Variable-resolution Boundary Detection *

Anup Basu Manoj Jain Xiaobo Li

Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2H1

Abstract

Computer Vision algorithms have relied primarily on either uniform resolution or multi-resolution techniques. In this paper, the question we try to answer is: How can varying resolution spatially within an image help in extracting information out of pictures? Specifically, we look at the problems of character thinning and boundary following. To deal with these problems we design variable resolution (VR) masks, whose centers are windows in normal resolution, with each of the peripheral cells used to keep some information at a reduced resolution. In this way, VR approaches effectively look at a large region of the original image at a cost only slightly higher than processing a small region using uniform resolution schemes. The proposed algorithms are demonstrated to perform better than other well-known methods. The VR procedures described here are inherently parallel in nature. They can also be efficiently implemented on a serial computer.

1 Introduction

The human visual system has a high resolution fovea and a low resolution periphery. The central viewing window allows detailed identification of objects while the peripheral low resolution field allows fast processing. In this paper, we design boundary following algorithms by proposing a variable-resolution approach to this problem.

The detection of object contours is one of the most important operations in the field of computer vision and pattern recognition. The contours of the objects link the edges, in the relatively featureless edge image, to give meaningful boundaries. The problem has been extensively dealt with in the literature, reflecting both the importance attached to the use of contours as image features and the difficulty of achieving an optimal method. Due to the presence of noise in the image, the task of locating a meaningful boundary becomes non-trivial. For example, the edge elements can exist in the absence of any boundary and may be absent at places where the boundary should be. These missing or extra edges make the task of boundary detection still more complicated.

Even though anthropomorphic visual systems and sensors have been fairly well studied and understood [11, 13, 10, 12, 16], their application to various vision problems still needs to be investigated. Here we consider an extremely simplified version of variable resolution. Specifically, we consider templates having two levels of resolution: a normal resolution 3×3 center, and a low resolution periphery of 8 cells, each of which represents a 3×3 region in normal resolution. The peripheral cells are used to take a "rough" view of a large neighborhood (a 9×9 window). This enables us to develop parallel algorithms which are also shape-preserving. Note that a short summary of the thinning algorithms has been presented in [7]. However, the previous implementation considered only vertical stroke preservation, and also lacked detailed experimental results and analysis.

The remaining portion of this work is organized as follows: Section 2 describes previous work in boundary detection. The basic concept behind the VR mask used here is discussed in Section 3. Section 4 outlines how VR techniques can be used to develop better boundary detection algorithms. Detailed experimental results and comparison with existing methods are presented in Section 5.

2 Previous work

Several boundary detection algorithms have been proposed by previous researchers. These methods, inherently sequential in nature, can be broadly classified into two categories: algorithms which incorporate *a priori* knowledge in order to link the edges to give a meaningful representation, and techniques which do not rely on prior information in the linking step. Thus a boundary detection algorithm can be categorized based on the amount of knowledge incorporated into it. The term *a priori* knowledge means implicit or explicit constraints on the likelihood of a given grouping [2].

Most of the boundary following algorithms have two objectives: to measure the degree of "edgeness" at a particular image point, and to link the groups of points to obtain closed contours. For clean images most of the existing algorithms are able to obtain closed boundaries of the objects in the image. However when the image is noisy, the task of locating a good boundary becomes difficult, and often disconnected contours are obtained. Many techniques

*This work is supported in part by the Canadian Natural Sciences and Engineering Research Council under Grants OGP 0105739 and OGP9198.

proposed in the literature use sequential segmentation which attempts to link together all the single contour elements to form either the whole object outline or a significant part of it. To achieve this, the algorithms use some amount of prior knowledge that maps the edge elements into meaningful boundaries. Ashkar and Modestino [1] list four approaches of incorporating *a priori* knowledge in sequential segmentation algorithms. These approaches are: exhaustive search, dynamic programming, structured tree search, and heuristics graph search.

Several researchers [5, 6] have used the contour following algorithm to detect boundaries. Contour following algorithms for gray-level image use edge strength and edge direction information to compute the next boundary point. Lacroix [6] used Kunt's contour following algorithm [5] and incorporated some *a priori* knowledge in the method to improve its performance. A test on the length of contours so obtained is used to remove short meaningless contours. The algorithm performs satisfactorily for less noisy images, but tends to give erroneous results if the image is contaminated with noise. Liu [8] proposed a boundary detection algorithm with a feedback loop for backtracking. The feedback loop was incorporated into the algorithm so that the results could be checked and refined using feedback whenever necessary. Backtracking is used whenever uncertainty arises in locating the next boundary pixel. This procedure enables the algorithm to recover from errors caused by noise, but the method failed to give good results when the quality of images was poor. Chen and Siy [3] improve this algorithm by using feedback to locate the noisy areas of the image, which are smoothed to remove irregularities. The feedback mechanism is activated only when the contours obtained by the conventional algorithm are not closed. The process continues until all the contours obtained are closed. The algorithm, tested on a 60×64 chromosome image, gave satisfactory results.

William and Shah [14] used the concept of scale space [15] to obtain contours. Contours are first detected at higher scales. If a closed contour is not obtained by this process then the next finer scale is chosen and the procedure repeated until the contour cannot be extended at the finest scale. The use of finer scales produces improved detection of weak edges and helps in obtaining closed boundaries.

All the above techniques are inherently sequential in nature and require extensive computational time. A non sequential technique for determining the boundaries is described in [9, 4]. The method incorporates knowledge in terms of a curve representing a typical shape (model) of the contour. The final contour is obtained by minimizing a function, radial inertia, defined over the gradient of the object image. Though the method is able to obtain closed contours, it requires a good estimation of parameters like centroid and orientation of the model. Errors in these parameters result in poor solutions.

3 Variable-resolution masks

We propose variable resolution approaches to improve current boundary detection algorithms. The basic concept behind our method is the variable resolution mask. These masks (templates) are partitioned into 9 equal square-shaped parts as shown in Figure 1. The center part is a regular 3×3 template. Each one of the eight peripheral parts also covers a 3×3 region, but at a reduced resolution. Thus the resulting 9×9 window is called a variable resolution (VR) window.

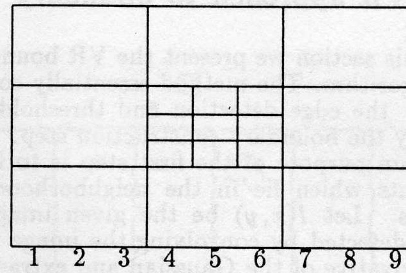

Figure 1: A variable resolution template

The VR masks are used as follows: First the high resolution center is examined. Then, if certain conditions are satisfied, some of the low resolution peripheral cells are used for further testing. Variable resolution templates have two advantages. First, the peripheral cells can be used to obtain a "rough" look at a large neighborhood around a given pixel. Second, the computational cost involved in the process is only marginally higher than the cost of using small uniform resolution templates. It is also important to note that the techniques described here are inherently parallel. That is, the decision taken in one neighborhood is completely independent of the decisions made in any of the adjacent regions.

4 Boundary detection algorithm

4.1 Limitations of previous approaches

There are two major drawbacks in most previous algorithms for boundary detection. First, they are inherently sequential in nature. Processing starts at a given pixel (or a set of pixels) and proceeds one pixel at a time according to a specified set of rules. This implies that similar operations cannot be performed simultaneously all across the image. The VR approach on the other hand is inherently parallel. The decision made at any pixel does not depend on the actions taken at the neighboring pixels. Thus we can process all image pixels in parallel without affecting the output. The second limitation of several boundary following methods lies in their inability to take a "rough look" at a large neighborhood of a pixel, while processing using a small mask. This shortcoming can result in noisy contours being detected, as well as meaningful contours being dropped. The VR algorithm is designed to avoid this drawback. Noisy contours are largely ignored by our method, which can also fill in small missing gaps in significant boundaries.

4.2 A VR approach to boundary following

In this section we present the VR boundary detection algorithm. The method essentially consists of two steps: the edge detection and thresholding step followed by the boundary construction step.

The main purpose of the first step is to highlight those points which lie in the neighborhood of the boundaries. Let $I(x,y)$ be the given image. The edges are detected by convolving the image with the second derivative of the Gaussian and extracting the zero crossings. An adaptive thresholding technique is used to delete the zero crossings with very low gradient strength. The advantage of using an adaptive threshold is that it is relatively independent of the quality of image. The pixels which survive thresholding are assigned weights depending on the gradient magnitude and gradient direction. For example, a pixel is assigned the maximum weight if it has a high edge strength and a low direction difference. In the second step, we link the pixels together, to get the meaningful boundaries, using the VR boundary detection algorithm. The VR boundary detection algorithm can be described by the following pseudocode:

procedure VR_Boundary_Detection

```
for all image neighborhoods  do

if ( center pixel has maximum weight)
  {
  check 3x3 neighborhood for edge in
  vertical, horizontal and other directions;
  if (no edge is found in the small window)
    {
    take a ''rough'' look at appropriate
                         peripheral cells;
    if (edge present in larger window)
      Mark appropriate pixels in
      smaller window as edge pixels
       to get a continuous boundary;
    }
  }
```

For the purpose of assigning weights, the edge image was thresholded to a three-valued image depending on the edge strength. We used an adaptive threshold to classify a certain percentage of pixels as *High*, *Medium*, and *Low*. Pixels in the percentile ranges [0,25], [25,60], [60,100] are classified as low, medium, and high, respectively. Extensive experimental results suggest the use of the above thresholds. Similarly, the gradient angle is digitized into 16 different directions. The total weight is defined as the sum of the following:

1. Strengths classified as *High*, *Medium*, and *Low* are assigned weights 3, 2, and 1, respectively.
2. A weight of 3 is assigned if the difference in gradient direction of the two adjacent pixels is less than 45 degrees. Similarly a weight of 2 is assigned if the direction difference is between 45 degrees and 90 degrees, a weight of 1 is assigned if it is between 90 degrees and 180 degrees, and a weight of zero is assigned for direction difference greater than 180 degrees.

Based on this scheme the maximum weight a pixel can have is 6 (it has a high edge strength and a low direction difference) and the minimum weight is 1 (it has a low strength and a high direction difference). The algorithm uses the high resolution central window to detect the boundary pixels in all the 16 possible directions. Only if it is unable to decide which boundary pixels are suitable, does it look at the outside low resolution periphery.

The information in the peripheral cells is reduced in a manner similar to VR thinning. By having a rough look at a larger neighborhood, we can fill in the small gaps and also avoid small noisy contours. The method is inherently parallel as each window is independent and can therefore be processed simultaneously. The source code in C for both the VR thinning and VR boundary detection algorithms is available on request from the authors.

5 Experimental results

Two-dimensional images were generated for testing the boundary detection algorithm. Test images were prepared using additive Gaussian noise at varying levels. The test images contained 256 gray levels and had objects of both regular and irregular shapes. In addition to the synthetic test images, the algorithm was tested on several real images to verify its robustness. All the images were 256 × 256 pixels large.

Both test images and real images were subjected to two versions of the boundary detection algorithm. The first version utilized only the mean gradient magnitudes in the computation of costs while the second version combined mean gradient magnitude and gradient direction. Both versions of the algorithm were tested with various values of σ and different thresholds. A value of $\sigma = 2$ was used for all the images.

In all, the algorithm was tested on nearly 100 different images. Typical results obtained are shown in the figures. It was observed that version 2 of the algorithm, incorporating both the gradient direction and the gradient magnitude, gave slightly better results on images with very high levels of noise ($\eta \geq 32$). For comparison with our method, two previous algorithms [14, 6] were chosen. The first module of the algorithm given in [6] was implemented using the local operator with unweighted masks, as described in the paper. In [6], a test on the length of the contour was performed to delete edges less than 4 pixels in length. We implemented the algorithm given in [6] without performing the aforementioned test in order to maintain compatibility with our algorithm which retains short contours. The source code for the other algorithm [14] was obtained from its authors. The same values of σ mentioned in the paper [14] were used. However, a different threshold of 0.25 had to be applied as the algorithm failed to give good results with a threshold of 0.08 (as stated in [14]).

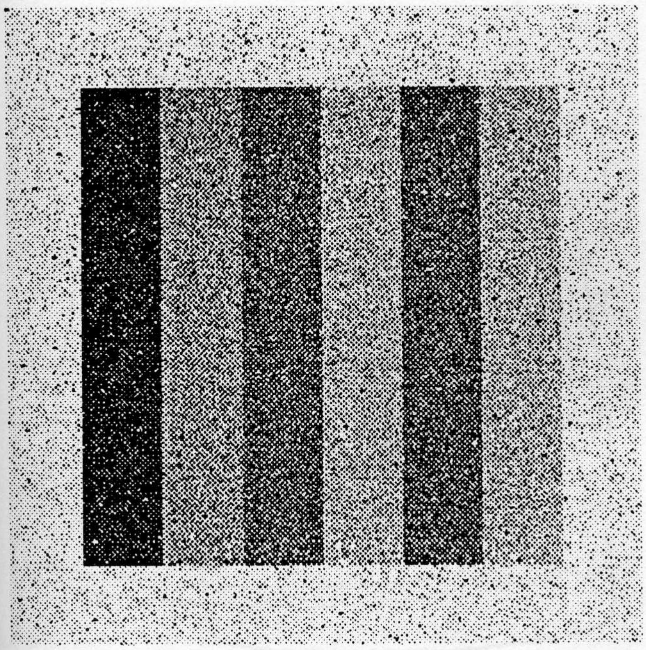

Figure 2: A set of columns of different grey levels.

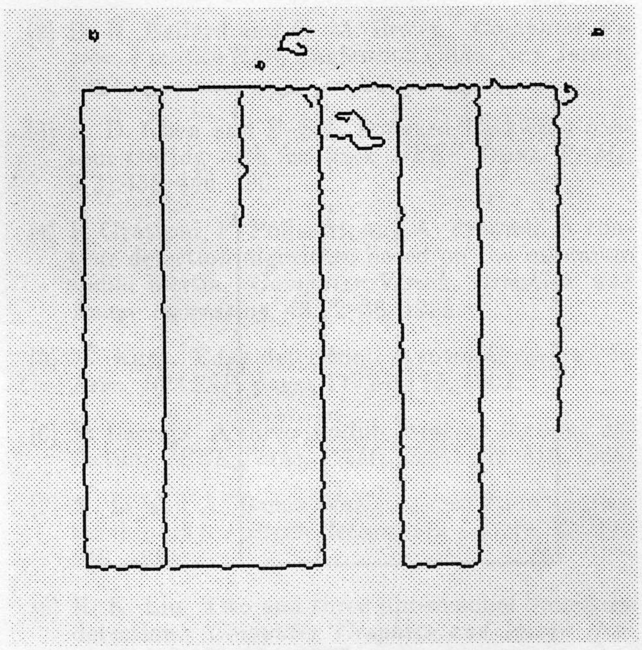

Figure 3: Contours using Williams' method.

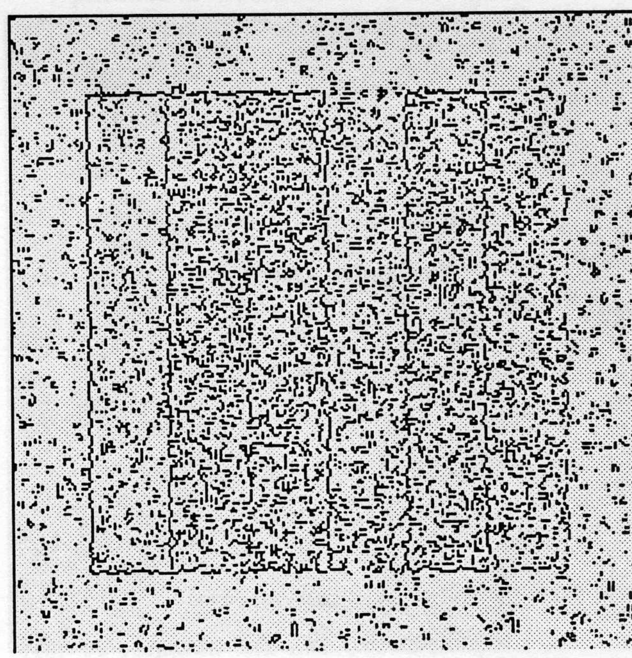

Figure 4: Contours using Lacroix's method.

Figure 2 shows the artificial image of a gray-level band, with noise having mean (μ) = 0 and standard deviation (η) = 32. For a clean image with noise of $\mu = 0$ and $\eta = 0$, all the algorithms were able to extract perfect boundaries. However as the noise level increased in the image, the boundaries obtained by the Lacroix and Williams methods did not give straight lines. On the other hand, the VR method was still able to obtain straight lines for even high noise levels ($\eta \geq 16$). For $\eta \geq 4$, Williams' method was unable to obtain a closed boundary. For noise levels greater than $\eta = 16$, both Williams' and Lacroix's methods gave poor results whereas the VR algorithm still retained all the boundaries. For very noisy images, the VR method obtained better results than the other two methods. Here the VR method was still able to obtain a sufficiently large number of contours unlike Williams' method. Lacroix's method generated several noisy contours in addition to the meaningful ones.

Figure 6 shows the image of a cup. The figure has several well-defined edges. Figure 9 shows the result of the VR algorithm.

The cup image as shown in Figure 6 was also subjected to Williams' and Lacroix's methods. The results obtained are shown in Figures 7 and 8. Lacroix's method failed completely on this image as several double contours were obtained. This happened because the number of pixels retained after thresholding was quite large and thus resulted in redundant contours — one drawback of the Lacroix method which we observed in all the images. The quality of the boundaries depended directly on the threshold. Though the paper claimed the use of an adaptive threshold where only x

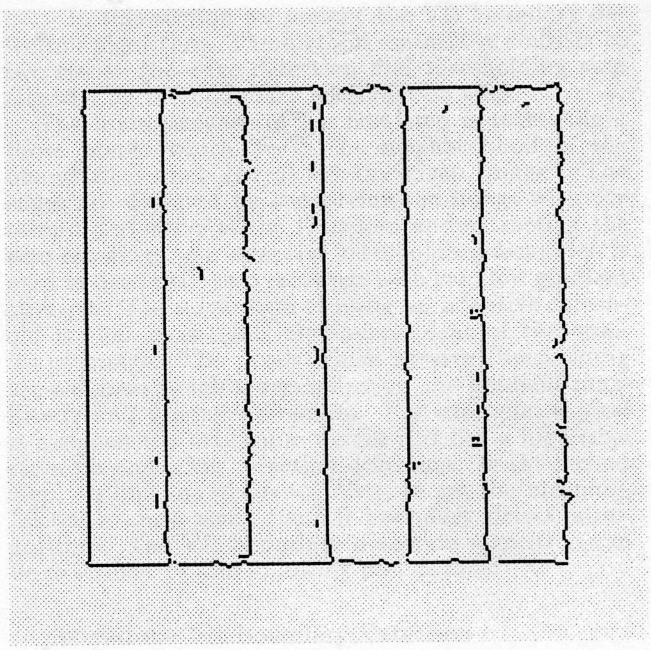

Figure 5: Contours using VR method.

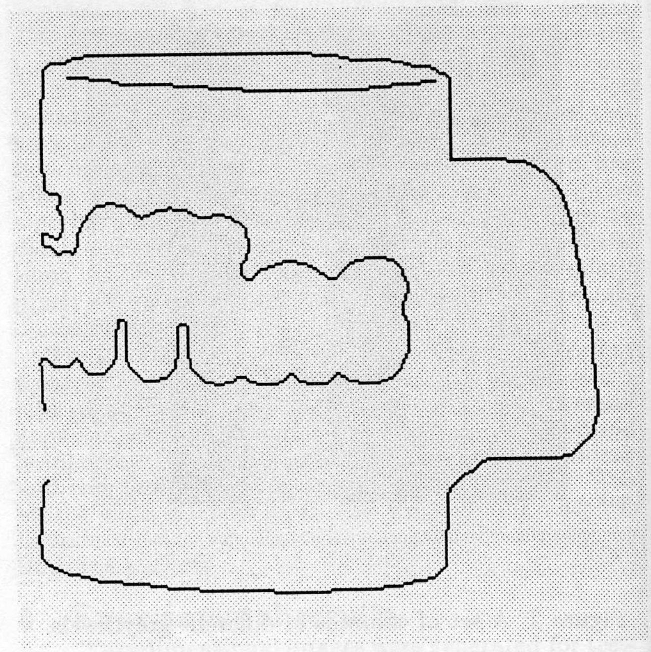

Figure 7: Contours using Williams' method.

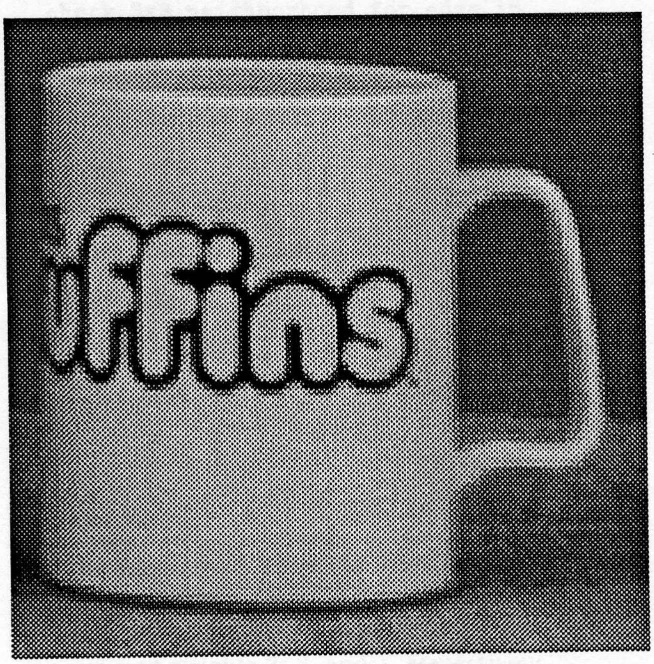

Figure 6: Original image of a cup.

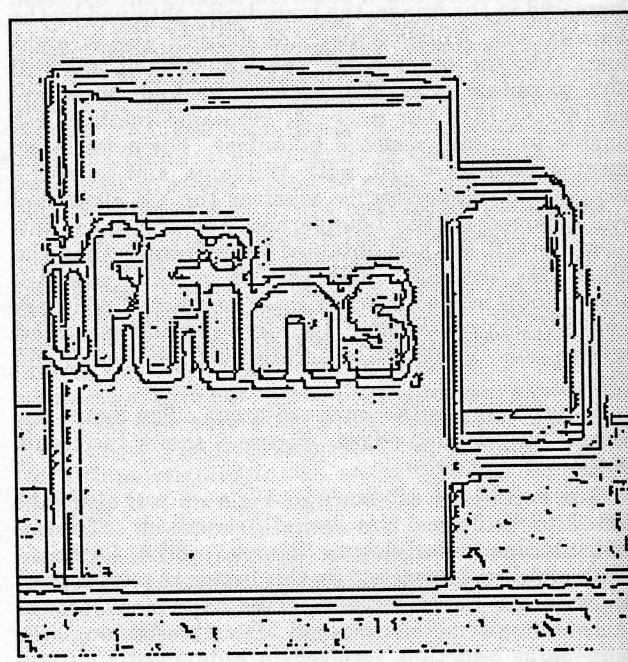

Figure 8: Contours using Lacroix's method.

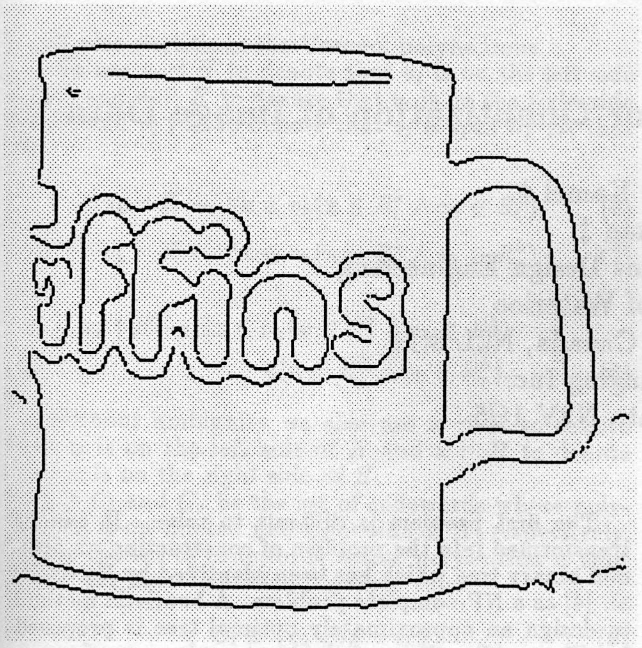

Figure 9: Contours using VR method.

percent of the pixels were retained after the edge detection phase, the value of x depended entirely on the quality of image. Most of the images gave good results when only 10-15 percent of the pixels were retained. However in some images it resulted in loss of information while in others, for example in the cup image, it resulted in double contours. Williams' method gave good results on this image though it missed out some of the details.

6 Conclusion and Acknowledgement

In this paper we present an edge and corner preserving and boundary detection scheme using a new variable-resolution approach. Our method takes a "rough" view of the peripheral region outside the central neighborhood when certain conditions exist in the inner window. The algorithms are parallel in nature, unlike methods based on contour following, which are inherently sequential. On average, the complexity of our scheme is only a few times higher than methods using small uniform resolution windows, even when implemented on a serial computer. Extensive experiments demonstrate that, compared to the best known existing techniques, the VR scheme obtains improved boundaries, as well as better outlines for thinned characters. The application of variable resolution techniques to other problems, such as vergence control and robot navigation, is currently being investigated.

The authors are grateful to D. J. Williams and M. Shah for providing the source code for *multiscale edge contour algorithm* given in [14].

References

[1] B. P. Ashkar and J. W. Modestino. The contour extraction problem with biomedical applications. *Computer Graphics and Image Processing*, 7:331–355, 1978.

[2] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall, Englewood Cliffs, New Jersey, 1982.

[3] B. D. Chen and P. Siy. Forward/backward contour tracing with feedback. *IEEE Trans. PAMI*, 9(3):438–446, 1987.

[4] P. Grattoni, F. Pollastri, and A. Premoli. A contour detection algorithm based on the minimum radial inertia. *Computer Vision, Graphics, and Image Processing*, 43:22–36, 1988.

[5] M. Kunt. Edge detection: A tutorial review. In *Proc. ICASSP*, pages 1172–1175, 1982.

[6] V. Lacroix. A three-module strategy for edge detection. *IEEE Trans. PAMI*, 10(6):803–810, 1988.

[7] X. Li and A. Basu. Variable-resolution character thinning. *Pattern Recognition Letters*, 12:241–248, 1991.

[8] H. K. Liu. Two and three dimensional boundary detection. *Computer Graphics and Image Processing*, 6:123–134, 1977.

[9] A. Premoli, P. Grattoni, and F. Pollastri. A nonsequential contour detection with a priori knowledge. *Pattern Recognition Letters*, 9:45–51, 1989.

[10] J. Rovamo and V. Virsu. An estimation and application of the human cortical magnification factor. *Experimental Brain Research*, 37:495–510, 1979.

[11] G. Sandini and V. Tagliasco. An anthropomorphic retina-like sensor for scene analysis. *Computer Graphics and Image Processing*, 14(3):365–372, 1980.

[12] E. L. Schwartz. Computational anatomy and functional architecture of striate cortex: A spatial-mapping approach to perceptual coding. *Vision Research*, 20:645–670, 1980.

[13] V. Virsu and J. Rovamo. Visual resolution, contrast sensitivity, and the cortical magnification factor. *Experimental Brain Research*, 37:475–494, 1979.

[14] D. J. Williams and M. Shah. Edge contours using multiple scales. *Computer Vision, Graphics, and Image Processing*, 51:256–274, 1990.

[15] A. Witkin. Scale-space filtering. In *Int'l Joint Conf. AI*, pages 1019–1021, 1983.

[16] Y. Yeshurun and E.L. Schwartz. Shape description with a space varying sensor: Algorithms for scan-path, fusion, and convergence over multiple scans. *IEEE Trans. PAMI*, 11:1217–1222, 1989.