# Comments on “Ground from Figure Discrimination”

Paul L. Rosin
Department of Computer Science
Cardiff University
UK
Paul.Rosin@cs.cf.ac.uk

## 1 Introduction

In a recent paper Amir and Lindenbaum [1] described a method for discriminating between foreground and background features. Over the last decade or two there has been considerable interest in computational solutions to this problem, much of which has focussed on alternative search techniques [2, 6, 9, 12] and neural network implementations [8, 5, 3, 4, 11].

The basis of Amir and Lindenbaum's algorithm is the commonly used characteristic of foreground shapes that they tend to be smooth in comparison with background objects. In their paper they take as input binary edge points which are then connected to their nearest $k$ neighbours, creating the so-called “underlying graph”. To enforce goodness of shape a binary smoothness function is applied to connected edgel pairs in the graph, and is computed as a combination of proximity, cocircularity, similar orientation, and low curvature. Only arcs satisfying all the constraints are retained to form the “measured graph”. Nodes with a low proportion of edgel neighbours that pass the smoothness test are considered to be part of the background. A form of relaxation labelling is applied to the two graphs in which background nodes are iteratively deleted and new arcs are inserted into the graphs so that nodes are connected to at least $k$ neighbours. The final “measured graph” contains the foreground edge set.

Our criticism of this scheme is that it is unnecessarily complex. For the sort of data that Amir and Lindenbaum used to demonstrate their algorithm (i.e. standard edge detector output) it is possible to use simpler, more efficient, and more robust edge thresholding methods. In particular, this is possible if the connectivity and gradient magnitude information generally readily available from the edge map is used, rather than discarded (or at any rate not required) as in Amir and Lindenbaum's scheme. It is also possible to use many other edge characteristics in the thresholding process such as clutter, lifetime, regularity, etc. [7]. For example, Venkatesh and Rosin [10] described an approach in which, prior to thresholding, all the connected edgel curves are processed as single elements. A plot of curve length versus the mean gradient magnitude over the curve shows a distinctive triangular cluster formed by the edges arising from background noise. Based on simple robust statistics (medians and median absolute deviations) the triangle can be reliably located and eliminated, leaving just the foreground edges. It is interesting to note that in both the Venkatesh/Rosin and Amir/Lindenbaum algorithms the same asymmetry is present – the background is determined first, and the foreground is simply whatever remains after the background is eliminated.

An important consideration in the practicality of an algorithm is the number of tuning parameters it contains. In Venkatesh and Rosin's scheme there is only one parameter which determines the cut-off line for eliminating the triangle, but in practise this is fixed to six. In contrast, each of Amir and Lindenbaum's four smoothness tests requires a threshold, while the node pruning phase needs yet another threshold to determine the transition value for the proportion of connected neighbours that distinguishes foreground and background nodes. In terms of complexity Amir and Lindenbaum's algorithm is also at a disadvantage, principally because it requires the $k$ nearest neighbours to be found. For $n$ edgels, if $m$ iterations are run then its complexity is $O(m(n \log n + kn))$. In comparison, once the lengths of the linked edge curves have been computed (which can be done on the fly during edge linking), Venkatesh and Rosin's algorithm is $O(e)$ where $e$ is the number of edge curves (assuming a linear median algorithm is used). Since the number of edge curves is relatively small compared to the number of edges, the total computation is negligible. This is confirmed by the run times. On a Sparc 10 Venkatesh and Rosin's algorithm took 48 seconds to process the sixty images described in section 3 (despite its inefficient I/O; this also includes the time for edge linking) while Amir and Lindenbaum's algorithm took 1606 seconds.

## 2 Experiments

Results are shown of applying the two algorithms. The first example images (figures 1–2) come from Amir and Lindenbaum's paper. Our copy of Amir and Lindenbaum's code required minor modifications to run on a different platform. Also, since the original parameters used are unavailable, it was not possible to produce results identical to those appearing in the original article. However, it can be seen that the new results (d) are close to the original ones (c). The original binary edge map was augmented with edge magnitude information and fed into Venkatesh and Rosin's algorithm. The results are somewhat underthresholded (f) as this algorithm assumes the full unthresholded edge set for input. When this is provided (using the Canny edge detector) then the results shown in (h) are different in detail to Amir and Lindenbaum's output, but comparable in the overall set of edges retained.

A second, larger scale and quantitative test was carried out using the South Florida data set of sixty images containing manually generated ground truth edges. As the images in the original paper show, the input used for Amir and Lindenbaum's algorithm had been pre-thresholded at some intermediate level. Therefore we ran two tests, one in which the full set of edges is provided, and the second in which weak edges are removed first by thresholding the gradient magnitudes at $T = 10$ where the gradients range in $0 - 255$. As before, Venkatesh and Rosin's algorithm is applied directly to the complete unthresholded edge map. Some example outputs are shown in figures 3–7. It can be seen that Amir and Lindenbaum's algorithm is prone to retaining small fragments of clutter – especially if the effects of noise are not reduced by thresholding out low magnitude edges.[1] Venkatesh and Rosin's algorithm produces results of comparable quality: it manages to successfully remove much of the background clutter while retaining the majority of the foreground structure. Since it operates on a more global level the small edge fragments are more successfully eliminated and many long connected curves are extracted. However, in some cases low strength insignificant features are retained if they form longer connected curves. Possibly incorporating some additional local shape information would improve on this.

From the ground truth edges the effectiveness of the two algorithms can be tested numerically. It should be noted that the ground truth is necessarily subjective. It tends to be sparse, missing out in many cases what appear to be significant details such as the writing on the meter in figure 3 and the trees in figure 4. The proportion of correctly classified pixels in the image is used as an estimate of the quality of the edge selection. Table 1 shows Amir and Lindenbaum's algorithm is improved when it is provided with fewer edges, while Venkatesh and Rosin's algorithm is rated just below. In comparison, simple thresholding at $T = 0$ and $T = 10$ produces substantially lower ratings. As a caveat, it must be remembered that edges only account for a small proportion of the image which skews the measurement. For instance, thresholding so as to eliminate *all* edges results in an even better overall classification rate! Nevertheless, visual examination of the results supports the conclusions drawn from table 1.

| method | mean | $\sigma$ |
|---|---|---|
| AL ($T = 0$) | 0.9020 | 0.0174 |
| AL ($T = 10$) | 0.9213 | 0.0174 |
| VR | 0.9175 | 0.0203 |
| $T = 0$ | 0.7064 | 0.0234 |
| $T = 10$ | 0.8463 | 0.0552 |
| no edges | 0.9641 | 0.0138 |
| all edges | 0.0359 | 0.0138 |

Table 1: Proportion of correctly classified (edge/non-edge) pixels. The results were calculated individually for each image, and the average scores (and standard deviations) for each method are shown over the set of 60 images. Binary edge maps were generated (at thresholds $T = 0$ and $T = 10$) and Amir and Lindenbaum's algorithm (AL) was run on each. Venkatesh and Rosin's algorithm (VR) was run on the full unthresholded edge map. As a control, an evaluation is included for testing a totally black or totally white image, i.e. where all pixels are classified as edges, or non-edges.

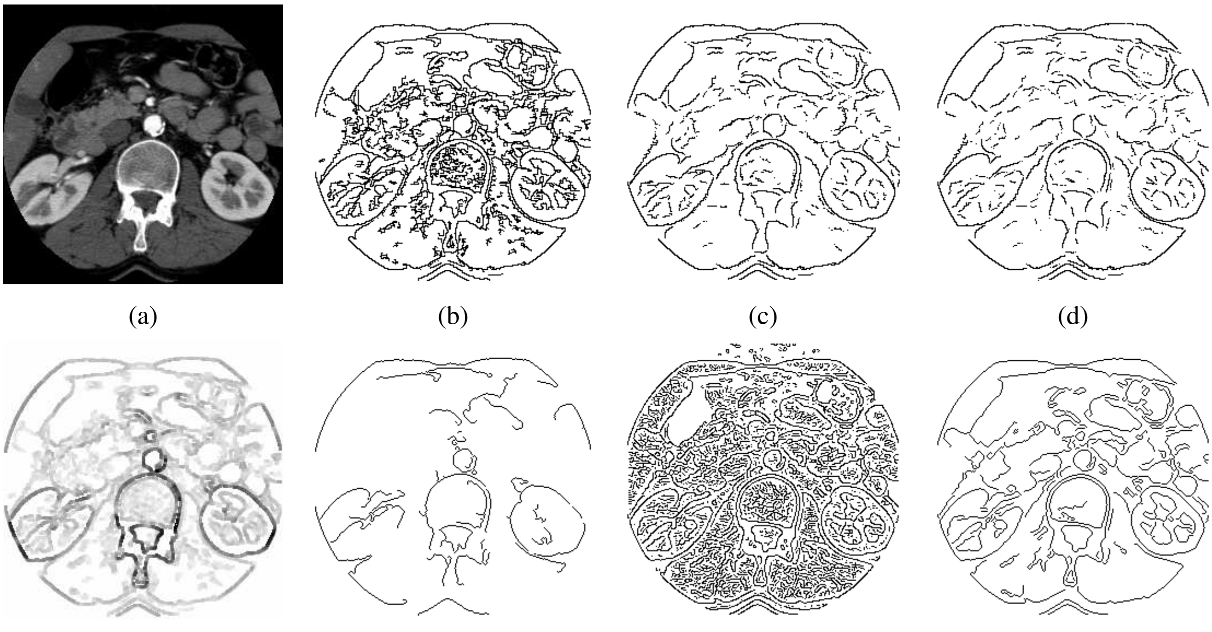
[1] It would be possible to eliminate some of the undesirable clutter generated by Amir and Lindenbaum's algorithm by setting a threshold on the minimum segment length retained. This post-processing step has not been investigated in this paper.

# 3 Conclusions

When applied to standard edge maps the recent work on ground from foreground discrimination by Amir and Lindenbaum appears to produce comparable results to the connected edge list thresholding algorithm by Venkatesh and Rosin that was described earlier in the same journal. If the edge attributes (e.g. curve length and gradient magnitude) are retained this then enables edge thresholding to be performed using an algorithm that is simpler and more efficient than Amir and Lindenbaum's, which also does not require the relatively large number of preset threshold values. The interested reader can download Venkatesh and Rosin's code from `http://www.cs.cf.ac.uk/User/Paul.Ros`


# Acknowledgments

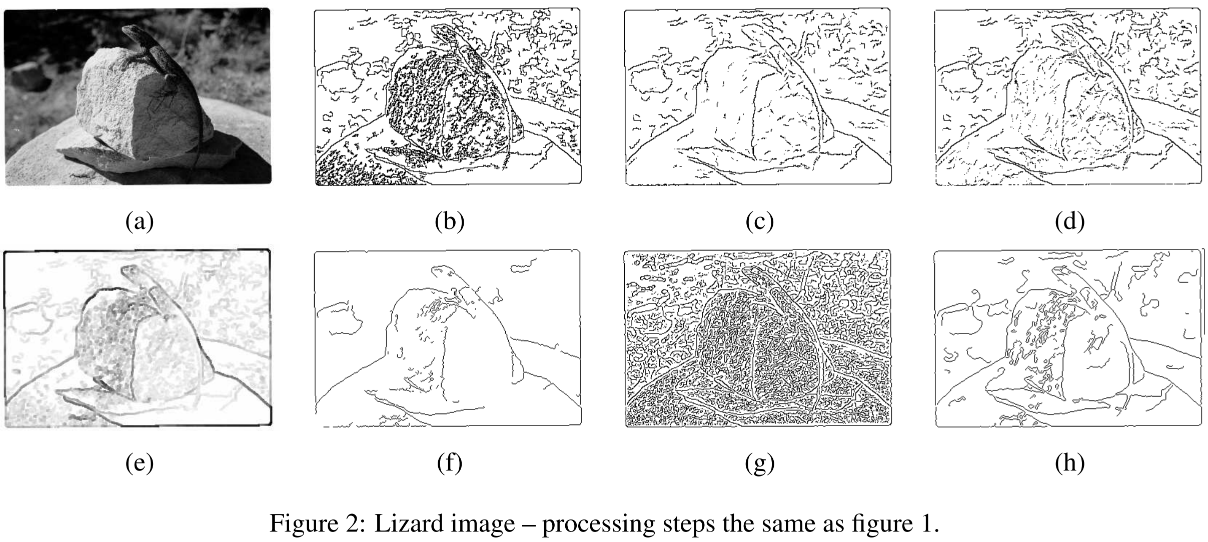
I would like to thank Arnon Amir for kindly providing me with a copy of his code, and Kevin W. Bowyer for making the data set of ground truthed images available.

(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 1: (a) CT image, (b) binary edge set, (c) AL published result, (d) regenerated AL result, (e) edge magnitudes, (f) VR output using e) as input (g) all Canny edges (shown thresholded for display purposes) (h) VR output using g) as input.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 2: Lizard image – processing steps the same as figure 1.

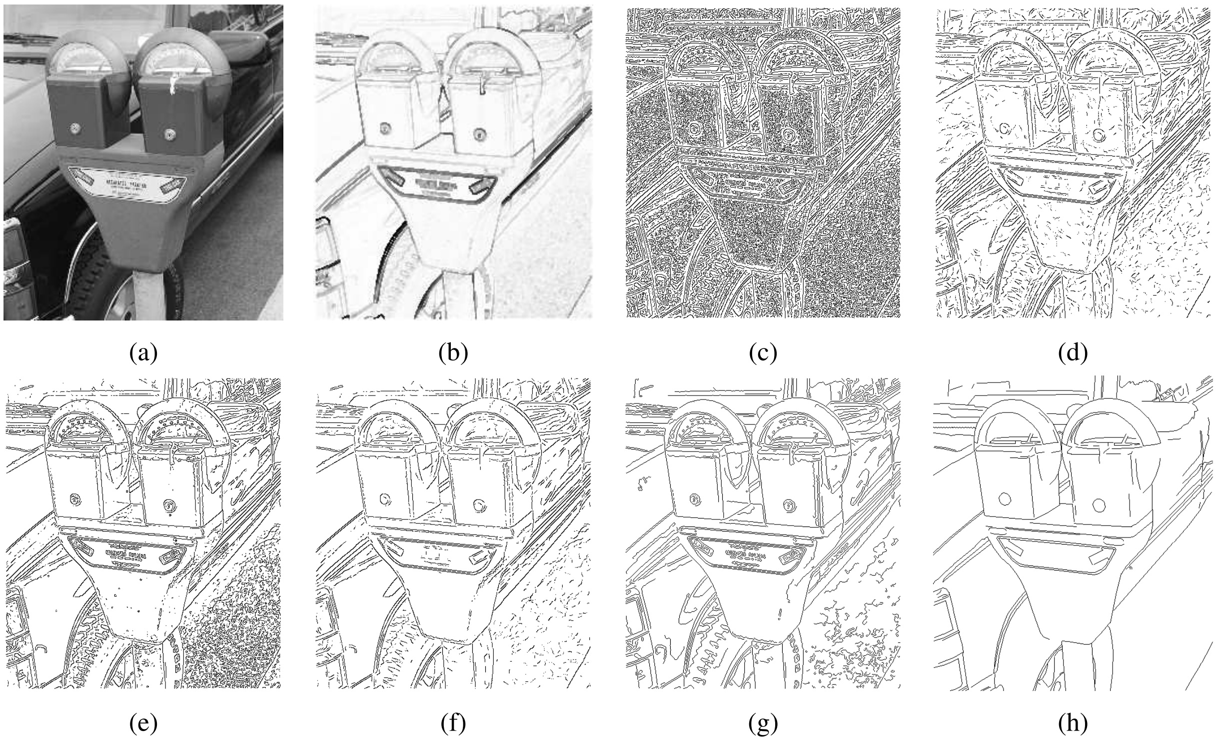



Figure 3: (a) Parking meter image, (b) edge magnitudes, (c) binary edge set (thresholded at $T = 0$), (d) AL result ($T = 0$), (e) binary edge set ($T = 10$), (f) AL result ($T = 10$), (g) VR result, (h) ground truth.

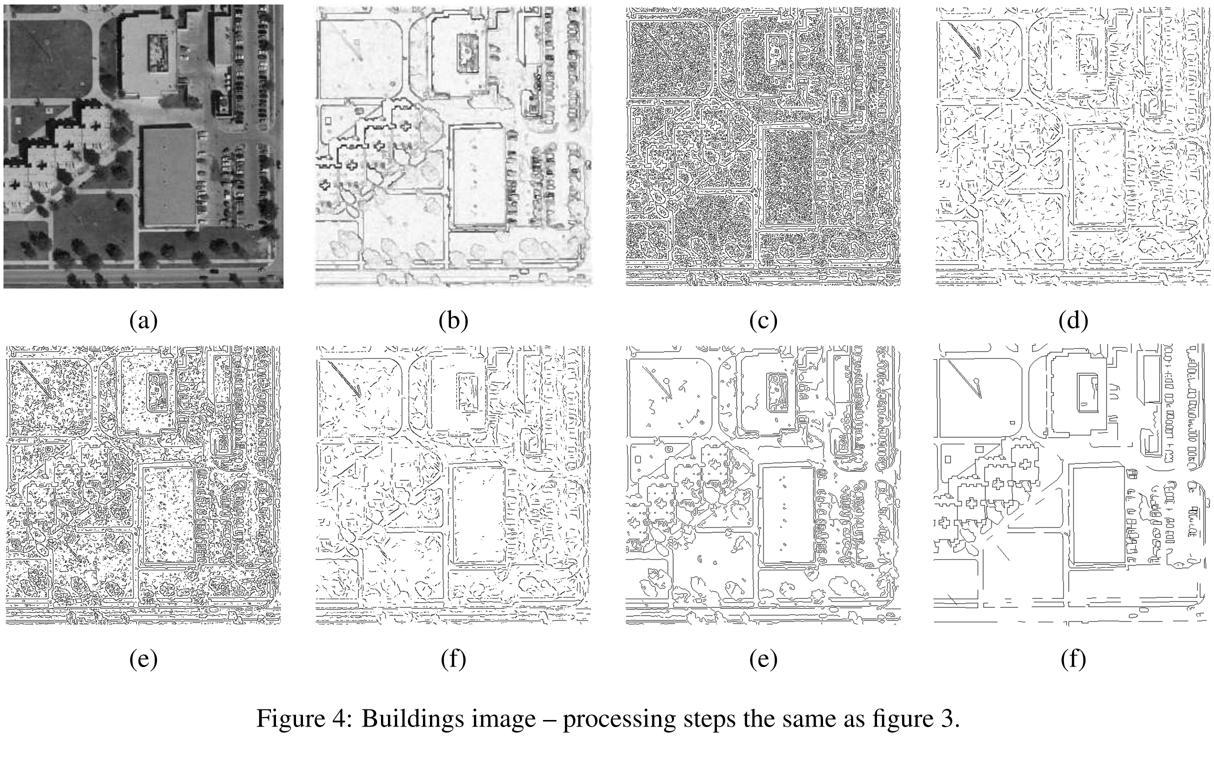
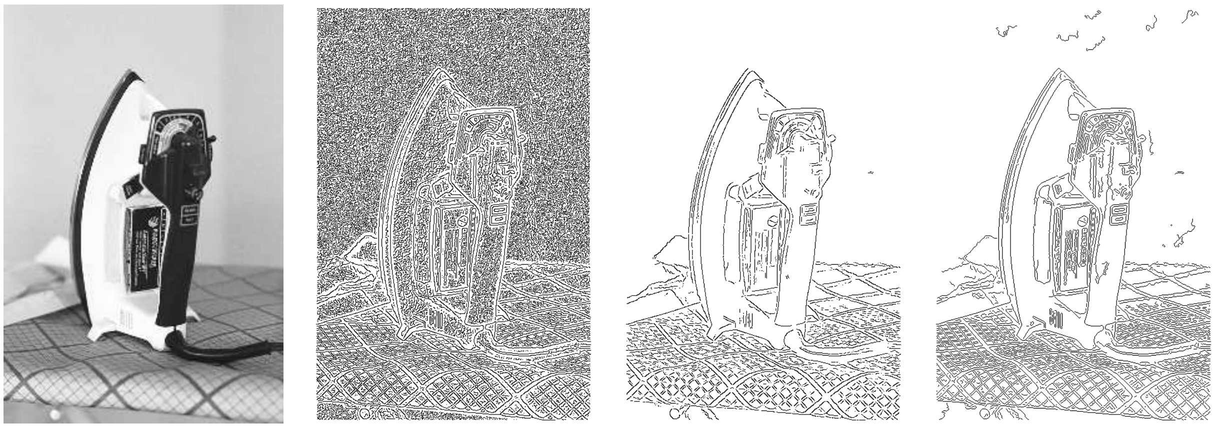
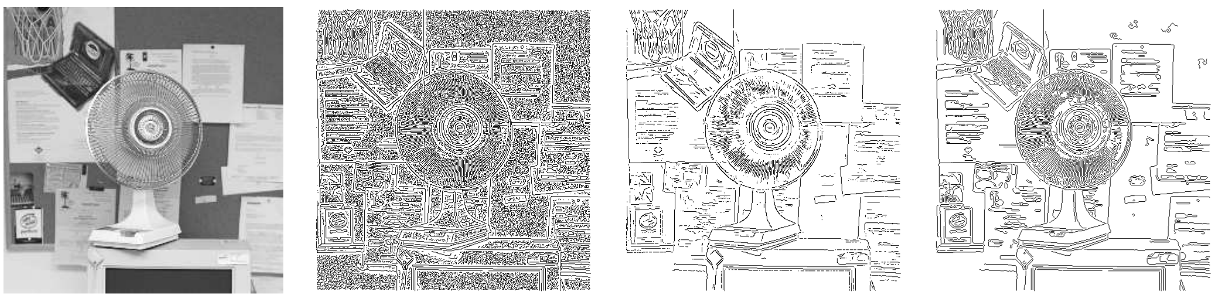



Figure 4: Buildings image – processing steps the same as figure 3.



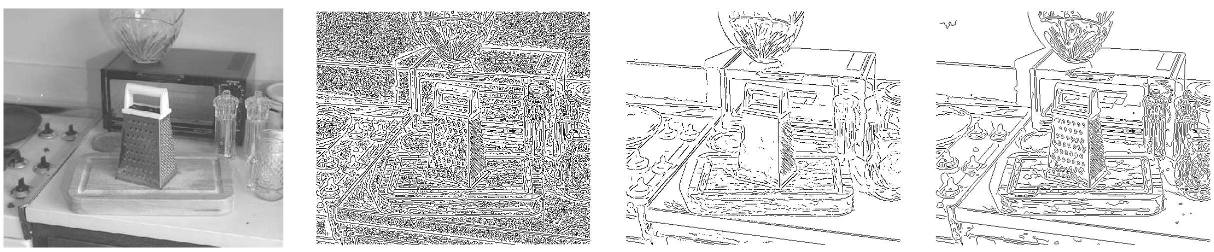
(a) (b) (c) (d)

Figure 5: (a) Image #101, (c) binary edge set (thresholded at $T = 0$), (c) AL result ($T = 10$), (d) VR result.



(a) (b) (c) (d)

Figure 6: Image #143 – processing steps the same as figure 5.



(a) (b) (c) (d)

Figure 7: Image #214 – processing steps the same as figure 5.

# References


[1] A. Amir and M. Lindenbaum. Ground from figure discrimination. *Computer Vision and Image Understanding*, 76(1):7–18, 1999.

[2] S.M. Bhandarkar and X. Zeng. Evolutionary approaches to figure-ground separation. *Journal of Applied Intelligence*, 11(2):187–212, 1999.

[3] S. Grossberg. A solution of the figure-ground problem for biological vision. *Neural Networks*, 6(4):463–484, 1993.

[4] S. Grossberg and L. Pessoa. Texture segregation, surface representation, and figure-ground separation. *Vision Research*, 38:2657–2684, 1998.

[5] S. Grossberg and L. Wyse. A neural network architecture for figure-ground separation of connected scenic figures. *Neural Networks*, 4(6):723–742, 1991.

[6] L. Herault and R. Horaud. Figure-ground discrimination: A combinatorial approach. *IEEE Trans. PAMI*, 15(9):899–914, 1993.

[7] P.L. Rosin. Edges: Saliency measures and automatic thresholding. *Machine Vision and Applications*, 9(4):139–159, 1997.

[8] T.J. Sejnowski and G.E. Hinton. Separating figure from ground with a Boltzmann machine. In M. A. Arbib and A. R. Hanson, editors, *Vision, Brain, and Cooperative Computation*, pages 703–724. MIT Press, 1987.

[9] M. Stricker and A. Leonardis. Figure-ground segmentation using tabu search. In *IEEE International Symposium on Computer Vision*, pages 605–610, 1995.

[10] S. Venkatesh and P.L. Rosin. Dynamic threshold determination by local and global edge evaluation. *Computer Vision, Graphics and Image Processing*, 57(2):146–160, 1995.

[11] H. Shimizu Y. Hirakura, Y. Yamaguchi and S. Nagai. Dynamic linking among neural oscillators leads to flexible pattern recognition with figure-ground separation. *Neural Networks*, 9(2):189–209, 1996.

[12] J. Zhang, J. Gao, and J. Liu. Figure-ground separation by a dynamical system. *IEEE Trans. Image Processing*, 8(1):115, 1999.