Pattern Analysis & Applications (1999)2:73–81

Algorithms for Fuzzy Segmentation*

Bruno M. Carvalho[1], C. Joe Gau[2], Gabor T. Herman[1] and T. Yung Kong[3]

[1]*MIPG, University of Pennsylvania, Philadelphia, PA, USA;* [2]*Graduate School & University Center, City University of New York, NY, USA;* [3]*Queens College, City University of New York, NY, USA*

Abstract: Fuzzy segmentation is an effective way of segmenting out objects in pictures containing both random noise and shading. This is illustrated both on mathematically created pictures and on some obtained from medical imaging. A theory of fuzzy segmentation is presented. To perform fuzzy segmentation, a 'connectedness map' needs to be produced. It is demonstrated that greedy algorithms for creating such a connectedness map are faster than the previously used dynamic programming technique. Once the connectedness map is created, segmentation is completed by a simple thresholding of the connectedness map. This approach is efficacious in instances where simple thresholding of the original picture fails.

Keywords: Dynamic programming; Fuzzy pattern recognition; Greedy algorithms; Medical imaging, Segmentation; Thresholding

1. INTRODUCTION AND OUTLINE

Segmentation is the process of recognising an object of interest in a picture. Thresholding is not an appropriate method of segmentation if there is some nonuniform shading in the picture, or if what distinguishes the object of interest is not the exact values assigned to the individual pixels, but rather some textural property. In such cases, one can usefully apply fuzzy segmentation.

We call a sequence of pixels in which consecutive pixels are adjacent a *chain*, and a pair of adjacent pixels a *link*. In fuzzy segmentation, the strength of any link is automatically defined based on statistical properties of the links within regions identified by the user as belonging to the object of interest. The strength of a chain is the strength of its weakest link. The fuzzy connectedness between any pair of pixels is the strength of the strongest chain between them. The fuzzy object containing a given pixel at a particular threshold is the set of all those pixels whose fuzzy connectedness to the given one exceeds or equals the threshold.

A potentially time-consuming step in fuzzy segmentation is the calculation of the fuzzy connectedness of all other pixels to the given one. Previously, this was done by a dynamic programming technique. We investigate the usefulness of replacing this by either of two greedy algorithms that have asymptotically better worst-case running times. We experimentally demonstrate that, even on quite small pictures, the greedy algorithms are many times faster than the dynamic programming technique.

Received: 22 October 1998
Received in revised form: 22 November 1998
Accepted: 2 December 1998
* Presented at ICAPR '98

2. THE THEORY OF FUZZY SEGMENTATION

The idea of fuzzy connectedness goes back to the seminal work of Rosenfeld [1]. Our approach is based on that advocated by Udupa and Samarasekera [2], but generalised to arbitrary digital spaces [3].

A digital space is a pair (V,π), where V is a set and π is a symmetric binary relation on V such that V is connected under π. We can obtain a digital space, for example, as follows. Let us consider a tessellation of the plane into identical regular hexagons. We select V as the set of all such hexagons whose centres lie within a much larger closed regular hexagon, and we say that two hexagons in V are in the relation π if, and only if, they are distinct and share an edge. In this case V is finite, as it is likely to be in most practical applications. We refer to elements of V as *spels* (short for 'spatial elements'), and we call two spels which are in the relation π to each other *proto-adjacent*. We observe that for a finite V, (V,π) can be interpreted as a connected graph [4], with V as the set of nodes and π as the set of arcs. We define a *real-valued picture* over the digital space (V,π) as a triple (V,π,f), where f is a function which maps V into the real numbers R. *Binary pictures* are real-valued pictures in which the range of f is the two-element set $\{0,1\}$. One way of identifying an object of interest in a real-valued picture is to produce from it a binary picture (over the same digital space), in which the spels of value 1 are exactly the spels that are contained in the object of interest. This is referred to as *segmentation*.

A commonly used method of segmentation is *thresholding*. For any $t\in R$, the *t-level set* of the real-valued picture (V,π,f) is defined as the set of spels $\{c|f(c)\geq t\}$. Each such t give rise to a *threshold picture*, which is the binary picture (V,π,f_t) with

$$f_t(c) = \begin{matrix} 1, & \text{if } f(c) \geq t \\ 0 & \text{if } f(c) < t \end{matrix} \quad (1)$$

(The selection of the threshold t may be based on principles of fuzzy pattern recognition; see, for example, Pal and Majunder [5, Section 4.5.6]. This is *not* what refer to in this paper as 'fuzzy segmentation'. In our terminology, the process of turning a real-valued picture into a binary picture according to Eq. (1) is called thresholding, irrespective of how the value of the threshold is selected.)

Thresholding is very often not an appropriate method of segmentation. Examples of this are if there is some nonuniform 'shading' in the picture (i.e. the interesting information is superimposed on a background whose darkness slowly varies, as is often the case in many physically obtained digital pictures), and if what distinguishes the object of interest is not the exact values assigned to the individual spels but rather some textural property. We illustrate below that in these cases thresholding fails to provide satisfactory segmentation, but an alternative approach, called *fuzzy segmentation*, produces better results.

A *fuzzy spel affinity* on a digital space (V,π) is a function $\psi : V^2 \rightarrow [0,1]$ (i.e. a function which assigns to each ordered pair of spels a real value not less than 0 and not greater than 1), such that

(i) for all $c \in V$, $\psi(c,c) = 1$, and
(ii) for all c, $d \in V$, $\psi(c,d) = \psi(d,c)$.

The intuitive idea is that the value of the fuzzy spel affinity indicates how close the relationship is (in some sense relevant to the purpose of segmentation) between the spels in question. In other words, it indicates our degree of confidence that after segmentation the two spels should belong to the same class (e.g. they should either both be identified as bone or both be identified as not bone), with the value 1 indicating absolute certainty. The kinds of things that might be considered in defining a fuzzy spel affinity include the nearness of the spels to each other, the similarity of the values assigned to them in the real-valued picture that is to be segmented, and the similarity of the variances of these assigned values in the immediate neighborhoods of the two spels. For the definition to be useful for segmentation in a specific application, it will usually have to take into consideration the overall purpose of performing the segmentation. We will return to this point below and will give specific examples.

The definition of a fuzzy spel affinity is, intentionally, quite general. In particular, there is no requirement of transitivity: it frequently happens that spels c and d have a high fuzzy spel affinity, d and e have a high fuzzy spel affinity, and yet c and e have a low fuzzy spel affinity. For this reason we do not expect a fuzzy spel affinity, by itself, to be immediately useful for segmentation. (It cannot be that both c and d are identified with the same type of tissue and d and e are identified with the same type of tissue, and yet c and e are identified with different types of tissue.) To do segmentation, we first associate with the fuzzy spel affinity ψ *a fuzzy connectedness* function μ_ψ: $V^2 \rightarrow [0,1]$, defined by

$$\mu_\psi(c,d) = \max_{\substack{\langle c^{(0)},\ldots,c^{(k)},\ldots c^{(K)}\rangle \in V^{K+1} \\ c^{(0)}=c,\ c^{(K)}=d}} \min_{1\leq k\leq K} \psi(c^{(k-1)},c^{(k)}) \quad (2)$$

Clearly, $0 \leq \psi(c,d) \leq \mu_\psi(c,d) \leq 1$ (the second inequality follows from the choice $\langle c,d\rangle$ for the sequence from c to d) and so, in particular, $\mu_\psi(c,d) = 1$ if $c = d$. (In the terminology of the fuzzy pattern recognition literature [5,6] $\mu_\psi(c,d)$ is the *grade of membership* of (c,d) in the fuzzy set of 'connected pairs of spels'.)

It is worthwhile to give an intuitive discussion of this last definition. We call an arbitrary sequence $\langle c^{(0)}, \ldots, c^{(K)}\rangle$ of spels, a *chain from* $c^{(0)}$ to $c^{(K)}$. We think of each pair of consecutive elements $(c^{(k-1)}, c^{(k)})$ as a *link* in the chain and of $\psi(c^{(k-1)}, c^{(k)})$ as the *strength of the link*. The *strength of the chain* $\langle c^{(0)}, \ldots, c^{(K)}\rangle$ is defined in Eq. (2) to be the strength of its weakest link, i.e. the minimum of $\psi(c^{(k-1)}, c^{(k)})$ over $1 \leq k \leq K$. Then the fuzzy connectedness of the pair of spels (c,d) is the strength of the strongest chain from c to d. The use of this concept in segmentation depends on the following quite general result. (The proofs of the theorems in this section can be found in Section 5.2 of Herman [3] and will not be reproduced here.)

Theorem 1. For any fuzzy spel affinity ψ on a digital space and for $0 \leq t \leq 1$, the binary relation $K_{\psi,t}$ on V defined by

$$(c,d) \in K_{\psi,t} \Leftrightarrow \mu_\psi(c,d) \geq t \quad (3)$$

is an equivalence relation.

This theorem says that any fuzzy spel affinity and any threshold t partitions the set of spels into equivalence classes. If the threshold is low, then the equivalence classes will tend to be large. In particular, $t = 0$ results in V being the only equivalence class. Generally speaking, as the threshold increases, so will the number of equivalence classes, and the individual equivalence classes will tend to be smaller. If the fuzzy spel affinity is strictly less than 1 for any two distinct spels, then $t=1$ will result in each equivalence class being a singleton set. The equivalence classes of $K_{\psi,t}$ are referred to as *ψt-objects* or, in general, as *fuzzy objects*. These objects have some very desirable properties which we now indicate without a precise definition of our terminology. (For the corresponding mathematically precise statements, see Section 5.2 of Herman [3].) First, a fuzzy object is always connected. Secondly, the boundary between a fuzzy object and a connected subset of its complement is always 'Jordan', in the sense that it is always the boundary between its 'interior' (a connected set containing the fuzzy object) and its 'exterior' (a connected set containing the connected subset of the complement), which partition V, are such that every path from one to the other crosses the boundary.

In practice, we are probably not interested in all the fuzzy objects that arise from a particular choice of fuzzy spel affinity. We are more likely to be concerned with a particular spel and ask which other spels belong to the same fuzzy object as the given one. For example, let us assume that we have found a fuzzy spel affinity appropriate for segmenting bone in Computed Tomography (CT) images. To use this, we may just wish to point at a particular displayed spel which we are pretty sure is bone-containing (probably because of its lightness), and then ask which other spels in the three-dimensional array belong to the same piece of bone. This is where the 'fuzziness' of the fuzzy segmentation comes into play: the fuzzy object containing the given spel is not uniquely determined by the fuzzy spel affinity (as we decrease the threshold, the same spel is a member of a fuzzy object of increasing size). This reflects the typical state of knowledge in a practical application. From our data alone, it is usually impossible to say with absolute certainty which spels belong to the same object as the selected spel. The larger thresholds give us smaller fuzzy objects, but also greater confidence that we are including few spels which in reality would not be judged to be in the same object. We formalise the intuitive discussion of the previous paragraph as follows. For any fuzzy spel affinity on a digital space (V,π) and any spel o, we define the *connectedness map for o* as the real-valued picture (V,π,f), where $f(c) = \mu_\psi(o,c)$. The usefulness of this concept is reflected in the following theorem.

Theorem 2. Let ψ be a fuzzy spel affinity on a digital space (V,π) and t be a real number, $0 \leq t \leq 1$:

(i) The ψt-object that contains a spel o is exactly the t-level set of the connectedness map for o.

(ii) If c is in the t-level set of the connectedness map for o, then the t-level set of the connectedness map for c is the same as the t-level set of the connectedness map for o.

We can restate this result as follows. First, we can find all fuzzy objects that contain a given spel by generating the connectedness map for that spel and then thresholding this map at various levels. Secondly, this procedure is robust: if, instead of finding the connectedness map for o and thresholding at t to get a fuzzy object, we find the connectedness map for any other element of this fuzzy object and threshold that at t, then we end up with exactly the same fuzzy object. In view of this, it is desirable to find an efficient algorithm which, given a fuzzy spel affinity ψ on a digital space (V,π) with a finite V and an $o \in V$, finds the connectedness map for o.

3. ALGORITHMS FOR FINDING CONNECTEDNESS MAPS

From now on we assume that V is finite, and that $\psi(c,d) = 0$ if c and d are distinct but not proto-adjacent spels. The first assumption is necessary for our algorithms to terminate. The second assumption allows us to simplify the description of our algorithms, but most of what follows would be true even without it.

The design strategy that has been adopted in the literature on fuzzy segmentation [2,3,7] in answer to the need stated at the end of the last section is based on the principle of *Dynamic Programming* [8]. Specifically, the following *Dynamic Program for Fuzzy Objects* has been used.

Dynamic Program for Fuzzy Objects

Auxiliary Data Structures: a spel queue O, and a real-valued array f with one element $f(c)$ for each spel c.

1. Initialisation:
   (a) Put o into O.
   (b) Set $f(o) = 1$, and set $f(c) = 0$ if $c \neq o$.
2. Remove an element d from O. For each spel c that is proto-adjacent to d, do the following:
   a Set $v = \min\{f(d),\psi(c,d)\}$.
   b If $v > f(c)$, then
   - Put c into O.
   - Set $f(c) = v$.
3. Check if O is empty.
   If it is, STOP.
   If it is not, go back to step (2).

The rationale for step 2(b) is that if $v > f(c)$ then we have discovered a chain from o to c that is stronger than any previously discovered chain from o to c; we put c in O so that extensions of the newly discovered chain will be considered. Essentially, the same algorithm is implemented in the software system 3DVIEWNIX [9]; the algorithm has successfully segmented a wide variety of medical images [2, 7,10]. It has been found to be an efficacious, robust and acceptably efficient technique, which can provide high quality segmented pictures in spite of the presence of shading and noise; practitioners seem to be quite satisfied with its performance. Nevertheless, it occurred to us that there is a possibility of replacing this algorithm by one which is significantly more efficient.

One possibility of improvement is offered by the use of so-called *greedy algorithms* [8]. In particular, the earlier observation that a digital space with a finite V is essentially a connected graph suggests that some known-to-be efficient graph algorithms may well be applicable here. Observing further that we may consider the cost of the arc (c,d) to be $1 - \psi(c,d)$, we see that there is a possibility of adapting algorithms which find paths of lowest cost to our problem.

One such algorithm which is particular near in its description to the Dynamic Program for Fuzzy Objects is *Dijkstra's algorithm* [8]. To obtain a version of this algorithm that solves our problem we need to make only two changes to the Dynamic Program for Fuzzy Objects: make O a set rather than a queue, and replace the first sentence of step (2) by 'Remove an element d from O for which $f(d)$ is maximal'. (Since O is now a set, in step 2(b), if c is already in O, then 'put c into O' is an empty action; but we still need to update $f(c)$.) An important effect of these changes is that spels which have been removed from O are never put into O again. As opposed to this, the Dynamic Program for Fuzzy Objects may find a stronger chain to d after d has been removed from O (once or repeatedly), in which case subchains starting from d have to be reinvestigated. This is the reason why Dijkstra's algorithm has the potential of being faster than the Dynamic Program for Fuzzy Objects. However, there is a price to be paid: each time we execute step (2) we must find an element d in O for which $f(d)$ is maximal.

In most applications, the number of spels proto-adjacent to a given spel is bounded by a small integer (six, in the case of the above example with hexagonal spels). As described by Cormen et al [8, p.530], in such a case one can use a binary heap implementation of O to achieve an $O(|V|\lg|V|)$ running time. We have been able to prove that even in such cases the worst-case running time of the Dynamic Program for Fuzzy Objects is $\Theta(|V|^2)$. More importantly from the point of view of practical applications, we demonstrate below that Dijkstra's algorithm performs much faster even on pictures of quite limited size.

Another greedy algorithm that we have investigated is *Prim's algorithm* [8] for finding a minimal spanning tree. This is because it can be shown that, if we consider the cost of each arc (c,d) *to be* $1 - \psi(c,d)$, then in any minimal spanning tree the unique path between any two spels is a chain of maximal strength. The performance of this algorithm was found to be very similar to that of Dijkstra's algorithm.

4. APPLICATION TO SEGMENTATION

One of the beauties of fuzzy segmentation is that in many applications an appropriate fuzzy spel affinity can be automatically created by a computer program, based on some minimal information supplied by a user. It is easiest to demonstrate this with an example.

Figure 1(a) is a real-valued picture defined for a V consisting of regular hexagons which are inside a large hexagon (with 60 spels on each side). The rectangular region in the upper half in which the brightness increases slowly from left to right cannot be segmented by thresholding; no choice of threshold would produce a much better result than the binary picture in Fig. 1(b). (We chose to use hexagonal spels partly because the previous papers reporting on fuzzy segmentation [2,7,10] illustrated its use only on pictures with square spels, and we wished to demonstrate the generality of the underlying concepts by choosing spels of a different shape. However, there are actually many reasons to prefer hexagonal spels to square spels; see Herman [3, Chapter 2] for a general discussion, and Preston [11] for why such spels are advantageous in pattern analysis).

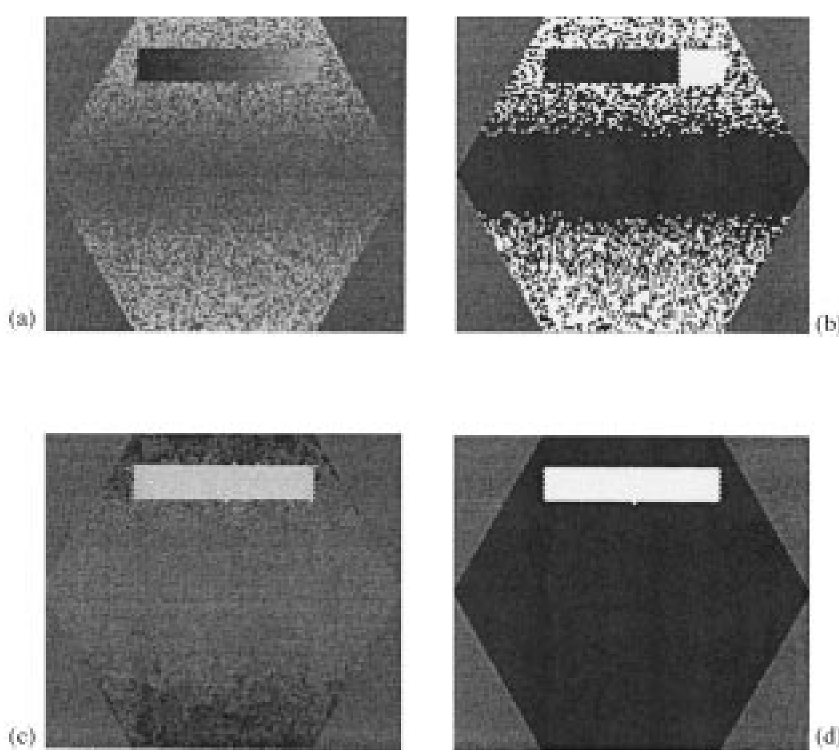


Fig. 1. Example of fuzzy segmentation. (a) A real-valued picture, (b) a binary picture obtained by thresholding, (c) a connectedness map, (d) a binary picture obtained by fuzzy segmentation.

The automation of finding an appropriate fuzzy affinity function for segmentation of such pictures depends upon the following observation: even though a user may not be able to describe the precise nature of the difference between the region of interest and its background in Fig. 1(a), it is easy to identify some spels which are definitely in that region. We can compute some statistics of the spels that are identified, and then the fuzzy spel affinity can be automatically defined on the basis of such information.

To demonstrate this on our example, let us assume (as is the case here and as is also the case in many applications) that the important distinguishing characteristics of regions have to do with the real values assigned to spels in them and also with the likely differences of these values between adjacent spels. Given a real-valued picture (V,π,f) we select the fuzzy spel affinity ψ such that, for $c \neq d$,

$$\psi(c,d) = \begin{cases} 0, & \text{if } (c,d) \notin \pi \\ \frac{1}{2}[g_1(f(c)+f(d)) + g_2(|f(c)-f(d)|], & \text{otherwise} \end{cases} \tag{4}$$

where, for $i \in \{1,2\}$,

$$g_i(x) = e^{-\frac{(x-m_i)^2}{2\sigma_i^2}} \tag{5}$$

It is easy to check that this ψ is indeed a fuzzy spel affinity on (V,π). Appropriate values of the m_i and σ_i can be obtained as follows. The user identifies some regions in the real-valued picture which definitely lie in the object that should be segmented out from its background (such as regions containing only heart muscle in a CT image). Then m_1 and σ_1 can be defined to be the mean and standard deviation, respectively, of $f(c) + f(d)$ over all protoadjacent spels c and d in the identified regions and m_2 and σ_2 can be defined to be the mean and standard deviation, respectively, of $|f(c) - f(d)|$ over all proto-adjacent spels c and d in the identified regions. This implies that for any pair of proto-adjacent spels c and d, their fuzzy spel affinity will be large if both $f(c) + f(d)$ and $|f(c) - f(d)|$ have values that are typical of the identified regions and will be low if neither $f(c) + f(d)$ nor $|f(c) - f(d)|$ has a value typical of the identified regions. In the programs which produced Fig. 1, the regions were defined to consist of a hexagonally-shaped spel selected by the user and the six spels adjacent to it. The user was asked to select two such spels, providing us with a total of 24 pairs of proto-adjacent spels in the identified regions (of seven spels each) to calculate the m_i and σ_i. (This is, of course, only one of many reasonable choices of fuzzy spel affinity; for alternatives, see the ideas presented by Pal and Majunder [5] and Kandel [6].

Figure 1(c) is the connectedness map produced by Dijkstra's algorithm when the two spels provided by the user were near the two ends the rectangular region in the upper half. The brightest spel is the o of the algorithm, which is also one of the two spels identified by the user. Figure 1(d) is obtained by thresholding the connectedness map. It is a fuzzy object that is a much better approximation of the rectangular region than can be obtained by thresholding the original real-valued picture, irrespective of the technique used for selecting the threshold.

Results such as the one shown in Fig. 1 and successful application of the approach to medical data [2,7,10] in cases where other segmentation approaches seemed to be inappropriate have convinced some practitioners of the value of fuzzy segmentation. Unfortunately, there is no publication reporting on a careful evaluation of fuzzy segmentation as compared to other methods of segmentation. The aim of the current paper is not to fill this gap; rather, we also assume that fuzzy segmentation is worth doing (further examples of its usefulness are presented in the next section) and concentrate on the problem of speeding up the process by choosing an appropriate search strategy for producing the connectedness map. We note that once the connectedness map has been obtained, the cost of fuzzy segmentation is the same as that of thresholding.

5. EXPERIMENTAL COMPARISON

In our experimental comparison we used five basic images, each digitized four diffeent ways. One of the basic images is illustrated in Fig. 1(a); the other four are shown in Fig. 2. (These were created to provide us with a variety of images of different natures so that the timing experiments reported in this section become indicative of the expected behavior of our algorithms on images in general.) In Figs 1 and 2 the images are digitized so that there are 60 hexagonal spels on each edge; in the other digitisations the numbers of hexagonal spels on each edge were 30, 40 and 50. On each of the 20 pictures each of the algorithms was run 49 times using all possible pairs of spels from a regular sampling of the picture. In Table 1 we report on the timings. Since the major influence on the running time of each algorithm was the number of spels, the numbers for each algorithm in Table 1 are the sums of the CPU times of 245 runs (49 runs on each of the five images which are digitised in the same way). Since all algorithms find the same connectedness maps, there is no difference in the quality of segmentations obtained by them; this is why they are compared only in terms of their CPU time requirements. (The memory requirements of the Dijkstra's and Prim's algorithms are essentially identical and are never much greater than that of the Dynamic Program for Fuzzy Objects.)

We see from the table that for our test images there is no significant difference between Dijkstra's and Prim's algorithms, but they are both several times more efficient than the Dynamic Program for Fuzzy Objects. We also see that the superiority of Dijkstra's and Prim's algorithms increases with the number of spels, but not as quickly as the theoretical worst-case running times might suggest. (If we used an affinity function that was less expensive to compute, then the superiority of Dijkstra's and Prim's algorithms probably would not be as great.) Since in many applications [2,7,10] the number of spels can be in the order of millions, we expect the running time to be reduced by more than an order of magnitude when either of the greedy algorithms is used instead of the Dynamic Program for Fuzzy Objects.

We illustrate in Figs 3 and 4 the performance of fuzzy

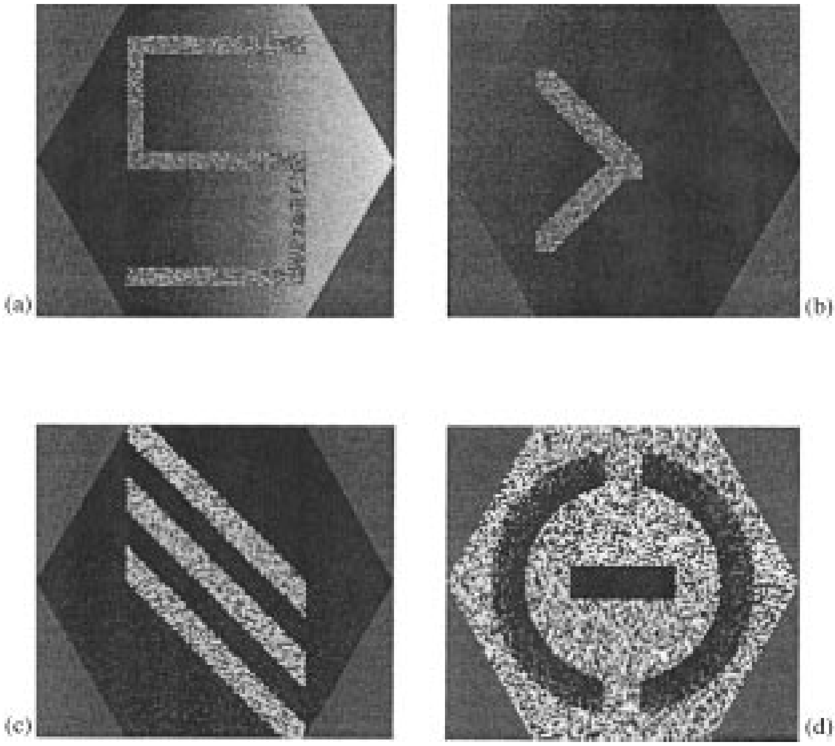


Fig. 2. Four real-valued pictures used in the experimental study.

Table 1. Total CPU time in seconds (measured on a SUN SPARCstation20 workstation) of the algorithms on five images (digitized so that the number of spels is as indicated in the left column) with 49 runs on each picture (DP: Dynamic Program for Fuzzy Objects, DA: Dijkstra's algorithm, PA: Prim's algorithm). Each time is for 245 runs

$\|V\|$	DP	DA	PA	DP/DA
2,611	247	37	36	6.7
4,681	472	67	65	7.0
7,351	837	110	107	7.6
10,621	1,350	164	163	8.2

segmentation on images obtained by Magnetic Resonance Imaging (MRI). We see that both the whole brain and just the white matter can be segmented out more efficaciously by fuzzy segmentation than by thresholding. The timings of the algorithms on these images are similar to those reported in Table 1.

6. DISCUSSION

The examples above, together with those given earlier [2,7,10], show that fuzzy segmentation is an effective approach. Nevertheless, it is not universally applicable. For example, if the region of interest is separated from another region of similar characteristics by a narrow wall, then noise in the image may cause a break in this wall and make the two regions 'leak' into each other. This is illustrated in the attempt to segment out a cardiac left ventricle in an MRI image in Jones and Metaxas [12]. Nevertheless, it is also shown there that a correct segmentation can be achieved when a 'deformable model' is combined with the fuzzy affinity concepts. In the rest of this discussion, we concentrate on the relative capabilities of the various algorithms.

An advantage of Prim's algorithm over the other two is that once a minimal spanning tree is found, it can be used to calculate the fuzzy connectedness between an arbitrary pair of spels at very little cost. However, since regions other than the one which is our current interest should be defined by their own spel affinities, this observation may be of limited practical use. It may be applicable when there are two or more objects with the same characteristics in the data set (such as the two kidneys). Then the minimal spanning tree can be used to inexpensively recalculate the connectedness maps with respect to a spel in each of the objects.

A simple alteration of any of the algorithms allows us to recover, for any spel c, a chain of maximal strength from c to o. This can have useful applications: it can be used to find a path entirely within a region between two user-selected spels in that region (see Fig. 5).

Typically, there will be many strongest chains between any two specified spels. It may be desirable in this case to

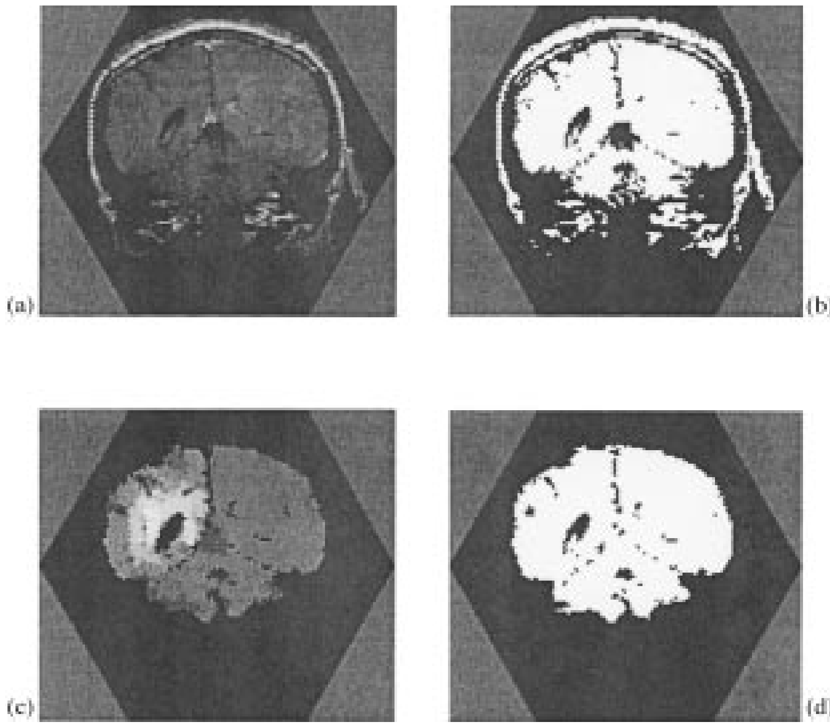


Fig. 3. Example of fuzzy segmentation on a coronal head section obtained from an MRI scan of a patient. (a) The real-valued picture (10,621 spels), (b) a binary picture obtained by thresholding which attempts to identify the brain, (c) a connectedness map obtained by specifying two spels in the brain, (d) a binary picture of the brain region obtained by fuzzy segmentation.

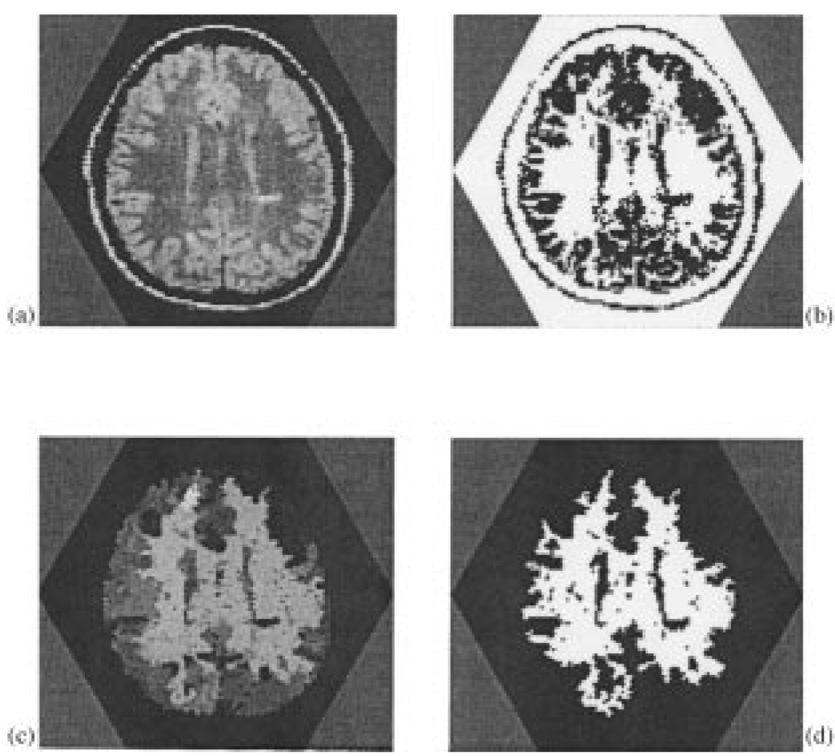
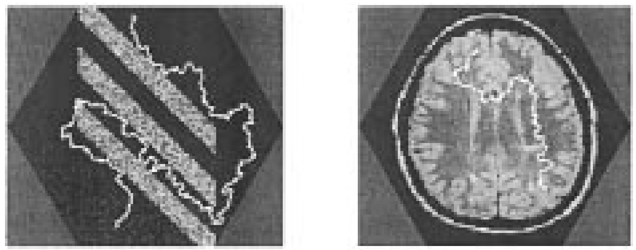


Fig. 4. Example of fuzzy segmentation on a transverse head section obtained from an MRI scan of a patient. (a) The real-valued picture (10,621 spels), (b) a binary picture obtained by thresholding which attempts to segment the white matter (values below the threshold are displayed as white in this picture), (c) a connectedness map obtained by specifying two spels in the white matter region, (d) a binary picture of the white matter region obtained by fuzzy segmentation.



(a) (b)

Fig. 5. The user points at two hexagonal spels of the real-valued picture. Based on these, a fuzzy spel affinity is calculated as explained in the text. In both images, the indicated chain is a strongest possible chain (for the calculated fuzzy spel affinity) between the two user-selected spels (it has been generated using a version of Dijkstra's algorithms that also tries to optimize with respect to a secondary criterion, as explained in the text).

select one among these which is optimal with respect to some secondary criterion. For example, we might try to find (among the strongest chains) a chain for which the product of the strengths of the links is as large as possible. (Note that if all the links between proto-adjacent spels have the same strength and this common strength is less than 1, this secondary criterion aims for the shortest path.) We do not have an efficient algorithm which is guaranteed to find such a chain, but both the Dynamic Program for Fuzzy Objects and Dijkstra's algorithm can easily be adapted to find an approximation to it. We just keep a record, for each spel c that has been visited, of the product of the strengths of links for the chain from o to c that is 'best so far', and we modify step (2) so that when $v = f(c)$ it puts c into O if the new chain to c has a higher link–strength product than the previously best chain to c. (In the case of Dijkstra's algorithm, the heap representation of O also needs to take into consideration the secondary criterion.) We have implemented this option as well; in fact, Fig. 5 was produced using it. We have found that the cost of both algorithms goes up, but while for Dijkstra's algorithm the increase is only 50%, the Dynamic Program for Fuzzy Objects becomes three to four times more expensive. Also, the chain output by Dijkstra's algorithm will always be at least as good with respect to the secondary criterion as the chain output by the Dynamic Program for Fuzzy Objects (but will sometimes be better).

Finally, we note that if our aim is not to get a fuzzy segmentation, but only to find the strongest chain between two given spels c and d, then we need not compute a connectedness map for the entire image. We can just interleave two instances of Dijkstra's algorithm which start at c and d, respectively, and stop both instances as soon as the two connectedness maps meet at any spel. If s is the spel where the two maps meet, then it is not hard to prove that a strongest chain from c to d is obtained when we combine the chains from each of c and d to s that have been found by the two instances of Dijkstra's algorithm.

7. CONCLUSION

The two greedy algorithms discussed in this paper are clearly superior to the currently used Dynamic Program for Fuzzy Objects in the fuzzy segmentation process.

Acknowledgements

This research has been supported by NIH Grant HL28432 (BMC and GTH), NSF Grant DMS96122077 (GTH), and CAPES-BRASILIA-BRAZIL (BMC). The authors are grateful to Eric Tsai and Samuel Matej for programs which were incorporated into the final package used in our experiments and to Michel Habib for suggesting the use of optimal spanning trees.

References

1. Rosenfeld A. Fuzzy digital topology. Information and Control 1979;40(1):76–87
2. Udupa JK, Samarasekera S. Fuzzy connectedness and object definition: Theory, algorithms and applications in image segmentation. Graphical Models and Image Processing 1996;58(3):246–261
3. Herman GT. Geometry of Digital Spaces. Boston, MA: Birkhauser, 1998
4. Harary F. Graph Theory. Reading, MA: Addison-Wesley, 1969
5. Pal SK, Majunder DKD. Fuzzy Mathematical Approach to Pattern Recognition. New Delhi, India: Wiley, 1986
6. Kandel A. Fuzzy Techniques in Pattern Recognition. New York: Wiley, 1982
7. Udupa JK, Wei L, Samarasekera S, Miki Y, Van Buchem MA, Grossman RI. Multiple sclerosis lesion quantification using fuzzy-conectedness principles. IEEE Trans Medical Imag 1997;16(5):598–609
8. Cormen TH, Leiserson CE, Rivest RL. Introduction to Algorithms. Cambridge, MA: MIT Press, 1990
9. Udupa JK, Odhner D, Samarasekera S, Goncalves R, Iyer K, Venugopal K, Furuie S. 3DVIEWNIX: An open, transportable, multidimensional, multimodality, multiparametric imaging software system. Proc SPIE Medical Imaging 1994: Image Capture, Formatting, and Display, vol 2164, SPIE Press, 1994, pp 58–73
10. Udupa JK, Odhner W, Yien J, Holland G, Axel L. Automatic clutter-free volume rendering for MR angiography using fuzzy conectedness. Proc SPIE Medical Imaging 1997: Image Processing, vol 3034, SPIE Press, 1997, pp 114–119
11. Preston Jr. K. Multidimensional logical transforms. IEEE Trans Pattern Analysis Machine Intell 1983;5(5):539–554
12. Jones TN, Metaxas DN. Segmentation using deformable models with affinity-based localization. Proc Conference on Computer Vision, Virtual Reality and Robotics in Medicine/Medical Robotics and Computer Assisted Surgery. Lecture Notes in Computer Science, vol. 1205. Berlin: Springer-Verlag, 1997, pp 53–62

Bruno M. Carvalho received the BSc degree in computer science from the Federal University of Rio Grande de Norte, Brazil, in 1992, the MSc in computer science from the Federal University of Pernambuco, Brazil, in 1995. From 1995 to 1997 he was a lecturer in the Department of Computer Science, Federal University of Rio Grande do Norte, Brazil. Since 1997, he has been a PhD student in the Department of Computer and Information Science, University of Pennsylvania, and his main interests are medical imaging, computer graphics and computer vision.

C. Joe Gau is an Associate Urban Designer for the New York City Department of Housing Preservation and Development, and a doctoral student in computer science at the Graduate School and University Center of the City University of New York. His doctoral research is directed by Dr T Yung Kong, and is focused on problems of digital geometry and digital image processing. He received a master's degree in computer science from Queens College of the City University of New York in 1994. He previously received a bachelor's degree in mathematics from Cheng–Kung University, Taiwan, in 1971, a master's degree in mathematics from the State University of New York at Buffalo in 1977, and a bachelor's degree in architecture, also from the State University of New York at Buffalo, in 1980.

Gabor T. Herman received the MSc degree in mathematics from the University of London, England, in 1964, the MS degree in engineering science from the University of California, Berkeley, in 1966, and the PhD degree in mathematics from the University of London, England, in 1968. From 1969 to 1981, he was with the Department of Computer Science, State University of New York at Buffalo, where in 1976 he became the Director of the Medical Image Processing Group. Since 1981, he is a Professor in the Medical Imaging Section of the Department of Radiology at the University of Pennsylvania. He is involved editorially with a number of journals – in particular he was for a while Editor-in-Chief of the *IEEE Transactions on Medical Imaging* – and is the author of numerous articles and books on medical imaging and related topics, including *Geometry of Digital Spaces*, published by Birkhauser in 1998.

T. Yung Kong is a Professor of Computer Science at Queens College of the City University of New York. His main research interest is in topological and geometrical problems of image processing and computer graphics, especially digital topology. A member of the Queens College computer science faculty since 1989, he previously held faculty positions in computer science at Ohio University, and in mathematics at City College of the City University of New York. He received a BA in mathematics from the University of Cambridge, England (with double first class honors) in 1981, and received a DPhil from the University of Oxford, England in 1986 for research on digital topology at the Oxford University Computing Laboratory.

Correspondence and offprint requests to: Gabor T. Herman, Medical Image Processing Group, Department of Radiology, University of Pennsylvania, Blockley Hall, Fourth Floor, 423 Guardian Drive, Philadelphia, PA 19104-6021, USA. Email: gabor@mipg.upenn.edu