REPRESENTING SPACE FOR PRACTICAL REASONING

Margaret M. Fleck

Dept. of Engineering Science, Oxford University
Jenkin Bldg, 19 Parks Road, Oxford OX1 3PJ

ABSTRACT

This paper describes a new approach to representing space and time for practical reasoning, based on space-filling cells. Unlike R^n, the new models can represent a bounded region of space using only finitely many cells, so they can be manipulated directly. Unlike Z^n, they have useful notions of function continuity and region connectedness. The topology of space is allowed to depend on the situation being represented, accounting for sharp changes in function values and lack of connectedness across object boundaries.

Algorithms based on this model of space are neither purely region-based nor purely boundary-based, but a blend of the two. This new style of algorithm design is illustrated by a new program for finding edges in grey-scale images. Although the program is based on a relatively conventional second directional difference operator, it can detect fine texture in the presence of camera noise, produce connected boundaries around sharp corners, and return thin boundaries without "feathering." New algorithms are presented for combining directional differences, suppressing the effects of camera noise, reconstructing image intensities from the second difference values and merging results from different scales (including suppression of spurious boundaries in staircase patterns).

I INTRODUCTION

In their daily lives, people frequently reason about the shapes and arrangements of objects in space. This *practical reasoning* goes on at a variety of levels, from low-level visual processing, through identifying objects, up to reasoning about how an object could be manipulated. All these types of reasoning depend on representations of 2D and 3D space. Similarly, a representation for time is needed for reasoning about the relative ordering of events. There has been much discussion recently about the proper representations for time and space in fields including AI, computer vision and robotics, linguistics, and philosophy (van Benthem 1983, Dowty 1979, Allen and Hayes 1985, Hayes 1978a, Lee and Rosenfeld 1986).

A reasoner will need to construct for himself various models or descriptions of the world. It is useful to distinguish *symbolic descriptions*, such as "There is a desk against the wall," from *concrete models* of these descriptions, e.g. the sets of points of R^3 which comprise the desk and the wall. Concrete models can be inferred from sensory input, such as camera images, or produced by the reasoner "in his mind's eye" from symbolic descriptions. Symbolic descriptions can be derived from natural language input or from parsing concrete models.

Both symbolic descriptions and concrete models are useful in practical reasoning. Symbolic descriptions can consisely capture the relevant facts about a situation. This consiseness is important for remembering situations, identifying objects, describing situations which are too complicated to visualize all at once, and communicating in natural language. However, it is often simpler to use a concrete model for geometric reasoning tasks such as checking topological connectivity or measuring distances between features. Robot path planning can done using digitized representations of space (e.g. Brooks and Lozano-Pérez 1985). Finally, it is almost impossible to verify the consistency of a symbolic description except by exhibiting a concrete model which satisfies it. In this paper, I will concentrate on the form of the concrete models.

Standard models of time and space, such as R^n and Z^n, do not account for the way people do practical reasoning. Subsets of R^n must be manipulated symbolically because they typically contain infinite numbers of points. They cannot be directly stored by a reasoner. Secondly, it is difficult to represent two regions which are touching, because it is unclear which region contains the points along the common boundary of the two regions. The objects must overlap along the boundary, or the boundary points must belong to neither object, or else the boundary must be assigned arbitrarily to one of the objects (Allen 1984, Allen and Hayes 1985).

Models based on the integers avoid these problems with R^n, but at the cost of having no useful notion of function continuity or region connectedness. All functions from the integers are continuous and no subsets of the integers are connected. Most integer-based models handle only regular arrangements of points. A good concrete model for practical reasoning should use a finite density of samples, like integer-based models, but it should allow irregular arrangements of samples and it should provide notions of region connectedness and function smoothness like R^n.

I propose an alternative approach, in which space is represented by a set of space-filling cells, as shown in Figure 1. Objects or regions are represented by subsets of these cells. Boundaries, such as the edges of objects are placed between cells and represent *topological* separations in space. The domain of a function is typically the set of cells. The value of a function at a cell represents a property computed for some neighborhood of that cell. This neighborhood contains at least the area filled by the cell. In some cases, e.g. texture descriptors, it may describe a much wider area.

This "model" of space is really a family of models. First, a given region of space can be represented by different sets of cells, e.g. at different densities. Secondly, a given set of cells can be endowed with different boundaries. For example, if an object is added to a region of empty space, boundaries around that object are created, changing the topology of space. Finally, the size of the neighborhoods over which a function value is computed may be varied.

Such models of space lead to algorithms which do not fit the usual patterns of region-growing or boundary-based algorithms. The topology of a situation, represented by the boundary locations, is one of its most basic features. However, representation and processing focus on cells rather than on boundary locations, because cells represent regions of space and functions have values at cells. The resulting algorithms tend to spread out over wide regions of cells, in service of locating boundaries and describing their shape. The

The research described in this paper was done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology and at the Department of Engineering Science, Oxford University. The author is supported by the Fannie and John Hertz Foundation and by the Bell Laboratories Graduate Research Program for Women.

AVC 1987 doi:10.5244/C.1.38

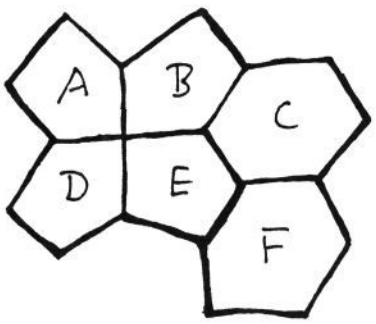

Figure 1. A sample set of space-filling cells. This set of cells has six adjacency sets of dimension 0: $\{A\}$, $\{B\}$, $\{C\}$, $\{D\}$, $\{E\}$, and $\{F\}$. The 1D adjacency sets are $\{A,B\}$, $\{A,D\}$, $\{B,E\}$, $\{D,E\}$, $\{B,C\}$, $\{C,F\}$, $\{C,E\}$, and $\{E,F\}$. The 2D adjacency sets are $\{A,B,D,E\}$, $\{B,C,E\}$, and $\{C,E,F\}$.

largest emphasis in processing falls on edge cells, i.e. cells next to boundaries.

The rest of this paper will describe the proposed models of space in more detail and then discuss an edge finder based on these models, called the Phantom Edge Finder. In this edge finder, the new view of regions and boundaries has been used to develop new methods for combining results from different directions and different scales and eliminating camera noise.

II CELL-BASED MODELS

I will model empty space using a set of space-filling cells, like those shown in Figure 1. These models are convenient idealizations for thinking about the relationship between physical space and sets of cells, e.g. those used in reasoning in a computer or a human brain. First, although physical space has extremely fine detail, we can only imagine limited amounts of it at once. Secondly, although brain or computer cells typically do not fill space, practical reasoning treats them as though they represented connected regions of space, i.e. as though they were space-filling. Finally, like cells in the human brain, these cells need not be arranged in a perfectly regular pattern.

I will, however, impose some conditions on the form of the cells. First, I require that the set of cells (plus their faces) form a regular cell complex (Munkres 1984). Secondly, I require that each cell X have a neighborhood homeomorphic to an N-ball, i.e. the space filled by the cells must be a manifold. Furthermore, I will require that this neighborhood contain all cells within a certain distance D (measured in cells) from X. In practical reasoning, a human or computer will only use a finite collection of cells and one might set D to be the diameter of this set of cells. Said another way, space may be curved, but we can only imagine flat portions of it. Thirdly, a given cell can only touch finitely many other cells and the intersection between two cells must be an M-ball, for some $M < N$. Finally, I will require that the cell sizes change only slowly as one moves around space.

Under these conditions, we can represent the essential features of a set of cells by specifying (1) how the cells touch one another, and (2) how large each cell is. Information about cell sizes is used to compare measurements from different parts of the cell structure. For example, cells in the human retina are much further apart towards the periphery of the visual field than they are in the center (fovea). Thus, if objects are to seem constant size as they move across the field of view, it must be possible to normalize for the changes in cell resolution. For local computations, however, this normalization will generally not be needed, because cell sizes change slowly.

Information about how cells touch one another is used to specify the topology of the set of cells. More specifically, the set of cells which touch at a point (such as A, B, C, and D in Figure 1), an edge (such as A and B), or a face will be called an *adjacency set*. In an N-dimensional situation, the *dimension* of an adjacency set is (N-M) if its cells touch along an M-dimensional face. For example, $\{A,B\}$ has dimension 1. For formal convenience, there is an adjacency set $\{X\}$ of dimension 0 for each cell X. A set of cells with adjacency sets and associated dimensions, will be called an *adjacency structure*. A pair of cells which belong to some common adjacency set (not necessarily one with two elements) are called *adjacent*. A *path from X to Y* or *connecting X and Y* is a finite ordered set of cells $X = W_0, ..., W_n = Y$, such that W_i and W_{i+1} are adjacent for every i $(0 \leq i < n)$. A set of cells $\mathcal{A}$ is *connected* if any two cells in $\mathcal{A}$ can be connected with a path.

The conditions on the sets of cells used in these models also constrains the form of their adjacency structures. In particular, if $\mathcal{X}$ is an adjacency set with dimension a and the adjacency set $\mathcal{Y}$ with dimension b is a proper subset of $\mathcal{X}$, then $b < a$. If there is no adjacency set $\mathcal{W}$ distinct from $\mathcal{X}$ and $\mathcal{Y}$ with $\mathcal{Y} \subseteq \mathcal{W} \subseteq \mathcal{X}$, then $b + 1 = a$. Dimension 0 is restricted to singleton adjacency sets and dimension 1 to sets of no more than two elements. A cell can belong to only finitely many adjacency sets and each adjacency set can contain only a finite number of cells.

Furthermore, it is possible to reconstruct the cell complex from its adjacency structure, up to homeomorphism. In order to do this, we first use the adjacency structure to construct the dual of the original cell complex, i.e. each cell of the adjacency structure corresponds to a vertex in the dual cell complex and each N-dimensional adjacency set specifies an N-cell. We can then reconstruct the original cell complex by taking the dual of the reconstructed dual complex. Thus the adjacency structure, unlike the pairwise adjacencies (Lee and Rosenfeld 1986), completely determines the topological structure of the space.

Space can obviously be represented by a number of different sets of cells, i.e. using cells of different sizes or different arrangements. I will say that one cell representation X is a *subdivision* of another cell representation Y if each cell of Y is the union of one or more cells of X. When handling adjacency structures or sets of receptor cells which do not actually fill space, it may be easier to use a definition in terms of adjacencies. That is, we partition the cells of X into a number of disjoint, non-empty, simply-connected, sets of cells $\{A_\alpha\}$, which will be the new cells of Y. To form the adjacency sets of Y, we replace each cell of X in some adjacency set with the set A_α to which it belongs. (If an adjacency set contains more than one element of a given A_α, the new adjacency set will have fewer elements than the old one.)

Changes in cell representation occur in three distinct forms in practical reasoning. First, when an object moves across the field of view, we can most easily analyze what is happening as a rigid motion of the cell representation onto itself. Similar cases occur for smooth motion of the eyes or camera. Secondly, if one sees an object or region of space on two separated occasions, it is easiest to relate the two representations via symbolic descriptions of the shape of the object. Finally, a reasoner may create a pyramid of coarser-scale versions of a cell representation in order to do reasoning quickly. In this case, the reasoner can ensure that the representation at one scale is a subdivision of the next coarser-scale representation.

III FUNCTION SMOOTHNESS

Many algorithms in practical reasoning, such as interpretation of motion sequences, surface reconstruction, and reasoning about the behavior of physical objects, depend on the assumption that functions are "smooth." The idea behind function smoothness is that the value of a property should not change "too fast" as one moves through space or time. For R^n, there are a number of mathematical definitions corresponding to this intuitive concept, including continuity, smoothness, and bounded derivatives.

The definition of function smoothness for adjacency spaces depends not only on the adjacency structure, but also on which cells *overlap*, i.e. sample overlapping patches of space. For example, in a CCD camera, the area sampled by a element overlaps areas sampled by elements which are several elements away, because of blurring and/or diffraction in the camera optics. Similar facts hold for the foveal area of the human visual system. This blurring before sampling reduces aliasing effects. In robot motion planning, it is essential that adjacent cells overlap, so that the entire area of space is covered by the cells and small objects cannot disappear from the representation. Texture descriptors must obviously be computed over regions many cells wide.

Following Poston (1971), I call the overlap relation on a set of cells the *fuzzy*. An adjacency space with a fuzzy is called a *fuzzy space*. This relation is symmetric and reflexive, but

not transitive. For a cell X in a fuzzy space, the *fuzzy neighborhood of X* is the set of cells overlapping X, including X itself. Each fuzzy neighborhood must be connected and have a finite number of cells. These definitions can be applied both to physical spaces and to abstract spaces, such as light intensities, temperatures, and distances. Two cells which overlap represent ranges of values which cannot reliably be distinguished. For example, I may have trouble distinguishing 10C and 15C, or 15C and 20C, but 10C and 20C are clearly different.

If $\mathcal{X}$ and $\mathcal{Y}$ are two fuzzy spaces, such as the visual field and grey-scale intensities, a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ is *smooth* if f(A) overlaps f(B) in $\mathcal{Y}$ whenever A and B overlap in $\mathcal{X}$. That is, in a region of smooth change, two overlapping cells in the visual field must have indistinguishable (overlapping) intensity values. If, for some overlapping A and B in X, f(A) and f(B) do not overlap, f has an *abrupt change in value* between A and B. This notion of smoothness depends on the fuzzies for the two spaces: if we extend the fuzzy on X so that more cells overlap, fewer functions are smooth.

Unfortunately, practical reasoning seems to distinguish two types of functions and the above analysis of function smoothness only works for one type. The first type of function is a mapping between two representations of physical space or time. Such mappings might be used in comparing cell representations of two regions, in analyzing symmetries of regions or textures, or comparing the temporal structure of two events. For these functions, the above definition of smoothness is adequate.

The second type of function maps from physical space or time into a space of values such as intensities, temperature, texture pattern, or type of material. These are functions for which many practical reasoning applications explicitly reason about first differences. For example, Forbus (1984) reasons about processes of change across time. Many others have tried to deduce surface shape from the slope of grey-scale intensity or texture descriptors. We can define the first difference DT of a function T along a one-dimensional set of cells $\{X_i\}$ by $DT(X_i) = T(X_{i-1}) - T(X_{i+1})$. If the domain is 2D, we define directional first differences at a cell by taking first differences along all straight paths through that cell.

A second property of these functions is that abrupt changes in the raw values are a poor indicator of a physical boundary. For example, smoothly shaded objects can have changes in intensity which are larger than those across low-amplitude physical boundaries. High curvature of a boundary can be caused by foreshortening. A better indicator of a physical boundary is an abrupt change in the first difference of the values. Thus, work on edge finding has looked for well-defined peaks in the first differences or zero-crossings of the second differences. Similar stories hold for finding sharp corners in curves (Asada and Brady 1984).

IV TOPOLOGICAL BOUNDARIES

Function smoothness and region connectedness are very important in practical reasoning. However, at a limited set of *natural boundaries* in a situation, such as at the edges of objects, functions can change abruptly and adjacent objects may not be, intuitively, connected. These locations of abrupt change are exactly the most interesting parts of the situation for practical reasoning. For example, computer vision programs extract locations of abrupt changes in intensity and the reason using only descriptions of these boundaries. In analyzing processes such as heating liquids, properties and rates of change of properties can change abruptly when processes stop or when a substance undergoes a phase transition. These locations of abrupt change, along with summaries of behavior within regions of smooth change, can be used to predict the behavior of these systems (Forbus 1984).

Not only can functions have abrupt changes in value at natural boundaries, but the objects to either side of the boundary are not perceived as connected to one another. For example, it is necessary to distinguish whether two adjacent metal bars are physically connected in order to determine whether one bar will move if one pulls on the other. Pieces of metal do not merge on contact, although other substances (e.g. water) do. Connectedness can also be used to "limit causality" (Hayes 1978b). For example, if one can surround the situation of interest with boundaries across with nothing of interest is likely to flow, then reasoning can be limited to the region thus surrounded. Similarly, an event can only cause another event if the two are connected by a sequence of events.

Boundaries can be characterized by which types of functions change at them. For example, changes in lighting are important for reading but not for motion planning. Two pieces of metal can be physically but not electrically connected. However, in a particular situation, *sharp changes in different functions all tend to occur at the same limited set of locations.* In other words, the world, at any fixed scale of resolution, exhibits natural boundaries which are separated by large regions with no sharp changes. Connectivity boundaries and locations of abrupt changes in function values tend to coincide. This suggests that these natural boundaries are *topological* boundaries in situations. The usual explanation, that each function is discontinuous at a small number of places, fails to account for the clustering of boundaries and for the connectivity facts.

We can add boundaries to a cell-based description of space by altering the adjacency structure of the cells. Specifically, a *boundary* is a set of adjacency sets, called the *boundary adjacency sets*. The non-boundary adjacency sets must be a well-formed adjacency structure, i.e. if $\mathcal{X}$ is a non-boundary adjacency set, no subset of $\mathcal{X}$ can be in the boundary either. A cell which is in a boundary adjacency set is called an *edge* cell. If some set of cells $\mathcal{A}$ is selected as a region, its boundaries are all boundary adjacency sets containing cells in $\mathcal{A}$, its edges are all edge cells in $\mathcal{A}$, and its borders are all cells not in $\mathcal{A}$ but in one of the boundaries of $\mathcal{A}$.

If we are given a regular cell complex representing the dual of a set of space-filling cells, and a set of boundary adjacency sets, we can construct a regular cell complex modelling space with these boundaries in it. This is done by "cutting space apart" between cells involved in boundary adjacency sets, creating duplicate copies of points where required, as shown in Figure 2. Specifically, each boundary adjacency set corresponds to some cell of the dual regular cell complex. For each point x in these dual cells, we create a distinct copy of x for each non-dual cell to which it belongs. This new space can be re-expressed as a regular cell complex each of whose cells is the intersection of one of the original cells and one of the dual cells.

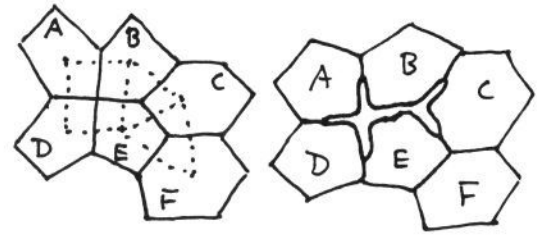

Figure 2. Left: the cells in Figure 1, with dual cells marked in. Right: the new model of space if the dual cells $\{B, E\}$, $\{A, B, D, E\}$, and $\{B, C, E\}$ are marked as boundaries.

This construction has the property that even very thin or small subsets of space, e.g. a ribbon one-cell wide, retain the dimension of the original space. Thus, the dimension of an object is not altered by changes in the number of cells used to represent it. Another subtle point is that distances between cells should be calculated on the original cell structure, even when boundaries are present. For example, the distance between two objects does not change when another object is put between them. I define a distinct concept *minimum path length* for the distance between two cells along the shortest connected path in the new space, with boundaries. Standard mathematical practice would be to use only the minimum path length, as it reflects the topology of the new space. However, practical reasoning about distance seems to involve both concepts.

Often, it is easiest to determine pairwise boundaries between cells, e.g. because there are sharp changes in properties between two overlapping cells. These pairwise boundaries are not boundaries in the above sense, but they can be used to construct boundary adjacency sets. That is, any adjacency set containing two cells with a pairwise boundary between them is marked as part of the boundary. These boundary adjacency sets may include sets containing more than two cells. So, for example, in Figure 1, if a boundary is placed between cells A and B, the adjacency set $\{A, B, C, D\}$ must also belong to the boundary. This method of specifying boundaries avoids the paradoxes described by Pavlidis (1977) and Lee and Rosenfeld (1986) involving cells touching only at a point.

In doing practical reasoning, a reasoner must choose which boundaries are active during a particular piece of topological reasoning. Certain boundaries may not be relevant to the task at hand. It may be necessary to consider several different models, e.g. in deciding which of several electrical connections is broken. There may be more than one way to parse a situation. For example, the region occupied by a marble inside a cup can be seen as overlapping the interior of the cup, or as disjoint from the free space inside the cup. In other words, *the topology of space is an integral part of the representation of a scene which the reasoner can manipulate dynamically.*

V REGIONS, BOUNDARIES, AND ALGORITHMS

The above analysis of boundaries in cell-based representations does not fit neatly into the usual division between region-based and edge-based representations. Regions (sets of cells) represent division of space into subsets, whereas boundaries represent the topology of space. In visual processing, one would first identify boundaries in a scene, using an edge finder. One can then use these boundaries to divide the scene up into regions. For example, a local symmetry shape analysis (Brady and Asada 1984, Fleck 1985, 1986, Connell 1985) picks out sets of edge cells as the borders of elongated or round regions. These regions are typically connected relative to the boundaries which define them (Hayes 1978a).

Although regions are built from boundaries, regions and boundaries are somewhat independent of one another. A region can contain internal boundaries, including ones which end abruptly in the middle of the region. For example, if you fold a bedsheet, there are still boundaries where it touches itself, and there is a boundary between two adjacent fingers on the same hand, even if they are pressed together. Conversely, two regions need not be separated by a boundary. For example, people identify hands and arms as separate entities, although a hand is smoothly connected to an arm. The walls of a tunnel define a region, although the ends are not closed. In particular, the two halves of the representation are not dual to one another, as is often assumed (cf. Blake 1983).

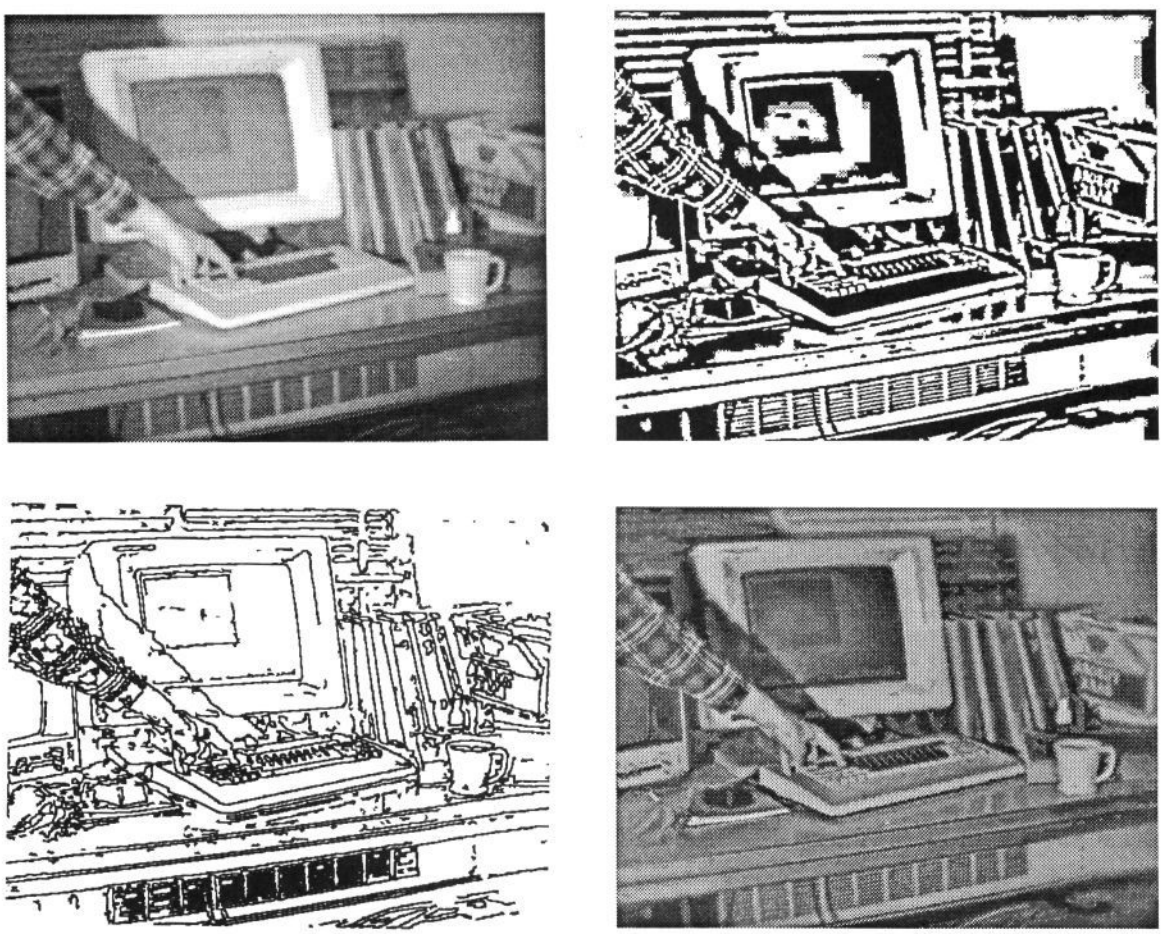

Figure 3. Output of the Phantom Edge Finder on an image of a Symbolics 3600 console. Top left: original image. Top right: cartoon, in which black cells are on the dark side of an edge, white cells on the light side, and grey cells both or neither. Bottom left: edge map. Bottom right: reconstructed intensities.

Pairs of regions can be related using analogues of James Allen's (1983,1984) interval primitives. For example, a region A *touches* a region B if A and B are disjoint and there are two adjacent cells a and b such that a is an edge of A and a border of B, and b is an edge of B and a border of A. This relation corresponds to Allen's *meets*. Analogues of his other primitives can be defined similarly, using also an order for 1D spaces. These definitions highlight the fact that the most important parts of the representation are often neither the boundaries nor the regions, but the edge cells.

The types of algorithms used for reasoning about cell-based representations also do not neatly fit into the usual typology of vision algorithms. Often, researchers restrict themselves to purely boundary-based or purely region-based representations and algorithms. Region-growing algorithms have problems with internal boundaries and/or slow changes. Processing only at boundary locations tends to leave processors idle in straightforward parallel implementations. Locations near one another may be far away in terms of boundary connectivity and it may be difficult to re-associate them. In a grey-scale image, useful information about a boundary is blurred over a strip several cells wide and it is not necessarily easy to condense it into facts at the boundary locations.

Figures 3 and 4 show results of a new edge finding algorithm, named the Phantom Edge Finder because of its resemblance to Watt and Morgan's (198X,1984) MIRAGE algorithm for one-dimensional boundaries. Its design also owes considerably to Pearson and Robinson's (1985) algorithm for detecting second difference peaks and producing cartoon representations of faces. The Phantom Edge Finder uses cell-based algorithms to find boundaries between cells. In these algorithms, computation is spread out over all cells with significant second difference responses, rather than just edge cells or boundary locations. However, these algorithms do not fit the pattern of the usual region growing algorithms, e.g. it can detect an internal boundary which ends abruptly in the middle of a region and does not mark boundaries in areas of slow change.

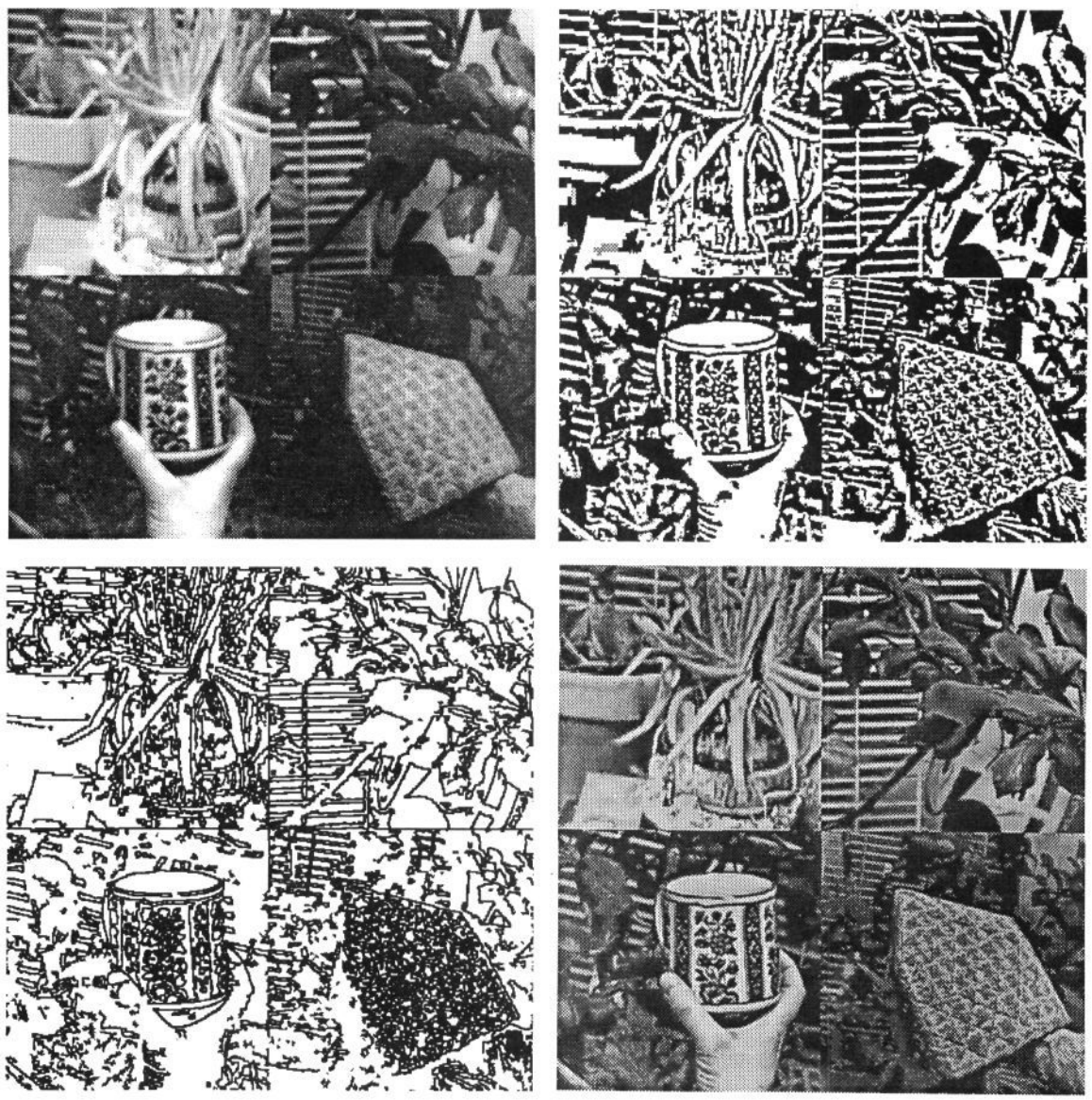

Figure 4. Output of the Phantom Edge Finder on an image of four textured patterns.

The input to the Phantom Edge Finder consists of grey-scale intensity values for an array of cells, currently supplied by a CCD camera and digitizer. The original scene shown to the camera has been first blurred by the camera optics and then sampled by the digitizer. Noticable additive noise is introduced during sampling. The edge finder's task is to identify and localize sharp changes in the original scene, eliminating the noise and blurring. The cell-based approach to algorithm design has lead to several new edge finder algorithms. First, the new algorithm successfully removes camera noise while preserving fine texture and sharp corners. Secondly, it can combine responses from different directions without feathering or multiple responses. Similarly, results from different scales are combined into one edge map and extraneous edges in staircase patterns are removed during this scale combination.

The edge finder algorithm is described in the following sections. I hope in the future to design algorithms in the same spirit for higher-level image analysis and reasoning. I see these algorithms as following a pattern similar to that of the edge finder: operating primarily on boundaries, but doing processing throughout wide bands of cells. For example, a texture analysis program might compute patterns of dark and light cell labels, rather than just patterns of edges. A motion planning program might use a propagation algorithm to compute the minimum distance from each cell to a obstacle boundary. It should also be possible to modify shape analysis algorithms to use the cells in a region, rather than just boundary locations.

VI COMPUTING THE EDGE RESPONSES

The Phantom Edge Finder detects boundaries using second differences along straight paths through the image. There are three reasons behind this decision. First, the sign of a first difference depends on the direction of motion along the 1D path and the second difference does not. Secondly, the second difference gives a high response near boundaries, whereas a first difference responds at the boundary itself. Since processing is done at cell locations and boundaries are located between cells, this is the most useful type of response. Finally, roof edges show up as peaks and valleys in the second differences and do not show up explicitly in the first differences.

For the rectangular arrays used by the current program, second differences are taken in four directions: horizontal, vertical, and two diagonal directions. The exact operator used is $[1, 0, -2, 0, 1]$. Because noise is suppressed later in the program, larger smooth operators are not needed and a minimal-sized operator can be used to preserve fine detail. One might use the yet smaller operator $[1, -2, 1]$, but this tends to pick up differences between the two interlaced pictures from our camera. When one of the cells involved in a difference would lie outside the image, it is given a value identical to the nearest cell on the boundary of the image.

A problem with the directional second difference values is that when the middle cell is in a thin bar, the response reflects two edges, rather than one. As a result, a thin bar typically produces twice as high a response as a step edge with the same intensity change. This gives the program an unfair impression of the intensity change for a thin bar or a sharp corner and causes problems in reconstructing image intensities from the second difference responses. To avoid this, the second difference response $a - 2b + c$ is normalized by $\frac{|max(a-b,c-b)|}{|(a-b)+(c-b)|}$. This function was chosen because it is relatively simple and has the right qualitative behavior. That is, it varies between 1 and 0.5; it is 1 if b is equal to one of the other two values; and it is 0.5 if a and b are equally different from b.

Another challenge in using these second difference responses is to combine the responses from different directions so as to produce thin boundaries. Extracting zero-crossings from each directional difference and putting them all into the boundary map produces "feathering," as shown in Figure 5. Feathering occurs because high responses are not limited to the difference perpendicular to the boundary, but can also come from differences at other angles to it. The zero-crossings of these responses need not lie along the same thin curve. To judge from what I have seen of output from previous directional difference algorithms, it seems to be difficult to eliminate these "feathers."

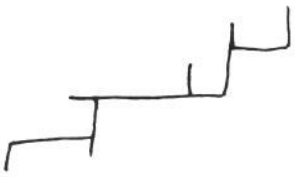

Figure 5. When zero-crossings are taken individually for differences in different directions, a boundary may have "feathering," i.e. short extra boundary markings off to each side.

Phantom gets around the problem of feathering by combining responses from different directions *before* extracting zero crossings. Specifically, it tries to classify each cell with a significant second difference response as either on the light side ("light") or on the dark side ("dark") of a boundary. Cells without significant responses are left unlabelled. It is, however, possible for a cell to be on the light side of one boundary and also on the dark side of another. This happens when the intensity surface has a saddle. Intensity saddles can be due to saddles in 3D surfaces, e.g. on a human face, or due to places where three or more regions meet at a point, which form saddles when smoothed. Such cells are labelled as both light and dark.

Classification of cells as dark or light is done by finding the maximum amplitude positive and negative responses over all directional differences. For an isolated step edge, the maximum amplitude response will reflect the difference in the direction closest to perpendicular to the boundary. If more than one boundary is involved, the maximum amplitude response(s) reflect differences perpendicular to the boundaries with the largest intensity changes. Since a cell can be on both the light and dark sides of different boundaries, separate positive and negative responses are computed.

This method of combining directional responses has been designed to give good performance on sharp corners and places where several regions meet at a point. For example, Figure 6 shows Phantom's performance on an image of a fork. The sum of the directional responses, other non-directional center-surround operators (Hildreth 1983 and Marr and Hildreth 1980), and the sum of the responses of the correct sign, are not a good indicators of the strengths of the boundaries involved in these cases. They tend to give overly high values to the insides of sharp corners and overly low values to the outsides. In theory (Berzins 1984) the zero-crossing of the Laplacian of a Gaussian is closed around a sharp corner but balloons out. In practice, the weak outside response of this type of operator causes the zero-crossing to merge into the background noise in some random manner. Phantom's method of combination also does not require that the directional responses fit any particular pattern, e.g. have a unique maximal response, peak responses in some directions, or responses that can be modelled as a linear transformation. Previous work has tended to depend on such assumptions (e.g. Canny 1983, Haralick, Watson, and Laffey 1983), although they break down at sharp corners and boundary intersections. Canny's edge finder, in fact, tends to leave small gaps at such points.

The maximum amplitude positive and negative responses are then used to classify each cell as dark or light. Presence of a non-noise response of one type is not sufficient grounds for assigning that label to the cell, since cells near the zero-crossing in a step edge may have both negative and positive responses. In such cases, one response is significantly larger than the other, unless the cell is centered on the boundary. Only when the two responses are of similar strength does the program assign both labels. When one response is more than 1.5 times the other response, Phantom gives the cell only a label reflecting the larger response. Cells with no significant response of either sign are left unlabelled.

VII EXTRACTING BOUNDARIES

From the dark/light classification of cells, the algorithm then extracts pairwise boundaries. A boundary is considered to exist between a pair of cells when both of them bear

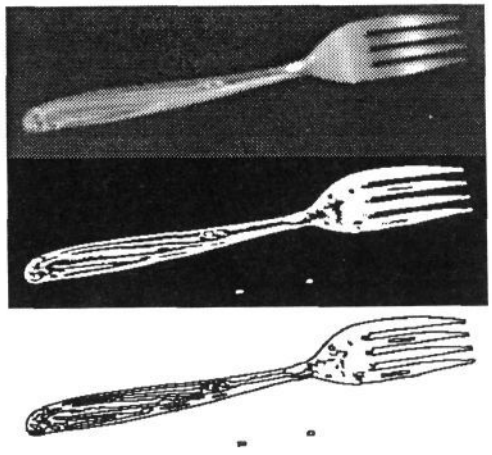

Figure 6. An image of a fork with sharply pointed tines, cartoon output, and edge map.

dark/light labels, but not the same ones. That is, boundaries occur between a dark cell and a light cell, and also between a dark or light cell and a cell bearing both labels. The boundaries detected in this way represent step edges. Thin bars, even those as small as one cell wide, show up as a pair of step edges. Because cells can be given no label or both labels, the boundaries can end abruptly when appropriate.

Roof edges show up as regions of dark or light response which are not near dark/light transitions. The current version of Phantom uses a very simple algorithm to find roof responses. First, cells next to boundaries are labelled "edge." Then, dark and light cells connected to edge cells, up to a radius of 8 cells, are labelled as "back." Any cells that have not been labelled are classified as "roof" responses. This is a simple 2D version of Watt and Morgan's (198X) criterion that a roof edge (they consider these a type of "bar") is a thin region of response with no response to either side of it. This algorithm does, in fact, pick up certain roof edges.

Unfortunately, this simple roof edge detector will only work for roof edges which are not near any step edges. If a roof edge occurs near a label transition, they form a combined response region which has a label transition to one side but where the response region is too wide and has peaks in the wrong places for a step edge. This is a resolution problem: if the image were represented at a finer scale, the two responses would be separated. In order to properly label the responses at the resolution given, it would be necessary to consider the shape of merged response region, e.g. detecting the locations of peak responses.

There is another problem with roof edges near step edges. Since the propagation used to label cells as the "back" of an edge reponse is non-directional, even roof edges which continue the line of a step edge are suppressed. Ideally, the "back" labelling should only be propagated perpendicular to the boundary. Since the propagation is moving out a substantial radius (8 cells), this must use the orientation of the boundary over several cell long stretches, rather than fine-scale orientations (e.g. the direction of the maximal directional difference response at each cell). Even worse problems occur if a roof edge intersects a step edge.

VIII ELIMINATING NOISE

To interpret edge finder output, it is necessary to determine which of its responses represent real features of the visual field and which are caused by camera noise. The amount of noise is illustrated in Figure 7, which shows the Phantom Edge Finder's output with the noise suppression turned off. Most edge finders used in computer vision eliminate camera noise by smoothing the image or by applying an operator with a large region of support. The result is that these edge finders cannot detect fine texture, like the pattern in Figure 8. But features as thin as one pixel wide are clearly visible in images and the psychophysical evidence (Marr, Poggio, and Hildreth 1980) seems to indicate that human visual processing also

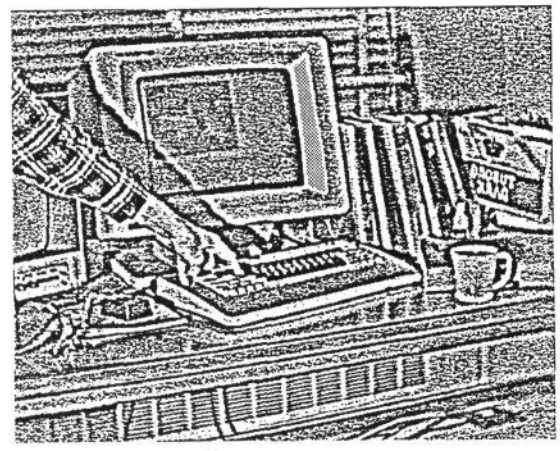

Figure 7. Output of the Phantom Edge finder with noise suppression turned off. The amount of high-frequency camera noise can be considerable, even for images which do not look noisy.

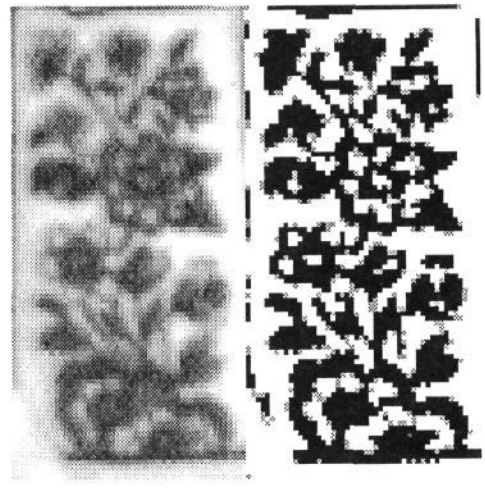

Figure 8. Enlargement of the pattern from the cup in Figure 3 and cartoon output.

operates at this fine a resolution. We must, therefore, find a way to suppress noise other than by smoothing.

In most vision applications, the same camera and digitizer setup is used for taking many pictures. Similarly, a human has only one fixed pair of eyes. Therefore, the noise suppression algorithm can be designed and tuned for the noise characteristics of a particular system. My algorithm is designed for a type of noise which has occurred in several setups that I have used, the most recent being a Panasonic WV-CD50 camera and a Datacube framegrabber. To a very rough approximation, the noise in these systems is not spatially correlated, vaguely Gaussian-like, and of zero-mean, and most cell values are displaced no more than 3 intensity units (of 256) from their correct values. Obviously, some adjustment may be needed if the noise differs substantially from this.

I also want to make a distinction between image features caused by camera noise and image features representing real texture or clutter in the scene. I believe that the edge finder should remove only effects of camera noise and it should report the location and properties of all scene features that it can distinguish from noise. Although some higher-level processing, e.g. shape analysis, may wish to suppress regions which are small or have low contrast, these regions are needed for other tasks, such as texture analysis.

The noise suppression technique used in Phantom depends on the fact that the camera noise is introduced *after optical blurring*. The blurring in our camera system is (very roughly) like that caused by a Gaussian of about $\sigma = 1$ *cell*. Thus, the blurring will cause the second difference response to a real boundary (in isolation) to be several cells wide on each side of the boundary, whereas noise responses are typically thin. I have not seen this observation used before in the computer vision literature. It is also characteristic of a noise edge that its responses are low amplitude and that the edge response does not continue for very long in a straight line. These two observations are widely used to filter out noise responses from edge operators, with and without smoothing. None of these three criteria separately is enough to identify noise responses. For example, there are real edges with very low amplitude responses, real edges with narrow responses (e.g. in a series of thin stripes), and real edges with short or sharply-curved responses (in texture; at sharp corners). However, a response which exhibits all three properties is almost certainly indistinguishable from the ambient noise.

These three criteria can be combined into the following single criterion:

> A cell response due to a real edge has a star-convex neighborhood of responses of the same sign, where the sum of the responses over that neighborhood is high.

This criterion is, in fact, slightly stronger than the combination of the three previous ones, because of the additional constraint that the neighborhood be star-convex. A neighborhood of a cell X is "star-convex" if there is a straight path from each cell in the neighborhood back to X which passes only through cells in the neighborhood. I use this notion rather than simple convexity because it is easier to implement. This criterion is similar to Watt and Morgan's (198X,1984) mass threshold for boundaries in 1D signals.

Noise suppression is done at two places in the current implementation of Phantom. It is done once on the directional difference responses for each of the four directions individually. It is then repeated after cells are classified as dark or light, using only maximum amplitude responses whose signs match the cell labels. This eliminates dark regions most of whose cells have been classified as light, and vice versa. The star-convex neighborhoods are computed within a 7x7 window centered on the cell of interest. If the sum of the detector magnitudes in a cell's neighborhood is less than a threshold, the cell response is considered to be noise and is set to zero. This threshold should be set to reflect the ambient level of noise in the camera setup in the absence of features.

One problem with this type of noise suppression is that it is sensitive to the classification of responses as positive or negative. I therefore include in the star-convex neighborhood any cell whose response is not more than T units from zero in the wrong direction, where T reflects noise amplitude. Cell values of the wrong sign do not contribute to (or subtract from) the sum over the neighborhood, but they allow a neighborhood to be built across cells which noise has caused to have slightly the wrong sign. This procedure ought to cause a cell straddling a boundary, which may have no response in either channel, to be given both light and dark labels, so that there is no gap between the light and dark responses for the boundary. This does not seem to be working and I do not know why. The current program explicitly fills in such one-cell gaps between response regions.

A side point which should be mentioned is in regard to smooth operators. Much effort has been devoted to finding smooth operators of optimal shape. In a sense, Phantom does not use any smoothing, but in another sense it does. Versions of the image at coarser scales are produced by smoothing and sampling the image. Thus, one might apply results about optimal smoothed operators to the problem of building the coarse-scale images. However, much of the interest in this problem revolves around detection and localization at the finest scale. A point which is frequently missed, however, is that we have *no control over smoothing at the finest scale.* The finest scale has already been blurred by the optics. As I have shown above, it needs no further smoothing. Thus, the designer of computer vision algorithms can only exercise control over the exact form of the blurring function at the finest scale by re-designing the camera.

IX MULTI-SCALE COMPUTATION

The multi-scale version of the Phantom Edge Finder is a relatively simple extension of the single-scale version presented above. Coarser-scale versions of an image are created by smoothing and sampling the original image, to form a pyramid structure. To form the image at a given scale, the image at the next finer scale is smoothed with a Gaussian of $\sigma = 2$ *cells*. Groups of four cells are then merged into one cell at the coarser scale, whose value is the average of their values. This produces a version of the image which is $\frac{1}{4}$ the area of the original.

Computation of second difference responses and cell labels is done independently at each scale. For each scale, the following outputs are produced:

- dark and light labels,
- edge, roof, and back labels, and
- net intensities.

The labelling algorithms have been explained above. The net intensity computed for a cell, at a given scale, is the signed sum of the maximum amplitude positive and negative difference responses for that cell, but only using the ones appropriate to the cell's dark/light classification. That is, responses of the wrong sign for the cell's classification are treated as if they were zero.

Responses from adjacent scales are then combined, moving from coarser scales to finer scales. That is, each scale is combined with the combination of all scales coarser than it. The final results are a unified fine-scale map of labels and a reconstruction of image intensities. The first step in combining the responses of two scales is to interpolate the coarse-scale responses: each of the four fine-scale cells corresponding to one coarse-scale cell is given its labels and intensity. The intensities are then smoothed with a Gaussian of $\sigma = 2$ *cells*.

In combining the coarse-scale and fine-scale symbolic labels, the fine-scale takes absolute precedence. When the fine scale has assigned symbolic labels to a cell, the cell is given only the fine-scale labels in the combined map. When no fine-scale label was assigned, the cell is given the coarse-scale labels. The result of this is that coarse scales can only add boundaries in areas where there were no previous fine-scale boundaries. When there was a fine-scale boundary, the coarse-scale responses widen the response regions, without changing the boundary location. Finally, the final dark/light labelling of the image provides an extremely vivid "cartoon" version of the image when the labelling is displayed as black and white, with small amounts of grey (= both dark and light or neither). As shown in Figure 9, it seems to provide, much more robustly, the same effect as Pearson and Robinson's (1985) combination of a valley detector and level thresholding.

In order for this label combination process to work correctly, it is essential that the entire wide response regions be used, rather than just locations of edge cells. As the image is smoothed and sampled, the location of a boundary can shift. If one looked at just the boundary locations, it would be necessary to match fine-scale boundaries to coarse-scale ones. However, the edge finder response regions for the two edges overlap substantially. As long as amount of the shift is less than the width of the response regions at the fine scale–which seems to be true for my sampling rate–only one boundary will be marked in the combined map.

Combining the intensity responses is slightly more interesting. The goal is to produce a smooth reconstruction of image intensities from the second difference responses, but with boundaries sharpened (de-blurred), as shown in Figures 3 and 4. Combination of the responses from two scales depends on the dark/light labelling at these scales. There are three cases:

- If the cell is not labelled at the fine scale, the coarse-scale intensity is used.
- If the coarse-scale and fine-scale dark/light labels are the same, then the intensity with larger amplitude is used. (Amplitude is signed!)
- Otherwise, the average of the two intensities is used.

In the average used in case (3) the values from the two scales are weighted equally. Since the coarse-scale intensity is really the combination of all the coarser scales, this means that the contribution from an individual scale counts less and less as the scale gets coarser. In other words, higher frequencies have been emphasized. This is a well-known technique for sharpening boundaries, though Phantom's fine-scale responses are not precisely the usual high-frequencies of a Fourier transform. Because the boundaries have been de-blurred, one can

extract the amount of contrast across an edge from the reconstructed intensities. The reconstructed intensities also contain information about smooth shading which is not reflected in the edge maps.

In their MIRAGE algorithm, Watt and Morgan (198X, 1984) propose a different method of scale combination. They suggest summing separately the positive and negative detector responses across scales and then looking for regions of zero response in either the positive or the negative summed response. Unfortunately, as far as I can tell, this method does not preserve fine-scale detail or fine-scale boundary locations. Nor will it work on a series of nested dark and light regions, e.g. a person's eyes, which are typically dark spots in light regions, surrounded by a dark shadowed area. The more structured combination technique in Phantom shares some properties with MIRAGE, e.g. fine scale texture can mask coarse-scale boundaries. However, if preserves fine detail and, as we will see in the next section, allows one to eliminate many "staircase phantom" edges.

X STAIRCASE PHANTOMS

There is a problem with the dark/light labelling algorithm presented above. Consider a grey region between a light region and a dark region, as in the righthand part of the terminal screen in Figure 3. As labels are added at coarser scales, the light and dark responses for the two sides of the grey region grow together, forming a "staircase phantom" label transition in the center of the grey region. Obviously, we do not want to consider such a label transition to be a region boundary! Similar spurious label transitions occur in trihedral vertices (Gennert 1986).

Phantom eliminates most staircase phantom boundaries during the process of combining labels from two adjacent scales. Specifically, the program propagates a marker out an 8-cell radius from all fine-scale labels. This propagation occurs only within cells with dark/light labels, but does not distinguish between dark and light cells. Coarse-scale edge or roof markings are removed from marked cells. This eliminates staircase phantom boundaries from appearing as the two scales are merged and also keeps fine-scale roof edge responses from being widened by the coarse-scale roof responses. If you compare the cartoon in Figure 3 with its edge map, you can see that a number of staircase phantom boundaries have been suppressed.

This algorithm has two problems, analogous to the problems with the roof edge detector. First, if the spurious label transition already exists at the finest scale, it cannot remove it. Since significant second difference responses to a boundary can be as wide as 6 cells on either side of it, this means that the grey region must be over about 12 cells wide. Figure 10 shows an enlargement of part of Figure 1, showing that the spurious boundary where three regions touch is only eliminated after the region gets wide enough. Secondly, the propagation of suppression marking should be limited to a direction perpendicular to the fine-scale boundary. Otherwise, coarse-scale responses which continue the line of a fine-scale boundary (e.g. the boundary gets more blurry) will be suppressed. I haven't yet found a simple, robust way to do this. Worse, although the orientation of a zero-crossing boundary, over a several cell wide radius, can be determined straightforwardly, it is less clear how to extract the direction of a wide roof edge response region.

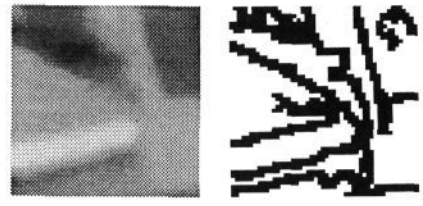

Figure 10. Staircase phantom boundaries can only be removed from a pointed region after it gets sufficiently wide.

Anther approach to eliminating staircase phantom edges is to check the first difference of the intensity along the putative boundary and make sure that its sign is consistent with the sign of the putative edge determined from the cell labels, as suggested by Clark (unpubl.) One might also compare the magnitudes of the two responses. Thus far, I have not gotten an algorithm of this type to work robustly. One problem with applying his differential calculus results is that Phan-

Figure 9. An image of a face and cartoon output.

tom prunes responses due to noise. Thus, label-transitions can appear at coarse scales where there was no fine-scale response, although infinite-resolution zero-crossings cannot do this. There can be slopes, in either direction, in the middle of a staircase region, so that it is difficult to prune the phantom boundaries in staircase regions while leaving weak real boundaries intact.

The current algorithms in Phantom for detecting roof edges and for eliminating staircase phantoms have a fundamental resolution limit: they can only separate responses if they are separated at the finest scale, i.e. about 12 cells apart. In fact, the whole representation of boundaries via dark/light and edge/back/roof labelling has problems representing staircase regions less than 4 cells wide, although it can represent alternating black and white stripes that are only one cell wide. It may be possible to sharpen up these algorithms so that they will work on smaller regions. However, I have also heard secondhand that humans have problems with very fine staircase patterns.

XI CONCLUSIONS

The edge finder algorithm described is implemented in C on a HP 9000 (Bobcat). It requires about 40 minutes to process a 500 by 500 image. Since most of computation involves extremely simple local pixel operations, it should speed up dramatically on special image-processing hardware (e.g. the DATACUBE convolution and systolic array processors). The program has no parameters which must be set by the user. The noise threshold, as well as parameters describing neighborhood sizes, mask imbalance, and the like, can be kept fixed for a given camera/digitizer setup. They reflect properties of the imaging system, not properties of any particular image.

The edge finder is currently implemented only for regular rectangular arrangements of cells, but the algorithm does not depend on this restriction. For non-regular arrangements, the edge finder will be provided with the adjacency structure as well as sizes of cells. This information would be used to find short paths (about 5 cells long) which are approximately straight and determine when two nearby paths are approximately parallel. The edge finder does not depend on a global definition of the relative orientation of line segments nor of straight lines.

I hope to extend the work described in the paper in two different ways, First, I intend to continue the mathematical development of the cell representations. Secondly, algorithms for higher-level reasoning need to be built, using the output of the edge finder. Such reasoning might include finding texture boundaries, shape representation, curve tracing or region filling, and simple motion planning.

ACKNOWLEDGEMENTS

Hal Abelson, Mike Brady, Bob Boie, David Chapman, Dave Forsyth, Liz Johnston, Alison Noble, and Andrew Parker gave me helpful technical comments. The other people who have given me more diffuse aid and support have become too numerous to mention.

REFERENCES

Allen, James F., "Maintaining knowledge about temporal intervals," CACM 26:11 (1983) 832-843.

———, 'Towards a General Model of Action and Time," *Artificial Intelligence* 23:2 (1984).

Allen, James F. and Patrick J. Hayes, "A Common-Sense Theory of Time," IJCAI 1985, 528-531.

Asada, Haruo and J. Michael Brady, "The Curvature Primal Sketch," MIT AIM-758, 1984.

van Benthem, J.F.A.K., *The Logic of Time*, D. Reidel, Dordrecht, 1983.

Berzins, Valdis, "Accuracy of Laplacian Edge Detectors," CVGIP (1984) 195-210.

Blake, Andrew, "Parallel Computation in Low-level Vision," Ph.D. thesis, University of Edinburgh, 1983.

Brady, J. Michael and Haruo Asada, "Smoothed Local Symmetries and Their Implementation," IJRR 3:3 (1984) 36-61.

Brooks, Rodney A. and Tomás Lozano-Pérez, "A Subdivision Algorithm in Configuration Space for Findpath with Rotation," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol SMC-15 (1985).

Canny, John F., "Finding Edges and Lines in Images," MIT AI Lab., TR–720, 1983.

Clark, James J. (unpubl.) "Authenticating Edges Produced by Zero Crossing Algorithms," unpublished manuscript.

Connell, Jonathan H., "Learning Shape Descriptions," MIT AI Lab. TR–853, 1985.

Dowty, David R., *Word Meaning and Montague Grammar*, D. Reidel, Dordrecht, 1979.

Fleck, Margaret M., "Local Rotational Symmetries," MIT AI Lab. TR–852, 1985.

———, "Local Rotational Symmetries," CVPR 1986.

Forbus, Kenneth D., "Qualitative Process Theory," MIT AI Lab., TR–789, 1984.

Gennert, Michael A., "Detecting Half-Edges and Vertices in Images," CVPR (1986) 552-557.

Haralick, Robert M., Layne T. Watson, and Thomas J. Laffey, "The Topographic Primal Sketch," IJRR 2 (1983) 50-72.

Hayes, Patrick J., "Naive Physics 1: Ontology for Liquids," unpubl. (1978a).

———, "The Naive Physics Manifesto," unpubl. (1978b).

Hildreth, Ellen C., "The Detection of Intensity Changes by Computer and Biological Vision System," CVGIP 22 (1983) 1-27.

Lee, Chung-Nim and Azriel Rosenfeld, "Connectivity Issues in 2D and 3D Images," CVPR (1986) 278-285.

Marr, David and Ellen Hildreth, "Theory of Edge Detection," *Proc. Royal Soc. London B* 207 (1980) 187-217.

Marr, David, T. Poggio, and E. Hildreth, "Smallest Channel in Early Human Vision," *J. Opt. Soc. Am.* 70 (1980) 868-870.

Munkres, James R., *Elements of Algebraic Topology*, Addison-Wesley, 1984.

Pavlidis, Theo, *Structural Pattern Recognition*, Springer-Verlag, Berlin, 1977.

Pearson, Don E. and John A. Robinson, "Visual Communication at Very Low Data Rates," Proc. IEEE 73 (1985) 795-812.

Poston, Tim, "Fuzzy Geometry", Ph.D. thesis, Univ. of Warwick, 1971.

Watt, R.J. and M.J. Morgan, "Spatial Filters and the Localization of Luminance Changes in Human Vision," *Vision Research* 24 (1984) 1387-1397.

Watt, R.J. and M.J. Morgan, "A Theory of the Primitive Spatial Code in Human Vision," appearing in *Vision Research* (198X).