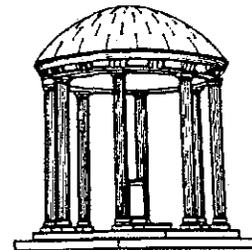


**Algorithms for 2D and 3D Image Description
Based on the IAS**

**TR90-034
October, 1990**

**Timothy J. Cullip
Robin E. Fredericksen
John M. Gauch
Stephen M. Pizer**

**Medical Image Display Group
Department of Computer Science
Department of Radiation Oncology
Department of Radiology
The University of North Carolina
Chapel Hill, NC 27599-3175**



The research reported herein was carried out with the partial support of NIH grant number P01CA47982.

Appeared in Proceeding of the First Conference on Visualization in Biomedical Computing, May, 1990.

UNC is an Equal Opportunity/Affirmative Action Institution.

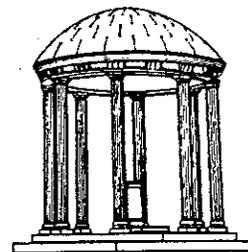
**Toward Interactive Object Definition in 3D
Scalar Images**

TR90-034

October, 1990

**Stephen M. Pizer
Timothy J. Cullip
Robin E. Fredericksen**

**Medical Image Display Group
Department of Computer Science
Department of Radiation Oncology
Department of Radiology
The University of North Carolina
Chapel Hill, NC 27599-3175**



The research reported herein was carried out with the partial support of NIH grant number P01 CA47982.

Appeared in Proceeding of the NATO Advanced Research Workshop. June, 1990.

UNC is an Equal Opportunity/Affirmative Action Institution.

Algorithms for 2D and 3D Image Description Based on the IAS

Timothy J. Cullip, Robin E. Fredericksen,
John M. Gauch, Stephen M. Pizer

Medical Image Display Research Group
University of North Carolina
Chapel Hill, NC 27599-3175, USA

ABSTRACT

This paper describes the algorithms used in computing the Intensity Axis of Symmetry (IAS) for 2D and 3D medical images. The basic 2D algorithms are described first, followed by the algorithms needed to incorporate scale space. Lastly, a brief discussion of the extensions needed to work with 3D images is given. The basic approach is to treat the image as a deformable intensity surface which is contracted onto the IAS. The primitive regions of the segmentation are identified by the branches in the resulting tree-like structure. A hierarchy is produced by following the simplification of the branching through scale space.

Introduction

A companion paper [Pizer, 1990] defined the Intensity Axis of Symmetry (IAS) and described its properties with regard to image segmentation. This paper describes the algorithms used for computing the IAS. A third paper [Fredericksen, 1990] describes a tool used in analyzing the effectiveness of the segmentation and hierarchy produced by the IAS method. The IAS is defined as a one parameter family of medial axes of the iso-intensity level curves of the image. When a 2D image is viewed as an intensity surface, the IAS produces a forest of branching and looping sheets which fit under the ridges of the surface with one branch per ridge. A hierarchy is produced by successively blurring the image and following the IAS through scale space. This hierarchy relates the annihilation

of one IAS branch into another to the annihilation of a subridge into its parent ridge.

Computing the IAS

To compute the Intensity Axis of Symmetry we first view the 2D image as an intensity surface in 3-space. We first define a parameterization of the 2D image as:

$$f(u,v)=(x(u,v),y(u,v),z(u,v))$$

with initial values:

$$\begin{aligned}x(u,v) &= u, \\y(u,v) &= v, \text{ and} \\z(u,v) &= I(u,v).\end{aligned}$$

The surface thus defined will be allowed to move in the spatial directions ($x(u,v)$ and $y(u,v)$ allowed to change values), but held fixed in intensity as it is moved toward the IAS. Think of this as a collapsing of the original surface onto the IAS skeleton. By not allowing motion in the intensity direction, we insure that the basic structure of the image, as defined by the ridges, is not lost.

In order to maintain surface continuity while migrating the surface points toward the IAS, we extend the concept of active contours [Kass, 1987] to this deformable surface. This active surface model uses an energy functional defined on the surface with terms that capture the smoothness constraints of the surface while driving the surface toward the IAS. The smoothness constraints aid in the identification of

individual IAS branches. The equation we are currently using is given by:

$$\text{Energy} = \iint [w_1(f_u(u,v)^2 + f_v(u,v)^2) + w_2(f_{uu}(u,v)^2 + f_{vv}(u,v)^2 + f_{uv}(u,v)^2) - w_3(g(f(u,v)))] du dv$$

where the u and v subscripts indicate the partial differentiation of the surface equation. The weighting factors determine the degree to which the surface behaves like a smooth membrane (w_1), or a spline (w_2). The attraction of the surface toward the IAS is given by the image symmetry function $g(f(u,v))$. An iterative relaxation technique is used to solve this energy minimization. We are also currently investigating other surface energy equations that better match the surface geometry of the IAS.

Because the IAS is defined in terms of centers of maximal disks tangent to the level curves of the image, the image symmetry function, g , needs to measure for each point (x,y,I) the distance to the nearest point on the original surface at the same intensity. Note that if the original surface has intensity I_1 at position (x,y) , then $g(x,y,I) = 0$ for all I greater than I_1 . The deformable surface is drawn in toward the IAS by seeking larger values of the image symmetry function.

Since the image symmetry function is used extensively during the iterative collapse routine, its values are precomputed for every point in the image at each intensity level. To limit the size of this 3D data array the image is first scaled to a reasonable number of discrete intensity levels, typically 50.

One approach to computing the image symmetry function is based upon a grass fire analogy. At each of the intensity levels, the iso-intensity contours are filled in with pixels of value one inside the contour (the grass)

and zero outside (burned out areas). Burning away one layer is accomplished by scanning the image once by rows and once by columns and removing any one valued pixels that have zero valued neighbors. The distance value corresponds to the iteration number at which a pixel is converted from a one to a zero. This method requires multiple passes of the entire 3D dataset.

A computationally more efficient method reverses the above approach. Rather than start at the contour and burn your way in, instead start at a point inside the contour and work your way out looking for the contour. Begin your search at the pixel you are interested in and spiral search your way out until you find a pixel of intensity equal to or lower than the contour level we are currently interested in. If you did this independently for each pixel at each contour intensity level it would be a very inefficient algorithm, but we make use of a precomputed spiral search path and spatial coherency to improve this. At a given pixel location, the distance value at a lower intensity must be no less than the distance value at a higher intensity ($g(x,y,I+1) \leq g(x,y,I)$). Therefore the spiral search at the next lower intensity can begin at the distance of the current intensity. Also, the distance value for pixel $(x-1,y,I)$ has to be within plus or minus one of the distance value for pixel (x,y,I) , so we can start the search for the current pixel based on the distance for the pixel to its left. This method also has the advantage of choosing between city block or Euclidean metrics (or any other metric) simply by changing the precomputed search path. This algorithm computes the image symmetry function in about 5 minutes (dec3100) for a 256x256 image scaled to 50 intensity levels, whereas the grass fire approach takes several times as long.

The surface is moved toward the IAS by an iterative relaxation algorithm. During each iteration each point on the

surface is examined to see if moving it to a neighboring location will lower its contribution to the surface energy. If so, the point is moved to that new location. Since the image symmetry function does not have local minima other than the global minimum, this minimization technique yields a surface representing the IAS after a finite number of steps.

One problem with the above method is that the surface continuity constraints should be limited to within an IAS branch, but not apply across them. For example, a valley between two ridges should be allowed to freely separate without the surface derivative constraints holding the two sides of the valley together (see figure 1). We have overcome this problem by first allowing the surface to freely move based only on the image symmetry function, noting those areas of the image that have moved away from each other, and then applying the above energy minimization with the derivative weights reduced in areas where the image separated. Now the continuity constraints result in a smooth surface patch for each IAS branch (which aids in the identification of the branch) while allowing individual patches to move independently. This movement takes about 10 to 20 minutes to converge.

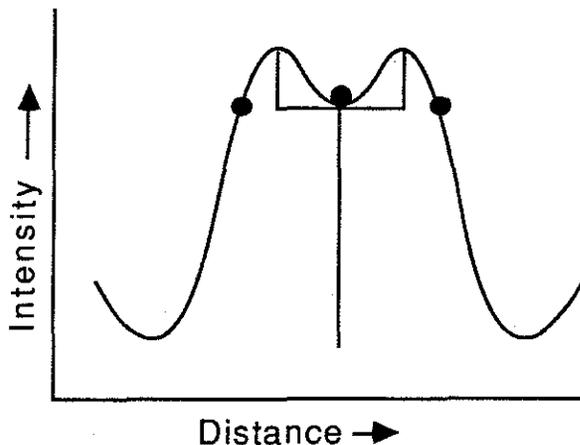


Figure 1. A cross-sectional slice of a double ridge. Areas of the image where the continuity constraints need to be relaxed are marked with dots.

Identifying Branches

Once the surface has been collapsed onto the IAS we need to identify the individual branches within this structure. We do this for each branch by identifying the two patches of the active surface which have been drawn to the opposite sides of the IAS branch. Before we do this, we must first identify for each point in the image its involute: the point on the other side of the IAS that mapped to the same point on the IAS (see figure 2). The collapsing of the surface onto the IAS has given us a mapping for each point from the original surface to the IAS. Note that each point on the surface maps to a single point on the IAS, but each point on the IAS may have several image points mapped to it. Determining involute pairs is achieved by first allocating a bucket for each IAS point, then filling the buckets with the original image points that map to each bucket location. A proper involute pair should satisfy the following conditions:

- 1) moved the same distance in getting to the IAS,
- 2) be distant from its involute to prevent linking points on the same side of the axis sheet,
- 3) moved to the same IAS point (bucket).

Due to the discrete nature of the data and the approximation of the IAS resulting from the surface minimization, these conditions may only be approximately satisfied. By using a weighting of the first two conditions, and an outward search based on the third condition (start with the current bucket and then search neighboring buckets) we achieved a reasonable linking of involutes. We now have for each point in the image two mappings, the first maps the point to its IAS point, and the second maps the point to its

involute. This step requires approximately one minute to compute.

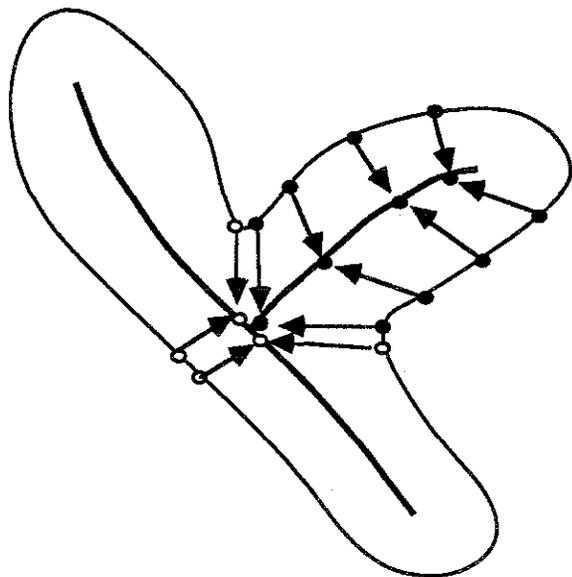


Figure 2 Involute pairs and their relationship at a branch point in the IAS. The outer curve represents a single isointensity contour.

The identification of individual branches of the IAS now becomes a simple region growing process. The process starts with an arbitrary point, placed in set A, and its involute in set B. You then examine all neighboring points of points in set A, adding them to A (and their involutes to B) if their involutes are in set B or neighbors of points in B. Likewise for points in B with their involutes in A. This process is repeated until neither set A nor B increase. The growing stops when the involute of a neighbor is not a neighbor of the other set, which occurs at the branch points of the IAS (see figure 2). The region growing typically requires on the order of 5 to 10 minutes.

Multiresolution IAS

By identifying the individual branches of the IAS, we have partitioned the image into primitive regions, one

region per branch. In order to group these regions in a hierarchical manner, we follow the simplification of the branching structure through scale space. Since the IAS branching structure simplifies through successive blurring levels, the goal is to track the branches from level to level, recording when individual branches annihilate.

Between two blurring levels, we need to be able to correspond the branches at the less blurred level (L1) to the branches at the more blurred level (L2), recognizing those branches that annihilate. Currently our method simply uses the surface patches corresponding to the IAS branches rather than the IAS branches themselves. For each region at L1 we determine the region at L2 that best contains it. If the L1 region overlaps more than one L2 region, we choose the region that contains the largest fraction of the L1 region. This results in each region at L1 mapping to a single region at L2, but a region at L2 may have more than one region at L1 mapped into it. If a region at L2 contains more than one region at L1 then those L1 regions have merged together into the L2 region (annihilation of their corresponding IAS branches). By repeating this process for each pair of blur levels, we form a hierarchy on the regions and therefore the branches of the IAS corresponding to those regions. We are currently investigating more robust methods of tracking the IAS branches themselves.

Extensions to 3D Images

The extension of the algorithms to 3D images is straightforward. The intensity surface now becomes a 3D surface in 4-space: $f(u,v,w) = (x(u,v,w), y(u,v,w), z(u,v,w), I(u,v,w))$. The IAS is now a branching structure in 4D. The image symmetry function is now defined over a 4D data set.

We have had some success with the 3D algorithms on small, simple test images, but are still investigating the process. We have found that the proper identification of involutes is both a more difficult task due to the added spatial dimension, and more critical to the proper termination of the branch growing algorithm. Also, the problem caused by the energy minimization preventing valleys from fully separating hinders the involute identification. We expect a similar solution of identifying those areas of the image in which to relax the smoothness constraints should help alleviate the problem, but we have not yet tested this. We are also currently modifying the algorithms to take advantage of a newly acquired MasPar SIMD parallel processing computer which should reduce the hindrances to our research caused by the large amounts of time and space the current sequential algorithms require.

Evaluation of Results

For this paper we have applied the algorithms to two different 2D images. The first one is an MRI image of a brain; the second one is a CT image through the chest (figures 3 and 4). The reader is also referred to the third paper in this series for further discussion on the results of IAS segmentation, especially as applied slice by slice to a 3D image.

The IAS segmentation generally results in meaningful primitive regions that correspond well to the ridge structure in the image. The branching structures within the CT image (figure 4b) are easily identified and followed through scale space. Another example of following a long narrow ridge is given in figure 3b, where the brain stem has been identified. The brain stem is composed of several minor ridges riding on the main ridge which results in multiple primitive regions. By following their annihilations through

scale space, the entire brain stem can be selected.

One area of disappointment was in trying to follow the convolutions of the brain itself. The primitive regions corresponded well to the individual folds, but moving up the hierarchy through scale space quickly brought in regions outside of the brain. We expect that improvements in the hierarchy will result by using non-stationary, anisotropic edge sensitive diffusion as opposed to the Gaussian blurring we have used so far. Also improvements can be made in our method for tracking the IAS branches through scale space.

The occasional failure in the identification of a meaningful primitive region results from the misidentification of a branch. At times the region growing algorithm manages to grow a single region that spans more than one IAS branch, and at other times it may produce regions that are smaller than what appears to be a single branch. This seems to be attributed mostly to improperly identifying involutes, which in turn is due to the surface contraction only approximating the IAS. By using a different surface energy function that better captures the geometry of the desired IAS we expect improvements in this area.

Acknowledgements

We thank Dr. Robert Gardner for his useful discussions on geometry. We thank Carolyn Din for her help in preparing the manuscript, Bo Strain for photography, Siemens, Inc. for providing the MRI image data, and Dr. Matthew Mauro for the CT image data. The research reported herein was carried out with the partial support of NIH grant number P01CA47982.

REFERENCES

- [1] Fredericksen, R.E., Coggins, J.M., Cullip, T.J., Pizer, S.M., "Interactive Object Definition in Medical Images Using Multiscale, Geometric Image Descriptions.", in this volume, 1990.
- [2] Pizer, S.M., Gauch, J.M., Cullip, T.J., Fredericksen, R.E., "Descriptions of Image Intensity Structure via Scale and Symmetry.", in this volume, 1990.
- [3] Kass, M, Witkin, A, and Terzopoulos, D, "Snakes: Active Contour Models." *Proc 1st Int. Conf. on Comp. Vis.*, IEEE Catalog #87CH2465-3:259-268, 1987.

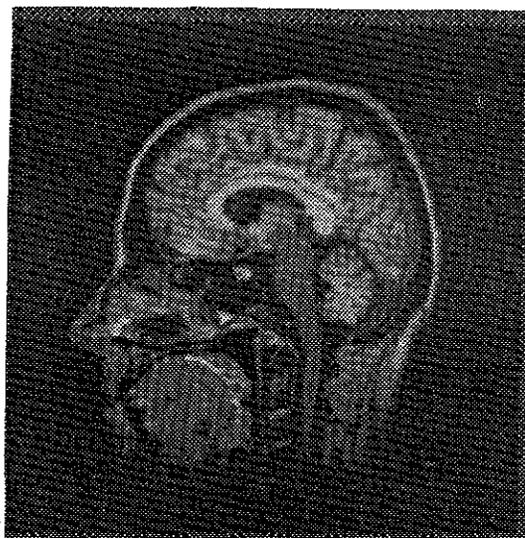
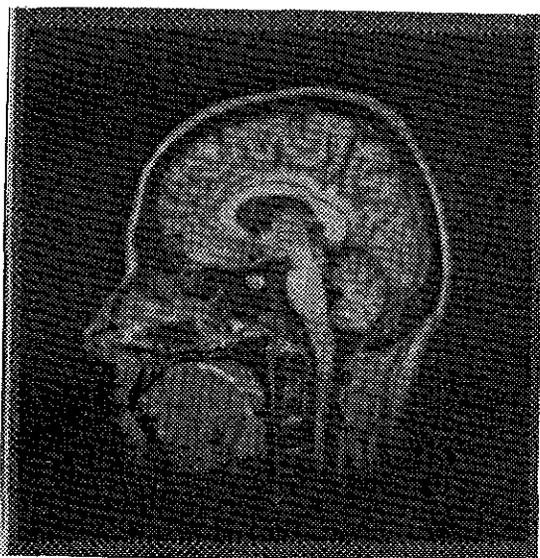


Figure 4. Brain MRI:

- a) Single primitive region in the lower brain stem.
 b) Capturing the brain stem by only moving up the hierarchy.

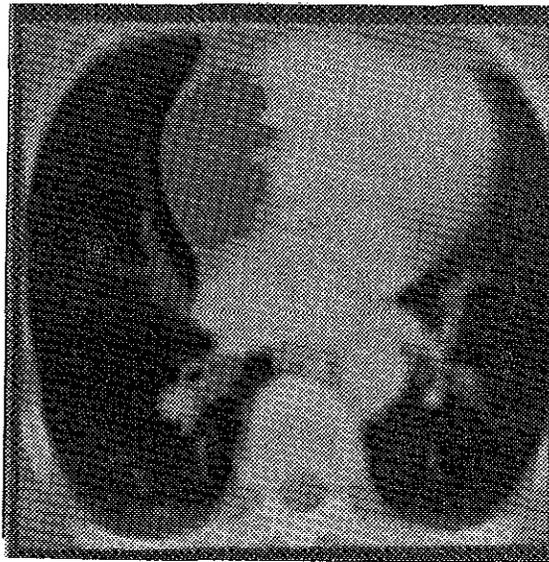


Figure 5.

- a) Chest CT image.
 b) Selection of branching sections of the bronchi in the left lung, and a large object within the mediastinum.

Toward Interactive Object Definition in 3D Scalar Images

Stephen M. Pizer, Timothy J. Cullip, Robin E. Fredericksen

Medical Image Display Research Group
University of North Carolina
Chapel Hill, NC 22599-3175, USA

Abstract

We present a method of object definition designed to allow fast interactive definition of object regions in 2D and 3D image data by a human user based on an automatically computed image description of sensible image regions. The image description is a quasi-hierarchy of ridges (or courses) and subridges (or subcourses) in the intensity surface corresponding to the image. Two methods of ridge computation are presented, one based on the intensity axis of symmetry and another based on flow lines in the intensity surface. A system for interactive object definition using this approach is described, and the use of this approach on a variety of medical images is evaluated. Generalizations of these descriptions and the interactive object definition tool to 3D are discussed.

Keywords: Segmentation, 3D display, object definition, multiscale, geometry.

1. Introduction

The most time-consuming step of methods involving general volume visualization is that of selecting regions within the 3D image data that correspond to the objects to be visualized. For example, in 3D scalar ("grey-scale") medical images the display of all but objects appearing at the highest contrast typically requires hours of hand selection, slice by slice, of regions enclosing the objects before the elegant rendering methods can be applied to the resulting regions to produce good visualizations of the organs or other anatomic objects of interest. For the so-called volume rendering methods the region selected does not need to fit closely to the surface of the object of interest but may include the object and part of its background, as long as that background does not include other objects that after rendering will occlude the

object of interest. For surface rendering, and also for the purposes of object measurement, the surface of the region indeed needs to follow the surface of the object. However, given the surface of the former type, the active contours method of Kass et al [1987] can be used to find the actual object surface with little user interaction. As a result this paper focuses on methods to determine the more loosely fitting region just described.

The definition of an object depends on both properties inherent in the image data and properties of the scene. The properties of the image data give geometric information about region structure (syntax) but do not use our knowledge of possible scenes (semantics). We suggest that both the early human visual system and a computer can derive the syntactic region structure by using geometric analysis, but while the human cognitive system also knows how to bring knowledge of possible scenes to bear, given the regions produced from the syntactic analysis, it is not very well known how to make computers do this. However, for the regions to be useful for rendering, they must be known not simply by the human but by the computer. We are thus led to design a method whereby we attempt to have the computer derive the syntactically defined regions and let these serve as a means of communication between the human and the computer such that the user interactively and quickly can specify to the computer semantically correct regions from the syntactically defined ones.

This paper is therefore divided into section 2 on how to produce an image description in the form of a graph of sensible regions and section 3 on an interactive method for producing a meaningful region based on the image description.

2. Image Description by Multiscale Geometry

We suggest that if the image is viewed as an intensity surface, where height corresponds to image intensity (see figure 1), then shape properties of this surface will determine the region definitions. Humans seem to use ridges of intensities as organizing features for light objects on dark backgrounds, or courses for dark objects on light backgrounds. In keeping with its being a good model for human vision, ridge/course analysis is largely insensitive to rotations of the co-ordinate system, spatial scaling, and monotonic transformations of intensity and involves a focus on curvature, orientation, and two-sidedness. In this paper we will focus on regions lighter than their background, but all of the

ideas will apply to the problem of finding regions darker than their background. The relation between these two sets of regions is also of interest if we are trying to imitate human visual behavior but is beyond the scope of this paper.

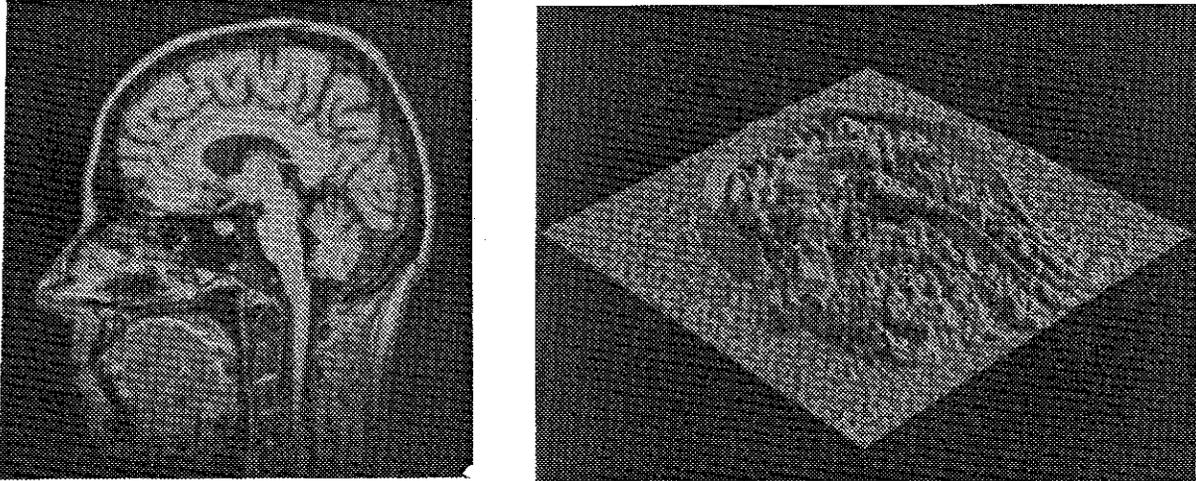


Fig. 1. MRI brain image and corresponding intensity surface

The fact that analysis into ridges may provide a reasonable model of human visual segmentation is relevant for 2D images but not necessarily for 3D images. Nevertheless, the effectiveness of this analysis for 2D images has led us to investigate its usefulness for 3D images. For each form of ridge analysis we will discuss first the ideas as applied to 2D images and then their generalization to 3D.

Ridges are structures with a ridge top and flanks falling to a course separating this ridge's flank from that of the next. The ridge top is defined by curvature properties of the surface there. Ridges have a multiscale structure, in the sense that small ridges may fall on the flank of or branch from larger ridges. For 2D (see figure 2), a) a ridge may fork into two ridges, and b) these two may rejoin, so that the pair surround an indentation with a course and even possibly a pit at its bottom. Furthermore, c,d) one ridge may appear on the flank of another. Finally, e) a ridge can begin on the flank of one ridge and end on the flank of another. The result is that the intensity surface forms a graph of ridges, with the directed arcs deriving from the child ridge being on the flank of or branching from the parent ridge. The fact that a ridge may fork and rejoin or connect two or more other ridge flanks mean that a single ridge can have more than one parent.

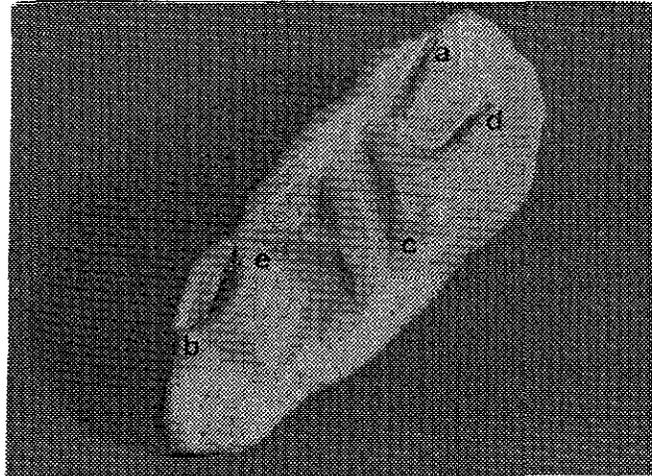


Fig. 2. Ridge types: a) ridge fork, b) ridge fork and rejoining, c) monotonic ridge on flank of another ridge, d) nonmonotonic ridge on flank of another ridge, e) ridge connecting two other ridges.

It follows that an analysis of an intensity surface into ridges must have a multiscale, geometric flavor such that ridge regions are identified and parent/child relations are generated among these ridge regions. Two different strategies suggest themselves. In the first, the quasi-hierarchical structure of ridges and subridges is captured by a quasi-tree of connected geometric elements with each element corresponding to a ridge, and the elements are followed to annihilation into other elements under increase in scale (blurring) of the image. The ridge corresponding to an annihilating element is then taken as the child of the ridge corresponding to the element into which the annihilation took place. In the second strategy the ridges (with their flanks) are analyzed directly via geometric properties, and the ridge/subridge relationship follows from the geometric properties of one ridge's flank containing the other ridge. Sections 2.1 and 2.2 explore these respective strategies.

2.1. Connected Structures with Annihilation

2.1.1. The Intensity Axis of Symmetry

We suggest that a basic property of "ridgeness" is symmetry, so that for each point on one flank of a ridge there is another point on the other flank to which

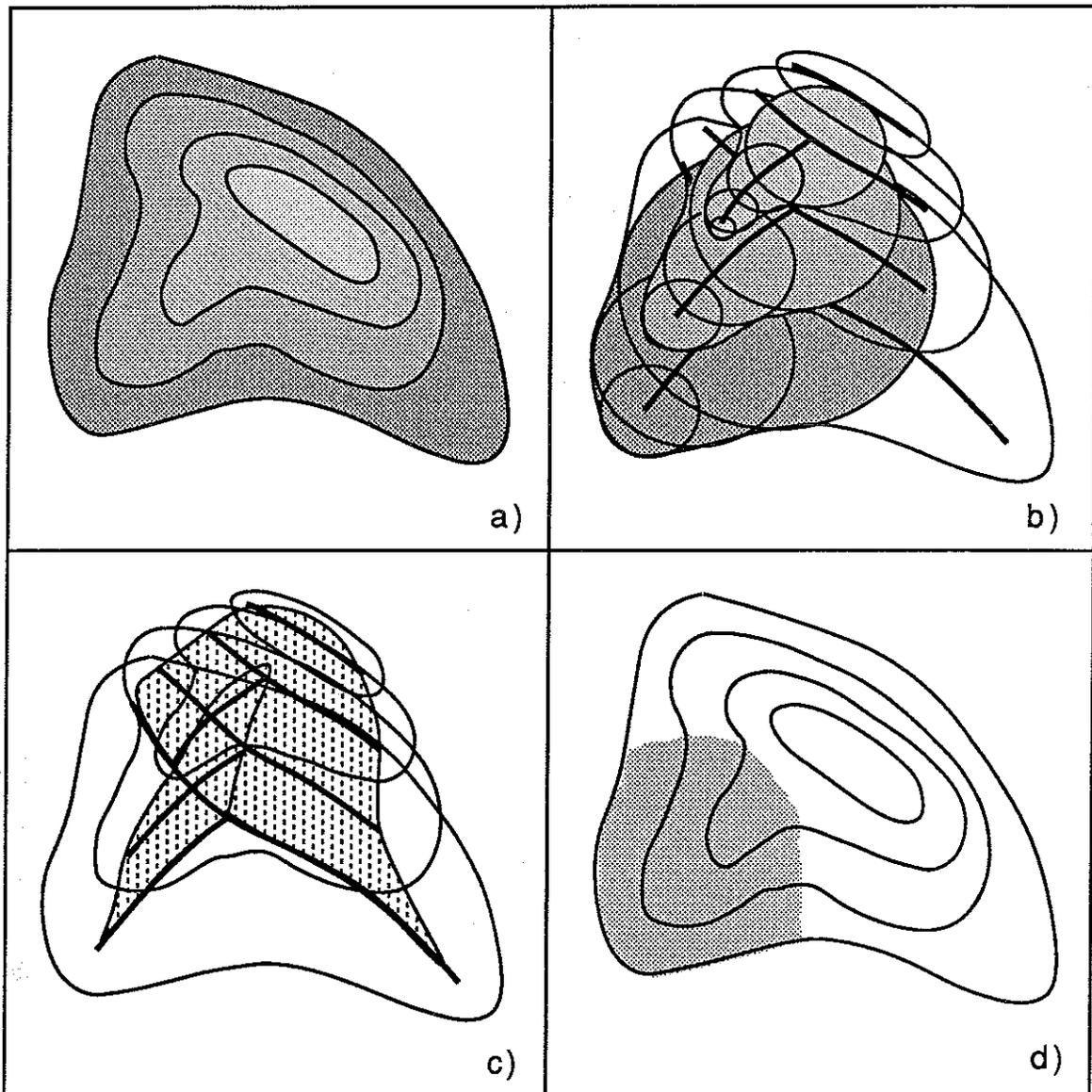


Fig. 3. The IAS as a one parameter family of slice axes of symmetry.
 a) Level curves of a simple image; b) Level curves on intensity surface, their medial axes, and selected maximal disks; c) Level curves on intensity surface, and IAS; d) Image region associated with an IAS branch.

and merging of the ridgetop results in a scoop in the IAS sheettop (see figure 4c), and a ridge connecting two other ridges corresponds to an IAS sheet connecting two other sheets (see figure 4d). The IAS under a nonmonotonic subridge forms a loop on the parent IAS sheet; at the intensity of the loop bottom the parent sheet tears (see figure 4e). More complicated forms arise if pits appear in inter-ridge valleys. In any case, we have the important relationship that although ridgetops may not form a connected structure, the

it has some sort of symmetry relation. Symmetry is an important visual property -- regions are seen as two sided, i.e., ridge flanks are visually grouped to form the objects that we see. An analysis in terms of this symmetry should thus generate regions and hierarchical relationships that provide a natural medium for human and computer communication for segmentation.

As a result we define an intensity axis of symmetry (IAS). This axis, made of a forest of branching or looping sheets, fits under the ridges midway between the two symmetric flanks of the ridge. In order to avoid sensitivity to any monotonic transformation of intensity and also to avoid the incommensurability of intensity and space, the image (intensity surface) must be considered as a one-parameter family of slices in the intensity dimension, with the IAS made by stacking axes of symmetry defined for each slice. Thus, the IAS is the one-parameter family of medial axes of the intersections of the intensity surface with a series of slicing surfaces (see figure 3).

For now we have been slicing at iso-intensity levels, so that each axis of symmetry in the family is the medial axis of a level curve of intensity, even though this slicing focuses too greatly on intensity levels and too little on local image structure. The present IAS has the advantage that there is a 1-1 relation between each branch and a ridge top as defined by the locus of maxima of positive curvature of intensity level curves (these loci are called *vertex curves*) [Gauch, 1988a, 1989]. However, it has the disadvantage that it is too directly tied to absolute intensity and ridge flanks go down only to the higher of the two courses bounding a ridge, so future consideration of slicing strategy is indicated.

As illustrated in figure 3, associated with each point on the IAS are (normally) two points in the image where the maximal disk centered at the IAS point is tangent to a level curve at its intensity. The basic sheets that are the leaves in the IAS quasi-tree thus have associated with them a set of image points that are taken as the primitive regions of the image.

2.1.2. Generating the Ridge/Subridge Relationships

Associated with a monotonic ridge on the flank of another ridge (its parent), then, is a branch of the IAS that grows from the IAS sheet lying under the parent ridge. If the child ridgetop branches from the parent ridgetop, the IAS child branch also begins at the top of the IAS sheet of the parent (see figure 4a); otherwise from the middle of the parent sheet (see figure 4b). A splitting

2.1.3. Computing the 2D IAS-Based Image Description

IAS Computation. To compute the IAS [Gauch, 1988b], we first view the 2D image as an intensity surface in 3-space. We define a parameterization of the 2D image as $f(u,v) = (x(u,v), y(u,v), z(u,v))$ with initial values $x(u,v)=u$, $y(u,v)=v$, and $z(u,v)=I(u,v)$. The surface thus defined is allowed to move in the spatial directions ($x(u,v)$ and $y(u,v)$ are allowed to change values) but held fixed in intensity as it is moved toward the IAS. Think of this as a collapsing of the original surface onto the IAS skeleton. By not allowing motion in the intensity direction, we insure that the basic structure of the image, as defined by the ridges, is not lost.

The surface is moved toward the IAS by an iterative relaxation algorithm. During each iteration each point on the surface is examined to see if moving it to a neighboring location will increase its distance, inside the intensity surface and at its intensity, from that surface, as measured by the image symmetry function, $g(x,y,I)$. If so, the point is moved to that new location.

This method breaks up surface continuity. In order to re-establish this continuity, at least at places where the surface should be continuous (see figure 6), we extend the concept of active contours [Kass, 1987] to a deformable surface, starting from the broken surface resulting from the unconstrained collapse. This active surface model iteratively minimizes an energy functional defined on the surface with terms that capture the smoothness constraints of the surface while also keeping the surface near the IAS. The smoothness constraints aid in the identification of individual IAS branches. The equation we are currently using is given by

$$\text{Energy} = \iint [w_1(f_{uu}(u,v)^2 + f_{vv}(u,v)^2) + w_2(f_{uuu}(u,v)^2 + f_{vvv}(u,v)^2 + f_{uuv}(u,v)^2) - w_3(g(f(u,v)))] du dv,$$

where the u and v subscripts indicate the partial differentiation of the surface equation. The weighting factors determine the degree to which the surface behaves like a smooth membrane (w_1), or a spline (w_2). To allow desired discontinuities of the type illustrated in figure 6, these weights are reduced in those areas of the image that have strongly moved away from each other during the original collapse. We are also currently investigating other surface energy equations that better match the surface geometry of the IAS.

corresponding IAS sheets do form a connected structure reflecting the subridge on ridge flank or ridge branching relationships. Because a given subridge may be on the flank of or branch from more than one other ridge, a given ridge may be the child of more than one parent.

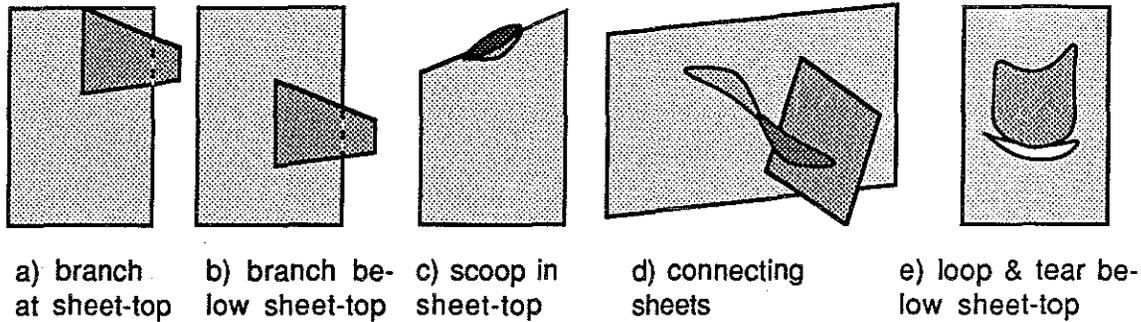


Fig. 4. IAS branchings

Region analysis thus involves computing a graph in which each node corresponds to a ridge, i.e. what we are taking to define sensible, coherent regions. The arcs in the graph are directed to ridges from subridges that describe sensible, coherent subregions of the parent ridge in question. In addition the parent ridge contains pixels that are on the parent's flank but not in any subridge.

Our problem is to decide which ridge is a subridge of which and to form the graph which describes these relationships. Our strategy is to compute the region containment relations induced by the connectivity of the IAS branches and by the annihilation of one branch into another as scale is increased (see figure 5). That is, one ridge is taken as a subregion of another ridge if the IAS branch corresponding to the former disappears into the IAS branch corresponding to the latter as the scale at which the image is considered is successively increased.

In the process of increasing spatial scale, it can be shown that a simple IAS branch (obtained by either ridge branching or by one ridge being on the flank of another ridge) will begin as a sheet but shrink from the top and bottom until it disappears smoothly into the sheet to which it is attached. This disappearance can be used to establish the child/parent relationship. Looping subsheets will normally disappear by narrowing down in their middle and breaking into two simple IAS sheets. An IAS sheet connecting two other

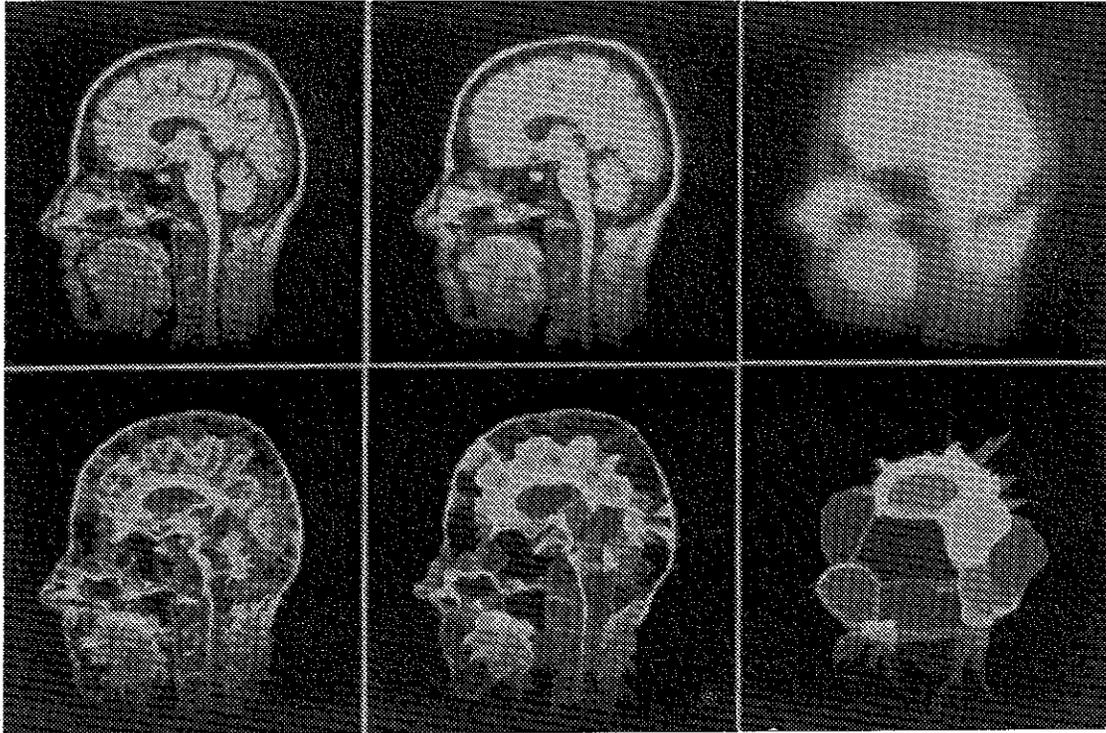


Fig. 5. An MRI brain image at various scales and corresponding IAS's. Scale change was obtained by edge-affected anisotropic diffusion.

sheets can detach first from one of its parents and then from the other, the intensity levels at which it is attached to one not being the same as those at which it is attached to the other.

Scale increase is performed by diffusion (Gaussian blurring), since this form of scale increase avoids most strongly the creation of structure [Yuille, 1983]. Simplicity suggests isotropic and stationary Gaussian blurring, but it seems preferable that the diffusion rate be matched to the scale of local objects. That is, diffusion across the interior of a large object should be faster than across a similar region made up of small objects or across the edge of the large object -- the scale change should be nonstationary. Moreover, diffusion along an edge should be faster than diffusion across it. To achieve this adaptively nonisotropic diffusion, we have begun using a program written by De Moliner [1989] at E.T.H., Zürich, in which a negative exponential function of intensity gradient magnitude (edge strength) is used as the local conductivity in a variable-conduction form of the diffusion equation [Perona, 1987].

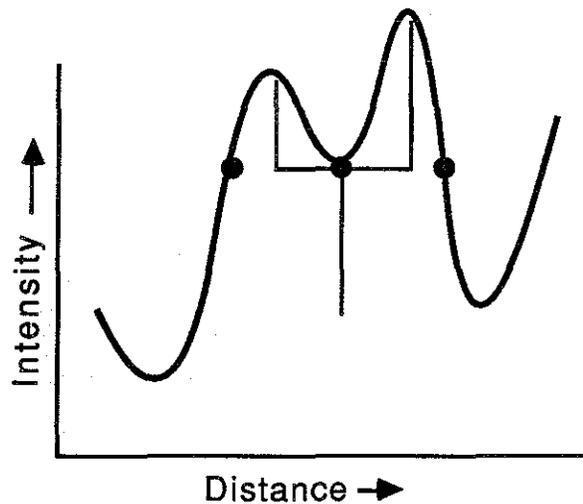


Fig. 6. A cross-sectional slice of a double ridge. Areas of the image where the continuity constraints need to be relaxed are marked with dots.

Since the image symmetry function g is used extensively during the iterative collapse routine, its values are precomputed for every point in the image at each intensity level. To limit the size of this 3D data array the image is first scaled to a reasonable number of discrete intensity levels, typically 50. The calculation can take advantage of the fact that at a given pixel location, the distance value at a lower intensity must be no less than the distance value at a higher intensity ($g(x,y,I+1) \leq g(x,y,I)$).

Identifying Branches. Once the surface has been collapsed onto the IAS, we need to identify the individual branches within this structure. We do this for each branch by identifying the two patches of the active surface which have been drawn to the opposite sides of the IAS branch. Before we do this, we must first identify for each point in the image its involute: the point on the other side of the IAS that mapped to the same point on the IAS (see figure 7). The collapsing of the surface onto the IAS has given us a mapping for each point from the original surface to the IAS. Note that each point on the surface maps to a single point on the IAS, but each point on the IAS may have several image points mapped to it. Determining involute pairs is achieved by first allocating a bucket for each IAS point, then filling the buckets with the original

image points that map to each bucket location. Each point in a proper involute pair should satisfy the following conditions:

- 1) moved the same distance in getting to the IAS,
- 2) be distant from its involute to prevent linking points on the same side of the axis sheet,
- 3) moved to the same IAS point (bucket).

Due to the discrete nature of the data and the approximation of the IAS resulting from the surface minimization, these conditions may only be approximately satisfied. By using a weighting of the first two conditions, and an outward search based on the third condition (start with the current bucket and then search neighboring buckets), we have achieved a reasonable linking of involutes. We now have for each point in the image two mappings, the first maps the point to its IAS point, and the second maps the point to its involute.

The identification of individual branches of the IAS now becomes a simple region growing process. The process starts with an arbitrary point, placed in set A, and its involute in set B. You then examine all neighboring points of points in set A, adding them to A (and their involutes to B) if their involutes are in set B or neighbors of points in B. You do likewise for points in B with their involutes in A. This process is repeated until neither set A nor B increase. The growing stops when the involute of a neighbor is not a neighbor of the other set, that is, at the branch points of the IAS (see figure 7).

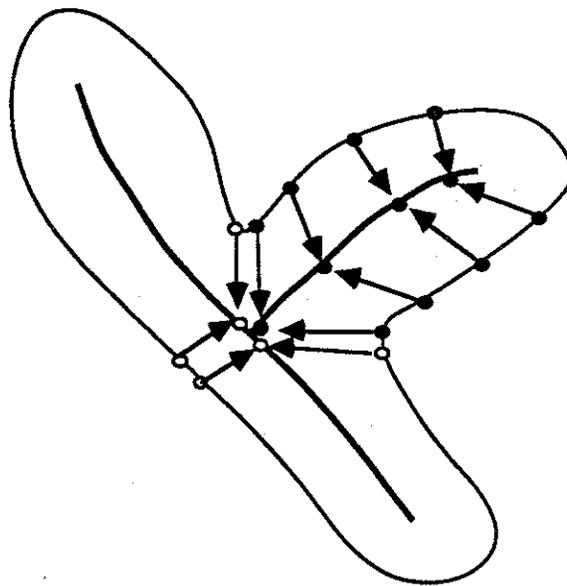


Fig. 7. Involute pairs and their relationship at a branch point in the IAS. The outer curve represents a single isointensity contour.

neighboring ridge. This results in a hierarchy that does not adequately reflect the ridge structure of the image.

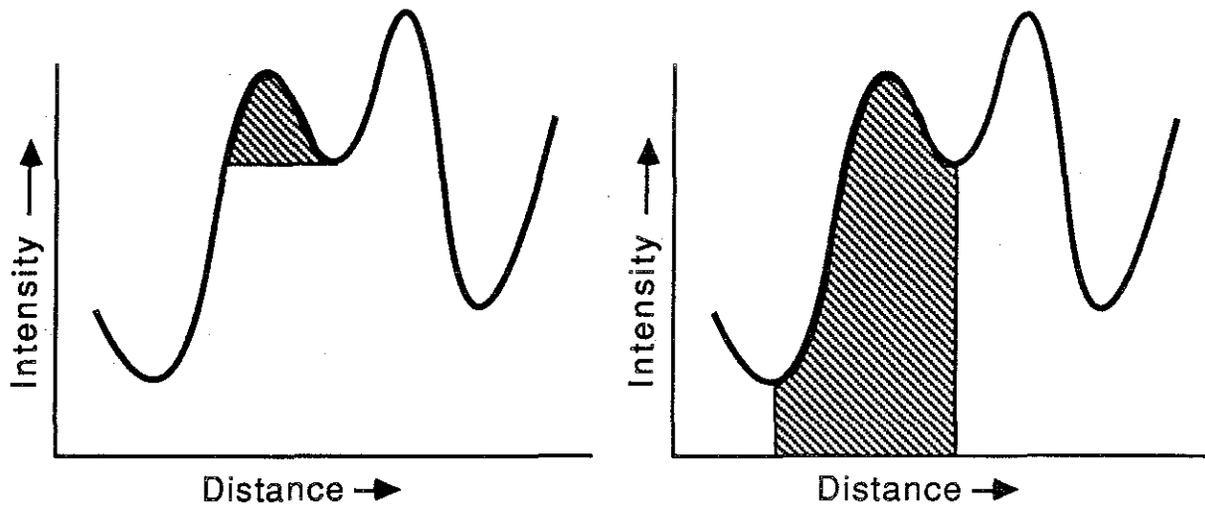


Fig. 8. IAS region (left) vs. reverse gravity watershed (right)

An alternative approach for producing a hierarchy on watershed regions has been tried with promising results. It examines surface properties rather than behavior under blurring. The concept is to hierarchically group together primitive regions based on a measure of the cost of moving from one patch to another. When you move from the maximum of one watershed, through the saddle point on the border of the two regions, to the maximum of the neighboring watershed, a reasonable cost measure is the effort spent going downhill and back up again. This would reflect a relatively low cost when moving along a ridge line (allowing watersheds along a ridge to be connected lower in the hierarchy) while moving from one ridge to another would be relatively more expensive (linking separate ridges higher in the hierarchy).

Partitioning the surface into watersheds is a relatively simple task. The image is scanned once initially to build a list of pixels for each discrete intensity. To grow each region from its highest intensity point downward these lists are processed one at a time. Starting with no primitive regions and the highest intensity list, you choose any pixel in that list as the seed for the first region. The region starts growing by repeatedly scanning the list, each time removing any pixels from the list and adding them to the region if they are neighbors of the region. If the region stops growing, but there are still pixels

Region/Subregion Identification. By identifying the individual branches of the IAS, we have partitioned the image into primitive regions, one region per branch. In order to group these regions in a hierarchical manner, we follow the simplification of the branching structure through scale space. Since the IAS branching structure simplifies through successive blurring levels, the goal is to track the branches from level to level, recording when individual branches annihilate.

At present we are associating each branch at a given blurring level (scale) with the branch at the successive scale with whose region its region has the greatest overlap. If more than one branch at the lower scale are associated with a single branch at the higher scale, a connection in the hierarchy is recorded. We are currently investigating more robust methods of tracking the IAS branches themselves.

2.1.4. Toward 3D IAS-Based Image Descriptions

All of the 2D definitions for IAS generalize to 3D. An intensity surface over three spatial variables has an IAS based on an intensity family of 3D medial axes based on maximal spheres. The resulting structure can be formed into a graph of connected elements, and these can be followed to annihilation into each other as the 3D image scale is increased by diffusion.

The extension of the algorithms to 3D images is straightforward. The intensity surface now becomes a 3D surface in 4-space: $f(u,v,w) = (x(u,v,w), y(u,v,w), z(u,v,w), I(u,v,w))$. The IAS is now a branching structure in 4D. The image symmetry function is now defined over a 4D data set.

We have had some success with the 3D algorithms on small, simple test images, but are still investigating the process. We have found that the proper identification of involutes is both a more difficult task due to the added spatial dimension, and more sensitive to the proper termination of the branch growing algorithm. We are currently modifying the algorithms to take advantage of a newly acquired MasPar SIMD parallel processing computer which should reduce the hindrances to our research caused by the large amounts of time and space the current sequential algorithms require.

2.2. Surface Regions with Containment

2.2.1. Reverse Gravity Watersheds

The goal of image segmentation is to produce visually meaningful primitive regions and a hierarchy on those regions that allows a sensible grouping of those regions into larger scale objects. The IAS achieves the first goal by associating regions of the image with individual branches of the IAS. It achieves the second goal by following the annihilations of the branching structures through scale space. Several aspects of the current implementation of IAS have led us to consider alternative approaches. One disadvantage of the IAS is that its dependence on iso-intensity slicing results in ridge flanks that do not necessarily reach down to valleys on both sides (see figure 8). Also the complex nature of the IAS branching and connectivity makes it difficult to follow the structure through scale space (consider that a loop at one scale can break into two branches at a larger scale). Finally, the computational cost of the IAS is significant, especially when considering the extension to 3D.

Instead of focusing on the medial axis of ridges, one can focus on their surface properties [Koenderink, 1990]. Consistency with the medial axis approach would suggest the use of vertex curves to define ridge tops, but this does not associate flanks with these tops. Another, not equivalent, means of characterizing a ridge is via properties of integral curves of principal curvature. Yet another, not equivalent, means of characterizing a ridge is via the properties of flow lines on the surface, that is curves of maximum slope on the surface. A weakness of this approach is its sensitivity to monotonic transformations of intensity. Nevertheless, the computational and theoretical simplicity of this approach has led us to work on its development and implementation.

Perhaps the simplest way of using flow lines to partition the surface into ridge/flank structures that span valley to valley is to group them into watersheds (with reverse gravity) (see figure 8). A reverse gravity watershed is a region of the image where if water were to fall on the region, the water would flow uphill and collect into a single maximum of that region. Once the surface has been partitioned into watersheds, you can produce a hierarchy on those regions by successively blurring the image and determining the order in which watersheds blur together. Unfortunately since a single long ridge may be composed of several watersheds (several local maxima along the ridge) we have found that they do not always blur together before the ridge blurs into a

in the list, then you choose one of these remaining pixels as a seed for a new region. Continue the process until the list is exhausted. Then start processing the next lower intensity list, adding neighboring pixels to the existing regions and creating new regions each time a list still has pixels that aren't neighbors to any existing region.

To compute the cost function between each pair of neighboring regions you need to determine the intensities at the maximum of each region and at the saddle point between the regions (cost equals the average of the two maximum intensities minus the saddle point intensity). The intensities at each maximum is easily determined since the maximum point is the seed point mentioned in the previous paragraph. The saddle point intensity can be found with two scans of the image, the first in row major order, the second in column major order, checking the intensity each time you cross from one region to another and retaining the largest of those intensities for each pair of regions. Once the cost function has been determined, the hierarchy is built simply by ordering the costs from lowest to highest. Typical results can be seen in figure 10.

Unfortunately, watersheds have significant disadvantages as the primitive region generator. It has already been mentioned that a single long ridge may be composed of several watersheds. Even worse, one watershed may contain more than one ridge structure. For example, a child ridge may branch off a parent ridge where the child ridge does not have its own local maximum. Therefore we are currently investigating other surface properties involving flow lines that may be used to partition the surface into ridge/flank primitive regions. For example, the separation and reconvergence of nearby flow lines may be utilized in defining a subridge. Also, the cost function used so far has been a very simple one and we plan to investigate more robust alternatives.

2.2.2. Towards 3D Watershed-Based Image Descriptions

Flow lines and their computation generalize directly to 3D. The definition of reverse gravity watershed regions in 3D, however, requires new concepts because in 3D there are more types of critical points and ridges/courses.

In a 2D image a critical point has either two negative principal curvatures (a peak, where the surface peaks along both principal directions), one negative and one positive principal curvature (a saddle, where the surface peaks along one direction but bottoms out along the other), or two positive principal curvatures (a pit, where the surface bottoms out along both

directions). In 3D there are four types of critical points, where the signs of the principal curvatures are respectively --- (peak), --+ (saddlepoint of type 1), -++ (saddlepoint of type 2), and +++ (pit). A reverse gravity watershed ridge in a 2D image goes from maximum to a saddlepoint to maximum. A reverse gravity watershed ridge in a 3D image goes from a maximum to a saddlepoint of the first type to a maximum. The role of the connections between saddlepoints of the two types needs to be investigated.

The 2D reverse gravity watershed algorithm extends naturally to 3D where each region is again grown from a maximum intensity seed point until it collides with a neighboring region. The cost function between pairs of regions is still the average of the maxima minus the largest intensity point along the border between the two regions (a saddlepoint).

2.3. Image Description Quality

The speed of the computation of the image description is not critical because it is totally automatic. Nevertheless, this speed affects the computing cost and the extendability of the method to 3D. We have implemented both 2D image description methods on a DEC3100 workstation (12 MIPS). Our programming has taken some care for speed, but we have not searched for optimal algorithms. Using 12 levels of blurring, the IAS-based method requires approximately six hours on a 256 x 256 image. Approximately 30% of this was taken by the distance calculations, approximately 20% by the surface collapses, and approximately 35% by the branch identifications. The reverse gravity watershed-based method is two orders of magnitude more efficient: the region growing and the hierarchy computation take less than a minute. The 3D generalization of this algorithm has been tested on a 128 x 128 x 32 MRI brain image. The computation time was approximately 15 minutes. We are in the process of transforming these to operate on a MasPar MP-1 4K-processor SIMD parallel computer (8000 MIPS). Speedups of a factor of approximately 100 are achieved by using the MasPar MP-1.

We have applied the algorithms for each of the two approaches to a number of 2D MR and CT images. These include objects that are elongated and possibly branching, such as the brain stem and the bronchial tree, and objects that are more compact, such as the cerebellum and kidneys. They also include objects with sharp edges and others with indistinct edges or portions of the organ boundary that have not been imaged as an edge. Examples of regions found on an MRI image of a brain are given in figures 9 and 10.

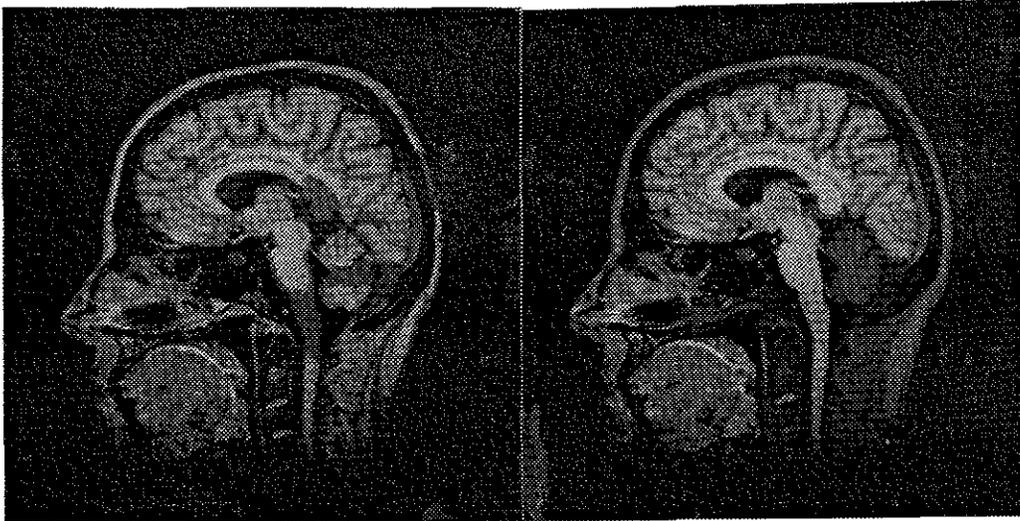


Fig. 9. Brain MRI regions defined via IAS
Left: Primitive regions. Right: The scalp and cerebellum selection each from a single point plus moving up the hierarchy. The blue region shows what would be added by moving once more up the hierarchy.

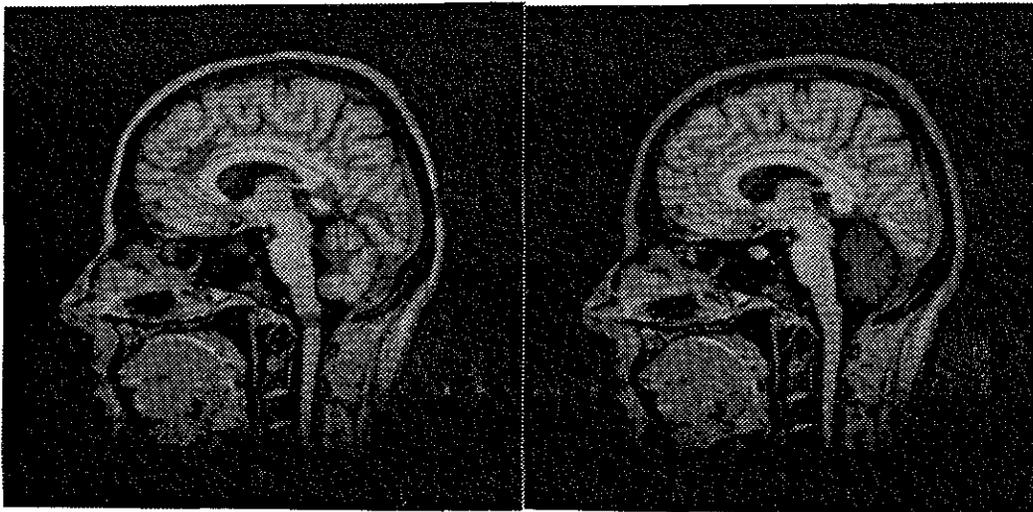


Fig. 10. Brain MRI regions defined from reverse gravity watersheds
Left: Primitive regions. Right: The scalp and cerebellum each from a single point plus moving up the hierarchy.

Most of the regions, both primitive ones and ones appearing higher in the quasi-hierarchy, from both approaches seem reasonable and provide a useful basis for a user to build meaningful object regions using the interactive tool described in section 3. For example, bronchial branches and their connections within the bronchial tree were well found in a CT chest image. The brain stem and the cerebellum in an MR brain image were each made up from just a few regions.

The IAS regions cover the full range of scale, and it appears that the regions as mathematically defined generate a reasonable hierarchy. The reverse gravity watershed primitive regions capture large sensible object regions more frequently than the IAS regions do. Our limited trial of the 3D generalization of the reverse gravity watershed calculation on a portion of a 3D MR image surrounding the cerebellum captures a number of reasonable 3D regions such as the scalp and most of the cerebellum, but the method sometimes failed to isolate visually sensible regions from neighboring regions.

Besides its inefficiency the main weaknesses of the IAS-based method have been that the resulting regions

- 1) sometimes are uncomfortably small,
- 2) connect image regions in an unnatural way, or
- 3) have boundaries that did not come up to the natural object boundary.

The small regions were frequently correct, but of a scale smaller than one desired to build a meaningful object. This is essentially a desirable situation because these regions are real ones that could for some small objects be just the regions needed to build these objects. However, for larger objects, some means of selecting only regions of a satisfactorily large scale may be usefully built into our interactive object region definition tool.

As for the unnaturally connected IAS regions and some of the small regions, it appears that these were largely the result of the computation not calculating the regions that the mathematics would define. Largely, this followed from improperly identifying involutes due to an imperfect surface contraction onto the IAS. An example of an area of disappointment was in trying to follow the convolutions of the cortex in an MR image. The primitive regions corresponded well to the individual folds, but moving up the hierarchy through scale space quickly brought in regions outside of the brain. We expect that improvements in the hierarchy and the primitive regions will result from

using non-stationary, anisotropic edge sensitive diffusion as opposed to the Gaussian blurring we have used so far and from using a surface energy function that better captures the geometry of the desired IAS. Also, improvements can be made in our method for tracking the IAS branches through scale space.

Finally, the unnatural IAS region boundaries seem to have two causes: problems with the computational algorithm and the fact, illustrated in figure 8, that region edges are tied to absolute intensity levels of one of the courses bounding the region.

The main weaknesses of the reverse gravity watershed method have been that sometimes small regions were simply incorporated into larger regions that were taken to be primitive and that some unnaturally connected regions were computed. We attribute both of these failings to two facts. First, only reverse gravity watershed regions and not regions reflecting flow line separations have been computed to date. Second, the cost function used to connect ridge pieces together requires improvement. Our short term research goals are to attack these problems.

3. Interactive Object Definition from Descriptions

3.1. 2D Interactions

In our approach to interactive object definition the user defines an object region using simple mouse-based interactions. The user causes a set of regions from the image description to be selected as an object region and displayed with a color wash to indicate their selection. While the option of editing an object region by pixel painting is provided, it is infrequently necessary, except when the object being defined really is not fully captured in the grey-scale differences in the original image data.

The simplest interaction of the user is that of pointing to and clicking on a pixel, which causes the smallest region containing the selected pixel to be chosen. If the user is in the mode of adding regions to the object set, the chosen region is added to the set. If the user is in difference mode, the chosen region is removed from the set. 'Region painting' is accomplished by holding the selection button down and sweeping the cursor over the region of interest. The union of the regions corresponding to all the selected pixels is selected for addition into or removal from the the object set, depending on the mode. A

"parent" button causes the presently selected description region to move to its parent in the description graph. This higher level region is then added to or removed from the object set, depending on the mode. It has proven quite useful to avoid the frequent use of the 'undo' operation by color washing, in a special color, additional pixels in the parent of the presently selected description region.

This method of object definition has proven to be very intuitive for physicians and to allow the definition of complicated anatomic objects in CT and MR slice images to take place in under a half-minute. The system implementing this method runs in an X environment on color workstations.

3.2. Speed of 2D Object Definition

The following examples of object definition speeds are typical using our interactive system. In our example MR brain image, using pointing and hierarchy traversals, the scalp could be identified in 10 seconds by selecting one primitive region followed by seven parent operations for IAS regions and eight parent operations for reverse gravity watersheds. The cerebellum required seven atomic actions over 8 seconds in each case, but the reverse gravity watershed method used one primitive region followed by six parent operations, whereas the IAS method required one primitive region selection followed by five parent operations followed by an additional primitive region selection. Selecting the cortex by groups consisting of the selection of a primitive region and a number of hierarchy traversal operations required six groups of 3-7 operations over 60 seconds for reverse gravity watersheds and 12 groups of 3-4 operations over 120 seconds for IAS regions. Thus the timings were about the same for the IAS regions and the reverse gravity watersheds except for the cortex, where the reverse gravity watersheds provided faster definition. In all these cases, with both methods, the object definition was rather speedy.

Region painting (see figure 11 for the selected regions) provided a considerable speedup for the more complex objects. For example, to define the cortex, a speedup by a factor of 3 was obtained for IAS and a factor of 2 for the reverse gravity watersheds. To define the cerebellum, no speedup was obtained.

Applying this object definition approach slice by slice to 69 MRI slices covering the cerebellum allowed us to specify a 3D region including the cerebellum and a bit of its background but no occluding organs. Only two of

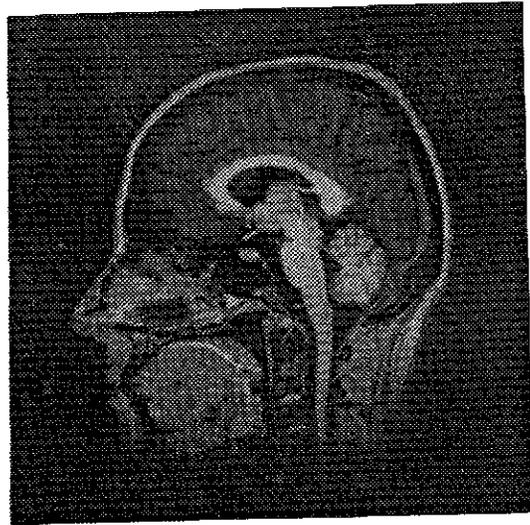


Fig. 11. A complex selected object region: the cortex

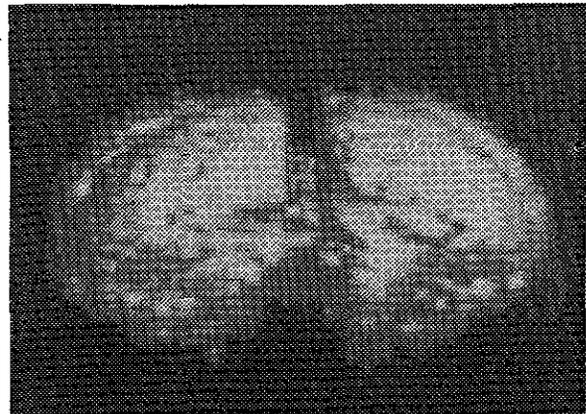
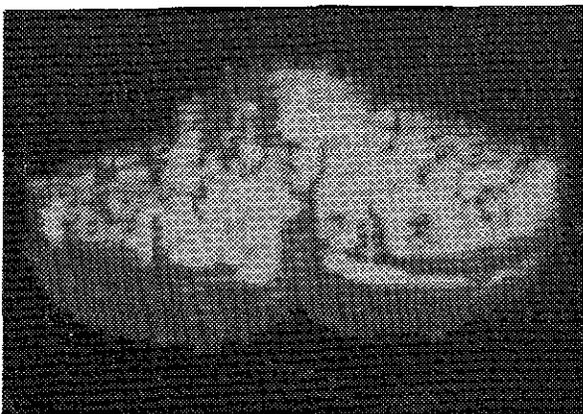


Fig. 12. Volume rendering of cerebellum from region selected from MRI study

these slices required editing by pixel painting. The application of Levoy's volume rendering technique [Levoy, 1988] to the image intensities in this 3D region produced the images in figure 12. Thus we have demonstrated the capability of this approach to select 3D regions of interest for the volume rendering of internal organs in 3D medical images.

3.3. Toward 3D Interactive Object Definition

We are working toward a system for interactive specification of 3D object regions based on quasi-hierarchical 3D image descriptions. We anticipate interactions essentially the same as the ones provided in our 2D system. In such an interactive system the user must be able to specify locations in 3-space, visualize the regions selected and unselected relative to the grey-scale values in the original 3D image data, and visualize the 3D objects in the selected region. The latter capability, with a speed of approximately one second from a 256^3 data set, will be provided by Pixel-planes 5, described in another paper in this volume [Fuchs, 1990]. The possibility of modification of viewpoint, rendering parameters, and clipping planes with optional display of the original grey-scale data on these planes, with a response via progressive refinement of about 0.1 seconds, is anticipated.

A major component of the system displays the slices making up the original image data as an array of small images ("postage stamps") in an X window on a CRT. Any postage stamp can be selected for display at full sampling in another X window. Pixel (voxel) pointing will be possible either on any postage stamp or on the fully sampled slice window. The presently selected 3D object region will appear as a color wash on the collection of slices, such that the selection of any pixel would cause a 3D region including pixels on many slices to be colored.

We are also considering region painting interactions using a 3D mouse controlling a 3D cursor that would appear on the volume rendered image as well as on the 2D postage stamp display. It remains to be seen whether a way can be developed to communicate in 3D the relation of the cursor to the 3D grey-scale data and whether such a means of communication is necessary.

In any case we are aiming to have such a system allow the volume rendered presentation of any anatomic region appearing in any 3D data set with

a selection time of under 2 minutes, after a considerably longer image acquisition and automatic image description computation stage.

Acknowledgements

We are indebted to John Gauch for his work on IAS and watershed algorithms and theory, on which much of this paper is based; to Drs. Fred Bookstein, Robert Gardner, and Jan Koenderink and Mr. Michael Tuchman for useful discussions on geometry; to James Coggins for his contribution in designing our object region definition system; Graham Gash for providing our image access, processing, and display environment; to Bo Strain for photography; and to Siemens, Inc. for providing the MRI image data. The research reported herein was carried out with the partial support of NIH grant number P01 CA47982.

References

- Fuchs, H, "Systems for Display of Three-Dimensional Medical Image Data", in this volume, 1990.
- Gauch, J, Oliver, W, Pizer, S, "Multiresolution Shape Descriptions and Their Applications in Medical Imaging." *Information Processing in Medical Imaging* (IPMI X, June 1987): 131-150, Plenum, New York, 1988a.
- Gauch, JM and Pizer SM, "Image Description via the Multiresolution Intensity Axis of Symmetry." *Proc. 2nd Int. Conf. on Comp. Vis.*, IEEE Catalog #88CH2664-1:269-274, 1988b.
- Gauch, JM, *The Multiresolution Intensity Axis of Symmetry and Its Application to Image Segmentation*, PhD Dissertation, University of North Carolina, 1989.
- Kass, M, Witkin, A, and Terzopoulos, D, "Snakes: Active Contour Models." *Proc 1st Int. Conf. on Comp. Vis.*, IEEE Catalog #87CH2465-3:259-268, 1987.
- Koenderink, JJ., *Solid Shape*. MIT Press, Cambridge, 1990.
- Levoy, M, "Display of Surfaces from Volume Data." *IEEE Computer Graphics and Applications* 8(3):29-37, 1988.
- De Moliner, R., *Kantenerhaltende Glättung*, Internal Report, Institut für Kommunikationstechnik, Fachgruppe Bildwissenschaft, Eidgenössische Technische Hochschule, Zürich, Switzerland, 1989.
- Perona, P and Malik, J., "Scale Space and Edge Detection Using Anisotropic Diffusion", *Proc. Workshop on Computer Vision*: 16-22, IEEE, 1987.
- Yuille, AL and Poggio, T, "Scaling Theorems for Zero-Crossings." A.I. Memo 722, MIT, 1983.