Hierarchical Segmentation Satisfying Constraints

L.D. Griffin, A.C.F. Colchester, S.A. Röll, C.S. Studholme

Dept. of Neurology, Guys Hospital, London SE1 9RT
l.griffin@umds.ac.uk

A new hierarchical segmentation algorithm is described. Its computational complexity and memory requirements are detailed, showing it to be practicably applicable to images of useful size. A simple modification of the algorithm adapts it to produce hierarchical segmentations that satisfy a constraint set. Results are given showing that this adapted algorithm can be used as the basis of a semi-automatic object definition tool or as the interface between a low-level image description module and a high-level module coding for knowledge and expectation.

1 Introduction

The goal of completely automated segmentation is still distant for most applications where the environment cannot be manipulated to produce especially simple images. Instead, either a purely manual technique, or an automatic stage followed by editing[9,15], is used. Even with a system that perfectly delivered the image objects seen by a naive, and otherwise average, observer; manual editing would still be necessary as sometimes domain (*e.g.* anatomical) knowledge or application requirements supplement or override what is present in the image. It is therefore essential to develop techniques for editing/refining/constraining automatic segmentations. In addressing this problem we are also addressing the very general problem of how high-level computational modules coding for knowledge and expectation should interact with low-level modules which build descriptions of the image.

We have previously argued[7,8] that an important property of an image description is that it should express object/sub-object relationships. This is particularly the case in medical imaging because much anatomical knowledge is expressed in this form. A description that fulfils this requirement is hierarchical segmentation, which describes an image as a set of regions, with each region being composed of sub-regions, which are themselves composed of sub-regions, and so on. In this paper we present our algorithm for computing hierarchical segmentations of images and then describe how it has been extended so that the resulting description satisfies constraints of quite a general character - *e.g.* there should be a region containing points $\boldsymbol{a}, \boldsymbol{b}$ and $\boldsymbol{c}$ but not containing $\boldsymbol{x}, \boldsymbol{y}$ and $\boldsymbol{z}$. We present applications of the technique with both user-supplied and automatically generated constraints.

BMVC 1994 doi:10.5244/C.8.13

2 Hierarchical Segmentation

A hierarchical segmentation of an image is a tree structure by inclusion of connected image regions.

2.1 Hierarchical Clustering

Hierarchical segmentation algorithms may be distinguished in various ways: they may be agglomerative or divisive; they may use local or global merging; and they may be binary or n-ary.

Agglomerative algorithms construct hierarchies bottom-up by iteratively clustering from pixels to root; divisive algorithms proceed top-down by recursive sub-division of the image[17]. We use an agglomerative approach, since divisive algorithms are of greater computational complexity due to their inherent non-locality.

Our algorithm, like other agglomerative algorithms, maintains a current partition of the image that is altered during the course of the clustering. Initially, the partition is the finest possible with a separate region for each pixel. The partition is then iteratively simplified by merging pairs of adjacent regions. Eventually, as a result of merging, the partition consists of a single region (the entire image) at which point the process is complete. During clustering, information of the form - "r is a sub-region of s" - is accumulated in a hierarchy structure. At each iteration the choice of regions to merge is dictated by an edge strength calculated for each pair of adjacent regions. Other authors[1,5] merge only the pair of regions separated by the globally weakest edge, which is a global merging strategy. We have opted for a local merging strategy, meaning that we merge those pairs of regions with locally weak separating edges. An edge between two regions is locally weak if it is weaker than any other edge involving one of the two regions. Local merging is computationally faster than global merging and is therefore preferred.

We have found a binary hierarchy (0 or 2 sub-regions) to be rather inflexible and the more general n-ary hierarchy (variable number of sub-regions) to be a more faithful representation of image structure. We explain this with an example. Consider a three element graph a-b-c to be clustered. There are three possible hierarchies -

H1	{a,b,c}		{a}	{b}	{c}
H2	{a,b,c}	{a,b}	{a}	{b}	{c}
H3	{a,b,c}	{b,c}	{a}	{b}	{c}

If we are restricted to binary hierarchies then **H1** is inadmissible. This is a problem if our hierarchy preference criterion has an associated uncertainty;

as a consequence of, for example, the finite precision of the input data. Such an uncertainty might make it illusory (and hence misleading) to distinguish between hierarchies **H2** and **H3**, in which case **H1** should be preferred.

Previously[7,8], we have constructed n-ary hierarchies using a multiple merge scheme, in which clusters of two or more elements can be formed. It is difficult to incorporate constraints within such a scheme. Instead, the current algorithm collapses a binary hierarchy, formed by pair-wise merging, into an n-ary hierarchy[5]. Collapsing is achieved by removing regions from a binary hierarchy, thus producing an n-ary hierarchy *e.g.* **H2**-{a,b}=**H1**. Our choice of regions to delete is based on consideration of the vertical pattern of merge costs (*i.e.* strength of edges overcome in forming regions) across the hierarchy. We require that these costs should be strictly increasing as one ascends the hierarchy *i.e.* the individual sub-regions of a region should be individually more homogenous than the entire region. It is simple to detect such cases and remove the corresponding regions.

2.2 Edge Strength Calculation

Two properties that are known to lead to 'good' regions are homogeneity and stability of the bounding edge[12]. Homogeneity can be defined in a relative form as the condition that parts of an object should be more similar to each other than they are to elements outside the region. Stable edges are those whose position does not change greatly with blurring (or similarly, increased viewing distance). Relative homogeneity is enforced by using an edge strength defined as the absolute difference between the mean luminances of the two regions. Locally weak edges, thus defined, are pairs of regions whose luminance level is more similar than that of either of the regions to any other of their neighbours. We define the stability of an edge by looking at its trajectory over (Gaussian) scale[11]. Edges that have a vertical trajectory (equating scale with height) tend to be perceptually strong[2]; edges with a flatter trajectory seem to have less perceptual significance. The angle of this trajectory (relative to the constant scale plane) is $\mathbf{atan}(|\nabla L| / |\nabla^2 L|)$, where L is the luminance. We combine these two considerations together into a single edge measure $|\nabla L|\,\mathbf{atan}(|\nabla L| / |\nabla^2 L|)$ which reflects both the strength and the stability of the edge. We have described this edge strength previously[7] and used it with our earlier hierarchical segmentation algorithm, it is also effective with the new algorithm. The gradient term ($|\nabla L|$) is the absolute difference between the mean luminances of the two regions, the Laplacian term ($\nabla^2 L$) is the average of the mean Laplacians of the two regions. To make this calculation efficient the size, mean luminance and mean Laplacian of each region is stored. When a new region is created these attributes are readily calculated from the attributes of its constituent regions.

2.3 Representation of Adjacency

Previously we have used an implementation of a general region-adjacency graph structure for representing the current partition that is simplified during clustering. This has the advantages of making it trivial to determine which regions are adjacent and guaranteeing that we need only perform the (computationally expensive) edge strength calculation once per edge. However, it has the disadvantages of being rather laborious to update as a result of merging and being inefficient in memory. The inefficiency is due to the fact that the data structures can represent any graph whereas they only need to be able to represent any planar graph. A simpler scheme is to label each pixel with the currently active region to which it belongs. Edges do not then have to be represented explicitly and can be detected by scanning the image looking for adjacent pixels with different labels. This is more memory efficient and leads to a simpler algorithm. The disadvantage is that the edge strength calculation is repeated for each pixel crack making up an edge between two regions. A synthesis of these two approaches is possible. It relies on the fact that given a planar graph it is possible to assign each link of the graph to one of its end-nodes in such a way that each node has at most three edges assigned to it. This constraint can be utilized by using a fixed memory approach to representing region adjacency. We allow three adjacency pointers per region. As the label image is scanned and an edge is discovered this fact is recorded by setting one of the pointers in one of the regions to point to the other region. If the same pair of regions is encountered later in the scan of the image then this fact is readily detected and the edge strength is not unnecessarily recomputed. This innovation has allowed us to decrease the memory usage of the algorithm without increasing the computational complexity. The average time required to construct a hierarchical segmentation of a 128^2 image is 10.4s on a *SPARC IPX*; approximately 74 bytes/pixel are required for this computation.

3 Satisfying Constraints

The form of constraint which concerns us is: "there is an object containing internal points ***a***,***b***,... but not containing external points ***x***,***y***,..." Such a constraint may be understood as a partial labelling: the internal points are labelled with one label, the external points with another label and all other points are unlabelled. The problem then is how to extend the partial labelling to a full labelling which will completely define the required object.

The hierarchical segmentation algorithm is easily adapted to perform this task by preventing the formation of regions containing both internal and external points. This is difficult to achieve in a multiple merging scheme but easy in the binary merging scheme that we use. It is accomplished by treating as non-existent those edges that lie between pairs of adjacent regions, one of which contains an internal point and the other an external point. The process terminates with a partition in which every region contains an internal or an external point, but not both, and adjacent regions are never both internal or both external. Collapsing the binary hierarchy into an n-ary hierarchy does not present any additional problems.

3.1 An Editing Tool

The manual interactions of semi-automatic segmentation tools are generally of two types: parameter setting to control the automatic stage of the segmentation and geometric editing to refine the results of the automatic stage. This section describes a method in which the distinction between the automatic stage and the editing stage no longer exists. We have reduced the interface to only two interactions: marking points inside the desired object, and marking points outside the object. As points are marked a region is displayed that satisfies the points marked so far. The region is computed using the constraint satisfaction method described above.

Although it is possible to delete and rebuild (with the new constraint) the entire hierarchy each time a point is added this is rather slow. Fortunately, we have found that it is possible just to delete the top portion of the hierarchy consisting of those regions that contain internal and external points, and then to rebuild from this partially deleted state upwards (whilst satisfying the constraints, including the new one). This produces almost identical results with greatly reduced computational cost. It is worth noting that inferior results are obtained if a binary hierarchy rather than an n-ary hierarchy is used. The reason is not that the merging proceeds differently given the same initial state (it does not), but that the removal of the inconsistent portion of the hierarchy leads to a different state depending on whether a binary or an n-ary hierarchy is used. This is supporting evidence that the rule that we have used to reduce the binary hierarchy to an n-ary hierarchy is a reasonable one.

Fig. 1 illustrates the editing tool in action. Panel 1 shows the original image, a 70×86 transaxial MR image showing the lateral ventricles. The task being performed is extraction of the left ventricle including the faint upper portion. In the second panel two points have been marked by the user: an internal point (black) and an external point (white); which is the minimum required to define an object. The region shown has been constructed by the technique described. Subsequent panels show the region changing as more points are added. With an image of this size (running on a *SPARC IPX*) the new region is drawn without noticeable processing lag. Notice that the target region can

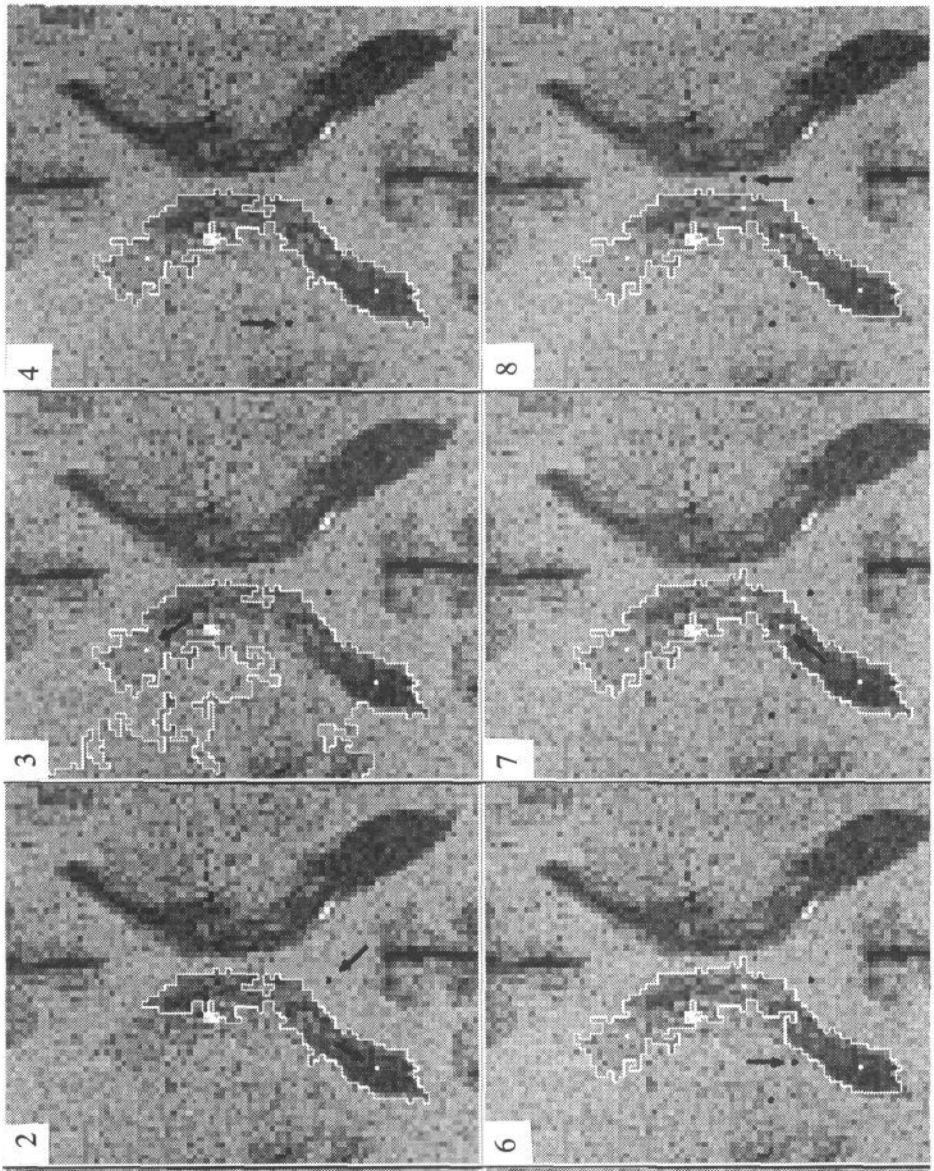

Figure 1

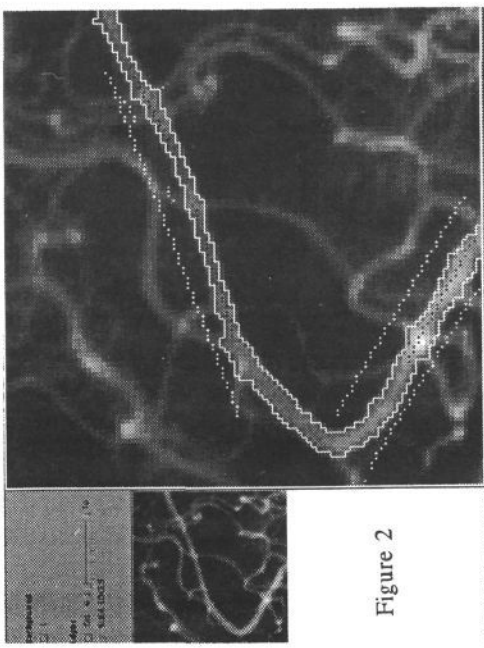

Figure 2

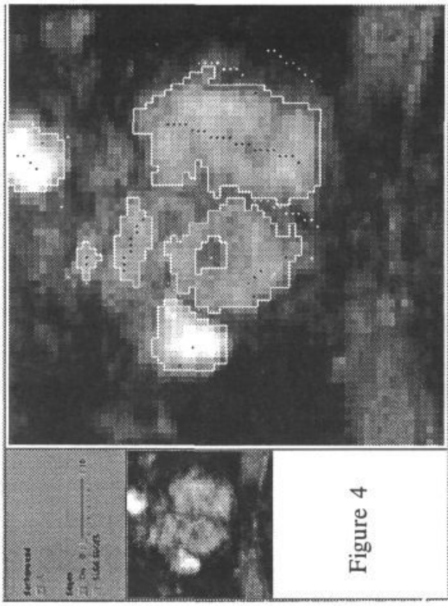

Figure 4

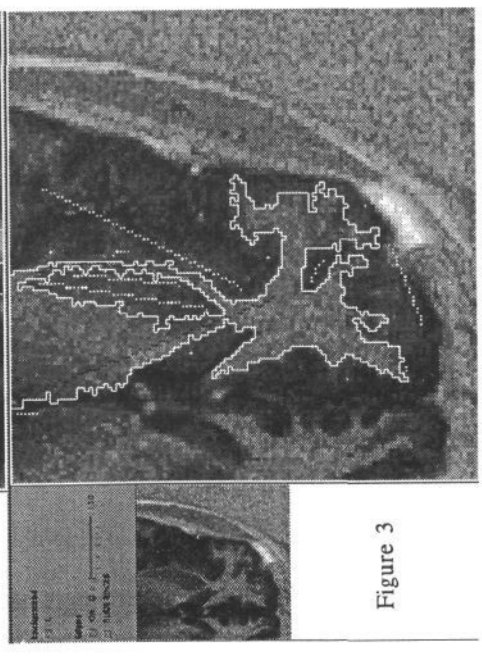

Figure 3

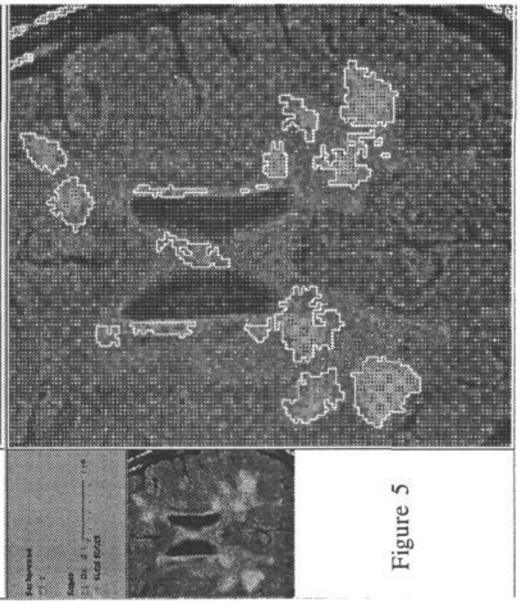

Figure 5

be enlarged or reduced by the process of adding points. Note also that the addition of a point can have a widespread effect. The sequence of images terminates with a reasonable segmentation of the left ventricle. The reader may see certain pixels that he would include or remove from the target object, actions which are easily performed by the addition of further points.

The following table shows timings for three different images (all times on a *SPARC IPX*). The resulting segmentations are shown in figures 2,3 and 4. The object extraction times are the actual recorded times and so are the sum of processing time, interaction time and looking time (*i.e.* deciding what to do). For these worked examples the user had the additional facility of being able to add a whole line of interior or exterior points. The manual segmentation time is the time taken to outline the object using point marking and line drawing. The assisted times are on average only 59% of the manual times. The manually segmented images are not shown but informal inspection suggests that the resulting segmentations are slightly inferior.

Image	**Size**	**Hierarchy Creation**	**Manual Segmentation**	**Assisted Segmentation**
Cerebral X-ray Angiogram	100^2	4.10s	90s	45s
Head MR	120^2	5.60s	120s	70s
Cardiac MR	70^2	1.65s	72s	50s

Fig. 2 shows the extraction of the peri-callosal artery from a cerebral angiogram. This illustrates how the editing tool can be used to select between multiple edges in an image. The need for this is particularly evident in a projection image such as this. Where two vessels cross there are (at least) two distinct objects that can be discerned. By placing external points along the edge of the desired vessel we cut off unwanted side-branches and define it throughout its entire course. The image also illustrates the utility of allowing lines of points to be placed. Fig. 3 shows the extraction of an area of white matter from a noisy transaxial MR image. This image demonstrates that no additional mechanism is needed to define holes in the target region. Fig. 4 show the extraction of several blood vessels (cross-sections) from a cardiac image. It demonstrates the extraction of a non-connected object.

3.2 Imposition of Anatomical Knowledge

The constraint satisfaction method that we have described can also be used with automatically generated constraints. One possible architecture for this would be coupled low- and high-level modules, with the low-level module computing descriptions of the image (*e.g.* hierarchical segmentations) and the high-level module guiding the low-level module by means of constraints suggested by expectation or partial matches to stored models[14]. What the form of such constraints should be and how a high-level system should generate them are open questions. In the following we present an example in which anatomical knowledge generates constraints on the segmentation.

Multiple sclerosis (MS) lesion extraction from MR images is an important and difficult application. Since lesions appear as bright areas on a darker background, it has been possible to use thresholding techniques for their extraction[16]. The difficulty is in choosing the appropriate threshold. Whilst pixels of sufficiently high value (say $L > T_f$) can be classified as being lesion with some confidence, and pixels of sufficiently low value (say $L < T_b$) can be classified as non-lesion with similarly high confidence: since $T_f > T_b$ there will be many pixels which cannot be easily classified in this manner. In practice it seems standard to accept a systematic under-estimation of lesion volume rather than risk the uncertainty of bringing T_f and T_b closer.

Fig. 5 shows an alternative approach. Pixels in this lesion image range in value from ***0*** to ***124***. We have set $T_f = 76$ and $T_b = 54$. These thresholds have been used to define a partial labelling of the image. The labels show up as white (background) and black (foreground) points. Pixels intermediate in value between the two thresholds are not labelled. The regions displayed satisfy the labelling and have been generated in the manner already described. The blob-like structures are all correctly identified MS lesions; the more elongated structures near the centre of the image are areas of peri-ventricular effusion which is also counted as abnormal for the purpose of lesion volume measurement. The only incorrectly identified structures are portions of skull and scalp; it is straightforward to identify and removed these. The outlines of the lesions are satisfactory and superior to the results of pure thresholding.

4 Concluding Remarks

The hierarchical segmentation algorithm that was presented in section 2 is an improvement over our previous techniques[6,7,8]. The use of a fixed memory approach to the representation of adjacency has allowed us to develop a fast implementation which is not excessive in its memory requirements.

As we have shown, using an agglomerative local merging scheme allows simple satisfaction of constraints. We have described how this may be used as an interface between imposed knowledge (top-down) and image description (bottom-up). Automatic constraints of the required form could be generated by processes more sophisticated than thresholding. Local methods which classify points on the basis of differential invariants at multiple scales[3] are one option, whilst multi-local shape information could be imposed using flexible templates[4] which attempt to locate object centres rather than object edges.

Finally, we mention that the original task for which the constraint satisfaction method was designed was multi-scale linking. This problem (being tacked by several centres[10,13,18]) is that of amalgamating descriptions at different scales into a single description. In such a process coarse scale descriptions are imposed on the interpretation of finer scale images[8]. We believe that constraint satisfying hierarchical segmentation is well-suited to this task.

references

[1] Beaulieu, J.-M. and Goldberg, M. (1989) Hierarchy in picture segmentation: a stepwise optimization approach. *IEEE Trans. Patt. Anal. Mach. Intell.* 11(2):150-163

[2] Bischoff, W.F. and Caelli, T.M. (1988) Parsing scale-space and spatial stability analysis. *CVGIP* 42:192-205

[3] Coggins, J.M. (1992) Statistical investigation of multiscale image structure. In: *Proc. VBC '92*, R.A. Robb (ed.) pp145-158

[4] Cootes, T.F. and Taylor, C.J. (1993) Active shape model search using local grey-level models: a quantitative evaluation. In: *Proc. BMVC '93*. J. Illingworth (ed.), pp639-648

[5] Flinchbaugh, B.E. and Chandrasekaran, B. (1981) A theory of spatial-temporal aggregation for vision. *Artif. Intell.* 17:387-407

[6] Griffin, L.D., Colchester, A.C.F. and Robinson, G.P. (1992) Scale and segmentation of grey-level images using maximum gradient paths. *Image Vis. Comput.* 10(6):389-402

[7] Griffin, L.D., Colchester, A.C.F., Robinson, G.P. and Hawkes, D.J. (1992) Structure-sensitive scale and the hierarchical segmentation of grey-level images. In: *Proc. VBC '92 SPIE vol. 1808*, R.A. Robb (ed.) pp24-32

[8] Griffin, L.D., Robinson, G.P. and Colchester, A.C.F. (1993) Multiscale hierarchical segmentation. In: *Proc. BMVC '93*, J. Illingworth (ed.), BMVA Press, pp289-298

[9] Höhne, K.H. and Hanson, W.A. (1992) Interactive 3D segmentation of MRI and CT volumes using morphological operations. *J. Comput. Assist. Tomogr.* 16(2):285-294

[10] Lindeberg, T. (1994) *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers.

[11] Koenderink, J.J. (1984) The structure of images. *Biol. Cybern.* 50:363-370

[12] Marr, D.C. and Hildreth, E. (1980) Theory of edge detection. *Proc. R. Soc.* B-207:187-217

[13] Morse, B.S., Pizer, S.M. and Liu, A. (1993) Multiscale medial analysis of medical images. In: *Proc. IPMI '93*. H.H. Barrett, A.F. Gmitro (eds.) pp112-131

[14] Mumford, D. (1992) On the computational architecture of the neocortex. II. The role of cortico-cortical loops. *Biol. Cybern.* 66:241-251

[15] Pizer, S.M., Cullip, T.J. and Fredricksen, R.E. (1990) Towards interactive object definition in 3D scalar images. In: *3D Imaging in Medicine*. K.H. Höhne, H. Fuchs, S.M. Pizer (eds.), Springer-Verlag, pp83-105

[16] Röll, S.A., Colchester, A.C.F., Summers, P.E. and Griffin, L.D. (1994) Intensity-based object extraction from 3D medical images inclusing a correction for partial volume errors. In: *Proc. BMVC '94*, E. Hancock (ed.)

[17] Toet, A. (1991) Hierarchical clustering through morphological transformation. *Patt. Recog. Lett.* 12:391-399

[18] Vincken, K.L., Koster, A.S.E., Viergever, M.A. (1992) Probabilistic multiscale image segmentation - set-up and first results. In: *Proc. VBC '92*, R.A. Robb (ed.) pp63-77