

# A Multiresolution Hierarchical Approach to Image Segmentation Based on Intensity Extrema

LAWRENCE M. LIFSHITZ, MEMBER, IEEE, AND STEPHEN M. PIZER, MEMBER, IEEE

**Abstract**—The aim of our research has been to create a computer algorithm to segment grayscale images into regions of interest (objects). These regions can provide the basis for scene analysis (including shape parameter calculation) or surface-based shaded graphics display. The algorithm creates a tree structure for image description by defining a linking relationship between pixels in successively blurred versions of the initial image. The image description describes the image in terms of nested light and dark regions. This algorithm can theoretically work in any number of dimensions; the implementation works in one, two, or three dimensions.

Starting from a mathematical model (developed by Koenderink) describing the technique, our research has shown that

- by explicitly addressing the problems peculiar to the discreteness of computer representations the segmentation described by the mathematical model can be successfully approximated;

- although the image segmentation performed sometimes contradicts semantic and visual information about the image (e.g., part of the kidney is associated with the liver instead of the rest of the kidney), simple interactive postprocessing can often improve the segmentation results to an extent sufficient to segment the region desired;

- the theoretical nesting property of regions, originally thought to hold in all circumstances, does not necessarily apply to all pixels in a region.

The interactive postprocessing developed selects regions from the descriptive tree for display in several ways: pointing to a branch of the image description tree (which is shown on a vector display), specifying by sliders the range of scale and/or intensity of all regions which should be displayed, and pointing (on the original image) to any pixel in the desired region.

The algorithm has been applied to approximately 15 CT images of the abdomen. While performance is frequently good, work needs to be done to improve performance and to identify and extend the range of images the algorithm will segment successfully.

**Index Terms**—Computer vision, hierarchical analysis, image segmentation, interactive graphics display, Morse theory, multiresolution image processing, pattern recognition.

## I. INTRODUCTION

### A. Segmentation as a Step Towards Interpretation

**A**N image as it is stored by a computer is just a multidimensional array of pixel values. Although we as humans may look at the displayed image and recognize it as meaningful, the computer must algorithmically analyze the array of pixel values before it can reach any conclusions about the content of the image.

A computer must deal with objects in an image, as objects and not just as unrelated pixels, for many reasons:

- 1) computer vision (e.g., robotics);
- 2) computer analysis of quantitative properties of objects;
- 3) computer manipulation of objects for image display via man-machine interactions;
- 4) object-based nonstationary image restoration.

Before any of the above mentioned object-related actions can be taken, one or two conditions must be met:

- 1) The object must be recognized as an entity distinct from other objects in the image (i.e., pixels belonging to the object must be understood to be related in some way).
- 2) The entity must be labeled. It must be understood that it is, in fact, the specific object that is being searched for.

Most image processing techniques perform step 1 first and independently of step 2. Step 1 is commonly called the *image segmentation* step.

### B. The Stack

There exists a large number of image segmentation techniques. Most techniques, however, fall into one of three broad categories: region growing, boundary detection, or multiresolution. Multiresolution techniques (e.g., the Pyramid [1], the DOLP transform [2], complete trees [3], edge focusing [4], [5]–[7]), attempt to gain a global view of an image by examining it at many different resolution levels. The lower resolution provides a global view of the image, and the higher resolution provides the detail.

The stack approach initially proposed by Koenderink [8] is a multiresolution image description and segmentation technique. The stack calculates image segments and an image description tree by associating every pixel in an image with a local intensity extremum [9]. The approach focuses on decomposing the image into light and dark spots, each, except for the spot representing the whole image, contained in others. Thus a face might be described as a light spot containing a light spot (a reflection from the forehead) and three dark spots (the mouth and the regions of the two eyes). In turn the eye regions would be described as containing a dark spot (the eyebrow), a light spot (the eyelid), and a dark spot (the eye), with the latter containing a light spot (the eyeball) which itself contains a dark spot (the iris) which finally contains a yet darker spot (the pupil). We call these light and dark spots,

Manuscript received August 15, 1988; revised August 11, 1989.

L. M. Lifshitz is with the Biomedical Imaging Group, University of Massachusetts Medical School, Worcester, MA 01655.

S. M. Pizer is with the Department of Computer Science and Radiology, University of North Carolina, Chapel Hill, NC 27514.

IEEE Log Number 9034377.

at whatever scale, *extremal regions*, since they each include a local intensity maximum or minimum.

The stack approach is promising for several reasons:

1) It is based on considering the image simultaneously at multiple levels of resolution and can therefore use global information.

2) It produces a natural tree structured extremal region description useful for matching purposes.

3) It has a firm mathematical description based upon differential geometry and Morse theory, giving investigation and extension firm footing.

1) *Hierarchical Descriptions from Multiresolution Processing*: The image description in terms of extremal regions can be produced by following the paths of extrema in a stack of images in which each higher image is a slightly blurred version of the previous one. Progressively blurring an image causes each extremum to move continuously and eventually to annihilate as it blurs into its background. An *extremum path* is formed by following the locations of an extremum across the stack of images.

Intensity change must be monotonic (increasing for dark spots and decreasing for light spots) as one moves along an extremum path from the original image towards images of increased blurring. As illustrated in Fig. 1, while following each extremum path one can associate each path point with the isointensity contour that is at that point's intensity and that surrounds that extremum in the original image [8]. The points (pixels) in the original image thus associated with each extremum path then form an extremal region (see Fig. 2). Equivalently, each contour (nonextremum) point in the original image can be associated with its extremum path by linking the point to the closest point with the same intensity at the next level in the stack and continuing this linking through the levels until the extremum path is reached (see Fig. 3). This process defines an *isointensity path*.

As indicated above, extrema annihilate when the blurring is sufficient to make the light or dark spot blur into an enclosing region. The amount of blurring necessary for an extremum to annihilate is a measure of the importance or scale of the extremal region, including the subregions that it contains. The intensity of the topmost point on an extremum path is the path's annihilation intensity. This is the intensity of the isointensity contour that forms the boundary of the associated extremal region. The annihilation intensity bounds from below (above) the intensities in the extremal region if the associated extremum is a maximum (minimum) and if there are no extremal subregions enclosed.

2) *A Tree of Extremal Regions for Image Description*: As illustrated in Fig. 1, when an extremum annihilates at some annihilation intensity, another region's isointensity contour at the intensity encloses the region associated with the annihilating extremum [8]. Thus, a containment relation among extremal regions is induced by the process. This set of extremal regions together with their containment relations can be represented by an *extremal region tree* in which nodes represent extremal re-

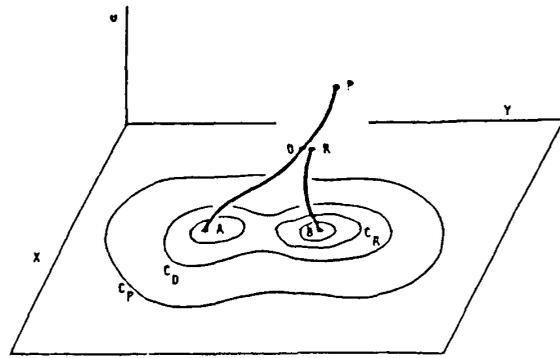


Fig. 1. Extremum paths and associated isointensity contours.

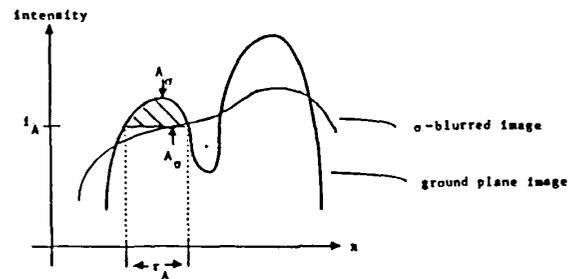


Fig. 2. Extremum annihilation by blurring and the consequent extremal region. If the extremum at  $A_0$  has moved to  $A_1$  at annihilation time,  $r_A$  gives the extremal region and  $i_A$  its minimum intensity.

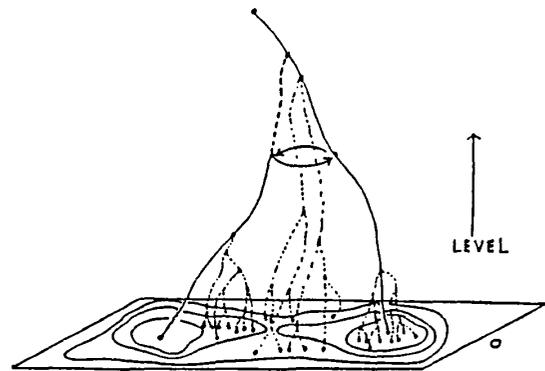


Fig. 3. Extremum paths (solid lines) and isointensity paths (broken lines). Isointensity contours are indicated in the original image (level 0). The left extremal region is a subregion of the right extremal region.

gions and a node is the child of another if the extremal region that it represents is immediately contained by the extremal region represented by the parent (see Fig. 4). The root of the description tree represents the entire image.

Each node in the extremal region tree can be labeled with its scale, i.e., the total amount of blurring necessary for its extremum to annihilate. Furthermore, each node can be labeled with the annihilation intensity of the associated extremum. Finally, the node can be labeled with its size, shape, orientation, location, or other spatial characteristics.

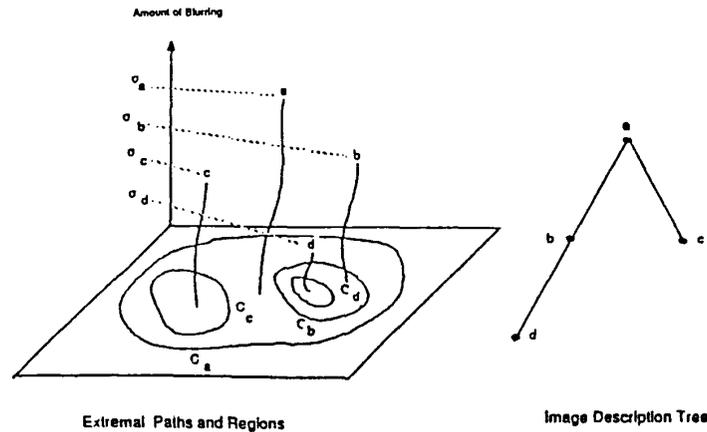


Fig. 4. Left: Extremum paths with their regions and scales. The vertical direction represents blurring. Right: The associated extremal region tree.

II. MATHEMATICAL PROPERTIES OF THE BASIC STACK

Our interest is in determining how maxima and minima behave when an image is blurred. Maxima and minima are of interest because these are the basic structures to which nonextremum points link and thereby define extremal regions in the original image. Several properties of extrema stem from basic results in Morse theory. One such result is that the number of minima, maxima, and saddle points cannot change (at least in the “typical” case) except through a bifurcation which causes both an extremum and a saddle point to appear (or annihilate). They both appear or annihilate at the same blurring level and same position (e.g., they must meet to annihilate). Similarly, the characteristic way in which the bifurcation occurs can be studied by examining simple “generic” cases. Morse theory precisely defines the meaning of the terms “typical” and “generic”. Some of the applicable results of Morse theory are presented below. Differences between the results for the canonical cases presented and our particular case (which has the additional restriction of having to satisfy the heat equation) have some important theoretical implications which will be presented following the canonical exposition.

A. Morse Theory Basics for the Generic Case

Fig. 5 shows the intensity of the extrema for  $I(x, t) = x^3 + tx$ . The way the maximum and minimum (when  $t < 0$ ) in this particular function move towards each other (as  $t$  increases) and annihilate (when  $t = 0$ ) is “typical”.  $I(x, y, t) = x^3 + tx \pm y^2$  is the “typical” or generic description in two dimensions (with a maximum or a minimum annihilating with a saddle). Although Morse theory considers this function to be equivalent to all other generic extrema annihilations, the theory’s notion of equivalence is too flexible for our purposes. It hides differences between functions that are of considerable importance to the extremal region definition. In the following section we describe the blurring scheme used to create the stack of

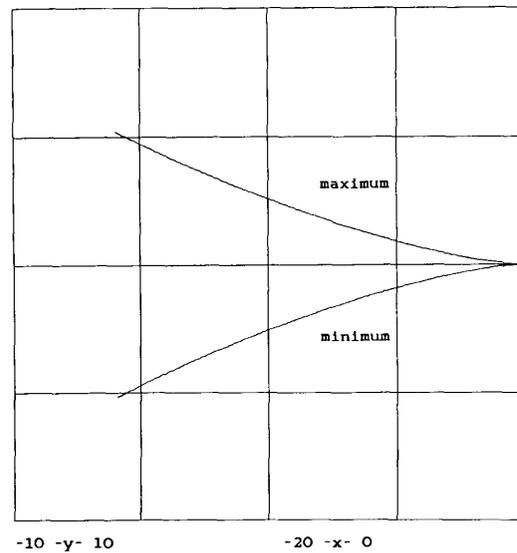


Fig. 5. Extremum intensities (ordinate) for  $x^3 + tx$  as a function of time (abscissa). The abscissa ranges from  $-20$  to  $0$ , the ordinate from  $-10$  to  $10$ . The two paths meet at time  $t = 0$ .

lower resolution images from an initial image. It is then shown that the generic annihilation presented above does not evolve over time in a manner consistent with this blurring scheme. Modification of the generic annihilation to create a new function that does satisfy this constraint yields a function with important differences in the way extremal regions are allowed to form.

B. Embedding an Image in a Family Based Upon Gaussian Blurring

The most important property guiding the choice of a convolution kernel with which to create a stack of images is one of causality. We are concerned with extremal paths when creating the tree representation from a stack of images; ideally an extremum should not be allowed to be

created (i.e., an extremal path to start) at any resolution level except the original one.

Another important property of the embedding is that it should be smooth. This means that intensity changes occur in a continuous manner as scale space is traversed. This implies that extremal paths and iso-intensity paths are smooth curves in position-scale space.

Koenderink [8] examines the specific case of intensity creation. No intensity should exist at a low resolution level which cannot be traced (in a continuous manner) to an identical intensity at a higher resolution. Applying the constraints of causality, homogeneity, and isotropy, he derives the following necessary and sufficient relationship at the locations of the extrema:

$$I_{xx}(x, y, s) + I_{yy}(x, y, s) = \alpha^2 I_s(x, y, s),$$

where  $I(x, y, s)$  is intensity,  $s$  is scale or resolution,  $\alpha$  is a constant, and a subscript denotes a partial derivative. If this equation holds everywhere in an image the constraints are also maintained. This is the heat (or diffusion equation).

Despite the fact that no new intensity levels are created, extrema can be created in images of dimension greater than one. An easy example to help in the visualization of how this can occur in two dimensions is as follows. Imagine two broad, high mountains with a deep wide valley between them. One mountain is higher than the other. Connect these two mountains with a thin ramp bridge between their tops. The heights of the mountains, the valley, and the bridge represent intensity levels. The shorter mountain is not a local maximum because the ramp connects it with the higher mountain. But as diffusion occurs on the intensity distribution represented by this geography, the intensities of points represented by the bridge will decrease, since the deep valley is on both sides of it. It will quickly turn into a bridge with a deep dip in the middle. This will turn the smaller mountain into a local maximum.

The diffusion (heat) equation seems to be the best alternative available to guide blurring, since it does not create new intensity levels. The ability to create extrema is undesirable; features exist at low resolution levels which do not exist at higher ones. Nevertheless, this method of extrema creation is preferable to others which may also create new intensity levels. There are several additional reasons for desiring that the embedding satisfy the heat equation everywhere. First, if the image satisfies the heat equation, the solution of the equation will depend continuously on the data, i.e., the original image and boundary conditions [10]. Second, if the image satisfies the heat equation,  $I(x, y, t > 0)$  is infinitely differentiable (in  $x$  and  $y$ ) even if  $I(x, y, 0)$  is not.

To blur an image, one should convolve it with the filter kernel that is the Green's function for the diffusion differential equation. The solution to a differential equation depends upon its initial conditions and boundary conditions. The initial condition is  $I(x, y, 0) = H(x, y)$ , where  $H(x, y)$  is the original image. If the original image has no

boundaries (i.e., takes up all of  $\mathbf{R}^2$ ), the solution yields a Gaussian filter kernel.

Unfortunately, if the image is not infinite in extent, the simple result of a Gaussian filter kernel is not necessarily correct. There are several ways the original image might be modified to eliminate its boundaries, and thereby maintain a Gaussian kernel solution to the diffusion equation. Mirroring of the image at its boundaries essentially creates an infinite image. An alternative approach to eliminating boundaries is to wrap-around an image at the boundary. This means that points outside a boundary are interpreted as being inside the boundary on the opposite side of the image. Both of these approaches create undesirable artifacts [11] and are therefore not used.

A third method of handling the boundary conditions is to solve the diffusion differential equation explicitly for the finite plane with the given initial conditions (the initial image). The general solution to the finite domain heat equation is [12]

$$\begin{aligned} I(\zeta, T) = & \int_{\Omega} K(x, \zeta, T) I(x, 0) dx \\ & + \int_0^T dt \int_{x \in \partial\Omega} \left[ K(x, \zeta, T-t) \frac{dI(x, t)}{dn} \right. \\ & \left. - I(x, t) \frac{dK(x, \zeta, T-t)}{dn} \right] dS_x. \quad (1) \end{aligned}$$

Here  $I(\zeta, T)$  is the image at time  $T$ .  $K(x, \zeta, T)$  is a Gaussian of width  $T$ .  $\Omega$  is the region the image is defined on, and  $\partial\Omega$  is its boundary.  $dS_x$  is just the length element along the boundary, and  $d/dn$  is the derivative in the direction normal to the boundary. This solution reduces to a convolution of the image with a Gaussian kernel if only the first term is nonzero. The conditions under which this is so will now be discussed.

The second term in the second integral will be equal to zero if the intensity of the image is zero along the boundary. An image which obeys the diffusion equation and has boundary values of zero can be created from the initial image (assuming the boundary conditions of the initial image are time independent). This is done by subtracting off an image which is invariant under blurring and has the identical boundary conditions.

The first term in the second integral will be nonzero unless an additional constraint is added, that the image be insulated, i.e., have  $dI(x, t)/dn = 0$  at the boundary. This is not an unreasonable constraint to impose. The image is already discretely sampled in space. We can therefore always assume that the unknown intensity distribution between a boundary pixel and its neighbor in the interior is such that the constraint is satisfied without contradicting any known data or forcing the interpixel intensity distribution to be unnatural.

We can conclude that convolution of a bounded image with a Gaussian kernel (whose contribution is set equal to zero if outside the image boundary) can indeed be considered to be an appropriate solution to the diffusion equation

for an *insulated* bounded image with zero intensity along its boundary. Since this approach is valid, it is the method used in this research. Toet [13] also uses this technique but did not show that it was in fact valid.

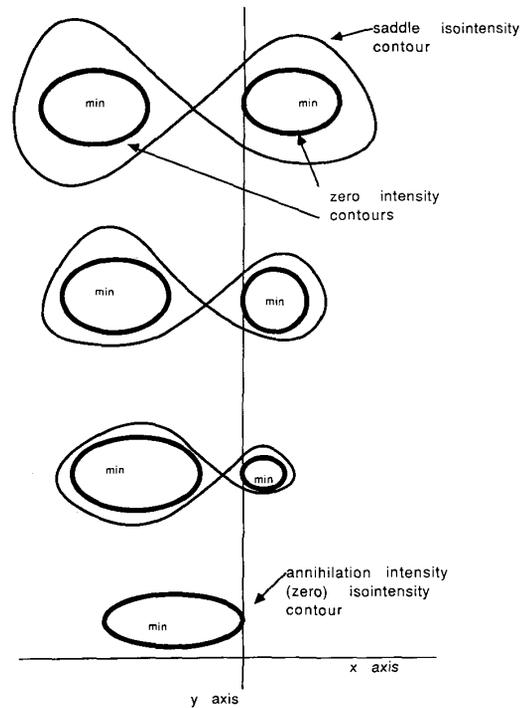
### C. Containment for Extremal Region Paths in the Generic Case

Deciding on an embedding (blurring) scheme for the original image in a family of lower resolution images is not enough. Points in the image at one resolution level must also be associated with points in the image at another resolution level. This will define a path through resolution space for each point in the original image. The way that these paths (links) join up with each other will induce a decomposition of the original image into nested regions. All those paths which link to the same extremal path define the extremal region associated with that extremal path. Criteria for linking both nonextremum points and extremum points at one level to the appropriate pixels at the next level are needed to guide the creation of this path structure.

There are two criteria the path of a nonextremum point should satisfy. First, the intensity should stay constant along the path; the path of a nonextremum is therefore also called an isointensity path. Second, the point should move along the isointensity surface (in  $x, y, s$  space) in a path of steepest ascent (where  $s$ , the resolution dimension, is up). Let the direction in which a path should move be defined as  $\vec{v} = (v_1, v_2, v_3)$ , where the three components are in the  $x, y$ , and  $s$  directions respectively. These constraints then imply [8]  $\vec{v} = (-l_x l_x - l_y l_y, (l_x)^2 + (l_y)^2)$ . This is the direction the path of a nonextremum point should take. At an extremum  $l_x$  and  $l_y$  are both equal to zero, so  $\vec{v}$  is a null vector. In other words, once a nonextremum path meets an extremum,  $\vec{v}$  no longer specifies the direction in which to proceed. Criteria for the path direction of an extremum hold from then on. The extremal region associated with an extremum path is just all those points whose paths eventually join with the extremal path.

The path criterion for extrema is very simple. Each extremum is isolated (in the  $x$ - $y$  plane) in the typical case. That is, there are no other extrema in some neighborhood around each extremum. In addition, their positions move continuously through scale space. It is therefore always immediately evident what path each extremum should take. Contrary to isointensity path criteria, intensity along an extremal path changes. Intensity along the path of a maximum will decrease with decreasing resolution, while intensity of a minimum will increase. When an extremum annihilates, its path continues on as an isointensity path (see Fig. 3). The path criteria for nonextremum and extremum points determine which regions in the original image become associated with each extremal path.

Let us now examine the generic case of extremum annihilation in an attempt to understand the rules guiding extremal region formation. The generic description of a saddle and a minimum annihilating was stated at the beginning of this section to be  $I(x, y, t) = x^3 + tx + y^2$ .



Time increases (more blurring) as we move downwards from one plot to the next.

Fig. 6. Saddle and zero intensity contours at various times until annihilation. Nondiffusion case.

The simplest way to visualize this (see Fig. 6) is to imagine a second minimum (which will not annihilate) existing on the other side of the saddle point. The saddle exists between the two minima so that the isointensity curve through the saddle point surrounds the two minima. Annihilation of the saddle with one minimum takes place at  $x = 0, y = 0, t = 0$ . The annihilation intensity is zero. At some initial time  $t_0 < 0$  the zero intensity contours surrounding the minima lie inside the isointensity contour of the saddle. As time progresses, the minimum that will annihilate and the saddle point move towards each other. The lobe of the saddle isointensity contour which surrounds this minimum gets smaller, as does the zero intensity contour which surrounds the minimum. The zero intensity contour remains inside the saddle intensity contour the entire time. The zero intensity contour surrounding the annihilating minimum and the lobe of the saddle contour surrounding the annihilating minimum become identical for the brief instant as annihilation occurs. To determine the nature of the extremal region such a scenario will produce, one must examine which nonextremum paths link to the annihilating extremum. Nonextremum paths are isointensity paths. Those starting out inside the zero intensity contour surrounding the extremum have intensities less than zero. They cannot cross the zero intensity surface represented by the zero contour in scale space. Yet the zero intensity contour eventually collapses into the extremum point, enclosing no area at annihilation

time. The only place for the nonextremum paths to go is to link up with the extremum path. This means that all isointensity paths through resolution space that start off inside the zero intensity contour *must* eventually link up to the annihilating minimum. This region represents the extremal region associated with the annihilating extremum.

#### D. Noncontainment for Extremal Region Paths in the Gaussian Case

Imposing the constraint of satisfying the heat equation (i.e., convolution with a Gaussian) modifies the conclusions reached above about the manner in which extrema annihilate. Once we restrict the embedding to a particular class of smooth embeddings, namely those satisfying the heat equation, it becomes very difficult to know which descriptions are typical in this restricted subclass. Nevertheless, specific cases can be examined which must fit into the diffusion framework.

Blurring  $x^3/6 + tx$  with a Gaussian produces the prototypical diffusion annihilation. In the one-dimensional case nothing changes significantly from the canonical (nondiffusion) description given above. The time parameter in the formalism gets replaced by  $\sigma^2/2$ , where  $\sigma$  is the total standard deviation of the Gaussian blurring done so far. The  $x^3$  term gets a factor of  $1/6$  in front of it so that the prototype equation satisfies the one-dimensional heat equation.

The two-dimensional annihilation via a diffusion process changes from the generic annihilation in a subtle but very important way. The annihilation equation representing the diffusion case is  $I(x, y, t) = x^3/6 + tx + t + y^2/2$ . We have been forced to add a term solely in  $t$  to preserve the diffusion characteristic! This has many important consequences. We will now examine this equation, in detail; identical conclusions can be drawn from looking at the  $-y^2/2$  case which is for a saddle and a maximum.

The intensities of these critical points differ from the nondiffusion case (compare Fig. 5 to Fig. 7).  $I(\hat{x}, 0, t) = \pm(-2t)^{3/2}/6 \pm t(-2t)^{1/2} + t$ , where  $\hat{x}$  is the position of a critical point, + used for  $\hat{x} > 0$  ( $I(\hat{x}) < 0$ ), whereas for  $\hat{x} < 0$  ( $I(\hat{x}) > 0$ ), - is used; the square root is interpreted as being the positive root. This equation is valid for  $t \leq 0$ . Now examine  $I(\hat{x}, 0, \hat{t}) = 0$ , i.e., those times when the critical point intensity equals zero. We find that, as before, both the saddle and the minimum intensities equal zero at  $t = 0$ , the annihilation intensity and time. But we also now have zero intensity at  $\hat{t} = -9/8$  for the saddle intensity! The saddle intensity, which starts out greater than zero (for time  $< -9/8$ ), gradually decreases becoming less than the annihilation intensity when  $t > -9/8$ . This is confirmed by setting  $\partial I(\hat{x}, 0, t)/\partial t = 0$  and noting that the saddle (but not the minimum) has a turning point in intensity at  $t = -1/2$ . From  $t = -1/2$  to  $t = 0$  the saddle intensity is increasing, as is the minimum intensity.

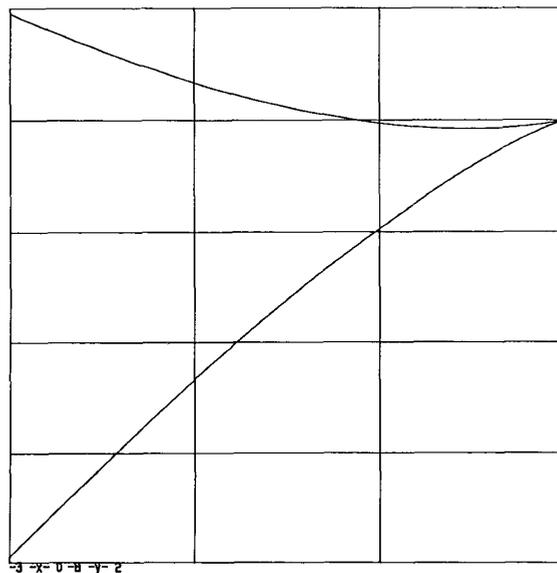


Fig. 7. Saddle (upper curve) and minimum (lower curve) intensities (ordinate) with respect to time (abscissa). The time axis ranges from  $-10$  to  $0$ , and the intensity axis ranges from  $-8$  to  $2$ . Diffusion case.

There are significant consequences of the saddle intensity dipping below the annihilation intensity and then rising back up to it. In the canonical case analyzed previously, when a saddle existed between two minima, the isointensity curve through the saddle point surrounded the two minima (see Fig. 6). The saddle intensity contour remained outside the zero intensity contour until annihilation time. This is not true for the diffusion case (see Fig. 8). The zero intensity contour, which starts out inside the saddle intensity contour, ends up outside the saddle intensity contour! The saddle intensity contour and the zero intensity contour become identical at the instant the zero intensity contour surrounding the annihilating minimum joins the zero intensity contour surrounding the other minimum. After this time the zero intensity contour surrounds the saddle intensity contour and both of the minima. The "lobe" of the zero intensity contour that surrounds the minimum that will annihilate gradually contracts and "pinches off" at  $x = 0, t = 0$  (see Fig. 8).

The fixed point of  $I(x, 0, t)$ , which does not change intensity with time, is now at  $x = -1$  (see Fig. 9). Intensities for all  $x < -1$  continuously decrease, while those for points  $x > -1$  continuously increase. Hence, if an isointensity path is at a particular  $x > -1$  at some  $\tilde{t}$ , then at time  $\tilde{t} + \delta t$  it will have to move *towards* the minimum, to compensate for the fact that all the intensities around it are rising. There are points with  $x < -1$  that have negative intensities. These points will move away from the minimum even though their intensities are less than the annihilation (zero) intensity! The annihilation isointensity surface in  $(x, y, t)$  space is still a concave cap, but now it has a hole in its side where it joins up with the other zero intensity surface; isointensity paths can escape from

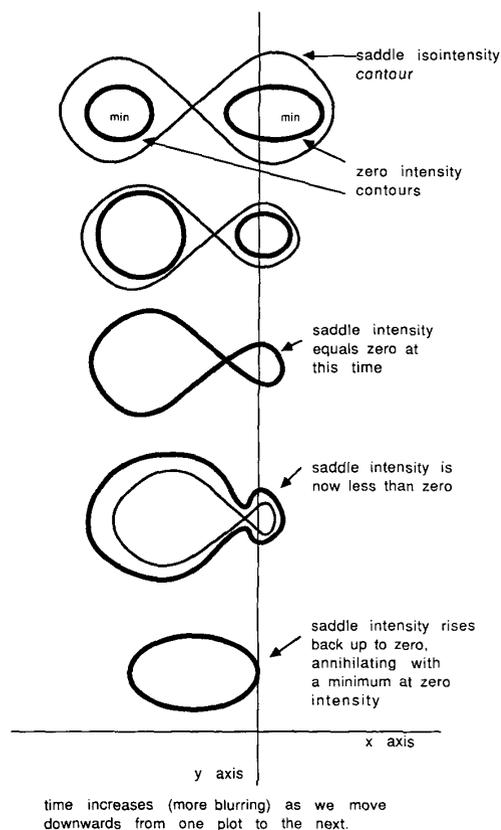


Fig. 8. Saddle and zero intensity contours at various times until annihilation. Diffusion case.

under the cap by sneaking out through this hole, so they are not forced to join the annihilating minimum's path!

It is difficult to describe mathematically the nonextremum paths which escape through this hole. The isointensity paths are integral curves of the vector field  $\vec{v} = (-I_x I_t, -I_y I_t, I_x^2 + I_y^2)$ . What needs to be done is to substitute into this the formula for  $I(x, y, t)$  and solve analytically for the isointensity paths. Unfortunately, an analytical solution of these differential equations is intractable. Nevertheless, qualitative examination of the vector field *directions* can yield a simple proof that some nonextremum paths do escape. The proof is as follows. For the prototypical diffusion case we have  $\vec{v} = (-(x^2/2 + t)(x + 1), -y(x + 1), (x^2/2 + t)^2 + y^2)$ . This means that any point with  $x < -1$  and positive  $y$  coordinate will have a positive  $y$  component in  $\vec{v}$ , and such points will move away from the  $x$  axis, which is where the saddle and minimum are. Furthermore, if such a point should happen to try to move towards an  $x$  coordinate greater than  $-1$ , it will not get there. This is because at  $x = -1$  both the  $x$  and  $y$  components of  $\vec{v}$  are zero and the point will stay fixed at  $x = -1$  for all future time. By showing that all points with  $x$  coordinates less than  $-1$  move away from the  $x$  axis (if not on it to begin with) and that these points can never move to an  $x$  value greater than

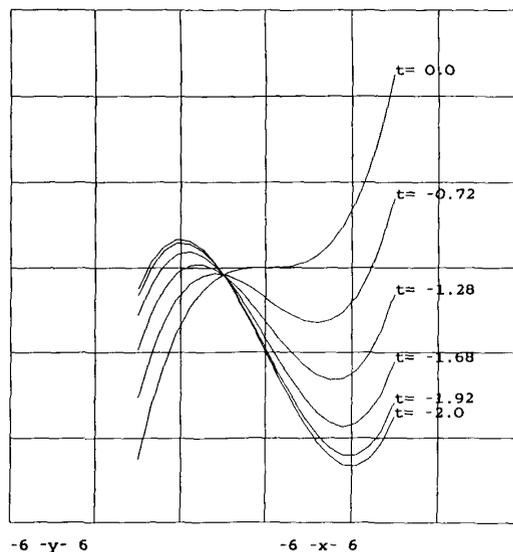


Fig. 9. The prototypical annihilation satisfying the diffusion equation.  $I = x^3/6 + tx + t$  is graphed for various values of  $t$ . The abscissa ( $x$  axis) and the ordinate (intensity axis) both range from  $-6$  to  $6$ .

minus one, we have shown that these points (paths) can never meet either the saddle or the minimum path. Yet many of these points have intensity between that of the minimum and the annihilation intensity (zero). Therefore, some points inside the annihilation intensity contour do not link to the minimum path.

By simulating the blurring and linking process, we have been able to display isointensity paths which start out inside an annihilation intensity contour, and yet do not link up to the extremum path of the annihilating minimum. In order to make the simulation as accurate as possible, an analytic model of the blurring process was used.

Despite the surprising theoretical result that nonextremum paths can escape from the extremal region they originate in, we have yet to observe any clear examples of escaping paths when working with actual images. This is most likely due to the fact that extremal regions are not sufficiently isolated from other regions to display the type of behavior shown in the "prototypical diffusion" annihilation case. The behavior of that case stemmed primarily from the fact that all points with  $x$  coordinate less than  $-1$  continually decreased in intensity. In an actual image this would not occur since there would be another minimum at some  $x$  less than  $-1$  and this would tend to cause intensities in the region to increase under blurring. In addition, our sampling resolution may be coarse enough to miss some small regions.

### III. THEORETICAL ISSUES DUE TO DISCRETENESS

The entire theory upon which the stack algorithm is based applies to continuous Morse images embedded continuously in resolution space. Unfortunately, images of this form cannot be handled by a digital computer. The continuous image must be approximated by one which is

spatially discrete (i.e., made up of pixels). The smooth embedding of the image in resolution space is approximated by a stack of images each derived from the previous one by convolution with a blurring kernel of non-infinitesimal width and finite extent. Nonextremum and extremum paths, which theoretically are continuous paths, become represented by a chain of links from a pixel in one image to a pixel in the next image to a pixel in the next image and so on. The best manner in which to create these approximations and the complications such approximations create have been investigated. See [11] for more details.

#### IV. EXTENSIONS TO THE STACK-EMBEDDING SCHEMES

The basic stack algorithm does not always segment an image in the most preferred manner. It can take pixels that, visually, all belong to one region and link them to different extremum paths (and hence different regions). It may also join pixels together in one region that visually are very similar but semantically should not be joined (i.e., they are in different organs which happen to abut each other). The algorithm works well, but not always well enough. It therefore seems reasonable to try to adjust it in a way which would improve its performance. The aforementioned problem is one of accuracy in segmentation. An accurate tree description implies the existence of subtree structure whose leaves represent completely an area of the image which we subjectively determine to be meaningful, and only that region (e.g., it would include all pixels in the liver, and none not in the liver).

The question is how the stack algorithm should be modified to produce a more accurate image segmentation. There are two modifications to the algorithm which come immediately to mind. One is to alter the way the original image is embedded in the multiresolution stack. A different embedding should cause different extremal regions to form. The second approach is to alter the linking criteria. Modification of the linking criteria has not been investigated but would be an area for promising research in the future. Of course, both of these modifications could be applied concurrently.

Intuitively one would expect that modifying the shape of the blurring kernel to reflect the shapes of the regions of interest would yield better segmentation results than always performing stationary, isotropic Gaussian blurring. Any blurring scheme adopted should still be required not to create any new intensity levels as blurring proceeds. Koenderink's main criterion for this is that  $I_{xx} + I_{yy} = \alpha^2(x, y, t)$  at the extrema. Of course, the embedding should also remain smooth. This allows for considerable flexibility in choosing a blurring strategy. We have examined the effects on extremal region formation and nesting of both stationary anisotropic blurring, and of nonstationary blurring.

##### A. Anisotropic Stationary Blurring

Theoretical calculations, simulations, and application to actual images all clearly show that iso-intensity paths

are able to move more quickly in the direction in which blurring is faster. In addition, some paths which before headed *away* from an extremum may now head *towards* it, thus the extremal region associated with an extremum is clearly different in the anisotropic case from the isotropic blurring case [11].

##### B. Nonstationary Blurring

Spatially variant blurring based upon local image content should improve the correspondence between the extremal regions and the semantically meaningful regions in the image. This statement is motivated by the human visual model of Cohen and Grossberg [14], which indicates that object perception is performed via an intensity diffusion process (i.e., blurring) which is moderated by edge strength measures in the image (i.e., it is nonstationary). Many nonstationary blurring schemes can satisfy the causality and smooth embedding constraints. There is no simple coordinate transformation between embedding produced by nonstationary blurring schemes and stationary blurring schemes. We have investigated several techniques, most of them based upon the premise that there should be minimal blurring across a region which is believed to contain an edge [11]. Region information can be supplied via a model tree (similar in format to that produced by the stack algorithm). Similarly, the stack algorithm can be applied iteratively, using its previous results (trees) to guide future iterations. No completely satisfactory method of using this information has been found yet.

#### V. USE OF THE STACK PROGRAM

##### A. Implementation Performance

The implementation of the stack and display algorithms has not been optimized for speed of execution or minimization of space requirements. The current version is an experimental one designed for flexibility and ease of modification and extension. Some indications of the speed and size of the implementation are nevertheless in order. The stack program has been applied to about 15 two-dimensional CT images of the upper abdomen. Approximately five one-dimensional and three-dimensional images have been analyzed also. To provide faster results for testing purposes, most of the CT images have been reduced from an initial size of 512 by 512 pixels to 64 by 64 pixels. Running on a moderately loaded VAX 780, it takes approximately 30 to 45 seconds to create each level in the stack, together with all of its associated data structures. The 64 by 64 images tend to need about 30 levels of blurring before only one extremum remains. Thus the program runs about 20 minutes. The image description tree created is approximately 250 kilobytes. All of the above numbers scale approximately linearly with image area. Of course, an image with a lot of noise or very many objects will tend to take longer and create a larger data structure. If it is known in advance that the structures of interest in an image are of small scale, the processing may be terminated before only one extremum remains, saving time.

It should be emphasized that all of the aforementioned processing is completely automatic; nobody's time is required.

Currently roughly half of the execution time is spent blurring. The blurring is so time consuming since it is written to handle several different nonstationary, nonisotropic methods. We believe that the stack program execution time can be decreased by at least a third to a half without the use of any special hardware if some of the experimental flexibility is removed.

### B. Interactive Display Based on the Image Description Tree

We have investigated several different interaction methods with the data structure produced by the stack algorithm. The display techniques developed permit easy specification of the region of interest to be examined and also yield insight into why certain subregions nest the way they do. The display program initially reads in the image description tree from a file produced by the stack algorithm. A user can then interactively control which regions in the image (subtrees in the data structure) are displayed. Several methods for specifying these regions are provided. These methods are discussed at length below.

The image is displayed on an Adage 3000 raster graphics system. A vector representation of the tree of extremum paths (along with the nonextremum paths they turn into) can be displayed simultaneously with the raster image. The vector image is displayed in three dimensions  $(x, y, t)$  on an Evans and Sutherland Picture System 300. This is a color vector graphics system. Paths of minima and maxima are displayed in different colors (see Fig. 10). This tree can be interactively rotated using knobs to specify the rotation about each axis. This display gives the user a very good feel for the way the extrema in the image have moved and merged together during the blurring process. This tree can be interrogated by picking any branch using a light pen. The extremal region associated with this branch will then be displayed on the raster display device. This has turned out to be a very powerful tool for visualizing the relationship between the tree structure produced and the original image. Frequently a major organ can be displayed quickly by using a light pen to choose the major tree branch (i.e., a branch that exists until very low resolution levels) in the location of interest. A visual examination of the tree structure sometimes focuses attention on high interest areas. For instance, noting that the extremum path representing the stomach region has a smaller branch (subregion) associated with it can focus attention on that subregion, which may be a tumorous mass.

The tree data structure can be examined more globally via various A/D devices. Two sliders are used to specify a scale range of objects which should be displayed. The two sliders specify the low and high scale limits of the range of interest. The scale of a region is defined to be the blurring level at which its associated extremum path annihilates. All extremum paths which annihilate within

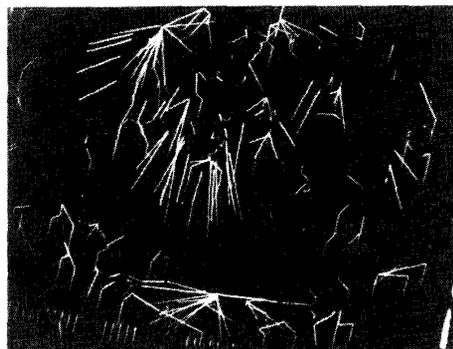


Fig. 10. Paths of maxima and minima. Roughly half of these paths are minima and half maxima. The actual display is in color; this clearly differentiates between the two types.

the scale range specified by the sliders have their associated extremal regions displayed on the raster display. This method of choosing regions has been very useful. Organs in the CT images can frequently all be chosen simultaneously simply by setting the sliders to display only regions of large scale. The length of time it takes to display the specified regions depends upon the number of regions specified. Typical display times range from about two to five seconds.

Similarly, two other sliders specify the intensity range of objects to be displayed. Even if the annihilation level of an extremum path lies within the specified scale range, it is only displayed if the average intensity of its associated region lies within the intensity window selected by the intensity sliders. Intensity windowing is useful when the regions of interest (or regions of disinterest) are most easily distinguished based upon their intensity. This would be used, for example, to select bright objects like the spinal column in the abdominal CT images or eliminate dark regions such as bowel gas. The display of extremal regions can also be constrained based upon the  $(x, y)$  position of the annihilation point of the associated extremum path. Four knobs are used to specify the range of spatial locations (maximum and minimum  $x$  and  $y$  coordinates) within which extremum paths must annihilate in order to be candidates for display. The chosen extremum paths can also be highlighted on the Picture System 300 display for better visualization. Intensity and spatial windowing have not been used extensively. It is not yet clear how useful they will be in a production (e.g., clinical) setting.

There is a difficulty in displaying some of the lowest resolution (largest scale) structures in an image. This is easiest to explain by use of an example. The spinal column and its associated musculature are oftentimes the largest object in an abdominal CT scan. In this case an intensity maximum in the spinal column will become the extremum path in the image which lasts the longest under blurring. If this extremum path is picked with the light pen and its associated extremal region is displayed, the entire image appears. This is so because all pixels in the image eventually link up to the last remaining extremum

path. There is no subtree which explicitly represents *just* the spinal column region. We have attempted to deal with this problem by providing more flexibility in the display mechanism. Entire regions associated with an extremum path do not have to be displayed. Subsections can be specified. Instead of following down all links from an extremum path, the user can specify that only links which join up before a certain blurring level be traversed. By picking this level low enough (high enough resolution), much of the image can be eliminated from the display. This often allows the isolation of the region of interest (e.g., the spinal column). The spinal region shown in Fig. 11 was specified in this manner (Fig. 16 is a schematic of organ positions in an abdominal CT scan). This method works since the true object of interest is usually spatially the closest to the extremum which forms the longest extremum path. As such, pixels in this region usually join the extremum path sooner than the other pixels in the image. Alternatively, if a resolution level is specified which is not as high as one used for the spinal region, the entire body in the CT image may be displayed without any of the surrounding image (e.g., the table the person is resting on).

Instead of displaying regions, the user has the option of displaying edges of regions superimposed over the original image. This is often useful since the interpretation of an isolated region displayed out of context can be difficult. Superimposing the edge of a region over the original image takes slightly more time than simply calculating the region itself. The time taken is highly variable depending upon the number of objects in the image, the number of objects specified to be displayed, and system load. The slower update rate does not significantly hamper interaction unless a large number of regions is specified.

Perhaps the most natural region specification method is via a cursor on the raster graphics console. By moving the puck on a data tablet, the user can position the cursor over a pixel in a region of interest (perhaps a pixel in the kidney). When a button on the puck is depressed, the smallest extremal region this pixel is in is displayed. Upon display, information about the region is shown on the user's terminal. This includes region size in pixels and average intensity of pixels in the region. The next larger extremal region is displayed when another button is pressed. Each successively larger region can be displayed until the root node of the entire tree is reached. This display method is the easiest to use, and will probably dovetail well with the future postprocessing techniques.

### C. Future Postprocessing Techniques

The difficulties with the algorithm as it stands now are:

- 1) A region of interest might not be precisely represented by an extremal region.
- 2) A region of interest does not always show up as one explicit subtree in the tree structure. It may be two subtrees with no common root except the last extremum.

An example of the first case might be an extremal region which includes the liver, but also includes part of the

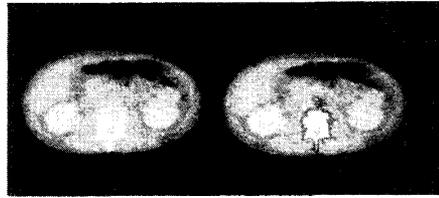


Fig. 11. *Left*: Original image. *Right*: Spinal region. A 128 by 128 pixel noisy image created by adding Gaussian white noise with a mean of 0 and a standard deviation of 30 to a CT image (which ranged in intensity from 0 to 1023).



Fig. 12. *Left*: Original image. *Right*: Segmented image. The liver and part of the chest wall are on the left side of the segmented image. The kidneys, intestine, pancreas, and a few blood vessels are also shown. A 128 by 128 pixel image.

chest wall near the liver (see Fig. 12). An example of the second type might be the kidney. Due to its shape, the two halves of the kidney may be represented as separate extremal regions. If the kidney is far enough from other organs, these two regions will join together (with one extremum path remaining) and be represented as one subtree. But proximity to the liver may cause each subtree to separately link to the liver extremum path instead of to each other (see Fig. 13). If this is the case, there is no single subtree which will display just the kidney.

One way of dealing with these problems is to modify the basic stack algorithm. Various blurring strategies based upon *a priori* and edge information seem promising. An interactive postprocessing step may be a simpler way to handle these difficulties. Various postprocessing capabilities might be provided. A few of the most promising ones are discussed below.

Difficulties presented by the first case listed above could be mitigated by use of a simple pixel editor. This editor would allow use of the cursor to delete or add pixels to a region displayed on the console. If the region displayed is accurate except for a few pixels, it would be a simple matter to delete or add those pixels to the display. If desired, it would also be simple to actually change the corresponding links in the data structure so that the data structure remains in accord with the display.

Problems arising from the second case (multiple disconnected subtrees) could be minimized by building a simple graphical editor for the tree data structure which is displayed on the Picture System 300. Using the light pen as a dragging device, the branch representing one of the kidney subtrees could be picked and dragged over to the other subtree and graphically joined to it. This graphical operation could then be used to guide a similar change

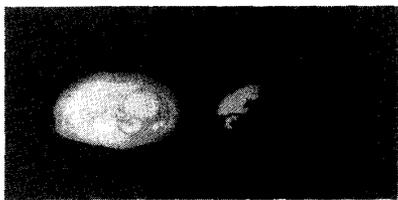


Fig. 13. *Left:* Original image. *Right:* The liver and a piece of the kidney. The other piece of the kidney links to the liver at a higher level and is not shown. A 64 by 64 pixel image.

in the actual data structure. Alternatively, disconnected subtrees can be dealt with by finding connected regions in the displayed image (as opposed to in the data structure). Even if the kidney is represented as two subregions which do not link to each other in any way, the capability currently exists to display both subregions simultaneously. This is done by picking each subregion separately while specifying that the previously picked region remain on the display. Once the entire kidney is displayed, it would be a simple matter to automatically find all the pixels which are in the connected region representing the kidney. As long as the pixels in the region of interest can be displayed on the console as a connected, isolated region, that region could be easily be defined to be one object and information about it calculated.

#### D. Results

The results obtained so far have been encouraging. Many correct image segmentations have been produced. Figs. 14 and 15 are two typical segmentation examples; the images on the left side of each photograph are the originals. CT scans are oriented so that the view presented is as if the viewer is standing at the foot of a table that the patient is lying on (face up), looking toward the patient's head. Thus the left side of the image is the right side of the person and the spinal column is towards the bottom of the image. High density objects (such as bone) show up as higher intensity (whiter) regions in the image. A schematic of the organ locations is shown in Fig. 16. Which organs are actually present in a particular CT slice, and their size, depends upon the axial position of the slice and whether disease is present.

Fig. 17 shows the result of applying a pyramid based segmentation algorithm [16] to the original image in Fig. 12, for comparison purposes. The segmentation was produced using a weighted linking scheme applied to a 128 by 128 version of the image and setting the desired number of segments to 16. For presentation purposes we have manually shaded the two segments which correspond to the major organs to make visualization easier; all the other segments are represented by their boundaries. Several points are worth noting in the pyramid segmentation. First, many different structures are in the same segment (e.g., the kidneys, spinal region, and part of the intestine are all part of the same segment). This happened despite the fact that 16 segments were specified, many more than



Fig. 14. The kidneys, liver, and some blood vessels (the aorta and inferior vena cava, running perpendicular to the image plane near the center of the image). The dark region near the kidney and the liver is a part of the intestine, as is the region near the left kidney (right side of image). The region extending out from the top of the left kidney is a large tumor, as is the very small region extending out from the bottom right part of the right kidney. A 128 by 128 pixel image.



Fig. 15. The kidneys, liver, aorta, and inferior vena cava. This image is the image in Fig. 14 with Gaussian white noise of mean 0 and standard deviation 30 added to it. The original image ranged in intensity from 0 to 1023.

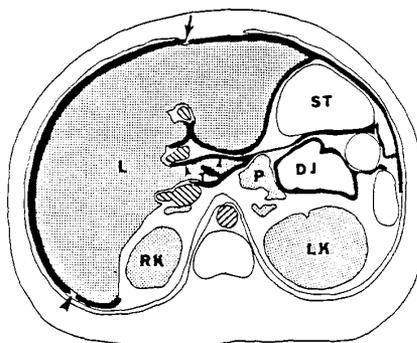


Fig. 16. An anatomical model. RK—right kidney. LK—left kidney. L—liver. ST—stomach. P—pancreas. DJ—duodenojejunal flexure (part of the intestine). The white object in lower center is the spinal column. The cross-hatched region above it is the aorta. This figure is from [15].

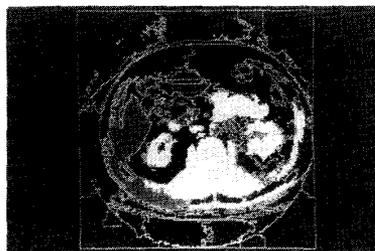


Fig. 17. The results of applying a pyramid based segmentation algorithm to the image in Fig. 12. Sixteen segments were specified. The intensities in the display were chosen to maximize contrast between the two segments of major interest, for easier visualization. Other segments are represented by their boundaries.

the number of anatomically distinct regions of interest. Second, several structures are part of two segments (e.g., the right kidney has sections in both of the shaded segments). Third, the pyramid algorithm has more trouble separating the liver from the chest wall than does the stack algorithm. It should also be emphasized that the pyramid approach either requires an *a priori* specification of the number of regions in an image or the setting of ad hoc and image dependent parameters which specify when a region is significantly different from another to prevent them from merging. The stack approach has a natural definition of a subtree based upon the annihilation of an extremum, there is no such natural definition in the pyramid algorithm.

## VI. CONCLUSIONS

Stack-based image segmentation correctly isolates anatomical structures in abdominal CT images. Both small vessels running perpendicular to the image plane and large organs are successfully identified in many instances. The inaccurate segmentations are frequently close enough to the desired result so that a simple interactive postprocessing step might produce a correct segmentation. Postprocessing is necessary because sometimes two nearby objects are represented as one extremal region, or a few pixels are missing from (or added to) a desired object. It remains to be seen whether a system with simple postprocessing abilities will be accurate enough, often enough, to be employed on a routine basis.

The main problem remaining is the incorporation of additional, probably imprecise, knowledge about the image into the stack segmentation algorithm. The knowledge might be edge strengths from the image itself, or perhaps model-based information. The challenge is to use this information in a manner which produces a more accurate segmentation without forcing the result to mimic the input knowledge.

## REFERENCES

- [1] P. J. Burt, T. Hong, and A. Rosenfeld, "Segmentation and estimation of image region properties through cooperative hierarchical computation," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-11, no. 12, pp. 802-809, 1981.
- [2] J. L. Crowley and A. C. Parker, "A representation for shape based on peaks and ridges in the difference of low-pass transform," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 2, pp. 156-169, 1984.
- [3] Y. Cheng and S. Lu, "Waveform correlation by tree matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, no. 3, pp. 299-305, 1985.
- [4] F. Bergholm, "Edge focusing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 6, pp. 726-741, 1987.
- [5] J. M. Beaulieu and M. Goldberg, "Hierarchy in picture segmentation: A stepwise optimization approach," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 2, pp. 150-163, 1989.
- [6] Y. Lu and R. C. Jain, "Behavior of edges in scale space," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 4, pp. 337-356, 1989.
- [7] S. L. Tanimoto, Ed., *IEEE Trans. Pattern Anal. Machine Intell.*, Special Section on Multiresolution Representation, vol. 11, no. 7, pp. 673-748, 1989.
- [8] J. J. Koenderink, "The structure of images," *Biol. Cybern.*, vol. 50, pp. 363-370, 1984.
- [9] S. M. Pizer, J. J. Koenderink, L. M. Lifshitz, L. Helminck, and A. D. J. Kaasjager, "An image description for object definition, based on extremal regions in the stack," in *Proc. 9th Conf. Information Processing in Medical Imaging*. Boston, MA: Martinus Nijhoff, 1986.
- [10] E. C. Zachmanoglou and D. W. Thoe, *Introduction of Partial Differential Equations with Applications*. Baltimore, MD: Williams and Wilkins, 1976.
- [11] L. M. Lifshitz, "Image segmentation via multiresolution extrema following," dissertation, Univ. North Carolina, Chapel Hill, 1987.
- [12] F. John, *Partial Differential Equations*. New York: Springer-Verlag, 1982.
- [13] L. Toet, J. J. Koenderink, C. N. deGraaf, and P. Zuidema, "The treatment of image boundaries in the case of progressive defocusing," Dep. Medical and Physiological Physics, State Univ. Utrecht, The Netherlands, Internal Rep., 1986.
- [14] M. A. Cohen and S. Grossberg, "Neural dynamics of brightness perception: Features, boundaries, diffusion, and resonance," *Perception and Psychophys.*, vol. 36, no. 5, pp. 428-456, 1984.
- [15] J. K. T. Lee, S. S. Sagel, and R. Stanley, *Computed Body Tomography*. New York: Raven, 1983.
- [16] T. H. Hong, K. A. Narayanan, S. Peleg, A. Rosenfeld, and T. Silberberg, "Image smoothing and segmentation by multiresolution pixel linking: Further experiments and extensions," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-12, no. 5, pp. 611-622, 1982.



**Lawrence M. Lifshitz** (S'82-M'87) received the B.A. degree in physics from Harvard University, Cambridge, MA, in 1980, and the M.S. and Ph.D. degrees in computer science from the University of North Carolina at Chapel Hill in 1983 and 1987, respectively.

Since then, he has been an Assistant Professor of Biomedical Imaging with joint appointments in Nuclear Medicine and Physiology at the University of Massachusetts Medical School in Worcester, MA. His research interests include image segmentation, pattern recognition, detection theory, and graphics.

Dr. Lifshitz is a member of the Association for Computing Machinery.

**Stephen M. Pizer** (M'84) received the A.B. and Sc.B. degrees in applied mathematics from Brown University, Providence, RI, in 1963 and the Ph.D. degree in computer science from Harvard University, Cambridge, MA, in 1967.

He is currently Professor of Computer Science and Adjunct Professor of Radiology at the University of North Carolina, Chapel Hill. For over 20 years, he has specialized in the display and processing of medical images, both two-dimensional and three-dimensional. He presently leads the UNC Medical Image Display Research Group, which consists of over 30 members of the UNC Departments of Computer Science and Radiology and which has as one of its focuses the interactive or automatic definition of objects in 2-D and 3-D medical images. Besides 19 years at UNC, for 11 years he was on the research staff at Massachusetts General Hospital, for a year he was Research Fellow in medical image processing at the University College Medical School, London, England, and in 1983-1984 was Visiting Professor doing research in visual perception at the State University of Utrecht, The Netherlands.

Dr. Pizer is a member of the Association for Computing Machinery, the Association of University Radiologists, and the American College of Radiology Computer Committee.