research Assistant at the Laboratory of Medical Informatics, Aristotle University, Thessaloniki, Greece. He is also a Teaching Associate at the Technological Educational Institution of Thessaloniki, Greece. His research interests include multidimensional image processing, computer vision and medical image analysis.

Serafim N. Efstratiadis (S'85-M'91) received the Diploma degree from Aristotle University, Thessaloniki, Greece, in 1986, and the M.S. and Ph.D. degrees from Northwestern University, Evanston, IL, in 1988 and 1991, respectively, all in electrical engineering.

He served as both Teaching and Research Assistant in the Electrical and Computer Engineering Department, Northwestern University, from 1987 to 1991. He was First Assistant and Head of the digital TV group at the Signal Processing Laboratory, Swiss Federal Institute of Technology, Lausanne, Switzerland, from 1991 to 1992. He was Research Associate at the Information Processing Laboratory, Department of Electrical and Computer Engineering, Aristotle University, Thessaloniki, Greece, from 1993 to 1995. He was adjunct Professor at the American College of Thessaloniki from 1995 to 1998. Since 1996, he has been Research Associate at the Laboratory of Medical Informatics, Medical School, Aristotle University. Since 1997, he has also been a Professor in the Department of Electronics, Technological Educational Institute of Thessaloniki. He has participated in various government and privately funded projects and co-authored many journal and conference articles. His research interests include multidimensional signal processing, motion estimation, monocular and stereo image/video coding, medical image processing and analysis, and multimedia applications.

Dr. Efstratiadis is a member of the Technical Chamber of Commerce of Greece. He has served as a reviewer for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON IMAGE PROCESSING, and *Signal Processing: Image Communication*. He was member of the organizing committee and proceedings co-editor of the International Workshop on Stereoscopic and Three-Dimensional Imaging, Santorini, Greece, 1995, and a member of the Technical Committees of EUSIPCO, SPIE VCIP, and IEEE ICIP conferences.
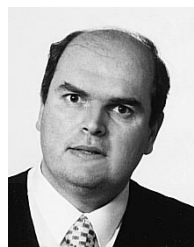
Nicos Maglaveras (S'80–M'87) received the Bachelor's degree in electrical engineering from the Aristotelian University of Thessaloniki, Macedonia, Greece, in 1982, and the Ph.D. and M.Sc. degrees from Northwestern University, Evanston, IL, in 1988, and 1985 respectively, all in electrical engineering, with emphasis in biomedical engineering.

He is currently an Assistant Professor in the Laboratory of Medical Informatics, Aristotelian University, Thessaloniki, Greece. His current research interests are in nonlinear biological systems simulation, cardiac electrophysiology, medical expert systems, ECG analysis, physiological mapping techniques, parallel processing, medical imaging, medical informatics, telematics and neural networks. He has also developed graduate and undergraduate courses in the areas of medical informatics, computer architecture and programming, biomedical signal processing and biological systems simulation. He has over 70 publications in refereed international journals and conference proceedings in the above areas. He has served as a reviewer in the CEC AIM technical reviews, and in a number of international journals.

Dr. Maglaveras has participated in national research projects, the HEALTH TELEMATICS, LEONARDO, TMR, and ESPRIT programs of the CEC. He has been a member of the Technical Chamber of Commerce of Greece, the New York Academy of Sciences, the CEN/TC251-WG5, and Eta Kappa Nu.

Aggelos K. Katsaggelos (S'80–M'85–SM'92) received the Dipl. degree in electrical and mechanical engineering from Aristotle University of Thessaloniki, Greece in 1979, and the M.S. and Ph.D. degrees, both in electrical engineering, from the Georgia Institute of Technology, Atlanta, in 1981 and 1985, respectively.

In 1985, he joined the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, where he is currently Professor, holding the Ameritech Chair of Information Technology. He is also the Director of the Northwestern-Motorola Center for Telecommunications. During the 1986–1987 academic year, he was an Assistant Professor in the Department of Electrical Engineering and Computer Science, Polytechnic University, Brooklyn NY. His current research interests include image and video recovery, video compression, motion estimation, boundary encoding, computational vision, and multimedia signal processing. He is the editor of *Digital Image Restoration* (Berlin, Germany: Springer-Verlag, 1991), co-author of *Rate-Distortion Based Video Compression* (Boston, MA: Kluwer 1997), and co-editor of *Signal Recovery Techniques for Image and Video Compression and Transmission* (Boston, MA: Kluwer, 1998).

Dr. Katsaggelos is an Ameritech Fellow, a Member of the Associate Staff, Department of Medicine, Evanston Hospital, and a Member of SPIE. He is a Member of the Steering Committee of the IEEE TRANSACTIONS ON MEDICAL IMAGING, the IEEE Technical Committee on Visual Signal Processing and Communications, and Multimedia Signal Processing, the Technical Chamber of Commerce of Greece, and Sigma Xi. He was a Member of the Steering Committee IEEE TRANSACTIONS ON IMAGE PROCESSING (1992–1996) and the Technical Committee on Image and Multi-Dimensional Signal Processing (1990–1998). He has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (1990–1992), an Area Editor for *Graphical Models and Image Processing* (1992-1995), the General Chairman of the 1994 Visual Communications and Image Processing Conference (Chicago, IL) and the Technical Program Co-chair of the 1998 IEEE International Conference on Image Processing (Chicago, IL). He is a member of the Board of Governors and the Publications Board of the Signal Processing Society, the IEEE TAB Magazine Committee, and Editor-in-Chief of the IEEE SIGNAL PROCESSING MAGAZINE.