

# Demarcation of Aneurysms Using the Seed and Cull Algorithm

Robert A. McLaughlin and J. Alison Noble

Medical Vision Laboratory, Dept. Engineering Science, University of Oxford,  
Oxford, England  
`{ram,noble}@robots.ox.ac.uk`

**Abstract.** This paper presents a method to demarcate the extent of an intracranial aneurysm given a 3-D model of the vasculature. Local shape descriptors are grouped using a novel region-splitting algorithm. The method is used to automatically estimate aneurysm volume. Results are presented for four clinical data sets.

## 1 Introduction

An intracranial aneurysm is a localised persistent dilation of the wall of a blood vessel in the brain. There are several possible treatments, including coiling and clipping. The treatment appropriate depends upon factors such as aneurysm volume and neck size. It is common to image the aneurysm using a 3-D modality such as 3-D X-ray angiography or magnetic resonance angiography (MRA). Such scans can be segmented to derive a 3-D model of the vasculature [1] [2]. Given such a 3-D model, it would be useful to automatically demarcate the aneurysm, identifying where it connects to the vessel. This could allow automatic estimation of volume and neck size, aiding the clinician to choose the appropriate treatment.

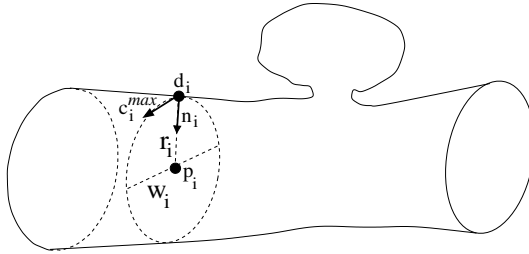
Several researchers have suggested methods to demarcate aneurysms by locating the aneurysm neck. Van Der Weide et al. [3] computed distances to the surface of the vasculature from a point selected inside the aneurysm, detecting the neck as a discontinuity in these distances. Wilson et al. [4] developed a variant of this idea, using a series of such distance functions from points along a user-defined spline. As noted in [4], such methods experience difficulties in aneurysms with wide necks. We propose a different approach, defining shape descriptors over a surface mesh and using region-splitting to identify the section of the mesh covering the aneurysm. The region-splitting algorithm has been termed the *Seed and Cull algorithm*. The method is applied to four clinical data sets.

## 2 Method

### 2.1 Local Shape Descriptors

The method begins with a surface mesh defined over a 3-D model of the vasculature. At each vertex in the mesh, a local description of vessel shape is computed,

illustrated in figure 1. Taking the unit surface normal  $n_i$  to the mesh at a particular vertex  $d_i$ , a ray is extended from  $d_i$  into the vessel, measuring the distance to the opposite side of the vessel. Halving this value gives an estimate of vessel radius  $r_i$  at  $d_i$ . Next, the algorithm estimates the vessel centre  $p_i$  as  $p_i = d_i + r_i \cdot n_i$ , and the direction of maximum absolute curvature  $c_i^{max}$ . A vector is extended from  $p_i$  in the directions  $c_i^{max}$  and  $-c_i^{max}$ , and the distance to the vessel surface is measured in each direction. Adding these two distances together gives an estimate of the vessel width  $w_i$  in a direction perpendicular to  $n_i$ . The two values  $(r_i, w_i)$  characterise the data point  $d_i$ , and are computed for every vertex in the mesh. The task is now to group points that lie on the aneurysm, and distinguish these from points on adjoining vessels.



**Fig. 1.** Local shape descriptors: vessel radius  $r_i$  and the perpendicular width  $w_i$ .

## 2.2 Seed and Cull Algorithm

Region-growing [5] and region-splitting [6] algorithms typically require the number of regions to be known a priori. Such methods are insufficient for aneurysm demarcation. When considering the entire vasculature, variation in size and shape between different sections of blood vessel tend to be greater than the variation between vessel and aneurysm, in all but the extreme cases of giant aneurysms. Hence it is unrealistic to group all vessel points into a single region. Thus we require an algorithm to segment points on the surface mesh into an unknown number of regions, where each region will correspond to a section of vessel or an aneurysm. One solution is to use an augmented Markov Random Field, where an extra region label is defined for new regions, and a parameter is pre-set to define the probability assigned to this extra state. Such a method was proposed in [7] for texture segmentation. We have adopted an alternative approach which adaptively uses Parzen windows [8] to estimate region statistics.

The Seed and Cull algorithm begins by assigning all points to a single region. A point is then selected somewhere on the mesh and a new region seeded, growing it as described in the next section. If the region does not grow, then it is culled and a different seed is chosen. The novelty of this algorithm lies in the mechanism by which regions are grown when appropriate, while being retarded when they will not improve the segmentation.

### 2.3 Growing Regions

Consider classifying the point  $d_0$  shown in figure 2. We add the restriction that it must be of the same class as one of the other five data points that lie within the neighbourhood of radius  $r_{classify}$ . Each point  $d_i$  has a vector  $v_i = (r_i, w_i)$  associated with it, where  $(r_i, w_i)$  are the local shape descriptors described in Section 2.1. For each class  $C_i$ , we can compute a probability distribution over these numeric values  $P(v_0|d_0 \in C_i)$ . Using this, we classify the point  $d_0$  using a Bayesian framework by computing the maximum a posteriori estimate for  $C_i$ .

$$P(d_0 \in C_i|v_0, D, C) \propto P(v_0|d_0 \in C_i, D, C).P(d_0 \in C_i|D, C) \quad (1)$$

where  $D = \{d_1, d_2, d_3, d_4, d_5\}$  denotes the set of nearby data points and  $C = \{d_1, d_2 \in C_1; d_3, d_4 \in C_2; d_5 \in C_3\}$  denotes which class each point is currently assigned to. By assuming that the probability assigned to the numeric value is statistically independent of data points in a particular neighbourhood (the  $D$  and  $C$  terms), we can replace  $P(v_0|d_0 \in C_i, D, C)$  with  $P(v_0|d_0 \in C_i)$ . The term  $P(d_0 \in C_i|D, C)$  is a prior probability that  $d_0$  belongs to class  $C_i$ . We have chosen a prior directly proportional to the number of data points of each class within the neighbourhood, although others are possible.

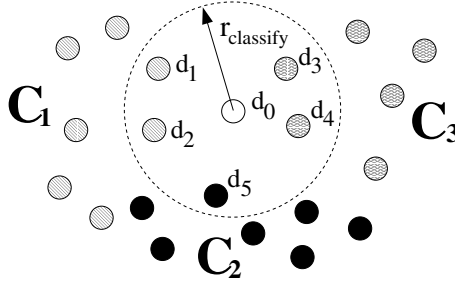
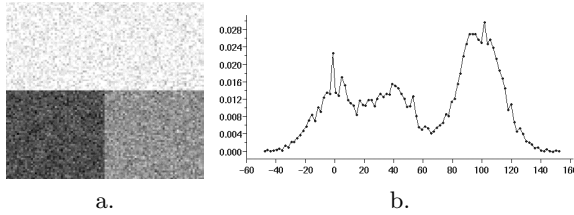


Fig. 2. Point and neighbourhood for classification.

### 2.4 Seeding New Regions

We will illustrate the seeding of new classes with the synthetic example shown in figure 3a, where each point  $d_i$  in the data set has an intensity value  $v_i$ . The method described here is directly applicable to the case of aneurysms and blood vessels, where we are segmenting points on a surface mesh, and where each point has a 2-D numeric value  $(r_i, w_i)$ . However, it is conceptually easier to understand the algorithm using the synthetic example shown in figure 3a.

The method begins by assigning all pixels to a single class  $C_0$  and evaluates the probability distribution over the intensity values  $P(v_j|d_j \in C_0)$ , as shown in figure 3b. Note that there is a peak in the distribution corresponding to each class. To generate this probability distribution, a histogram of the numeric



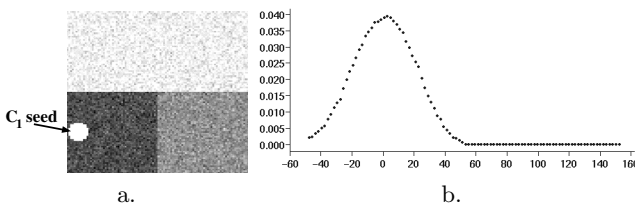
**Fig. 3.** a.) Three groups synthetic data. b.) Initial probability  $P(v_j | d_j \in C_0)$ .

values is computed and then smoothed using Parzen windows [8]. This involves smoothing the histogram by convolving the values with a kernel function. A common choice of kernel function is the Gaussian, although others are possible.

An important issue arises in the selection of variance for the Gaussian kernel function. This will greatly affect the probability distribution produced. When the histogram comprises few values, it is appropriate to use a large variance, resulting in heavy smoothing. If the histogram consists of a large number of values, then less smoothing is desirable and a small variance is appropriate.

In the Seed and Cull algorithm, the variance is a function of the number of values in the histogram, which equals the size of the class. A large variance is used for small classes, and a small variance for large classes. We have chosen the variance to equal the inverse square of an affine function of class size, although other functions are possible. As a new class  $C_i$  grows in size, progressively smaller variances are used in evaluating the probability  $P(v_j | d_j \in C_i)$ . It is this change in variance that will allow us to promote the growth of a class under certain conditions, and retard the growth of unnecessary regions.

Returning to the data shown in figure 3a, a new class is seeded by choosing a point, defining a neighbourhood of radius  $r_{seed}$  around it and assigning all points within the neighbourhood to the new class  $C_1$ , as shown in figure 4a. In figure 4b, we show  $P(v_j | d_j \in C_1)$ , the distribution over intensity values for this new class. Note that the probability distribution is much smoother than that shown in figure 3b, as a much large variance was used for this new, small class.

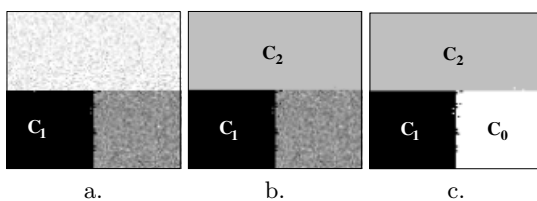


**Fig. 4.** a.) Seed for class  $C_1$ . b.) Initial probability  $P(v_j | d_j \in C_1)$ .

As the new class only contains points from one distribution, the probabilities assigned to this distribution are larger than those in  $C_0$ , which divides its probabilities between three distributions. Note that the maximum in figure 4b

for  $P(d_j \in C_1|v_j, D, C)$  is 0.040, while the corresponding value at  $v_j = 0$  for  $P(d_j \in C_0|v_j, D, C)$  is only 0.024, as shown in figure 3b. Choosing a data point  $d_j \in C_0$  that comes from this distribution and re-evaluating its classification will re-assign it to  $C_1$ , provided the prior probabilities are approximately equal.

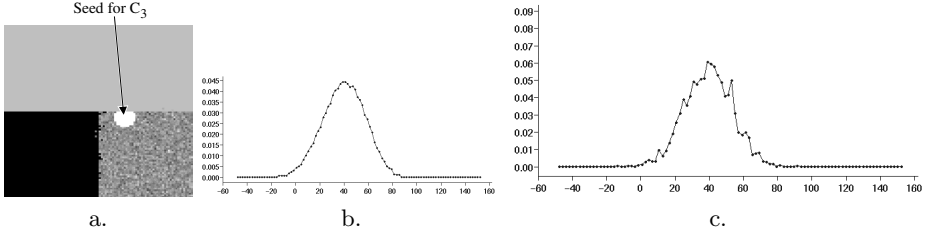
The algorithm now proceeds to ‘grow’ this distribution. Recall from Section 2.3, the assumption that the class of a data point was only directly affected by points within a neighbourhood of radius  $r_{classify}$ . Thus all points  $\{d_j\}$  within a radius  $r_{classify}$  of the new class  $C_1$  are tested to decide whether they should be re-classified. This is recursively repeated for each point  $d_j$  that is re-classified to class  $C_1$ . Note that only points currently assigned to class  $C_0$  will be tested. Once a point is re-classified as belonging to  $C_1$ , it will not be changed. This process is continued until no more points are added to class  $C_1$ , and no more remain that need to be tested. The result at convergence is shown in figure 5a.



**Fig. 5.** a.) Classification after  $C_1$  converges. b.) Classification after  $C_2$  converges. c.) Final classification.

As each point is removed from  $C_0$  and added to  $C_1$ , the probability distributions are re-evaluated. The variance used when computing  $P(v_j|d_j \in C_0)$  increases as the class shrinks, and decreases for  $P(v_j|d_j \in C_1)$  as the class grows. Thus  $C_1$  will improve its model of the distribution of numeric values  $v_j$ , and this distribution will be removed from the three distributions shown in figure 3b for class  $C_0$ . This process is then repeated, seeding a new class  $C_2$ . Once  $C_2$  has converged, the data will be classified into classes  $C_0$ ,  $C_1$  and  $C_2$  as shown in figure 5b. It is important that the segmentation algorithm recognise that no further classes should be introduced. The algorithm achieves this because of its adaptive choice of variance as a function of class size.

The algorithm will seed a new class  $C_3$  as shown in figure 6a, and the initial points in the neighbourhood will give a distribution for  $P(v_j|d_j \in C_3)$  as shown in figure 6b. However, this class will fail to grow in the way that  $C_1$  and  $C_2$  did. First note that because  $C_3$  contains less points than  $C_0$ , the probability distribution was generated by convolving with a Gaussian with a larger variance than was used for  $C_0$ . Hence  $P(v_j|d_j \in C_3)$  is more smoothed, resulting in lower probabilities for values from the distribution. Note that the maximum for  $P(v_j|d_j \in C_3)$  shown in figure 6b is 0.045, while the maximum for class  $C_0$  is greater than 0.06, as shown in figure 6c. This in turn reduces the probabilities  $P(d_j \in C_3|v_j, D, C)$ . As the algorithm attempts to grow  $C_3$ , most data points



**Fig. 6.** a.) Seed for class  $C_3$ . b.) Initial probability  $P(v_j | d_j \in C_3)$ . c.) Probability  $P(v_j | d_j \in C_0)$  when  $C_3$  is seeded.

will not be re-classified from  $C_0$  to  $C_3$ , instead remaining in  $C_0$ . In this implementation, we set a threshold that requires a class to grow to at least three times its original seed size. Classes that fail to meet this criterion are ‘culled’ and discarded. In this way, we avoid introducing excess classes. The algorithm will continue attempting to seed new classes on each point left in  $C_0$ , but each new class will be culled. The final segmentation is left as shown in Fig 5c.

The above algorithm extends, without conceptual change, to the case of segmentation using local shape descriptors  $(r_i, w_i)$  instead of intensities, and where the points are defined on a surface mesh instead of as pixels in an image. At completion, the mesh will be separated into regions, with the aneurysm separated from its adjoining vessels.

### 3 Experiment

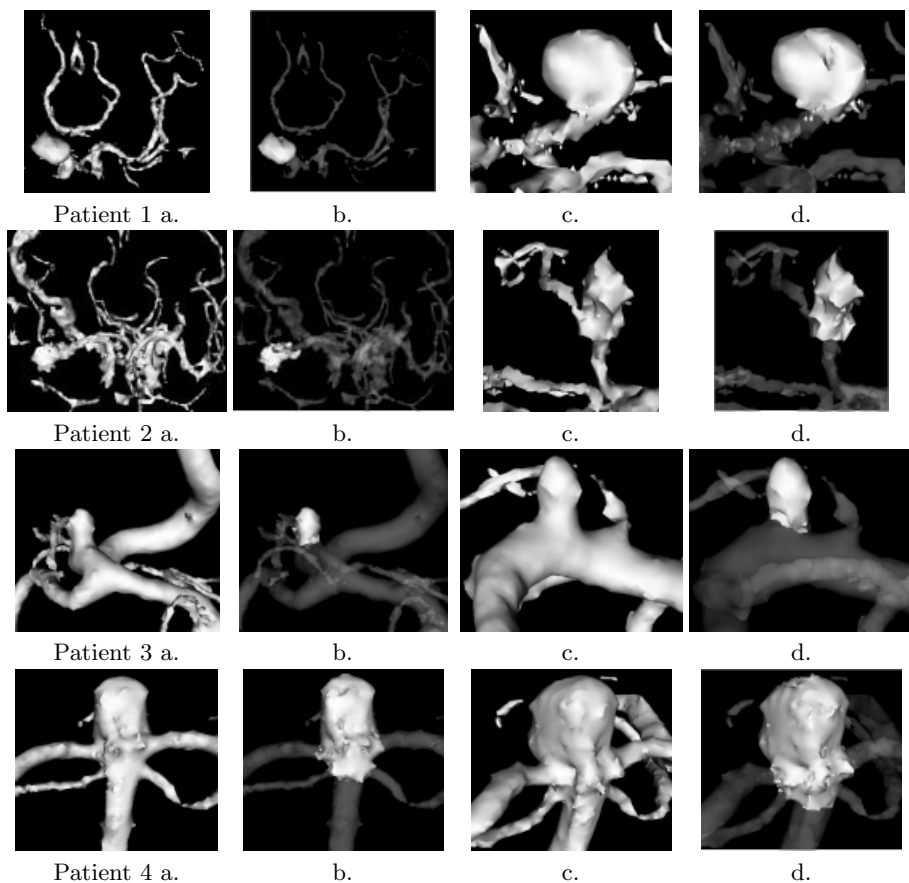
We applied the method to four clinical examples shown in figure 7. The data sets for Patients 1 and 2 were segmented from phase contrast MRA data with voxel size  $0.78 \times 0.78 \times 1.5\text{mm}$ . They were segmented using both flow speed and direction information, as detailed in [1]. The data sets for Patients 3 and 4 were generated from 3-D X-ray angiographic data with voxel resolution  $0.25 \times 0.25 \times 0.25\text{mm}$ , and segmented using thresholding. For each data set, a surface mesh was generated and segmented by applying the Seed and Cull algorithm to local shape descriptors. Each voxel within the vasculature was then assigned to the same group as the closest mesh point. The group covering the aneurysm was identified by a user and the volume of that group computed. These results were compared against aneurysm volumes obtained from a manual segmentation, where each slice was segmented by hand.

### 4 Results

Results of the aneurysm demarcation algorithm are shown in figure 7, where the voxels forming each aneurysm have been highlighted. Two different views are given of each example. Table 1 compares the automatic volume estimates with those from manual segmentation. In each of the four data sets, the aneurysm

was identified as a single region. Note that with Patient 4, the edge of the giant aneurysm was identified as approximately halfway through the intersection of the five adjoining vessels. The poorly defined neck of this aneurysm made this example particularly difficult. Wilson et al. [4] noted that neck-based demarcation methods will exhibit similar problems with such examples.

The parameter  $r_{classify}$  was chosen empirically, with one value being set for the MRA data sets, and a second value appropriate for the two 3-D X-ray angiographic data sets. The value was found to depend upon both the resolution of the data, and the scale of the features being demarcated. Results were found to be robust under minor perturbations of the value of  $r_{classify}$ .



**Fig. 7.** Aneurysm demarcation for four patients. a, c.) 3-D visualisation of vasculature. b, d.) Aneurysm highlighted in white.

**Table 1.** Results of volume estimation, with error as a percentage of the manual volume estimate.

Patient #	Automatic volume (mm <sup>3</sup> )	Manual volume (mm <sup>3</sup> )	% error
1	2120	2180	2.8
2	915	991	7.7
3	13.7	14.5	5.5
4	305	270	13.0

## 5 Conclusion

We have outlined a method to demarcate intracranial aneurysms. Using a surface mesh defined over a 3-D segmentation of the vasculature, local shape descriptors were computed and grouped using a novel region-splitting method referred to as the Seed and Cull algorithm. Results for four clinical data sets have been presented, generated from both 3-D X-ray angiographic data and phase-contrast MRA data. The results have demonstrated the applicability of this method to aneurysms of different sizes and shapes. Volume estimates from this method were compared against those obtained manually, with an average error of 7.2%.

## Acknowledgements

The 3-D X-ray angiographic data was kindly provided by GE Medical Systems. This work was supported by EPSRC grant GR/M55008.

## References

1. Chung, A., Noble, J.: Fusing magnitude and phase information for vascular segmentation in phase contrast mr angiograms. In: Proc. MICCAI. (2000) 166–175
2. Chung, A.C.S., Noble, J.A.: Statistical 3D vessel segmentation using a Rician distribution. In: Proc. MICCAI. (1999) 82–89
3. van der Weide, R., Zuiderveld, K., Mali, W., Viergever, M.: CTA-based angle selection for diagnostic and interventional angiography of saccular intracranial aneurysms. IEEE Transactions on Medical Imaging **17** (1998) 831–841
4. Wilson, D., Royston, D., Noble, J., Byrne, J.: Determining x-ray projections for coil treatments of intracranial aneurysms. IEEE Transactions on Medical Imaging **18** (1999) 973–980
5. Adams, R., Bischof, L.: Seeded region growing. IEEE Transactions on Pattern Analysis and Machine Intelligence **16** (1994) 641–647
6. Horowitz, S.L., Pavlidis, T.: Picture segmentation by a directed split-and-merge procedure. In: Second International Joint Conference On Pattern Recognition, Copenhagen, Denmark, IEEE (1974) 424–433
7. Kervrann, C., Heitz, F.: A markov random field model-based approach to unsupervised texture segmentation using local and global spatial statistics. Technical Report 2062, INRIA, France. Available from <http://www.inria.fr> (1993)
8. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford, England (1999)