Spectre: an Improved Phantom Edge Finder

Margaret M. Fleck

Oxford University,
Dept. of Engineering Science,
Oxford, OX1 3PJ.

This paper describes an improved version of the Phantom edge finder, called Spectre. Spectre is simpler to implement, faster, and does more extensive image processing. The crucial topological sum operation is improved in two ways: a new tesselation for representing images simplifies sums along 1D paths and a simpler, directional method is used to extend 1D sums to 2D.

Both edge finders use the topological sum to remove noise from second difference responses, so that zero-crossings in these responses represent reliable boundaries. Spectre applies the same techniques to a combination of first and third differences, producing a new map that reliably distinguishes real boundaries from spurious boundaries appearing in staircase intensity patterns. Similar processing is used to mark response regions not associated with zero-crossings (including roof edges) and to compute boundary strengths.

The dominant problem in designing edge finders is how to suppress the effects of camera noise while retaining as much image detail as possible. For example, the Phantom edge finder [1,2] reports boundaries where second directional differences change sign. However, when these differences are near zero, their signs can be altered by camera noise, causing the edge finder to report spurious boundaries. The edge finder identifies regions in which the sign has been corrupted by noise and re-classifies them as having near-zero response, so that they will generate no boundaries.

The Phantom edge finder identifies noisy responses using a *topological sum.* This operation integrates response amplitude over connected regions in which the response does not change sign. So, in Figure 1, the sum at location x will be the area of the shaded region. Cells whose topological sum is below a threshold are re-classified as near-zero. This technique suppresses the effects of noise with less smoothing than previous algorithms (e.g. Canny's [3]) and thus retains more image detail [2].

Previous uses of topological sums [4,5,6] have typically considered only the 1D case. The 2D extension in [7,8] is unclear and untested. The Phantom edge finder sums responses over *star-convex* 2D regions, i.e. regions in which each cell is connected to the starting cell by a straight path. This 2D extension produced good results, but proved difficult and inefficient to implement. In this paper, I present a simpler and faster 2D extension, together with a streamlined overall algorithm called *Spectre.*

A second difficulty with second difference edge finders, including the Phantom edge finder, is that they report spurious boundaries in regions of uniform intensity when two boundaries of the same polarity occur next to one another, as in the top of the righthand console in Figures 2–4. These boundaries should not cause serious problems for low-level algorithms such as stereo matchers,[1] but they will confuse higher-level processing. These boundaries differ from real boundaries in their behavior across scales [9,10]. However, no robust method of using this observation has been proposed and it cannot, even in principle, classify features visible only at the finest scale.

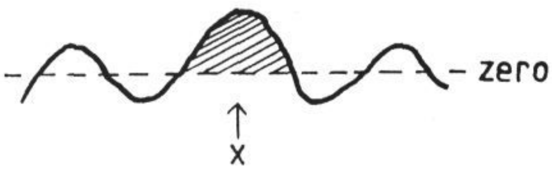

Figure 1: ***The topological sum at location*** x ***is the area of the shaded region.***

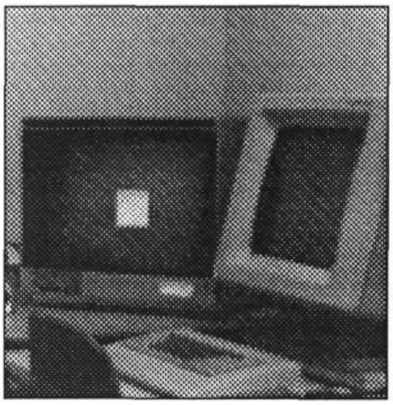

Figure 2: ***A 300 by 300 image of a console and display.***

It is well-known theoretically that the first and third differences at these spurious boundaries have the same sign, whereas they have opposite signs at real boundaries. However, previous implementations of this *sign test* seem to work only on extremely low-resolution images [11,12] or for constrained boundary shapes [13,14]. I will show that the same methods used to remove noise from second differences can also be used to produce a reliable sign test map, even in dense texture.

[1] The spurious boundaries are unstable only under those changes in viewpoint that also change the apparent size of regions. Some researchers [9] view the scale instability as fatal to low-level matching. However, real boundaries are also unstable under scale changes, except in special cases such as ideal step edges.

AVC 1989 doi:10.5244/C.3.22

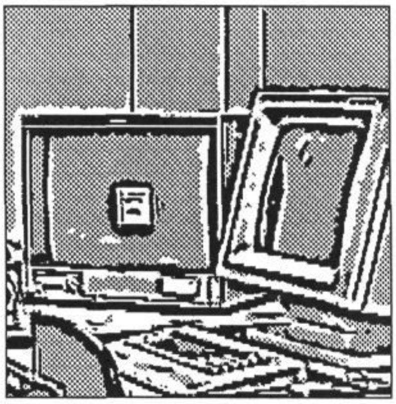

Figure 3: ***Second difference labels for the image in Figure 2. Cells labelled*** DARK ***and*** LIGHT ***are shown in black and white, cells with both labels or neither are shown in grey.***

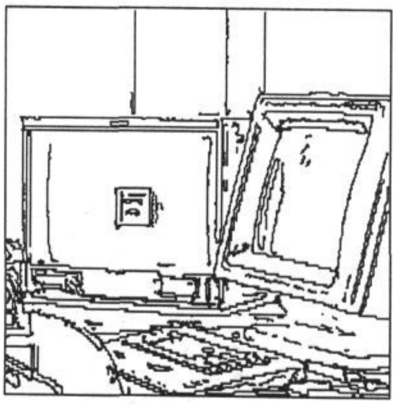

Figure 4: ***Zero-crossings from the second differences in Figure 3.***

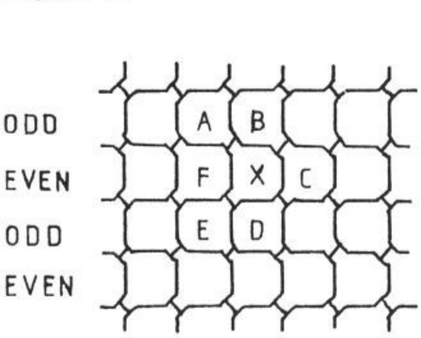

Figure 5: ***In the pseudo-hexagonal tesselation, cells have almost the same size and locations as in a rectangular array. However, each cell (e.g. X) touches only six other cells (A, B, C, D, E, and F) all along extended edges.***

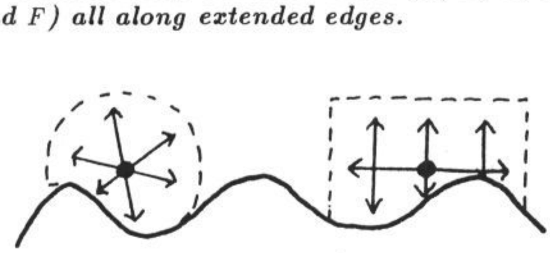

Figure 6: ***Left:** a star-convex connected neighborhood. **Right:** a directional connected neighborhood.*

Angle	Steps	Cells in even rows (even y)	Cells in odd rows (odd y)
0	all	(1,0)	(1,0)
34	odd	(1,0)	(1,0)
	even	(0,1)	(1,1)
64	all	(0,1)	(1,1)
90	all	(0,1)	(0,1)
116	all	(-1,1)	(0,1)
146	odd	(-1,0)	(-1,0)
	even	(-1,1)	(0,1)

Table 1: ***The (x,y) displacements used to move between successive elements of paths.***

1D TOPOLOGICAL SUMS

Topological sums in 2D are based on 1D topological sums taken along straight paths. Most computer vision algorithms represent images using rectangular or hexagonal tesselations. However, in a rectangular tesselation, diagonally adjacent cells touch at just a vertex. A same-sign region containing such a pair of cells fails to be connected if the other cells meeting at that vertex have contrasting signs. Thus, all four cells must be examined in building an integration region.[2] Hexagonal tesselations lack this connectivity problem but require re-sampling the image.

I use a new *pseudo-hexagonal* cell arrangement, shown in Figure 5, that combines the advantages of rectangular and hexagonal tesselations. Because cell locations and shapes are almost rectangular, digitized intensity values from standard camera systems can be used without re-sampling. However, two adjacent cells always meet along an extended edge, never at just a vertex, so building integration regions only requires examining cells actually on the integration path.

Suppose that we have taken the second differences of image intensities in some direction. Each cell can be labelled DARK if its response is above zero, LIGHT if it is below zero, and NEAR-ZERO otherwise. Processing described in later sections may give some cells the special label IMPASSABLE. Two labels *contrast* if (a) either one is IMPASSABLE or (b) one is DARK and the other LIGHT. Integration along a path stops when two cells in it have contrasting labels.

Integration along a path proceeds by moving response amplitudes and labels along the path, cell by cell, out to some desired path length. At each step, the shifted and unshifted labels at each cell are examined. If they do not contrast, the shifted response amplitude is added to the output sum at that cell. If they contrast, the shifted response amplitude is zeroed.

To compute the full topological sum, this shift-and-add process must be repeated twice, moving in opposite directions along the path. This technique is easily generalized to any set of mutually-exclusive labels, plus the special labels NEAR-ZERO and IMPASSABLE. Although the labels in this example reflected the signs of the responses, the topological sum computation does not depend on the existence of any such relationship.

The current implementation uses paths in 6 directions. However, in the tesselation of Figure 5, not all diagonal moves are possible. Table 1 shows how the algorithm moves from one cell to the next one on the path. For example, to

[2] Defining connectivity as in [1,2]. Similar facts hold for other definitions [15].

move along a path at 34 degrees, starting from an cell in an odd row, it moves one cell right, then one cell up, then one cell right, then (since the current cell is now in an even row) diagonally up and to the right. These paths were chosen so that the entire image can be shifted at once in the path direction (convenient for parallel implementations) and so that they approximate the directions in which differences are taken (Appendix A). To produce negative displacements along these paths, one must not only reverse the signs of the displacements shown in Table 1, but also exchange the roles of even and odd rows.

SECOND DIFFERENCES

Spectre's first task is to classify cells based on the signs of the second differences of image intensity. Spectre uses directional differences taken in six directions about each cell (Appendix A). Based on these differences, each cell is labelled DARK, LIGHT, NEAR-ZERO, or IMPASSABLE. This method performs better at corners and junctions than isotropic operators [1,2]. The 1D topological sum is used to remove the effects of noise from this labelling.

Spectre classifies second differences in much the same way as the Phantom edge finder [1,2]. However, a simpler gap filling algorithm and judicious rearrangement of operations allows Spectre to use fewer topological sums and to use directional, rather than isotropic sums. Specifically, Spectre first *constructs a preliminary labelling, using differences from all* directions. This labelling is used to guide removal of noise from the responses in individual directions.

Responses in each direction are labelled as DARK, LIGHT, or NEAR-ZERO, depending on their sign. The *global label* at each cell summarizes its pattern of labels over all directions. Many cells are labelled both LIGHT and DARK, in different directions. However, the cell is given both labels in the global map only if the amplitudes of the largest positive and smallest negative differences are comparable. In the current implementation, the smaller must be at least 40 percent of the larger. Otherwise, the cell is given the label of the strongest directional response, or NEAR-ZERO if all responses are zero.

Global labels due to image noise are identified by considering each set of directional responses separately. First, the directional labels are reconciled with the global labels. Specifically, if the directional label is DARK and the global map contains only the label LIGHT, or vice versa, the cell is re-labelled IMPASSABLE. Also, where the directional map is NEAR-ZERO, it inherits any labels present in the global map.

Using these modified directional labels, Spectre takes two cascaded 1D topological sums. The first sum integrates the responses in the same direction in which differences were taken, proceeding out 3 cells each direction (i.e. a maximum of 7 cells contribute to the sum at each cell). These summed responses are then integrated along paths perpendicular to the difference direction, out to a radius of 5 cells (i.e. 11 cells total). Cells whose response sum is below a threshold (currently 600) [3] are re-labelled as NEAR-ZERO.

Figure 6 compares this directional integration to the star-convex integration used in the Phantom edge finder. Directional integration is easier to implement and requires fewer operations. However, since it is not isotropic, it must be applied to data that specifies a natural direction of application, such as the second differences used in Spectre. For other tasks, an isotropic operator can be reconstructed by applying the directional sum in all six directions and taking the maximum response.

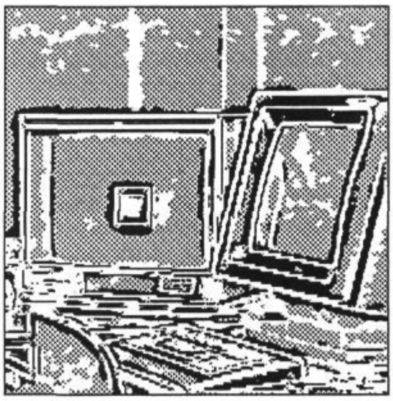

Figure 7: ***Sign test results for the image in Figure 2. Cells passing the sign test are shown in white, those failing it in black, and those*** NEAR-ZERO ***or receiving both labels in grey.***

After removing noisy labels in each direction, very small response regions are pruned. The resulting clean directional labels are OR-ed together into a new global labelling. Finally, small gaps in the global labelling are filled, yielding a clean response map, like that in Figure 3. The pruning and filling code is a simple adaptation of techniques from digital morphology, described in Appendix B.

Sign changes in the second difference map now represent potential boundaries (Figure 4). For later processing, cells labelled both DARK and LIGHT are re-interpreted as IMPASSABLE. Boundaries then consist of all cells labelled IMPASSABLE, together with all cell edges separating a LIGHT cell from a DARK cell.[4] As discussed in the introduction, some of these boundaries may be spurious.

FURTHER PROCESSING

To eliminate spurious boundaries and provide more detailed image descriptions, Spectre computes three further image maps. The *sign test* map shows where the first and third differences have opposite sign. It is used to prune spurious boundaries. Another map marks significant areas of second difference response not associated with zero-crossings (e.g. "roof" edges). Finally, amplitudes are computed for cells near zero-crossings or in non-zero-crossing responses.

In each direction, I define the sign test function to be positive exactly when the first and third directional differences have the same sign. Thus, a boundary is real if the sign test is negative and spurious otherwise. I define the magnitude of the sign test to be the minimum of the magnitudes of these the first and third differences. Unlike the product used in [9,11], this allows the magnitude of the sign test to reflect the reliability of its sign.

The sign test function is processed in exactly the same way as the second differences, but using a noise threshold of 300. This produces clean maps, as in Figure 7. Figure 8 shows boundaries induced by second difference labels from cells that pass the sign test. Notice that the spurious boundaries in the righthand console have been pruned. Figure 9 shows the effects of pruning on complex intersections.

[3] Including all IMPASSABLE cells

[4] See [1,2] for formal models of these boundaries.

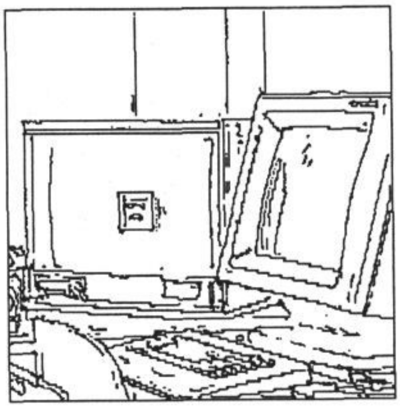

Figure 8: ***Zero-crossings from second differences (Figure 3) from cells that pass the sign test (Figure 7).***

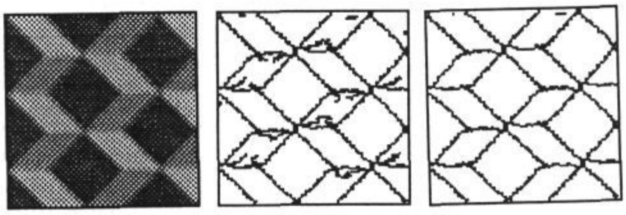

Figure 9: ***A 100 by 100 image of intersections with zero-crossing boundaries before (left) and after (right) sign test pruning.***

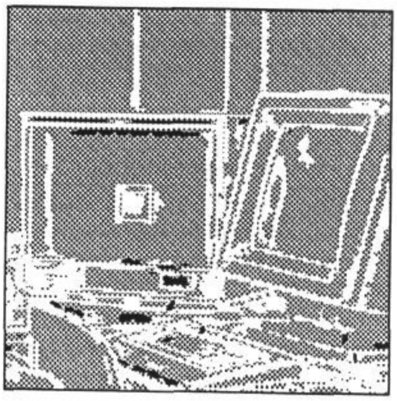

Figure 10: ***White: cells in Figure 2 that are associated with zero-crossings (i.e. cells labelled*** NEAR-ZC***). Black: significant areas of response not associated with zero-crossings (i.e. cells labelled*** NOT-ZC***). Cells given both labels are also shown in black.***

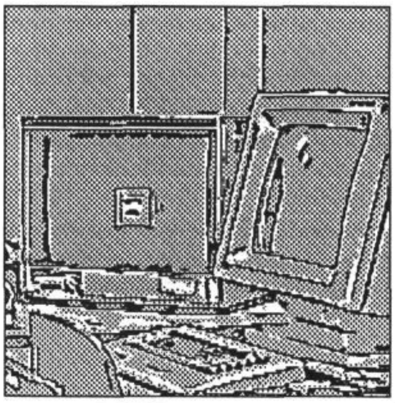

Figure 11: ***Second difference responses (as in Figure 3), but only for cells given either the label*** NEAR-ZC ***or the label*** NOT-ZC.

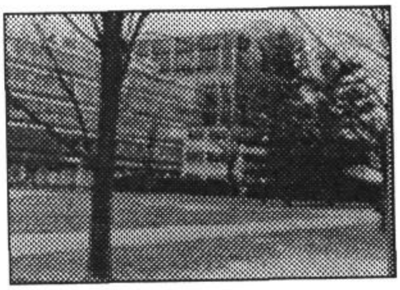

Figure 12: ***A 240 by 170 outdoor scene.***

The combination of the second difference map and the sign test map introduces a variety of new features that could supplement boundary locations in vision algorithms. For example, the width of the sign test regions might be useful in estimating boundary blur [16]. Locations where second differences fall to NEAR-ZERO could be used in stereo matching and might explain the data in [17,18]. Finally, regions of failing sign test often separate the second difference responses of closely-packed features, making it easier to classify responses.

It has long been recognized that images contain second difference responses not associated with zero-crossings, e.g. "roof" edges, shadowed object edges, and bright patches on smoothly-shaded objects. Figure 10 shows some examples. However, many of these responses are connected to zero-crossings in the second difference map, making them impossible for previous proposed methods [6,19] to detect.

To identify these regions, Spectre uses a modified set of second difference labels, in which cells with no second difference or with a failing sign test are re-labelled as NO-RESPONSE. Cells next to zero-crossings are identified as NEAR-ZC, as are

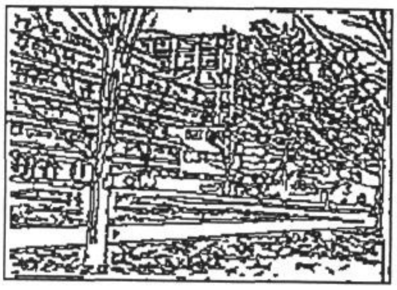

Figure 13: ***Boundaries for the image in Figure 12 before sign test pruning.***

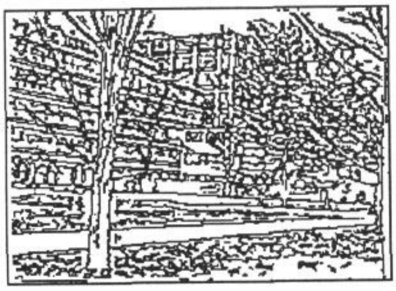

Figure 14: ***Boundaries for the image in Figure 12 after sign test pruning.***

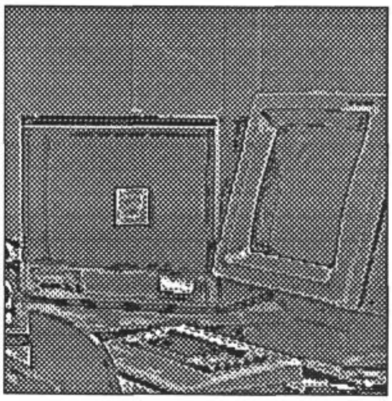

Figure 15: ***Boundary strengths for the image in Figure 2, shown only for cells assigned the label*** NEAR-ZC ***or*** NON-ZC.

cells in the middle of zero-crossings (i.e. IMPASSABLE cells). In each direction, the 1D topological sum (radius 7) is then used to spread NEAR-ZC labels to the entire connected response regions. Small holes are filled as described in Appendix B.

In each direction, second difference responses for cells labeled NEAR-ZC or NO-RESPONSE are deemed to be adequately explained by the zero-crossings. These responses are re-set to zero. To remove noise, a directional topological sum is run on the remaining responses, still using the modified second difference labels. Cells with sum above 600 are labelled NON-ZC and small holes in this labelling filled (Appendix B). The map shown in Figure 10 was produced by OR-ing labels from all directions. Figure 11 shows second difference labels for cells assigned either NEAR-ZC or NON-ZC labels.

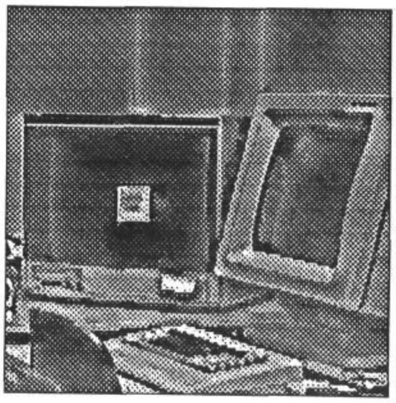

Figure 16: ***Boundary strengths for the image in Figure 2, interpolated to give local intensity values for the whole image.***

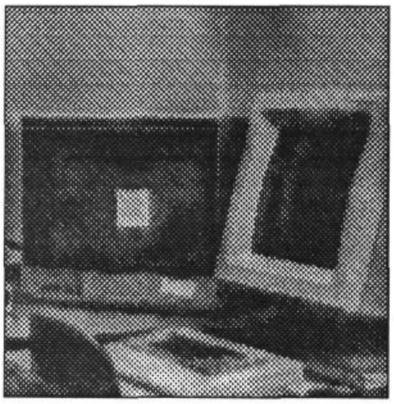

Figure 17: ***Intensities for the image in Figure 2, reconstructed from the boundary strengths shown in Figure 15.***

This processing also works well on natural, finely-textured images, as illustrated by Figures 12–14. Some examples of boundaries failing the sign test can be seen on the tree trunk and at the top of the lawn. In general, this class of spurious boundaries seems not to be very common: the image in Figure 2 contains an exceptionally large number.

The final processing step computes boundary strengths. In each direction, the 1D topological sum of second difference responses is taken parallel to the difference direction, using the second difference labels.[5] Output strengths are the maximum amplitude responses over all directions. Figure 15 shows these strengths for the NEAR-ZC and NON-ZC cells.

Edge finder results at coarser scales can be produced by running Spectre on a pyramid of smoothed and sampled versions of the image. In this implementation, each dimension of the image was reduced by a factor of 2 between successive scales. Sampling continues until either dimension falls below a minimum size (10 cells). This process generates a discrete set of scales, typically 5–7 per image.

Interestingly, a full map of image intensities can be reconstructed from the amplitudes at NEAR-ZC and NON-ZC cells at all scales. Strengths at these cells are first interpolated over the whole image (using a pyramid data structure) as in Figure 16. Results at all scales are added together, by smoothly expanding the coarser-scale results, producing the reconstruction in Figure 17. This reconstruction cannot be exact. For example, this algorithm should mimic the Craik-O'Brian-Cornsweet illusion found in human perception. However,the similarity to the original image suggests that the boundary locations and amplitudes contain most of the important information in the image (contrast [20]).

CONCLUSIONS

We have seen that the new edge finder, Spectre, is simple to build and can do more extensive image processing than the Phantom edge finder. Spectre is implemented in C and requires 13.5 minutes on a Sun 4/260C workstation to process a 512 by 512 image at the finest scale (18 minutes for all scales). Of this, extracting the clean second difference map (the processing done by the Phantom edge finder) requires only about 4 minutes (5.4 minutes for all scales). Since all operations are simple and local, it should run much faster on parallel hardware.

[5] The original, unmodified ones, as shown in Figure 2.

ACKNOWLEDGEMENTS

The author is supported by a junior research fellowship funded by BP. Mike Brady and David Forsyth supplied helpful comments.

APPENDIX A: DETAILS OF DIFFERENCES

Spectre uses first, second, and third differences in 6 directions. The image is first smoothed with the mask $[1, 2, 1]$ in both axis directions, to remove interlace differences. First differences are taken by applying the mask $[1, 0, -1]$ to the values at 3 cells. Depending on the desired direction, the lefthand cell would lie at one of the following displacements from the middle cell: $(1, 0)$, $(1, 0.5)$, $(0.5, 1)$, $(0, 1)$, $(-0.5, 1)$, and $(-1, 0.5)$ and the righthand cell in the opposite direction. Fractional displacements are achieved by averaging values from two adjacent cells. Second and third differences are computed similarly, using the mask $[1, -2, 1]$ for second differences, and $[1, -2, 0, 2, -1]$ for third differences.

To retain all available precision, all steps in this computation are implemented as weighted sums, with no division by mask norms. Thus, the first differences are effectively multiplied by 64, the second differences by 128, and the third differences by 96.

APPENDIX B: GAP FILLING AND PRUNING

To fill small gaps in response regions, Spectre uses a technique similar to morphological dilation followed by erosion [21]. Specifically, the label at each cell is propagated to any of its six neighbors that is labelled NEAR-ZERO. Some cells may receive both LIGHT and DARK labels. Labels are then stripped from any cell adjacent to a NEAR-ZERO cell.

Small response regions can be pruned by applying these operations in the reverse order. However, in directional responses, cells centered on boundaries may be labelled NEAR-ZERO. Raw pruning will unfairly remove labels near such cells. Therefore, in building second difference and sign test maps, directional labels are filled (propagate then strip) before pruning (strip then propagate). However, any new labels introduced by the filling are erased before directions are combined.

Similarly, in detection of non-zero-crossing response, gaps between NEAR-ZC and NO-RESPONSE regions must be filled, without filling small non-zero-crossing regions with NO-RESPONSE labels. In this case, NEAR-ZC and NO-RESPONSE labels are filled, but then any new NO-RESPONSE labels introduced by filling are erased.

References

[1] **Fleck, Margaret M.**, "Representing Space for Practical Reasoning," *Im. and Vis. Comp.* 6/2 (1988) pp. 75–86.

[2] **Fleck, Margaret M.**, "Boundaries and Topological Algorithms," Ph.D. thesis, MIT, also AI Lab. TR–1065 (1988).

[3] **Canny, John F.**, "A Computational Approach to Edge Detection," *IEEE Trans. Patt. Analy. Mach. Intell.* 8/6 (1986) pp. 679–698.

[4] **Huertas, Andres and Gerard Medioni**, "Detection of Intensity Changes with Subpixel Accuracy Using Laplacian-Gaussian Masks," *IEEE Trans. Patt. Analy. Mach. Intell.* 8/5 (1986) pp. 651–664.

[5] **Huttenlocher, Daniel P.**, "Three-Dimensional Recognition of Solid Objects from a Two-Dimensional Image," Ph.D. thesis, MIT (1988).

[6] **Watt, R.J. and M.J. Morgan**, "A Theory of the Primitive Spatial Code in Human Vision," *Vision Research* 25/11 (1985) pp. 1661–1674.

[7] **Huang, K., D. Lee, and T. Pavlidis**, "Edge Detection through Two-Dimensional Regularization," *Proc. IEEE Work. on Comp. Vis.* (1987) pp. 225–227.

[8] **D. Lee, T. Pavlidis, and K. Huang**, "Edge Detection through Residual Analysis," *Proc. IEEE Conf. Comp. Vis. Patt. Recog.* (1988) pp. 215–222.

[9] **J. J. Clark**, "Singularity Theory and Phantom Edges in Scale Space," *IEEE Trans. Patt. Analy. Mach. Intell.* 10/5 (1988) pp. 720–727.

[10] **Y. Lu and R.C. Jain**, "Behavior of Edges in Scale Space," *IEEE Trans. Patt. Analy. Mach. Intell.* 11/4 (1989) pp. 337–356.

[11] **Clark, James J.**, "Authenticating Edges Produced by Zero Crossing Algorithms," *IEEE Trans. Patt. Analy. Mach. Intell.* 11/1 (1989) pp. 43–57.

[12] **Ulupinar, Fatih and Gérard Medioni** "Refining Edges Detected by a LoG Operator," *Proc. IEEE Conf. Comp. Vis. Patt. Recog.* (1988) pp. 202–207.

[13] **Haralick, Robert M.**, "Digital Step Edges from Zero Crossings of Second Directional Differences," *IEEE Trans. Patt. Analy. Mach. Intell.* 6/1 (1984) pp. 58–68.

[14] **Chen, J. S. and G. Medioni**, "Detection, Localization and Estimation of Edges," *IEEE Trans. Patt. Analy. Mach. Intell.* 11/2 (1989) pp. 191–198.

[15] **Pavlidis, Theo**, *Structural Pattern Recognition*, Springer-Verlag, Berlin (1977)

[16] **Watt, R.J. and M.J. Morgan**, "The Recognition and Representation of Edge Blur: Evidence for Spatial Primitives in Human Vision," *Vision Research* 23/12 (1983) pp. 1465–1477.

[17] **Bülthoff, Heinrich H. and Hanspeter A. Mallot**, "Interaction of Depth Modules: Stereo and Shading," *Jour. Opt. Soc. Amer.* 5/10 (1988) pp. 1749–1758.

[18] **Mayhew, John E.W. and John P. Frisby**, "Psychophysical and Computational Studies towards a Theory of Human Stereopsis," *Artif. Intell.* 17 (1981) pp. 349–385.

[19] **Marr, D. and E. C. Hildreth**, "Theory of Edge Detection," *Proc. Roy. Soc. London B* vol. 207 (1980) 187–217.

[20] **Hummel, Robert A.**, "Representations based on Zero-Crossings in Scale-Space," *Proc. IEEE Conf. Comp. Vis. Patt. Recog.* (1986) pp. 204–209.

[21] **Serra, J.** *Image Analysis and Mathematical Morphology*, Academic Press, New York (1982).