photographs. PhD thesis, Computer Science Department, SUNY at Buffalo, August 1992.

[92] Shriram Revankar and David Sher. Constrained contouring in polar coordinates. In *Proceedings of CVPR'93, Omni Park Central – New York City*, 1993.

[93] Shriram Revankar, David Sher, Valerie Shalin, and Maya Ramamurthy. Supervised interpretation of echocardiograms with a psychological model of expert supervision. In *Proceedings of the IS&T/SPIE International Symposium on Electronic Imaging Science and Technology, February'93*, 1993.

[94] David Sher, Shriram Revankar, Valerie Shalin, and Maya Ramamurthy. A multimodal system for interpreting echocardiograms. In *Proceedings: Workshop on Next Generation Information Technologies and Systems (NGITS' 93), Haifa, Israel June 28-30*, 1993.

[83] Berthold K.P. Horn. Understanding image intensities. *Artificial Intelligence*, 8:201–231, 1977.

[84] James F. Havlice and Jon C. Taenzer. Medical ultrasonic imaging: An overview of principles and instrumentation. *Proceedings of the IEEE*, 67(4):620–641, April 1979.

[85] Shriram Revankar and David Sher. Maximal detection of myocardium in echocardiograms for supervised refinement. In *Proceedings of the IEEE Eighteenth Annual Northeast Bioengineering Conference*, March, 13-14 1992.

[86] Shriram Revankar and David Sher. Supervised binarization of echocardiograms by adaptive thresholding. Technical Report 92-11, Department of Computer Science, SUNY at Buffalo, 1992.

[87] Shriram Revankar. Computer drafting of hand drawn line sketches. Master's thesis, Department of Computer Science and Engineering, Indian Institute of Technology, Madras, India, August 1987.

[88] Hassan Ghassemian and David Landgrebe. An unsupervised feature extraction method for high dimensional image data compaction. In *Proceedings, IEEE Conference on System, Man and Cybernetics*, pages 540–544, 1987.

[89] N. S. Friedland and A. Rosenfeld. Compact object recognition using energy-function-based optimization. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, PAMI-11(7):770–777, July 1992.

[90] Terzopoulos D. and Fleischer K. Deformable models. *The Visual Computer*, 4(6):306–331, 1988.

[91] Venugopal Govindraju. *A Computational theory for locating human faces in*

[74] Theo Pavlidis and Y.T. Liow. Integrating region growing and detection. In *Proceedings the International Conference on Computer Vision and Pattern Recognition*, 1988.

[75] J.P. Gambotto. A parallel and hierarchical algorithm for region growing. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 649–652, 1985.

[76] Leonard Zusne. *Visual Perception of Form*, chapter Contemporary theory of visual form perception:I. The psychology of contour perception, pages 16–20. Academic Press, 1970.

[77] A. Rosenfeld and A.C. Kak. *Digital Picture Processing*. New York: Academic Press, 1976.

[78] A. Martelli. Edge detection using heuristic search methods. *Computer Graphics and Image Processing*, 1:169–182, 1972.

[79] A.K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, 1989.

[80] Amir A. Amini, Terry E. Weymouth, and Ramesh C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 12(9):855–867, September 1990.

[81] Donna J. Williams and Mubarak Shah. A fast algorithms for active contours and curvature estimation. *CVGIP, Image Understanding*, 55(1):14–26, January 1992.

[82] Alan Yuille, David Cohen, and Peter Hallinan. Facial feature extraction by deformable templates. Technical Report 88-2, Harvard Robotic Laboratory, 1988.

[65] T. Pavlidis. Segmentation of pictures and maps through functional approximation. *Computer Graphics and Image Processing*, 1:360–372, 1972.

[66] S.L. Horowitz and T. Pavlidis. Picture segmentation by directed split and merge procedure. In *Proceedings: Second International Joint Conference on Pattern Recognition*, pages 424–433, 74.

[67] J.N. Gupta and P.A. Wintz. Computer processing algorithms for locating boundaries in digital pictures. In *Proceedings of the Second International Joint Conference on Pattern Recognition*, pages 155–156, 1974.

[68] Robert M. Haralick and Linda G. Shapiro. Survey: Image segmentation techniques. *Computer Vision, Graphics and Image Processing*, 29:100–132, 1985.

[69] T. Aasano and N. Yokoya. Image segmentation schema for low-level computer vision. *Pattern Recognition*, 14:267–273, 1981.

[70] Y. Yakimovsky. Boundary and object detection in real world image. *Journal of Association of Computing Machinery*, 23:599–618, 1976.

[71] M.D. Levine and S.I. Shaheen. A modular computer vision system for picture segmentation and interpretation. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, PAMI-3, 1981.

[72] P Zamperoni. Analysis of some region growing operators for image segmentation. In *Proceedings of the International Conference on Advances in Image Processing and Pattern Recognition*, 1986.

[73] H.M. Rafaat and A.K.C. Wong. A texture information directed region growing algorithm for image segmentation and region classification. *Computer Vision, Graphics and Image Processing*, 43:1–21, 1988.

[55] N. Ahuja, A. Rosenfeld, and R.M. Haralick. Neighborhood graylevel as feature in pixel classification. *Pattern Recognition*, 12:251–260, 1980.

[56] J. Kittler and J. Foglein. Contextual classification of multispectral pixel data. *Image and Vision Computing*, 2(1):13–29, February 1984.

[57] J. Hyde, J.A. Fullwood, and D.R. Corall. An approach to knowledge driven segmentation. *Image and vision computing*, 3(4):198–205, November 1985.

[58] P. Zamperoni. Model based segmentation of graytone images. *Image and vision computing*, 2(3):123–133, August 1984.

[59] David Sher, Shriram Revankar, and Steven Rosenthal. Computer methods in quantitation of cardiac wall parameters from two dimensional echocardiograms: A survey. *International Journal of Cardiac Imaging*, 8:11–26, 1992.

[60] R.W. Smith. Computer processing of line images: a survey. *Pattern Recognition*, 20(1):7–15, 1987.

[61] M.D. Levine and A.M. Nazif. Dynamic measurement of computer generated image segmentation. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, PAMI-7:652–655, 1985.

[62] Steven Zucker. Region growing: Childhood and adolescence. *Computer Graphics and Image Processing*, 5:382–399, 1976.

[63] J.L. Muerle and D.C. Allen. Experimental evaluation of techniques for automatic segmentation of objects in a complex scene. In G.C. Cheng *et al.*, editor, *Pictorial Pattern Recognition*, pages 3–13. Thompson, Washington, 1968.

[64] C. Brice and C. Fennema. Scene analysis using regions. *Artificial Intelligence*, 1:205–226, 1970.

IEEE SMC, SMC-9(1):38–52, January 1979.

[46] S. Watanabe and CYBEST group. An automated apparatus for cancer prescreening: Cybest. *Computer Graphics and Image Processing*, 3:350–358, 1974.

[47] J.S. Weszka, J.A. Verson, and A. Rosenfeld. Threshold selection techniques. Technical Report 260, University of Maryland, Computer Science Center, 1973.

[48] J.R. Ullman. Binarization using associative addressing. *Pattern recognition*, 6:127–135, 1974.

[49] R.N. Wolfe. A dynamic thresholding technique for quantization of scanned images in automatic pattern recognition. *National Security Industrial Association, Washington D.C.*, pages 143–162, May 1979.

[50] H.Ogawa and K. Taniguchi. Thinning and stroke segmentation for hand written chinese character recognition. *Pattern recognition*, 15(4):298–308, 1982.

[51] L.T. Watson, K. Arvind, R.W. Ehrich, and R.M. Haralick. Extraction of lines and regions from graytone line drawing images. *Pattern recognition*, 17(5):493–507, 1984.

[52] T.W. Ridler and S. Calvard. Picture thresholding using iterative thresholding method. *IEEE SMC*, SMC-8(8):629–632, August 1978.

[53] Z.M. Wojcik. A natural approach in image processing and pattern recognition: Rotation neighborhood technique, self adapting threshold, segmentation and shape recognition. *Pattern Recognition*, 18(5):299–326, 1985.

[54] C.K. Chow and T. Kaneko. Boundary detection of radiographic images by a threshold method. In *Proceedings IFIP Congress - 71, Booklet - TA-7*, pages 130–134, North Holland, Amsterdam, 1972.

for intuitive user interfaces. In *IEEE International Conference on System Man and Cybernetics, Charlottesville, Virginia*, pages 1109–1114, 1991.

[36] R.D. Riecken. Adaptive direct manipulation. In *IEEE International Conference on System Man and Cybernetics, Charlottesville, Virginia*, pages 1115–1120, 1991.

[37] Erika Rogers and Ronald Arkin. Visual interaction: A link between perception and problem solving. In *IEEE International Conference on System Man and Cybernetics, Charlottesville, Virginia*, 1991.

[38] Joan S. Weszka. A survey of threshold selection techniques. *Computer Graphics and Image processing*, 7:259–265, 1978.

[39] W. Doyle. Operations useful for similarity invariant pattern recognition. *JACM*, 9:259–267, 1962.

[40] J.M.S. Prewitt and M.L. Mendelson. Analysis of cell images. *Annals of NY Acad. Sci.*, 128:1035–1053, 1966.

[41] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on System Man and Cybernetics*, SMC-9(1):62–66, 1979.

[42] K. Fukunaga. *Introduction to statistical pattern Recognition.* New York Academic Press, 1972.

[43] J.S. Weszka, R.S. Nagel, and A. Rosenfeld. A threshold selection technique. *IEEE Transactions Computers*, 23:1322–1326, 1974.

[44] D.P. Panda and A. Rosenfeld. Image segmentation by pixel classification in graylevel gradient space. *IEEE Transactions Computers*, 27:875–879, 1978.

[45] J.S. Weszka and A. Rosenfeld. Histogram modification for threshold selection.

[27] G.T.Toussaint. The use of context in pattern recognition. *Pattern Recognition*, 10:189–204, 1978.

[28] F.C. Billingsley. Review of digital image processing. In Douglas Lewin, editor, *Interactive Systems*, pages 293–318. Online Conferences Ltd. Uxbridge, London, 1975.

[29] T. Orhaug, J.O. Eklundh, S. Grambo, H. Marmolin, and S Nyberg. A system for digital processing of pictorial information. In Douglas Lewin, editor, *Interactive Systems*. Online Conferences Ltd. Uxbridge, London, 1975.

[30] D.A. Linkens. Interactive computer graphics analysis of digestive track medical data. In Douglas Lewin, editor, *Interactive Systems*. Online Conferences Ltd. Uxbridge, London, 1975.

[31] E.H. Shortlife. Medical consultation systems: Designing for doctors. In M.E. Sime and M.J. Combs, editors, *Design for Human Computer Communication*, pages 209–238. Academic Press, 1983.

[32] M.J. Fitters. Doctors using computers: A case study. In M.E. Sime and M.J. Combs, editors, *Design for Human Computer Communication*, pages 239–260. Academic Press, 1983.

[33] J.M. Carroll, editor. *Designing Interaction: Psychology as the Human Computer Interface*. Cambridge University Press, 1991.

[34] A.F. Norcio. Adaptive interfaces: Modeling tasks and users. In *IEEE International Conference on System Man and Cybernetics, Charlottesville, Virginia*, pages 1099–1103, 1991.

[35] M. Manhartsberger, F. Penz, and M. Tscheligi. Interaction objects - a paradigm

1985.

[18] D. LaBerge. Spatial extent of attention to letters and words. *Journal of Experimental Psychology: Human Perception and Performance*, 9:371–379, 1983.

[19] Howard C. Hughes and Lynn D. Zimba. Spatial maps of directed visual attention. *Journal of Experimental Psychology: Human Perception and Performance*, 11(4):409–430, 1985.

[20] Gordon L. Shulman, Roger W. Remington, and John P. Mclean. Moving attention through visual space. *Journal of Experimental Psychology: Human Perception and Performance*, 5(3):522–526, 1979.

[21] Howard S. Bashinski and Varne R. Bacharach. Enhancement of perceptual sensitivity as a result of selectively attending to spatial locations. *Perception and Psychophysics*, 28(3):241–248, 1980.

[22] Yehoshua Tsal. Movement of attention across visual field. *Journal of Experimental Psychology: Human Perception and Performance*, 9(4):523–530, 1983.

[23] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. In *Proceedings: IEEE 1st International Conference on Computer Vision, London England, June 8-11*, pages 259–267, 1987.

[24] P.K. Sahoo, S. Soltani, and A.K.C Wong. A survey of thresholding techniques. *Computer Vision, Graphics and Image Processing*, 41:233–260, 1988.

[25] J.S. Weszka and A. Rosenfeld. Threshold evaluation techniques. *IEEE Transactions on System Man and Cybernetics*, SMC-8:622–629, 1978.

[26] T. Pavlidis. A critical survey of image analysis methods. In *Proceeding of the International Conference on Pattern Recognition*, pages 502–511, 1986.

performance, 14(2):188–202, 1988.

[9] Shriram Revankar, David Sher, and Steven Rosenthal. Collaborative processing to extract myocardium from a sequence of two dimensional echocardiograms. In *Proceedings of the SPIE conference on Extracting Meaning from Complex Data: Processing, Display, Interaction II*, pages 268–273, February 1991.

[10] Shriram Revankar and David Sher. Constrained contouring in the polar coordinates. Technical Report TR-92-25, Department of Computer Science, State University of New York at Buffalo, October 1992.

[11] C. Marlin 'Lin' Brown. *Implementation of human-computer interface guidelines*, chapter 10, pages 171–188. Ablex Publishing Corporation, 355 Chestnut Street, Norwood NJ 07648, 1988.

[12] G. Boy. *User Interfaces*, chapter 10, pages 241–246. Academic Press, San Diego, CA 92101, 1991.

[13] William Reinhart and Richard Marken. Control system analysis of computer pointing devices. In *Proceedings of the Human Factors Society-29th Annual Meeting*, pages 119–121, 1985.

[14] M.I. Posner, C.R.R. Snyder, and B.J. Davidson. Attention and detection of signal. *Journal of Experimental Psychology: General*, 109:160–174, 1980.

[15] J.E. Hoffman and B. Nelson. Spatial selectivity in visual search. *Perception and Psychophysics*, 30:283–290, 1981.

[16] Posner M.I. *Orienting*, pages 185–215. Hillsdale, NJ: Erlbaum, 1978.

[17] C.W. Eriksen and Y.Y. Yeh. Allocation of attention in visual field. *Journal of Experimental Psychology: Human Perception and Performance*, 11:583–597,

References

[1] Edward Cohen, Jonathan Hull, and Sargur Srihari. Understanding hand written text in a structured environment: Determining zip codes from addresses. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(1-2):221–264, 1991.

[2] Peter Cullen, Tin Kam Ho, J. Hull, and S.N. Srihari. Contextual analysis of machine printed addresses. In *Proceedings of the Fourth USPS Advanced Technology Conference*, pages 779–793, November 1990.

[3] Theo Pavlidis, Jerome Swartz, and Ynjium Wang. Fundamentals of bar code information theory. *IEEE Computer*, pages 74–86, April 1990.

[4] C. Marlin 'Lin' Brown. *General human-computer interface concepts: Allocation of functions*, chapter 1, pages 4–8. Ablex Publishing Corporation, 355 Chestnut Street, Norwood NJ 07648, 1988.

[5] Jefferey C. Jackson and Renate J. Roske-Hofstrand. Circling: A method of mouse-based selection without button presses. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 161–166, 1989.

[6] C. Marlin 'Lin' Brown. *Control and Display Devices:Mouse*, chapter 8, pages 138–140. Ablex Publishing Corporation, 355 Chestnut Street, Norwood NJ 07648, 1988.

[7] G.R.R.Mangun and A. Hillyard. The spatial allocation of visual attention as indexed by event related brain potentials. *Human Factors*, 29(2):195–211, 1987.

[8] Cathryn J. Downing. Expectancy and visual-spatial attention: Effects on perceptual quality. *Journal of Experimental Psychology: Human Perception and*

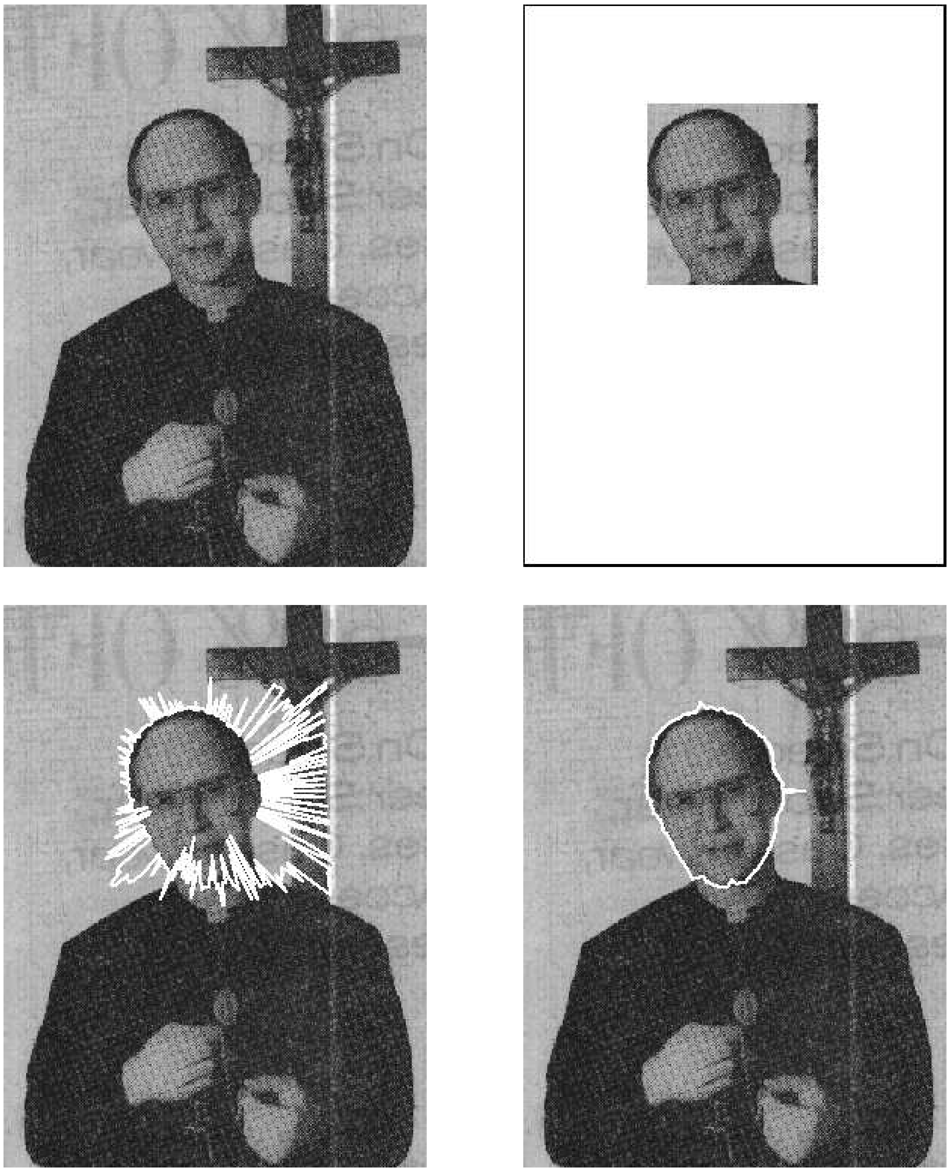


Figure A.4: Image scanned from *The Buffalo News, January 6, 1991.* The center image shows the output of Venugopal's[91] face location algorithm. The right image shows the contour trace obtained from our algorithm.

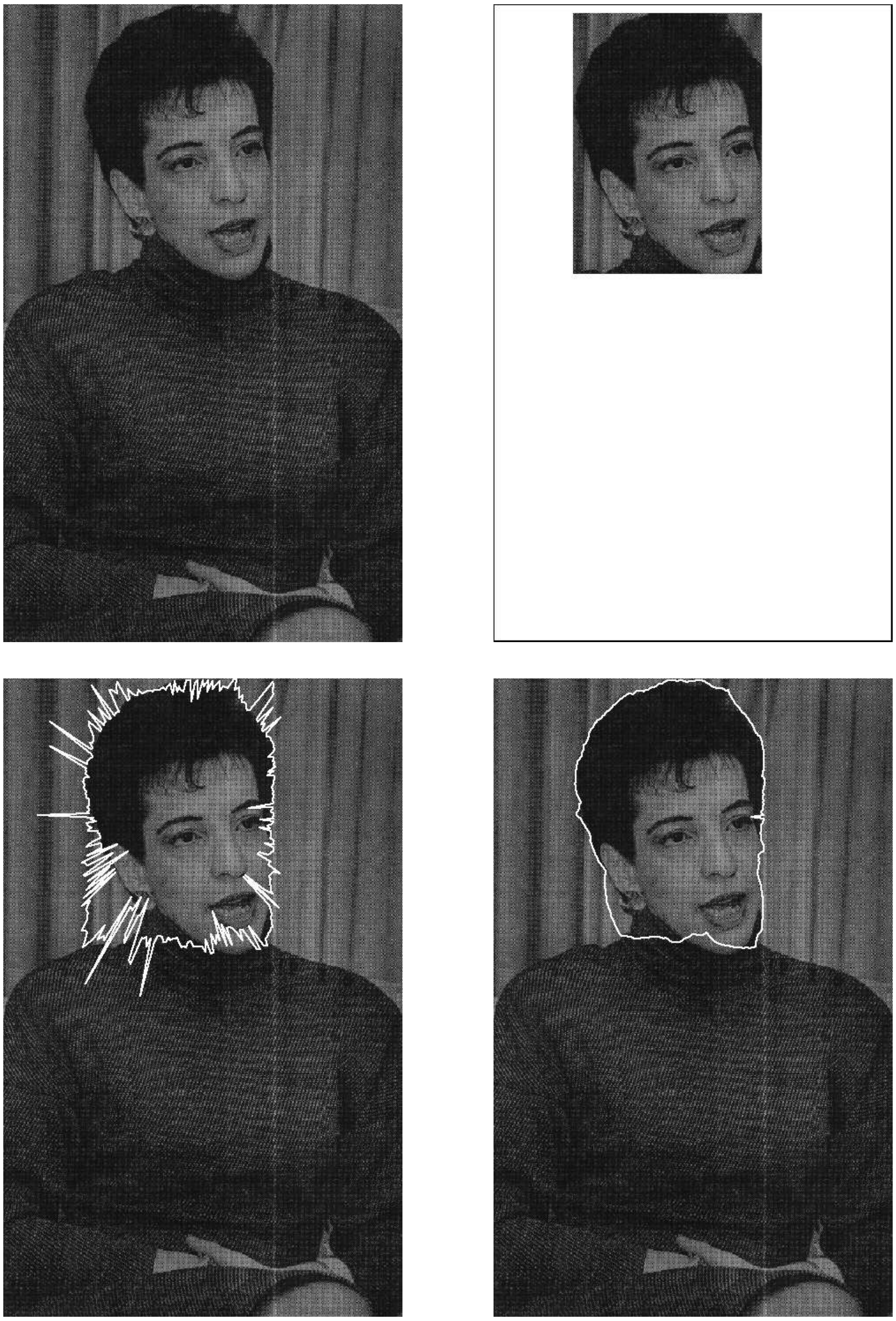


Figure A.3: Image scanned from *The Buffalo News, January 15, 1991.* The center image shows the output of Venugopal's[91] face location algorithm. The right image shows the contour trace obtained from our algorithm.

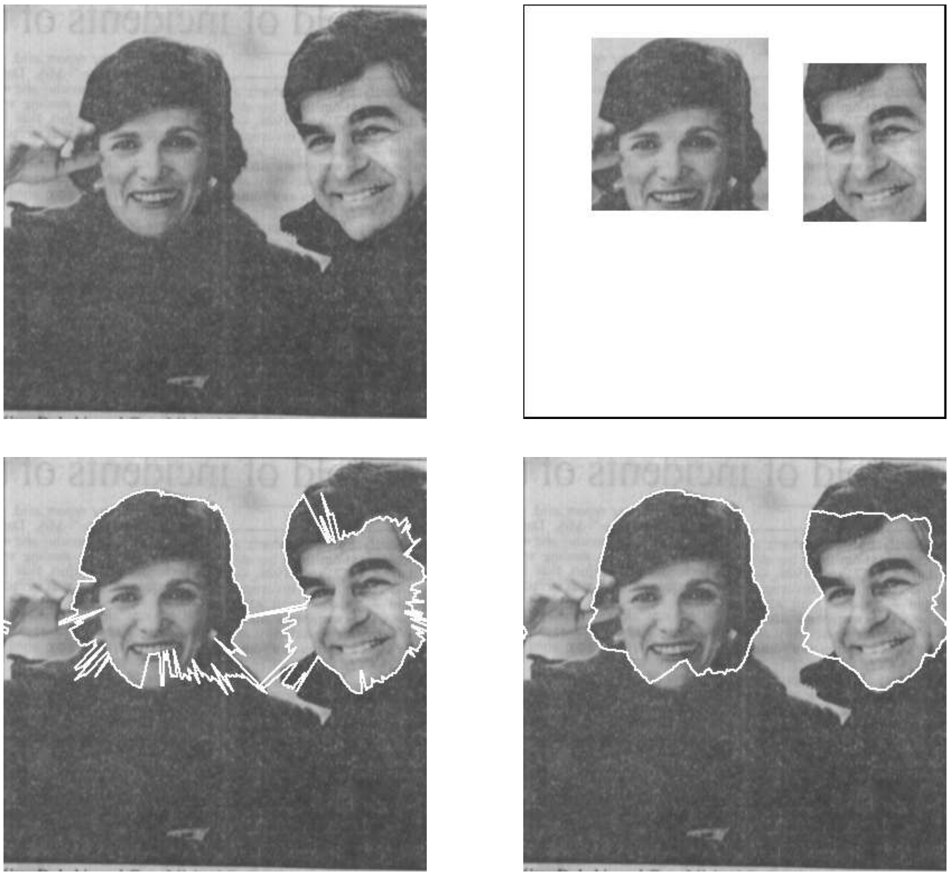


Figure A.2: Image scanned from *The Buffalo News, March 8, 1989.* The center image shows the output of Venugopal's[91] face location algorithm. The right image shows the contour trace obtained from our algorithm.

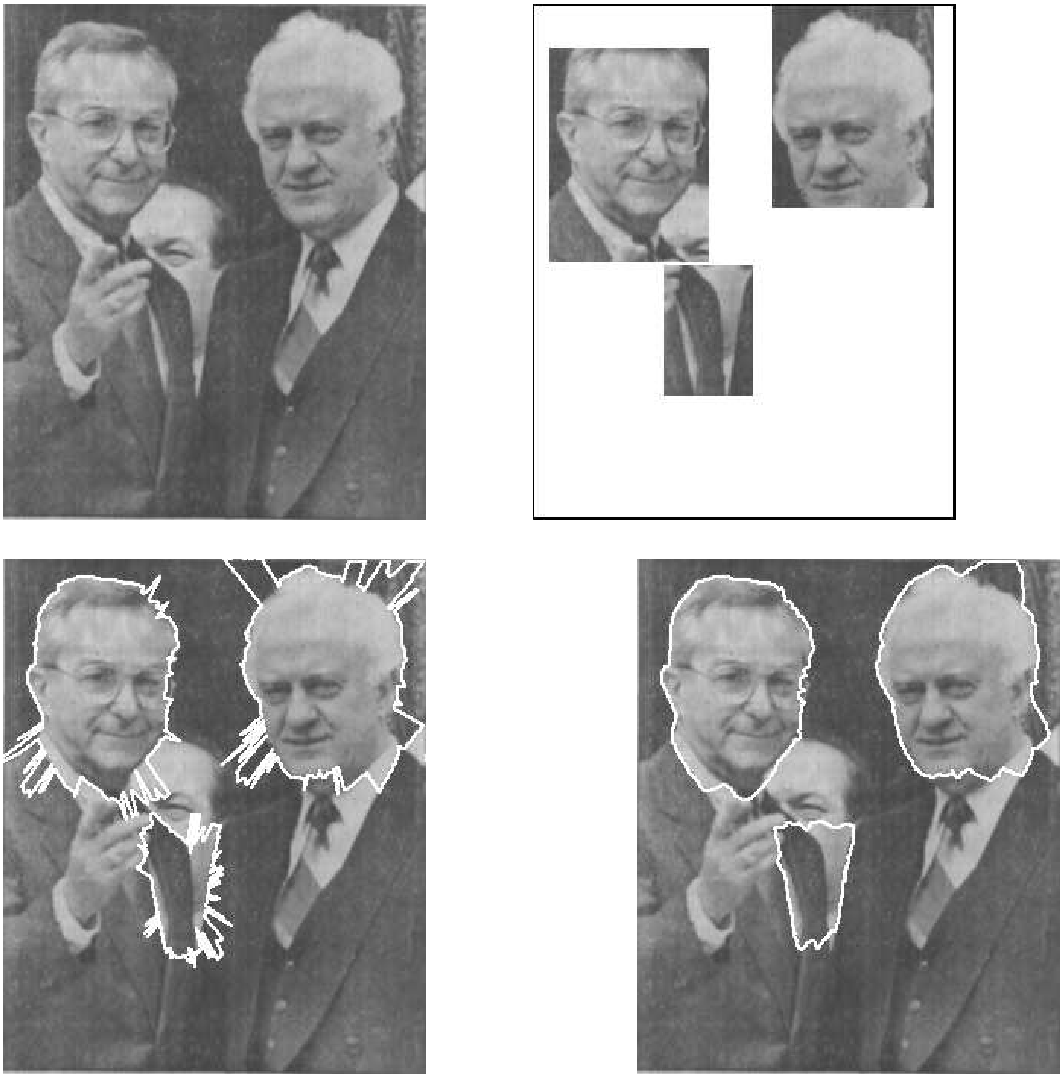


Figure A.1: Image scanned from *The Buffalo News, Feb 23, 1989.* The center image shows the output of Venugopal's[91] face location algorithm. The right image shows the contour trace obtained from our algorithm.

inal and the second image shows the approximate face locations in the original image. The face location results were generated by Venugopal for his PhD thesis [91]. The third image shows the contours generated by our constrained contouring algorithm by using the rectangular boxes as the rough input contours.

Appendix A

Contouring the Approximately Located Human Faces

In Figures A.1 through A.4 we show the results of our constrained contouring algorithm. Through these results we demonstrate the possible use of the contouring algorithm. To generate these results we have used a constant bending constant. These results may show substantial improvement if bending values are set to reflect the bendings in a human face outline. One of our future goal is to develop a scheme that would set these bending values given the outline curvature in all the radial directions.

The following test images were scanned from a news paper, and are noisy and highly textured. The effect of noise and texture can be seen on the extremely spiky contours joining the LOCLs directly. The rough input contour is a rectangle fully or partially covering the subject's face. I have selected a few example that clearly show the way our algorithm works.

Each of the following figures have four images. The first image is the origphisticated than the published interactive segmentation schemes, because the interaction are approximate natural gestures, and are mapped to the relevant image processing algorithms. The algorithms account for both the image data constraints and the approximations in the human gestures by incorporating the human visual attention map. Furthermore, all our algorithms account for the non-rigid motion inherent to ultrasound heart images, and supervisor's correction on one frame is propagated to all the previous and subsequent frames. This reduces the tedium of making the same correction in every frame.

We have already started extending our supervised segmentation scheme to facilitate more complex psychological models of visual thought process [93]. Through such a system a supervisor will be able to provide richer advise to the image analysis system. We are also working on extending this scheme to facilitate multimodal communication through speech and gesture interfaces [94]. Such an advanced collaborative scheme, augmented with suitable expert knowledge may provide a sound frame work for a wide variety of *Intelligent Active Assistance Systems*(IAAS).

rough input contour to yield the boundaries of the intended object.

Because of their widespread use and some supporting research, we chose *pointing* and *circling* as the two desirable gestures for supervised refinement of the computer generated segmentation. In the fourth chapter we describe a region growing method to process corrections by pointing. The new feature in this method is incorporating the computational model of the human visual attention map in region growing. The described region growing algorithm collects all the points of similar nature around the picked point. It accounts for various image features such as edge strengths, edge crossings and intensity values, in addition to the human visual attention map.

The circled inputs are processed by a constrained contouring algorithm. The problem here is generating a precise outline, given a rough outline of the region of interest. Several active contouring methods have been proposed to solve this problem. We provide an interesting alternative method. The novelty is mapping the energy minimization problem into a graph search problem [92]. We trace the contour in polar coordinates. The polar coordinate transformation has some inherent advantages, such as providing quadratic interpolation of missing traces, relating the roughness with the size, and avoiding unexpected behaviors such as shrinking. This process can be viewed as being equivalent to deforming a circle, and it is natural for tracing closed contours. We have promising results both on optical and ultrasound images. We have also tracked the non-rigid motion of the heart using our constrained contouring algorithm.

3. Overall, our supervised image segmentation scheme is significantly more so-

bution, while computing the measures for image segmentation.

The important contributions of this research are:

1. *Development of a image segmentation algorithm that facilitates a priori specification of width of a limb for adaptive thresholding of elongated patterns, in the presence of nonuniform illumination and background.*

   It is desirable to generate a thin and smoothly connected object patterns for processing text or line drawings. In medical image processing, the physical size of the generated pattern is also important. Our thresholding algorithm provides all these desirable characteristics, by allowing an expert to specify the width of the pattern at a region, and then propagating the threshold computed at that region to the rest of the image. In the third chapter we have described a new method for pattern extraction by adaptive propagation of a known regional threshold. A regional threshold is computed either by analyzing the supervisor suggested section of the image or by automatically selecting a threshold that generates the patterns of specified width and length. The regional threshold is then propagated over the entire image (or the image sequence of a dynamic scene), such that the ratio of threshold to slope of the intensity transition is maintained constant. The thresholded patterns are usually rough. Therefore we vary the threshold by the average gray value of the best fit morphological enclosure to yield smooth thresholded patterns. We show the outputs generated by this algorithm for both echocardiograms and optical images of hand written text and a line diagram.

2. *Development of an efficient constrained contouring algorithm, that refines a*

Chapter 5

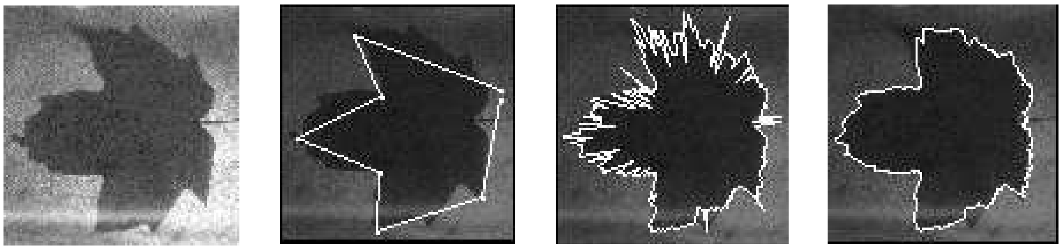
Conclusions and Future Scope

In this thesis we have developed a scheme for supervisory verification and modification of the computer generated segmentation. This supervised segmentation scheme combines the reliability of a human expert with the precision of the automatic image segmentation algorithms.

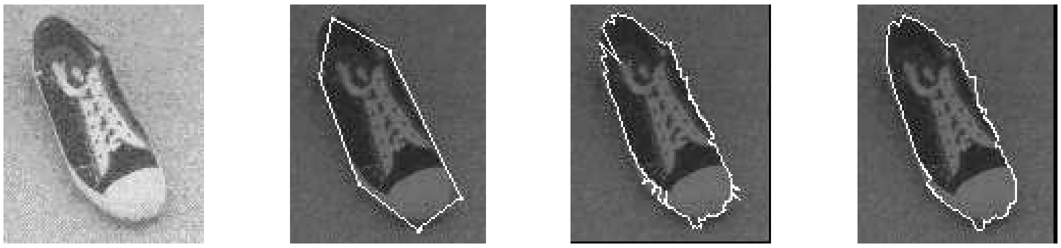
Researchers have shown that humans and machines have several complementary capabilities, but little has been published on collaborative image segmentation systems that attempt to combine those capabilities. In our system we incorporate a psychological model of human visual attention map for collaborative image segmentation. This enables an active and natural interaction between a user and the machine.

We facilitate two common modes of natural interactive gestures, namely *pointing* and *circling*. The supervisor's interactions by pointing activate a region growing algorithm, and interactions by circling activate a constrained contouring algorithm. These algorithms account for human interaction inaccuracies, and attention distrias edge strengths, edge crossings and intensity values. It also accounts for human visual attention map.

Our constrained contouring method deforms the input rough contour to trace the precise contour of the region of interest. Tracing salient contours of the regions of interest in two dimensional images is important for analyzing shape, size and motion of the region. Several active contouring methods have been proposed to solve this problem. We provide an interesting alternative method. The novelty is mapping the energy minimization problem into a graph search problem [92]. We trace the contour in polar coordinates. The polar coordinate transformation has some inherent advantages, such as providing quadratic interpolation of missing traces, relating the roughness with the size, and avoiding unexpected behaviors such as shrinking. This process can be viewed as being equivalent to deforming a circle, and it is natural for tracing closed contours.



a. Stages in contouring an optical image of a leaf.



b. Stages in contouring the optical image of a shoe.

Figure 4.11: The left most column of images shows the originals and the second column shows the rough initial contour. The third column shows the trace of LOCLs, and the right most column shows the optimum contour traced by setting a constant smoothing constraint.

are outputs of Venugopal's Ph.D. dissertation on locating human faces [91].

4.5 Summary of supervised refinement

We have described a scheme for supervised refinement of image segmentation. An initial estimate of the desired segmentation is obtained by a adaptively propagated threshold and the result is refined through a gestured dialogue with an expert operator. The operator either picks points on the misclassified regions or circles them. The precise boundaries of the corresponding regions are found by using region growing based algorithm or constrained contouring algorithm.

The described region growing algorithm collects all the points of similar nature around the picked point. The algorithm accounts for various image features such diogram displaying a different cross section of the left ventricular chamber. The algorithm traces smooth chamber walls even through the valves and papillary muscles that occlude the wall contours. Further more, in the places where no gradient is found, and in the places where smoothness criterion is not met, the contour interpolates the region with smooth circular arcs, because our algorithm traces the contour with a linear interpolation of radius between two LOCLs that belong to the optimum contour.

Our algorithm can be used to trace the contours of a sequence of images of a dynamic scene. In Figure 4.10 we show the contours generated by tracing the contours of a sequence of short axis view, two dimensional echocardiograms. A rough contour is input in one of the frame. The optimum contour of that frame is used as the rough initial contour for the neighboring frames. The procedure is repeated for generating the contours in all the frames.

Contouring in optical images

In Figure 4.11 we show the results of our contouring algorithm on two optical images. Both images have various textures and grey level gradients. The left most images in each row are the original images. The second image in each row shows the rough input contour and the third images are the contour joining the LOCLs without smoothness constraints. The final images are generated by tracing the optimum contour that satisfy a constant smoothing constraint.

In Appendix A, we have used this constrained contouring to trace human faces, given the approximate location of the face by rectangular boxes. The box locations

Figure 4.8: Input contour on the original image (left), The contour generated by connecting LOCLs (middle), and the trace of the optimum contour generated by setting a constant smoothness constraint.

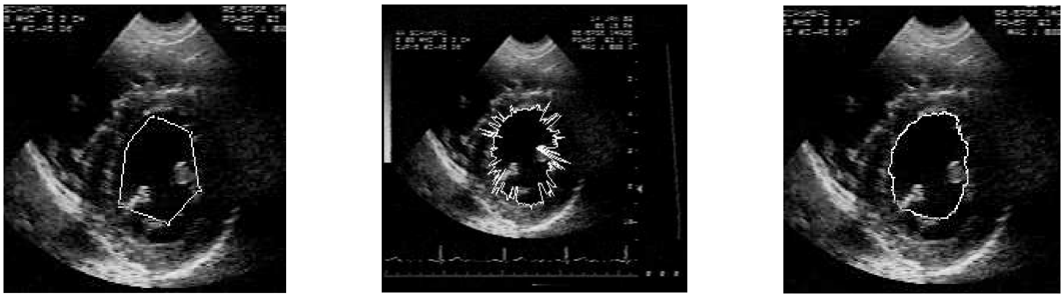


Figure 4.9: Long-axis two-dimensional echocardiograms with an input contour (left) the traced contour.

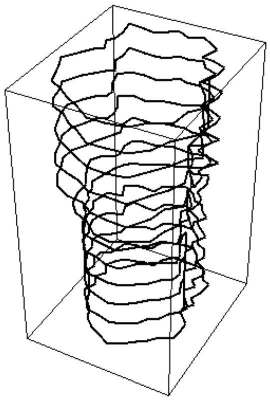


Figure 4.10: Contours of a sequence of short axis views of the left ventricle in two dimensional echocardiograms.

endfor

endfor

$P_{opt} = P_{n+1,j}$ if $W_{n+1,j} \geq W_{n+1,i}, \forall i \in$ neighborhood of q_1''

4.4.4 Results

Our algorithm provides promising results. We ran this algorithm on ultrasound images of the human heart to trace the boundaries of the heart chambers, and also on optical images of objects of varying smoothness. The displayed results are for a constant smoothing constraint τ. We are currently working on improving the results by setting the $\tau(i)$ values using *a priori* shape information.

Contouring in ultrasound images

In ultrasound images, objects of interest are brighter than the background, and the objects have smooth contours. So we set τ small so that contour traced is smooth. In addition, we need to trace either increasing or decreasing image gradients. In Figure 4.8 we show contouring of left ventricular chamber of a long-axis two dimensional echocardiogram. The left image shows the original image and the contoured image is shown in the right. The middle image shows the contour generated by simply connecting the LOCLs. Clearly, our algorithm has bypassed sharp spikes and distant LOCLs. In contrast to energy minimization based algorithms, the LOCLs that are not on the final contour, have no influence on contour location.

In Figure 4.9 we show input output pair of a short-axis two dimensional echocarspeedup. Similarly, we have a *linear time* algorithm when we know possible locations of all the LOCLs.

Optimum-Contour(V, P_{opt})

$\forall_{ij} W_{ij} = \infty$

$\forall_j W_{1,j} = 1, j$ in the neighborhood of q_1

$W_{1,j} = 0$ if $q_{1j} = q_1$

$q_{n+1} = q_1$

for each $q_i \in \{q_2, q_3, \ldots, q_n, q_{n+1}\}$

    for each q_{ij} in the neighborhood of q_i

        for each $q_{i-1,k}$ in the neighborhood of q_{i-1}

            if $|q_{ij} - q_{i-1,k}| < \tau i \cdot r_i$

                if $q_{ij} = q_i$

                    if $W_{ij} > W_{i-1,k}$

                        $W_{ij} = W_{i-1,k}$

                        $P_{ij} = append(P_{i-1,k}, \{q_{ij}\})$

                    endif

                else

                    if $W_{ij} > W_{i-1,k} + 1$

                        $W_{ij} = W_{i-1,k} + 1$

                        $P_{ij} = append(P_{i-1,k}, \{q_{ij}\})$

                    endif

                endif

        endfor

If there are two or more contours that pass through the same number of LOCLs, we pick the contour that is the smoothest. This can be achieved, by computing the sum of Sm_i values, selecting the contour with the minimum sum.

4.4.3 Algorithm modification for known discrete neighborhood

The above algorithm is of computational complexity $O(n^3)$. But, sometimes it is possible to know the range of movement for at least one radial direction. For example, if the image is of heart, knowing the dynamics of heart motion, it is possible to know the limits displacement of a wall section. If for a point q_i on the contour, we know a set of values $\{q_{i1}, q_{i2}, q_{i3}, q_{i4}, \ldots, q_{im}\}$ that it can take, then we can rewrite the optimum-contour algorithm with reduced complexity.

Optimum-Contour(G, P_{opt})

$P_{opt} = []$

length=$|P_{opt}|$

for each $q_{ij} \in \{q_{i1}, q_{i2}, q_{i3}, \ldots, q_{im}\}$

    Generate a graph G_{ij} replacing q_i by q_{ij}

    Conditional-Optimum-contour(G_{ij}, P)

    if(length $< |P_{n+1}|$)

        length $= |P_{n+1}|$

        $P_{opt} = P_{n+1}$

    endif

endfor

This is clearly a $O(n^2)$ algorithm, and for small m's this provides considerable and from the correctness of shortest path algorithm, we have $w_P \leq w_{P'}$

$\Longrightarrow (\Lambda' \leq \Lambda)$ **(contradiction)** $\square$

This algorithm is of asymptotic complexity $O(n^2)$, with n proportional to the length of the contour. The algorithm generates the optimum contour with a condition that q_1 belongs to the optimum contour, but q_1 may not belong to the optimum contour. Therefore we run this algorithm with all the points as the source points, and pick the contour that passes through the maximum number of LOCLs. The following algorithm describes the selection of the optimum contour:

Algorithm:

- *Input:* $G = (V, E, W)$ is a graph
- *Output:* P_{opt} is the optimum path.

Optimum-Contour(G, P_{opt})

$P_{opt} = \{\}$

length=$|P_{opt}|$

for each $q_i \in \{q_1, q_2, q_3, q_4, \ldots, q_n, q_1'\}$

  Generate a graph G_i with q_i as the starting node

  Conditional-Optimum-contour(G_i, P)

  if(length $< |P_{n+1}|$)

    length $= |P_{n+1}|$

    $P_{opt} = P_{n+1}$

  endif

endfor

proof: Follows directly from the correctness of the shortest path algorithms for directed acyclic graphs. The algorithm finds the shortest path between the nodes q_1' and q_1'', and both q_1' and q_1'' represent the same point, therefore the shortest path is a cycle. $\square$

claim 2: The algorithm returns a smooth closed contour that passes through maximum possible number of LOCLs starting from q_1.

proof:

1. The algorithm returns a closed contour : from (*claim 1*).
2. The contour is smooth because:
   (a) all finite weight links in the graph are constrained by specified smoothness factor τ.
   (b) the contour passes through finite weight links only, and
   if $w_p = \sum_i w_{pi}$ is the total weight of the path,
   $w_p \leq (n-1)$ because the weight of the link $q_1' \longrightarrow \; q_1''$ is $(n-1)$.
3. The contour passes through maximum number of LOCLs: (proof by contradiction)
   - Let the contour P_{n+1}, with total weight w_p pass through Λ number of LOCLs. If there is a path P' with total weight $w_{P'}$, that passes through Λ', number of LOCLs, such that $(\Lambda' > \Lambda)$,**(assertion)**
   From equation (4.18) we know that w_p is the number of nodes skipped along the path from q_1' to q_1'', *i.e.* $\Lambda = n - w_P, \quad and \quad \Lambda' = n - w_{P'}$

path algorithm.

Algorithm:

- *Input:* $G = (V, E, W)$ is a graph
- *Output:* $P = \{P_1, P_2, \ldots, P_n, P_{n+1}\}$ are the lists that hold the trace of the vertices, and $L = \{L_1, L_2, L_3, \ldots, L_n, L_{n+1}\}$ are the sum of the weights of the corresponding traces. P_{n+1} holds the shortest cycle.

Conditional-Optimum-Contour(G, P, L)

$\forall i, L_i = \infty$

$L_1 = 0$

$P_1 = q_1$

$q_{n+1} = q_1''$ for each $q_i \in \{q_2, q_3, q_4, \ldots, q_n, q_{n+1}\}$

  for each $q_j \in \{q_1, q_2, \ldots, q_{i-1}\}$

    if $(L_j + W_{ij}) < L_i$ then

      $L_i = L_j + W_{ij}$

      $P_i = P_j$

      $P_i = append(P_i, q_j)$

    endif

  endfor

endfor

claim 1: The algorithm returns P_{n+1}, the shortest cycle that passes through q_1.

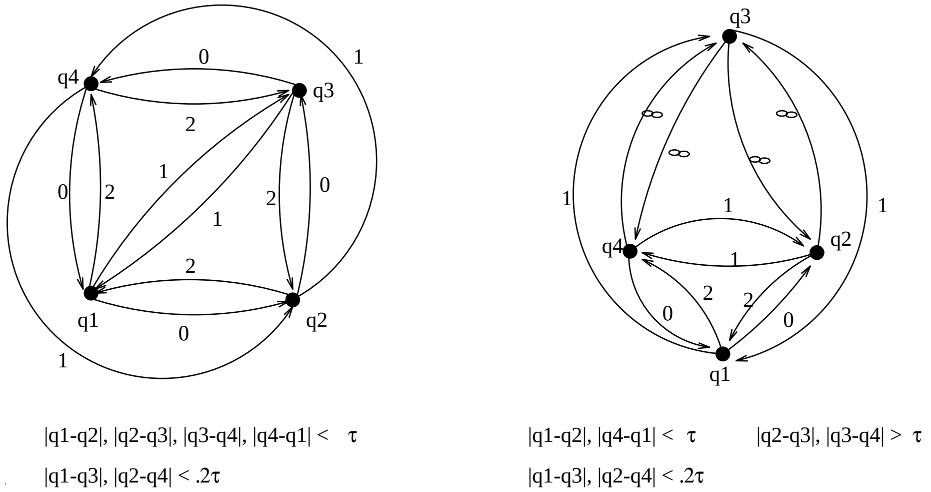


Figure 4.7: A schematic of graph showing the assignment of weights.

wise successors are of zero weight, and the rest of the arcs have the weight equal to number of nodes they bypass. In the right side graph, the point q_3 does not satisfy the smoothness constraints with its neighbors, however the link $q_1 \Longrightarrow q_3$ satisfies the smoothness constraint. So the links between those two points have unit weight, indicating that each link bypasses one intermediate point.

4.4.2 Selecting the optimum contour

To find the shortest cycle of the graph, we find the shortest cycle assuming that the node q_i is on the optimum contour, and then select the shortest of all the conditional shortest cycles. For example if q_1 is the start node, we split the node into two nodes q_1' and q_1''. The node q_1' inherits all the outgoing links of q_1, and q_2'' inherits all the incoming links of q_1. We also add a new link of weight $n-1$ from q_1' to q_1'', indicating that the link skips $n-1$ number of intermediate nodes. The shortest path from q_1' to q_1'' is the shortest cycle that includes the point q_1. Therefore we find the optimum contour containing the node q_1, using the following modified shortest smoothness constraints, and

2. $|P_o| \geq |P_a| \; \forall P_a \in \Pi$.

4.4 Contour selection algorithm

We solve the problem of selecting the optimum contour P_o by a graph search algorithm. The LOCLs form the nodes of a directed graph. A link between two nodes is finite if a smooth contour can pass through both the LOCLs. The weight is equal to the number of LOCLs missed if the contour passes through without touching any of the intermediate LOCLs. The shortest cycle of such a graph is a contour that is smooth and passes through the most number of LOCLs, and that is the optimum contour.

4.4.1 Construction of an equivalent graph

To find the optimum contour we assign weight $w_{ij} \in W$, to each link $e_{ij} \in E, \forall_{i,j \in [1 \ldots n]}$ as follows:

$$W_{ij} = \begin{cases} (j-i-1) & if\ (i \leq j) \wedge (\sum_{k=i}^{j} Sm_k < \sum_{k=i}^{j} \tau_k) \\ (n+j-i-1) & if\ (i > j) \wedge (\sum_{k=i,0}^{n,j} Sm_k < \sum_{k=i,0}^{n,j} \tau_k) \\ \infty & o/w \end{cases} \tag{4.18}$$

We show examples of this weighting scheme in Figure 4.7. According to equation (4.18), the links of the graph have 0 value if they are adjacent to each other and satisfy the smoothness constraint. If a link skips w number of nodes then the weight of that node is w. In Figure 4.7, the left side graph represents the nodes that completely satisfy the smoothness constraint, therefore all the links to anti-clock

To visualize the idea behind this characterization, in Figure 4.6 we show the optimum contour for the displayed set of points. In the figure, solid line contour

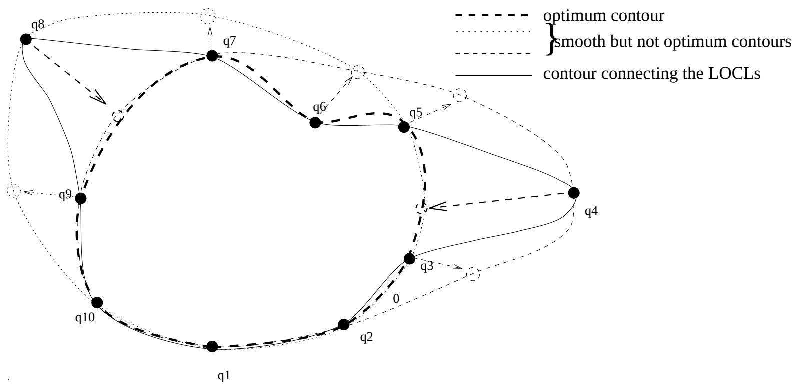


Figure 4.6: Schematic of the optimum contour.

is the rough contour that does not satisfy the smoothness constraint. There are three broken line contours that satisfy the imposed smoothness constraints. Two of them are shown in thin broken lines, and both pass through 6 LOCLs. The third smooth contour, traced by a thick broken line passes through 8 LOCLs, and it is the optimum contour. Formally, we define the optimum contour as follows:

Optimum Contour: let $G = (V, E, W)$ be the graph with a set of vertices $V = \{q_1, q_2, q_3, \ldots, q_{n-1}, q_n\}$ be a set of LOCLs located at increasing angles with respect to the origin and a reference radial direction. Let $E = \{E_{i,j} | i, j \in [1 \ldots n]\}$ be the set of links connecting the vertices and $W = \{W_{ij} | i, j \in [1 \ldots n]\}$ be their weights. Let Π be the set of all ordered subsets of V (set of valid paths in V).

The optimum contour is a subset $P_o \in \Pi$, $P_o = \{q_i, q_{i+k_1}, q_{i+k_2}, q_{i+k_3}, \ldots, q_{i+k_l}\}$, where $(k_1 < k_2 < k_3 \ldots < k_l)$, such that:

1. a closed curve joining $(q_i \Longrightarrow q_{i+k_1} \Longrightarrow q_{i+k_2} \Longrightarrow \ldots q_{i+k_l} \Longrightarrow q_i)$ satisfies the

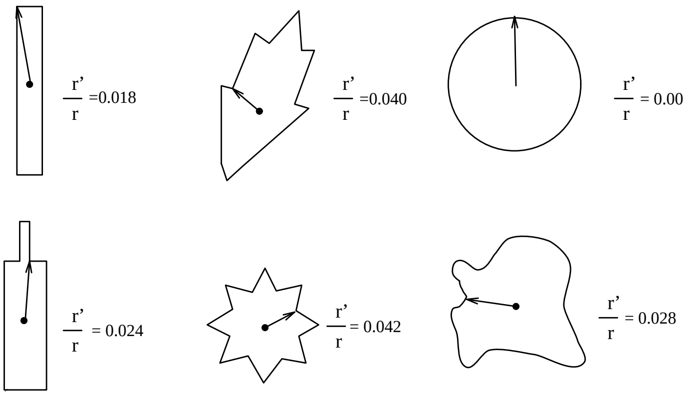


Figure 4.5: Digitized contours and their maximum $\frac{r'}{r}$ ratios.

also be interpreted as the ratio of thickness of the contour to the minimum radius of the object at a corner.

Therefore, we assume that r' is much smaller than r, and so is r''. From equation 4.12 we have $s_i \approx r'_i$, and from the curvature equation (4.14) we have:

$$k_i \underset{\frac{r'}{r} << 1}{\approx} \frac{1}{r_i} \tag{4.16}$$

Therefore in polar domain, if $\frac{r'}{r}$ is small, we can control the smoothness of a contour by constraining the sum:

$$Sm_i = \alpha r'_i + \beta \frac{1}{r_i} \tag{4.17}$$

where α and β are empirical constants.

Characterization of the optimum contour

Given an ordered set of LOCLs $Q = \{q_1, q_2, \ldots, q_n\}$, we define the optimum constrained contour as the *smoothest contour that passes through the maximum number of LOCLs $\{q_i\}$, while satisfying the smoothness constraint $Sm_i \leq \tau(i)$.*

We have transformed contour so that the angular difference between two consecutive contour points is $(\theta_i - \theta_{i-1}) = \Delta\theta$, which is small. Therefore $\cos(\theta_i - \theta_{i-1}) \longrightarrow 1$, and we approximate the above equality as follows:

$$s_i = \sqrt{r_i^2 + r_{i-1}^2 - 2r_i r_{i-1}} = (r_i - r_{i-1}) \tag{4.12}$$

Bending at a point on the curve is characterized by its curvature at that point. Curvature of a contour at a point is defined as the rate of change of inclination of the tangent to some base line ϕ, with respect to the arc length s, and is given by:

$$k = \frac{d\phi}{ds} \tag{4.13}$$

This relation can be written in polar coordinates as follows:

$$k = \frac{r^2 + 2r'^2 - rr''}{[r^2 + r'^2]^{3/2}} \tag{4.14}$$

where

$$r' = \frac{dr}{d\theta} \qquad r'' = \frac{d^2 r}{d\theta^2}$$

For digital contours represented with equiangular points we can approximate $r'|_i$ by $r'_i = (r_i - r_{i-1})$ and $r''|_i$ by $r''|_i = (r_{i+1} + r_{i-1} - 2r_i)$. Therefore:

$$|r''_i| \leq 2 \cdot \max(|r'_{i+1}|, |r'_i|) \tag{4.15}$$

Unlike Cartesian domain, in polar domain we can relate the smoothness with respect to size of the object. A large class of natural and artificial objects have r' components that are much smaller than their size, in their digital representation. In Figure 4.5 we show some objects and their largest $\frac{r'}{r}$ ratio. These numbers can

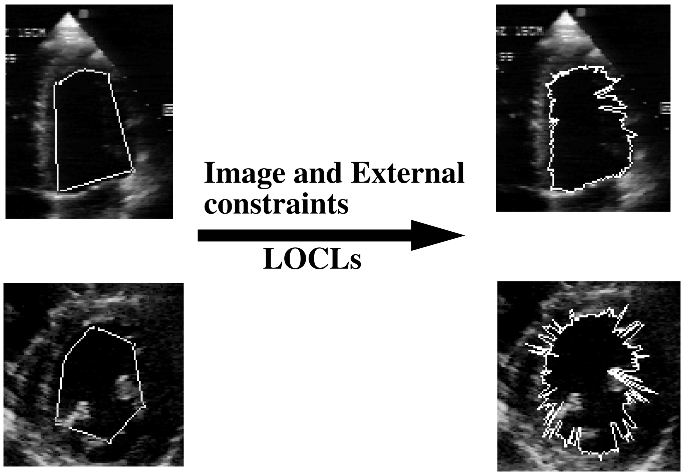


Figure 4.4: Image on the left shows the externally provided approximate contour of the left ventricular chamber of a two dimensional echocardiogram, and the right image shows the corresponding LOCLs

a spiky contour generated by joining all the LOCLs. Our goal is to find a smooth contour that passes through the maximum number of LOCLs.

4.3.3 Smoothness constraints

Smoothness of a contour can be evaluated by studying its bending(k) and shearing(s) components [90]. For digital contours in rectangular coordinates, researchers have used the distance between the adjacent points as the reasonable approximation of *shearing* [80; 81].

$$s_i = [V_s(s)]|_i = \sqrt{[x_i(s) - x_{i-1}(s)]^2 + [y_i(s) - y_{i-1}(s)]^2} \tag{4.10}$$

By substituting, $x_i = r_i \cos\theta_i,\ y_i = r_i \sin\theta_i$, for polar coordinates:

$$s_i = \sqrt{r_i^2 + r_{i-1}^2 - 2r_i r_{i-1} \cos(\theta_i - \theta_{i-1})} \tag{4.11}$$

focuses on the input contour, there is a finite band within which the contour could be relocated without surprising the supervisor. Relocating the contour at any point away from this spot-light band would incur linearly heavier costs. *i.e.*

$$E_s(r,\theta) = \begin{cases} 1 \quad if \quad |r - r_s| < spot - light \\ \frac{c}{(r_s - r)^2} \quad |r - r_s| \geq spot - light \end{cases} \tag{4.7}$$

where c is an empirical constant.

We also define $E_c(r,\theta)$, that constrains the final contour to be in the vicinity of the input approximate contour. $E_c(r,\theta)$ can be used to improve the accuracy of contouring in an iterative refinement system, by increasing its weight proportional to the number of current iteration.

We define the *external constraint* function $E_{ext}(r,\theta)$ as the sum of the individual constraint functions.

$$E_{ext}(r,\theta) = E_s(r,\theta) + E_c(r,\theta) \tag{4.8}$$

Locally optimum contour locations

Once the image feature constraint function, $E_{image}(r,\theta)$, and external constraint function, $E_{ext}(r,\theta)$ are evaluated, we determine the LOCLs at the maxima of the sum of $E_{image}(r,\theta)$ and $E_{ext}(r,\theta)$ at each radial direction, and we describe a LOCL q_i as follows:

$$q_i = (r_i,\theta_i) \ni \forall_{j \in [0,\ldots,r_{max}]}[E_{image}(r_i,\theta_i) + E_{ext}(r_i,\theta_i)] \geq [E_{image}(r_j,\theta_i) + E_{ext}(r_j,\theta_i)] \tag{4.9}$$

In Figure 4.4 we show LOCLs on a two dimensional echocardiogram. The left image shows an externally provided approximate contour, and the right image shows

2. If in the original image, there is a positive gradient perpendicular to the direction θ, then the value of the constraint function at that point is 1.

This edge strength constraint function is derived from the idea that the ideal contour location is likely to be closer to the contours of initial segmentation. Further, the optimum location is more likely to lie on the gray level transition points than on flat points. This potentially noise sensitive edge strength function is made robust by directional search and perpendicularity check, facilitated by the polar coordinate analysis. A gradient is considered to be perpendicular to the search direction if the gradient magnitude is maximum along the search direction, and minimum along the tangential direction. The maximum and minimum are obtained by rotating a 3X3 difference operator at the point of interest. If the maximum and minimum are not unique, the gradient is not considered perpendicular to the search direction.

For optical images of nonbinary scenes we do not make use of the boundaries of reference segmentation, because our thresholding method is developed for segmenting a binary scene. The edge function for such images is simply defined as the natural log of the gradient perpendicular to the radial line under consideration.

External constraints

External constraints can be domain dependent, or can be derived from the process that places the rough contour. In our system, the supervisor observes the approximate segmentation and inputs a contour to refine the segmentation. Recalling the *human visual attention map* described in section (1.3.2), although the supervisor

Hence, it is desirable that the final contour passes through the selected features, or at a close vicinity of the features. Therefore we form feature constraint functions so that the function value is maximum at the feature point, and it decreases at points increasingly away from the feature point. For each radial direction we compute these values and express the image constraint $E_{image}(r, \theta)$ as the sum of constraint functions of image features.

Constraint evaluation in polar coordinates allows selecting only the relevant features from observable features. For example one may want to constrain a contour by the edges that are orthogonal to the radial direction of interest. Furthermore, a large number of image domains yield regions of interest brighter or darker than the background. For such images, the gradients being considered can be limited to either increasing gradients or decreasing gradients only, as observed from the origin of the polar coordinate space. This enables contouring along a thin object limb, without the final contour zigzagging through either sides of the limb, depending on the local gradient strength.

In our prototype implementation we express the image constraints in terms of gradients and distance from reference segmentation boundaries. For example to process the ultrasound data we computed the edge strength at any radial line as the sum of the following two components:

1. In any direction (r, θ), if in the reference thresholded image, there is an increasing edge at distance r_1, then the value of the constraint function at point (r_1, θ) is 1 and $1/(r - r_1)^2$ elsewhere in the same radial direction θ.

where r_{max} is the maximum radial distance in the image from the selected origin, and the corresponding radial resolution is given by:

$$\Delta r = \lim_{x \text{ or } y \longrightarrow (>>1)} \sqrt{x_1^2 + (y_1+1)^2} - \sqrt{x_1^2 + y_1^2} \geq \frac{1}{\sqrt{2}} \tag{4.4}$$

Therefore, by choosing $\Delta r = \frac{1}{\sqrt{2}}$ prevents two pixels in rectangular coordinates mapping onto the same discretized polar coordinate. We use an internal point $(\overline{x}, \overline{y})$, as the origin of transformation, where:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.5}$$

$$\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{4.6}$$

The rectangular to polar transformation yields a contour $[P' = \{p'_1, p'_2, p'_3, \ldots, p'_n\}, p'_i = (r_i, \theta_i)]$ such that the contour points p'_i have equiangular separation of $(\theta_i - \theta_{i-1}) = \Delta\theta$.

4.3.2 Selection of locally optimum contour locations(LOCLs)

A set of LOCLs is selected from all the radial directions by independent image feature and external constraint analysis in each direction. Both the *image feature constraints* and the *external constraints* are expressed as continuous functions. The location where the sum of these two constraint functions is the maximum is considered a LOCL.

Domain dependent constraints

Depending on the task at hand, image features such as brightness or color gradients, lines, brightness levels, corners and terminations, etc. indicate contour locations.

4.3 Constrained contouring method

Our constrained contouring method is summarized by the following three steps:

1. the data is transformed into polar coordinates.

2. a set of Locally Optimum Contour Locations (LOCLs) is determined according to the constraints of context, distance from the externally specified approximate contour, and the relevant image features.

3. a contour that passes through the maximum number of LOCLs, while satisfying the imposed smoothness constraint is selected as the optimum contour.

4.3.1 Transformation to polar coordinates

Let $P = \{p_1, p_2, p_3, \ldots, p_{N-1}, p_N\}$ be a simple polygon with vertices $p_i = (x_i, y_i)$ that represents the initial contour input by some other process. In our system the polygon P is the circled input. To analyze in the polar coordinates, we transform the initial contour along with the image data to polar coordinates (r, θ) so that no data is lost.

For no loss transformation, we select the angular resolution:$\Delta\theta$, and the radial resolution:Δr, such that a unit translation in the rectangular coordinates at any point in the image causes at least a $\Delta\theta$ angular change or Δr radial change in the polar coordinates. The angular resolution is given by:

$$\Delta\theta = tan^{-1}(\frac{1}{r_{max}}) \tag{4.3}$$

- many interesting contouring problems involve tracing closed contours enclosing a region [89; 86].

- a large fraction of the regions of interest are star-like, so the radial lines from an internal point intersect the border only once (*e.g.* See Figure 4.5). Such contours form single valued curves in polar coordinates.

- The smoothest closed contour is a circle, with curvature or allowable bending being inversely proportional to the radius. Hence it is natural to visualize deforming a circle to yield a closed contour.

- In active contouring, researchers have observed that shrinking and bunching-up of contour points where image forces are large while processing in rectangular coordinates [81]. In polar coordinates, such problems are avoided, because for a closed contour deformation, a point need be free to move only in its radial direction.

- Analysis in polar coordinates:

  - allows simple approximations to bending and shearing components of a contour.

  - aids in the selection of relevant image features because of the available orientation information.

  - allows tracing the boundary of a thin limb without zigzagging through prominent feature points on both sides of the limb.

straints are usually set according to the *a priori* domain knowledge. Kass, Witkins and Terzopoulos [23] suggested a deformable contouring model to select the best set of edges from the all possible edge locations, so that a set of global constraints are maximally satisfied. The basic premise of their work is that an approximate contour is placed near the desired boundary by some external mechanism.

They use variational calculus based energy minimization method to refine the shape and location of the approximate contour under the constraints of connectivity, bending and domain dependent image features. A dynamic programming algorithm for efficient energy minimization is suggested by Amini, Weymouth and Jain [80]. A greedy algorithm for faster evaluation of the minimum energy contour is suggested by William and Shah [81]. A similar work where *a priori* knowledge of the shape of the object is used to develop deformable templates has been reported by Yuille *et al.* [82].

In our *constrained contouring* algorithm we treat the above internal, external and image energy terms as the constraints on the possible contour location and shape, and we select a smooth contour that satisfies the constraints. To find the optimum contour we enumerate the all the contours that satisfy the constraint, and select the one that passes through the most number Locally Optimum Contour Locations (LOCLs).

4.2.1 Motivation to contouring in polar coordinates

The following observations motivated using polar coordinates to compute the contour.

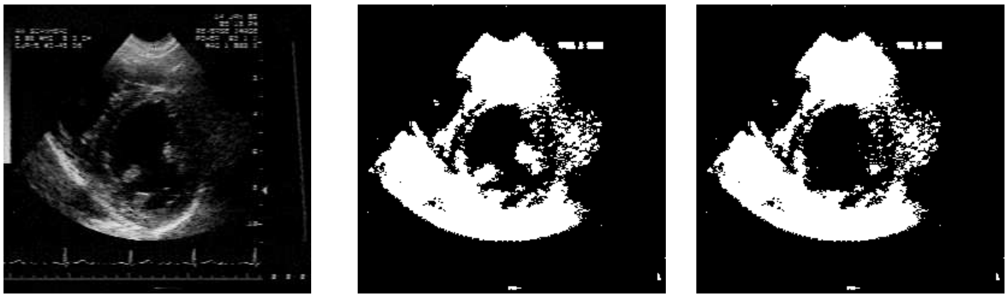


Figure 4.3: A short axis two dimensional echocardiogram (left), the thresholded output, and its refinement by pointing(right).

4.2 Segmentation refinement by circling

If the initial segmentation is erroneous, or if there are too many isolated misclassified regions, the supervisor may refine the region by circling. Due to the imprecision inherent in the human-machine interface, the circled gesture yields a rough approximation of the boundaries of the indicated region. Our goal is to modify the circled input so that it traces the precise boundaries of the intended region. This problem is closely related to the problem of *active contouring* proposed by Kass *et al.*[23].

We have developed a constrained contouring method that analyses the data in polar coordinates, and traces the precise boundaries of the indicated region. Initially the data is transformed into polar coordinates, and the optimum contour locations are computed for each radial direction. We develop a graph search based algorithm to select the optimum contour that passes through the maximum number of locally optimum locations, under a smoothness constraint.

Recently, to trace satisfactory contours researchers have proposed using external constraints, and smoothness restrictions to edge selection. The smoothness con-

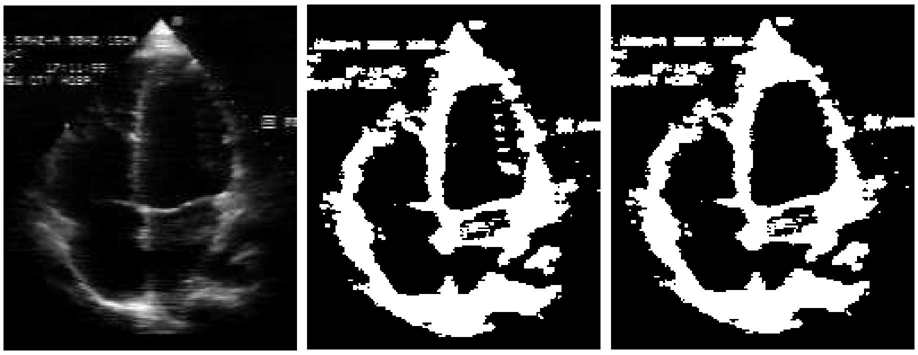


Figure 4.2: A two dimensional echocardiogram (left) and the thresholded output(center). The right image is the refined segmentation by pointing at the fragments in the left ventricular chamber.

4.1.2 Results: refinement by pointing

Figure 4.2 shows a four-chamber echocardiogram, its thresholded output, and the refined output. The top right chamber in the image is left ventricle. The intention was to get uninterrupted boundaries of the wall. The isolated bright spots inside the chamber do not belong to the wall. Those points are indicated by pointing. In the right most image, we display the refined segmentation. Our region growing algorithm also tracks the indicated region over time.

A similar result for a different data set (short axis view - two dimensional echocardiogram) is shown in figure 4.3, where the left image is the original, the center image is the thresholded output, and the right image is the refined output after picking two points on the valves inside the cavity.

sure the *edge transgression distance* and it is defined recursively as follows:

$P_e(z_0, z_0) = 0$

$p_e(z_0, z_1) = k_3. \min\{(P_e(z) + |g(z) - g(z_1) - g_{avg}(z) + g_{avg}(z_1)|)$

$\quad |z \in \{neighborhood\ of\ z_1\}\}.$

where $g_{avg}(.)$ is the average local image intensity, and k_1, k_2, k_3 are experimentally determined normalizing constants.

All these distance measures are limited by the supervisor's visual attention. In Chapter I, we introduced the *human visual attention model.* From the model we derived that there is a constant high attention spot around the point of focus, and the at all other points the attention falls linearly with respect to their angular distance from the observer. We incorporate the model into our region growing algorithm by defining the overall distance measure as follows:

$$P_i(z_0, z) = \begin{cases} 0 & if \quad |z_0 - z| < \ spot - light \\ min\{P_d(z_0, z), P_t(z_0, z), P_e(z_0, z)\} & otherwise \end{cases} \tag{4.1}$$

In other words, we describe the intended region $L_i(x_0)$ as:

$$L_i(z_0) = \{z | P_i(z_0, z) < C \ \wedge z \in I\} \tag{4.2}$$

where C is an experimentally derived constant, and $z = (x, y)$ for a static scene, and $z = (x, y, t)$ for a dynamic scene, t being the temporal coordinate.

$\{P_i\}$ being less than a predefined constant C [88], i.e.

$I = \{L_0 \cup L_1 \cup ....L_i \cup \ldots L_n\}$, and $L_i = \{z | P_i(z) < C \wedge z \in I\}$.

Our goal is to find the intended region L_i around a picked point z_0, such that $P_i(z_0) < C$. This requires that all the regions in the image and their features are known *a priori,* which is often not true. To solve this problem we model an intended region using the following assumptions:

1. Since each region L_i in the image is contiguous, and the study shows that visual sensitivities fall off steeply with the distance from the point of attention [8], the physical distance (P_d) defined by $P_d(z_0, z_1) \;=\; k_1.|z_0 - z_1|$, between any two points in an intended region should be small.

2. A change in the image intensity normally indicates a change in the imaged object characteristics, therefore intensity transitions (P_t) between any two points in an intended region must be small. We call this measure the transition distance, and it is recursively defined as follows:

   $P_t(z_0, z_0) = 0$

   $P_t(z_0, z_1) = k_2. \min\{(P_t(z_0, z) + |g(z) - g(z_1)|)$

   $\qquad\qquad | \; z \in \{neighborhood \;\; of \;\; z_1\}\}.$

3. Since the image intensity characterizes the regions, there must be a intensity gradient between two neighboring regions. Therefore intensity gradients (P_e) between any two points in an intended region must be small. We call this mea-

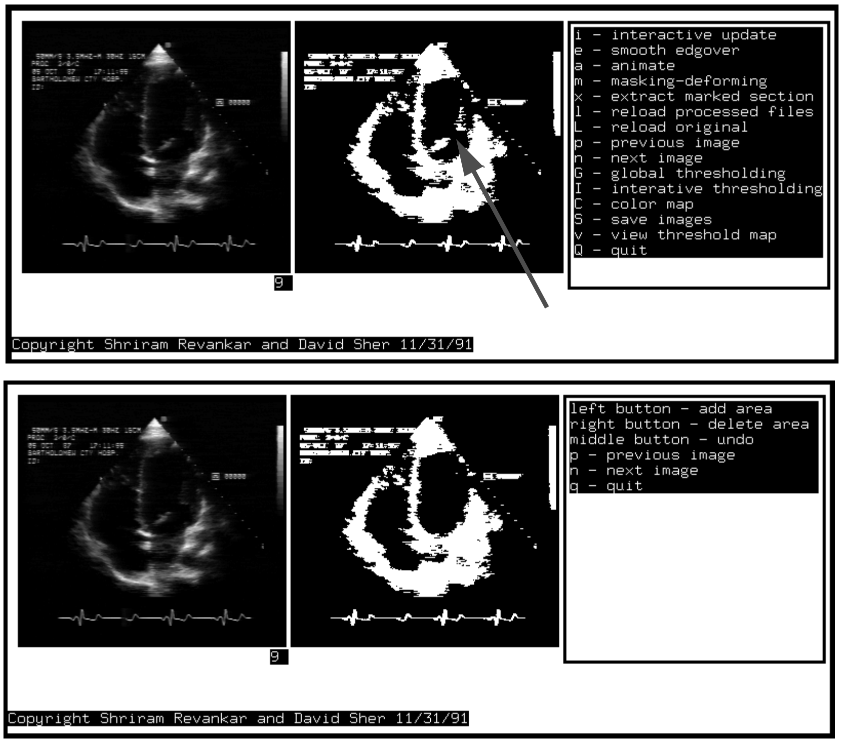


Figure 4.1: Graphics displays of refining by pointing. The top frame shows the supervisor pointing at the approximate locations of error, and the bottom frame shows the refined segmentation.

find the boundaries of the pointed locations and classify them as suggested.

The assumption is that the precise boundary of the intended region is in the vicinity of the pointed location, and the region is a contiguous area. The human visual attention map introduced in section (1.3.2) supports this assumption. This type of problem of collecting similar points is conventionally solved using region growing techniques. In this thesis we develop a new region growing based algorithm to find the boundary of the pointed region. The algorithm uses the visual attention map for weighting the distance measure used for gathering the points in the region. The extent of a region is determined by the supervisor's expectancy, in addition to the image features such as intensity and intersecting edges.

In Figure 4.1 we show snap shots of our graphics interface while the segmentation is being refined by pointing. The display contains the original and the processed images side-by-side, and the supervisor can indicate the regions on both of them. The supervisor compares the original image with the processed one, and indicates the misclassified regions by picking points in those regions. The computer uses a *region growing* technique to obtain the precise group of neighboring pixels to be added to or deleted from the computer estimated object region. This refinement is repeated till the results are satisfactory.

4.1.1 Supervised region growing

The goal of region growing is to extract the precise boundaries of the region indicated by the picked point. To achieve this we model the image data I as a collection of spatially contiguous regions $\{L_i\}$, that are described by some measurable features

Chapter 4

Supervised Segmentation Refinement

In the previous chapter we have described a threshold propagation method to generate an initial segmentation of the image. If some regions in the initial segmentation are misclassified, the supervisor indicates the corrections either by *pointing* or by *circling* the erroneous regions. The supervised refinement algorithms update the segmentation according to suggested corrections. For a dynamic scene, the corrections suggested on one frame are propagated to the entire temporal sequence of frames. For of refinement by pointing, we have developed a three dimensional region growing algorithm, and for refinement by circling, we have developed a constrained contouring algorithm.

4.1 Segmentation refinement by pointing

If in the initial segmentation there are only a few misclassified regions, it is natural for the supervisor to point at those regions and suggest that the regions be reclassified. But the pointing is an approximate gestures. The goal of the refinement is to

3.6 Summary of initial segmentation

We have described a new method for pattern extraction by adaptive propagation of a known regional threshold. A regional threshold is computed either by analyzing the supervisor suggested section of the image or by selecting a threshold that generates patterns of specified width. The regional threshold is then propagated over the entire image (or the image sequence for a dynamic scene), such that the ratio of the threshold to slope of the intensity transition is maintained constant. The thresholded patterns are usually rough. Therefore we vary the threshold by the average gray value of the best fit morphological enclosure to yield smooth thresholded patterns.

We show the outputs generated by this algorithm for both echocardiograms and optical images of hand written text. This method is useful for medical image processing where direct measurements of the extracted pattern is of diagnostic interest, and for the fields that desire extraction of thin continuous lines and strokes from digitized images of line drawings and text.

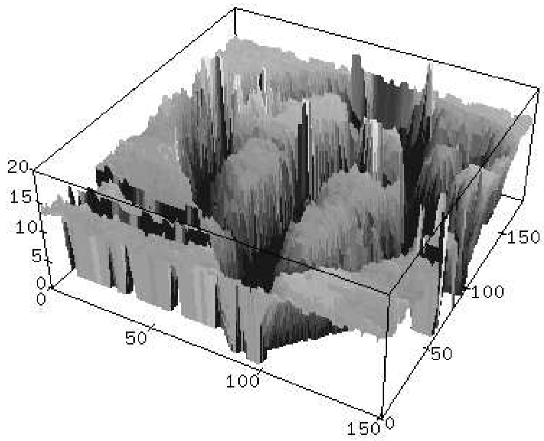


Figure 3.16: Threshold map generated by the adaptively propagated threshold for the second image shown in Figure 3.13.

threshold has valleys at the object regions and peaks at the background, indicating that the threshold suppresses formation of holes and noise specks. In addition, it also extracts the patterns from barely visible regions.

There are some drawbacks in this scheme. A brief description of these drawbacks and possible ways to overcome them are explained below.

- In images with large illumination variation, there may not be a global low threshold that detects the complete pattern.
- If there is a large regional blur, the morphological biasing may suppress the detected patterns, because the bias makes the threshold higher on the low intensity areas.
- Although selection of s is not critical, if it is too high, one may lose the finer details of a thresholded pattern.

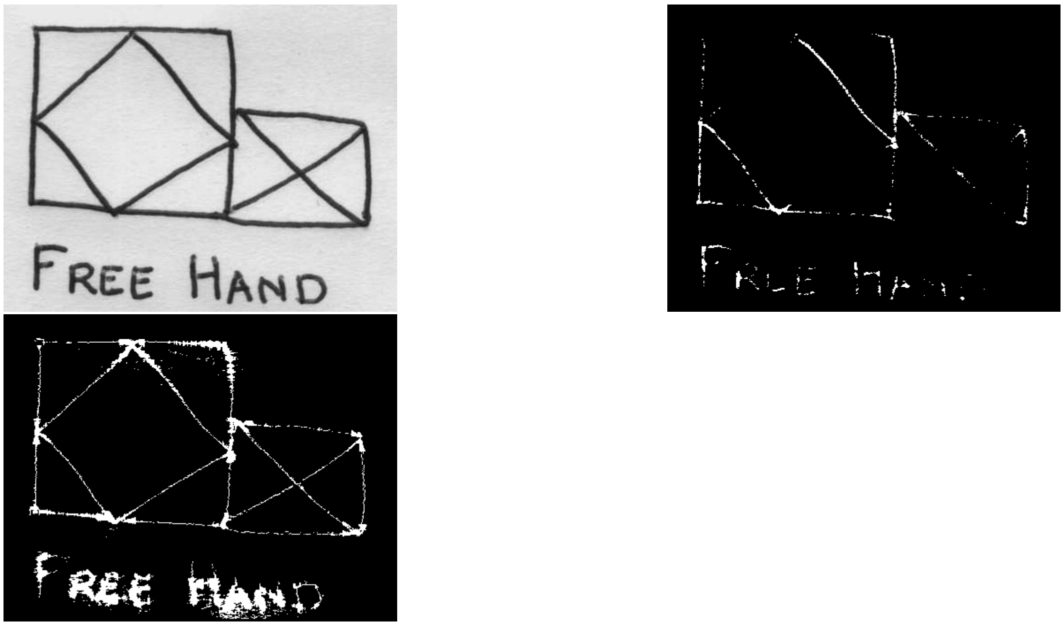


Figure 3.15: Optical image of a hand drawn line sketch(left), was thresholded by the global threshold T_{ref} selected so that thin line is detected. The bottom image is the output by adaptive propagation of T_{ref}.

image in the center is generated by the global threshold T_{ref}. It is selected such that it yields thin strokes. In the bottom row we show the output generated by our adaptively propagated threshold. It clearly generates contiguous thin and uniform patterns. A second example of processing an optical image is shown in Figure 3.15, where an image of a free-hand line drawing is being thresholded. In the top right image we specified the line width to be two pixels, and did global thresholding. Although the pattern generated is desirably thin, it is quite disconnected. In the bottom image we show the result of our adaptive thresholding. Clearly, the line segments are of the same thickness, and are better connected.

To visualize the adaptation of our propagated threshold, in Figure 3.16, we show a three dimensional rendering of the threshold values at all the image points. The

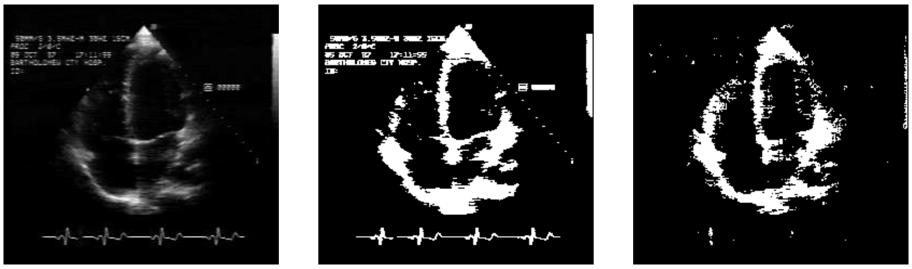


Figure 3.12: Two dimensional echocardiograms of Figure 1 (left and center), with adaptively thresholded image in the right.

Figure 3.13: A two dimensional echocardiogram (left) thresholded by $T_{ref} = 27$ without adaptation (center) and with adaptation (right).

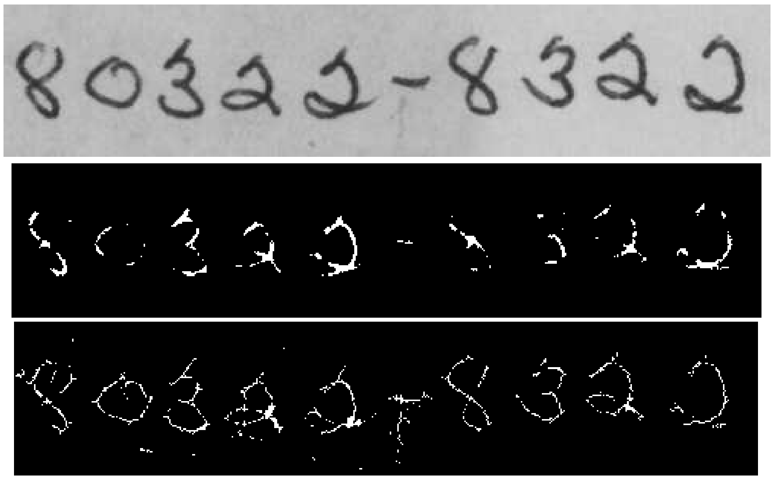


Figure 3.14: The top image is the grey tone image obtained from a USPS mail piece. The middle image is the binary pattern generated by the global threshold T_{ref} and the bottom image is the binary pattern generated by adaptive propagation of the threshold.

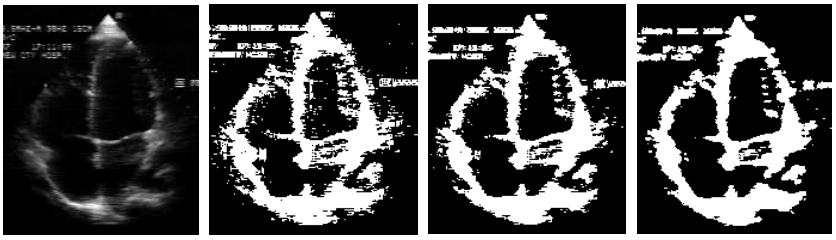


Figure 3.11: Thresholded output at smoothing factors s = 0.0, 0.5 and 2.0.

- **Smoothing:** The *smoothing factors*, controls the extent of adaptiveness (local influence), and hence the smoothness of the output. When the smoothing factor $s = 0$, the threshold is equal to the propagated local threshold T_p, and there is morphological bias. In Figure 3.11, we show the results of thresholding for three different values of smoothing factor. For $s = 0$, the thresholded pattern is noisy, but as s is increased, the thresholded patterns become smoother.

3.5 Results

In Figure 3.12, we compare the outputs generated by our algorithm with and without adaptive propagation. The image in the center was generated by the global threshold T_{ref}. This image has a threshold that detects the patterns of correct width for the most part, but it has missed some wall sections. In the right image, this threshold was adapted by maintaining the constant W_{diff} over the entire image, and the walls are detected even in the barely visible sections of the original image. Thresholding of a second echocardiogram frame is shown in Figure 3.13.

This method works well on optical images of hand written text. In Figure 3.14 we compare the thresholding with and without adaptive propagation. The original image in the top row is a digitized image obtained from a USPS mail piece. The

3.4.1 Morphological biasing of the propagated threshold

We define the morphologically biased threshold as follows:

$$T(x, y, t) = T_p + s(T_p - N(x, y, t)) \tag{3.20}$$

We call the constant s, *smoothing factor,* since it controls the effect of the morphological bias on the propagated threshold.

From the above equation we observe that when N(x, y, t) is large, the threshold is low, and when $N(x, y, t)$ is low, the threshold is high. This morphological bias on the propagated threshold T_p gives some desirable properties to the threshold. They are:

- **Suppression of noise:** Noise specks are formed when false positives are observed on the background or when false negatives are observed on the object. These noise specks can be suppressed by making the threshold high on background and low on object region. Our threshold has this adaptiveness because $N(x, y, t)$ is low on the background and high on the object.

- **Edge preservation:** The three dimensional morphological enclosures account for all possible surface orientations at the pixel under consideration. Therefore by selecting the best fitting enclosure for threshold biasing, we preserve the most likely edge. In cases where the neighboring object and background enclosures differ by the same value from the local propagated threshold, classification of the pixel as either background or object does not significantly alter the accuracy of classification.

We define the *best-fit* enclosure as the one with an average gray value that deviates the most from the threshold at that point. It is to be noted that, if the temporal resolution and spatial resolution of an image are different, then the best-fit enclosure is found by computing a weighted average.

Let $\{S_1(x,y,t), S_2(x,y,t), S_3(x,y,t), ..., S_{27}(x,y,t)\}$ be the enumeration of the set of three dimensional morphological enclosures at any point (x, y, t).
Let $\{m_1(x,y,t), m_2(x,y,t), m_3(x,y,t), ..., m_{27}(x,y,t)\}$ be the weighted-average gray levels of the corresponding enclosures. The weighted-average m_i of the ith enclosure enclosing the pixel at (x,y,t) is given by:

$$m_i(x,y,t) = \frac{1}{17} \sum_{\forall (j,k,l) \in S_i(x,y,t)} (a.(l=t) + b.(l \neq t))g(j,k,l) \tag{3.18}$$

where $g(.)$ is the intensity or grey level of the pixel, and a, b are empirically chosen constants that account for the difference between spatial and temporal resolutions. The constant a is the weight of the pixels in the template that are on the same time slice as the central pixel, and b is the weight of the pixels that are on a different time slice.

We define the morphology function N(.), as follows:

$$N(x,y,t) = m_i(x,y,t) \ni_{\forall j \in \{1,2,3,...,27\}} \{|T_p - m_i(x,y,t)| \geq |T_p - m_j(x,y,t)|\} \tag{3.19}$$

i.e. the morphology function N(.) gives the average gray value of the best-fit enclosure.

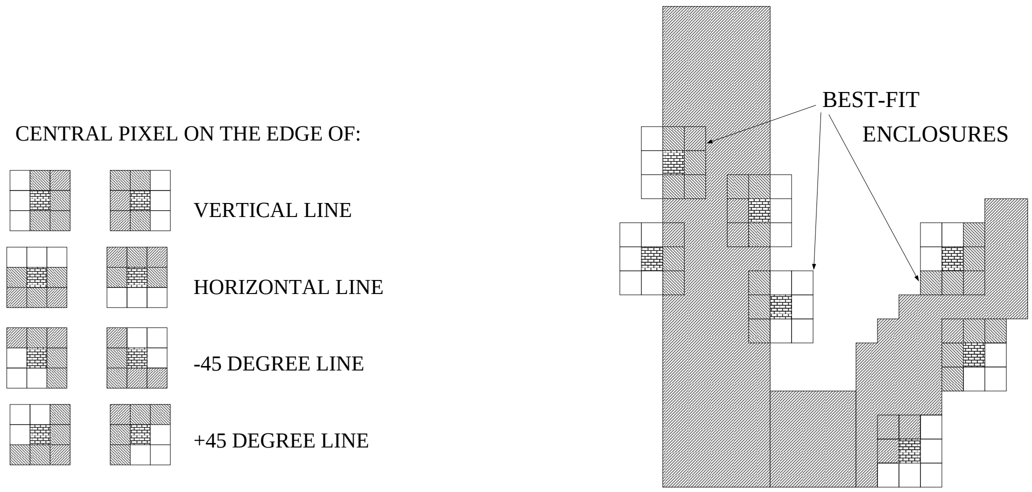


Figure 3.9: Two dimensional morphological enclosures, and their best-fit positions are shown for a binary image.

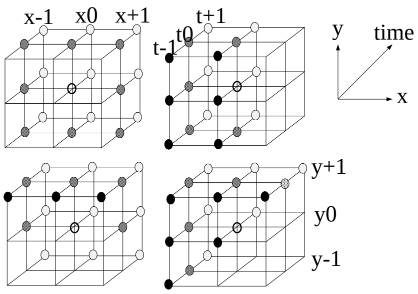


Figure 3.10: Samples of three dimensional morphological enclosures.

have developed a new morphological operator that varies the propagated threshold so that it yields smooth binary patterns directly. Morphological operators exploit the local coherency of the data. We have designed a morphological operator based on the following two assumptions:

1. **Assumption:**A pixel surrounded by object pixels is more likely to be an object pixel than a pixel that is surrounded by background pixels, and, similarly, a pixel surrounded by background pixels is more likely to be a background pixel.

2. **Assumption:** A pixel on the object and background boundary is more likely to belong to the region with which it has the lower contrast.

If the scene is dynamic both spatial and temporal coherency can be exploited, by selecting three dimensional morphological enclosures. The morphology function $N(.)$ searches through a set of enclosures defined in a $(3 \times 3 \times 3)$ neighborhood, and returns the average gray value of the *Best Fit* enclosure. These enclosures are three dimensional extensions of two dimensional enclosures defined by Revankar [87]. The significance of these enclosures can be explained with the help of their two dimensional counterparts, which are a set of possible *homogeneous enclosures* of a pixel at the object-background boundary, as shown in Figure 3.9.

Some examples of the three dimensional enclosures are schematically shown in Figure 3.10. These can be viewed as enclosures at a pixel at the boundary surface created by a sequence of two dimensional boundaries. There are 27 such templates, and each enclosure contains 17 contiguous pixels. These templates enumerate all possible surface orientations at the pixel under consideration.

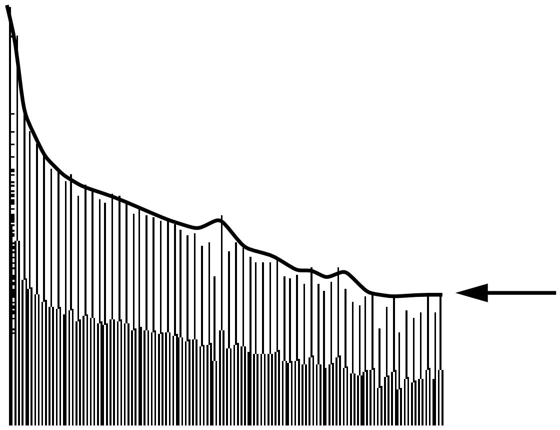


Figure 3.8: Simulation of histogram smoothing by traversing regional maxima.

The level at which there is an abrupt change in area increase, the magnitude of $\frac{da^2}{dt^2}$ is the maximum. Therefore we define the low global threshold T_{low} as the highest intensity level at which the abrupt change is observed.

$$T_{low} = \max_i \ni \{\forall t, j < T_{ref} \left|\frac{da^2}{dt^2}\right|_{t=i} \geq \left|\frac{da^2}{dt^2}\right|_{t=j}\} \tag{3.16}$$

This threshold computation is sensitive to noise, and scaling used in imaging instruments. For example, if the image is histogram equalized or scaled, it may not have pixels with some grey levels. This may result in locating T_{low} at one of such valleys. We can reduce the noise sensitivity by smoothing the histogram. We simulate the histogram smoothing by computing $\frac{da^2}{dt^2}$ with respect to regional maxima of the histogram as follows: (see Figure 3.8).

$$\frac{da^2}{dt^2}|_t = H[t] - \max_{i=t+1}^{T_{ref}} H[i] \tag{3.17}$$

3.4 Morphological smoothing

The boundaries of the thresholded patterns are usually not smooth. Usually the patterns are smoothed by various morphological methods after binarization. We

QED.

Therefore, for a general line pattern with a radius of curvature greater than 1.57 times the width, our method computes the pattern width fairly accurately. Furthermore, the proposed width computation accounts for the patterns that are oriented parallel to a coordinate axis.

3.3.4 Computation of the global low threshold

To determine the low threshold T_{low} we scan through the binary patterns of the image generated by a series of decreasing threshold levels, starting from the reference threshold T_{ref}. Consistent with our assumed model of the image, the object area increases as the threshold is decreased. The level at which the threshold falls below the background level, the object area increases abruptly, and we pick the previous level as the global low threshold T_{low}. This method can be implemented efficiently through simple histogram analysis.

Let $H[i]_0^{T_{max}}$ be the histogram of the entire image. Then the area a_t of the thresholded pattern at any threshold t is given by:

$$a_t = \sum_{i=t}^{T_{max}} H[i] \tag{3.14}$$

and the rate of change of the area is given by:

$$\frac{da}{dt}|_t = (a_t - a_{t+1}) = H[t]$$

$$\Longrightarrow \frac{da^2}{dt^2}|_t = H[t] - H[t+1] \tag{3.15}$$

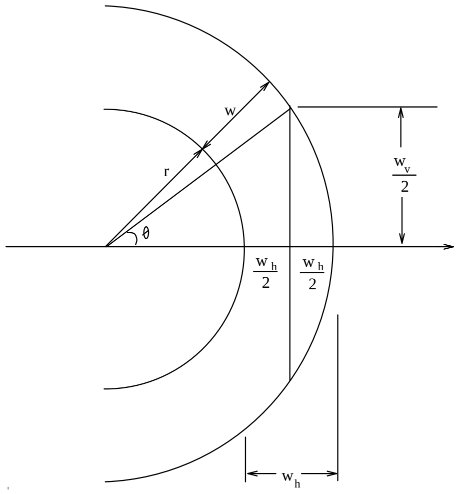


Figure 3.7: Computing the width of an uniform curved pattern

pattern of radius of curvature r and width of the pattern is w. Let the vertical and horizontal run lengths be w_v and w_h correspondingly. From Figure 3.7 we have:

$$w_v = \sqrt{3w^2 + 4rw} \tag{3.8}$$

$$w_h = w \tag{3.9}$$

Therefore the estimated width w' of the pattern from equation 3.7 is

$$w' = \frac{1}{\sqrt{(\frac{1}{3w^2+4rw} + \frac{1}{w^2})}} = w\sqrt{\frac{3w+4r}{4w+4r}} \tag{3.10}$$

and the error $\epsilon = \frac{w-w'}{w}$ is:

$$\epsilon = 1 - \sqrt{\frac{3w+4r}{4w+4r}} \tag{3.11}$$

From equation 3.10, $w' < w$. Therefore to limit the error to $\epsilon < .05$ we must have:

$$\sqrt{\frac{3w+4r}{4w+4r}} > 0.95 \tag{3.12}$$

$$i.e. \quad r > 1.564w \tag{3.13}$$

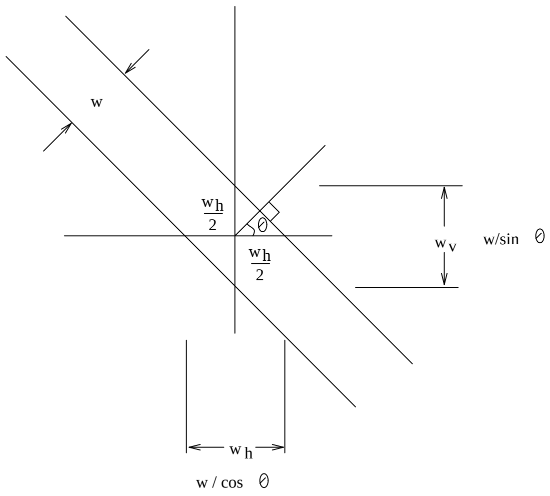


Figure 3.6: Computing the correct width of a uniform object pattern with arbitrary orientation θ.

have $w_v = \frac{w}{sin(\theta)}$ and $w_h = \frac{w}{cos(\theta)}$, where θ is the orientation of the pattern.

If $w_h/w_v > 3.2$, the width is set to w_v.

i.e. $w_h/w_v = tan(\theta) > 3.2$, or $sin(\theta) > 0.955$. Therefore, from Figure 3.6 $w_v = w/0.955$, and the error ϵ is:

$$\epsilon = |(w - w_v)/w| < |1 - 1/0.955| = 0.047$$

i.e. the width is over estimated by less than 5%. A similar argument can be made when $w_h/w_v < 0.31$.

QED.

Claim: Equation (3.7) evaluates the width of a curved pattern within the error bound if the radius of curvature is greater than 1.57 times the width.

Proof: Figure 3.7 illustrates the situation of measurement of width for a curved to corresponding row or column flags, a run is detected and its length and location are noted, and the flag is set to the background value. On the other hand if there is a transition from background to object, the start of a run is noted by starting the corresponding run-length counter and setting the flag to the object value. If there is no transition, both row and column run-length counters are incremented if the pixel is an object pixel. This algorithm computes and stores all the runs in the image in a single scan while making two checks at each pixel.

Let w_v, w_h be the vertical and horizontal run-lengths at the center of a pattern cross section. Then the width of the pattern at that cross section is given by: $\frac{1}{\sqrt{\frac{1}{w_v^2}+\frac{1}{w_h^2}}}$ (see Figure 3.6). This equation does not cover the case where patterns are parallel or nearly parallel to a coordinate axis. Therefore we modify the equation as follows:

$$w = \begin{cases} \min(w_v, w_h) & \text{if } (0.31 > w_v/w_h) \bigvee (3.2 < w_v/w_h) \\ \frac{1}{\sqrt{\frac{1}{w_v^2}+\frac{1}{w_h^2}}} & \text{otherwise} \end{cases} \tag{3.7}$$

i.e. if a run is longer than thrice that of its orthogonal counterpart, the shorter run-length is used as the correct width of the pattern at that cross section. This avoids the errors that might occur when the object pattern is nearly parallel to one of the coordinate axes.

Claim: Equation (3.7) introduces a maximum of 5% error in the estimation of the width of an uniform straight pattern.

Proof:

The method approximates the correct answer only when the ratio w_v/w_h is outside the specified range. In all other cases it correct because, from Figure 3.6 we

```
W_low = width at T_low;
high = T_max;
low = T_low;
T_p = high;
While( (high-low) > 1 ) do{
     W_T = width at T;
     if( W_T > W_low − W_diff )
          low = T_p;
     else
          high = T_p;
     endif
     T_p = (high + low)/2;
     }
```

The propagated local threshold T_p is computed at all points on the image. All pixels with intensities higher than T_p are classified as the pattern pixels, and the rest are classified as background pixels. At the points on the background where no pattern is detected by T_{low}, we set threshold $T_p = T_{ref}$ so that the formation isolated noise spots are suppressed.

3.3.3 Efficient determination of pattern width from run length

Width of a pattern is the distance between it external boundaries that are parallel to its major axis. It is difficult compute the pattern width, without the knowing the pattern orientation. We have developed a method to efficiently compute a good approximation of pattern widths from the vertical and horizontal run lengths[86].

We find the horizontal and vertical runs in a binarized image in one raster scan. While scanning, we maintain a flag and a run-length counter for each column, and for the current row of the image. The flags hold the value of the traversed image area in their respective columns and in the current row. In the raster scan (row-by-row scanning), at each pixel if there a transition from object to background with respect

Let $w_{low_{ref}}$ and w_{ref} be the widths of the pattern at the reference region, generated by thresholding at T_{low} and T_{ref} respectively. The widths are the thickness of the pattern measured perpendicular to the orientation of an elongated pattern. Therefore the slope of the intensity transition, m_{ref}, at the reference region is given by:

$$m_{ref} = \frac{2(T_{ref} - T_{low})}{(w_{low_{ref}} - w_{ref})} \tag{3.4}$$

$$\Longrightarrow \quad \frac{2(T_{ref} - T_{low})}{m_{ref}} = w_{low_{ref}} - w_{ref} \tag{3.5}$$

i.e. twice the ratio of threshold to slope at the reference section is equal to the width difference at that location. For threshold propagation we need to maintain the threshold proportional to the local illumination. In other words we need to maintain the same threshold to slope ratio. Therefore using the equality of equation (3.5) we can propagate the threshold by computing a threshold T_p at any pattern location in the image such that the difference in the pattern width $w_{T_p} - w_{T_{low}}$ is equal to the width difference at the reference section. *i.e.* the *propagated threshold* T_p is:

$$T_p = T \ni \{(w_{low_T} - w_T) = (w_{low_{ref}} - w_{ref})\} \tag{3.6}$$

3.3.2 Threshold propagation algorithm

We compute the reference width difference w_{diff} as the difference between the average pattern widths at T_{ref} and T_{low}, at the reference section of the pattern. The following is an efficient algorithm that uses a binary search to select a local threshold that maintains the constant width difference w_{diff}.

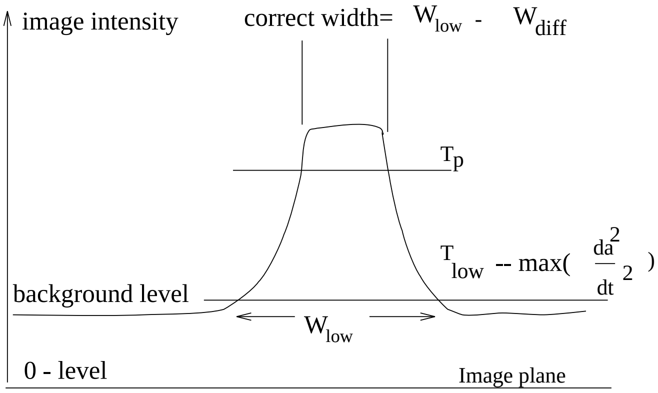


Figure 3.5: A schematic diagram that shows the location of the global low threshold T_{low} and computation of the propagated threshold T_p.

that it is higher than the background intensity level, but it is low enough to detect even the dimmest object pattern. In Figure 3.5 we schematically show the level of the low threshold T_{low}.

3.3.1 Computation of the propagated threshold

As mentioned in the previous section, we need to maintain a constant slope to threshold ratio to propagate the known reference threshold. The slope of the transition can be computed only if all the legitimate transitions are located, and if the transition ramp value is sampled at at least two different intensity levels perpendicular to the orientation of the pattern. That is computationally complex and in our experiments the method yielded poor results in a noisy data.

We avoid the computation of the slope by finding the local thresholds at all pattern locations so that the same *width difference* is maintained. This method can be shown to be equivalent to maintaining the constant *threshold to slope ratio*, as follows:

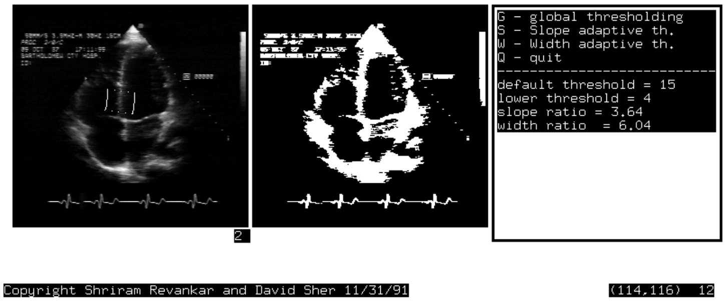


Figure 3.4: A snap shot of the graphics display. The left image is a long-axis two dimensional echocardiogram. The supervisor has picked the section of the septum (indicated by two parallel lines either side of the central limb) and suggested the thickness to be 8 pixels. The right image shows the thresholded pattern that shows correct average thickness at the selected wall section.

thickness ($W.l$).

In Figure 3.4 we display supervisor's selection of a section of septum in four-chamber view of an echocardiogram. The right image in the figure displays the pattern generated by the corresponding threshold computed so that the binary pattern of the indicated wall section has an average thickness of 8 pixels.

3.3 Threshold propagation

In Figure 3.1 we show a schematic diagram of a ramp transition between the background and a limb of the object. Given the object and the background levels, clearly the correct threshold be placed at $T_p = \frac{I_o\rho_b + I_o\rho_o}{2}$. But, it is not possible to find the local object and background levels unless a threshold level is established. To get around this problem we estimate the slope of the ramp at the reference section by thresholding the pattern at two levels T_{ref} and T_{low}. Then maintain the same *threshold to slope ratio* for the entire image, thereby making the threshold at any location proportional to the illumination at that location. The threshold T_{low} is computed so

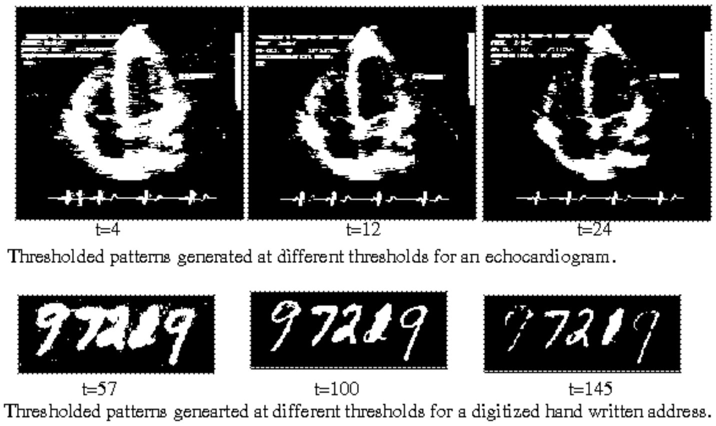


Figure 3.3: Pattern thickness is sensitive to the selected threshold 't'. Both echocardiogram and the hand written text images show an increase in the pattern width a decrease in the threshold.

we do a binary search to select the highest threshold that detects a line segment of specified width and length. For two dimensional echocardiograms, a cardiologist can point to a section of the wall with known thickness, and a regional threshold can be determined such that it generates a pattern of thickness close to the specified thickness[85]. Let l be the length of the indicated (or automatically) section of the limb in the image, and W be its specified width. We compute the regional threshold T_{ref} such that the area of the thresholded pattern closely approximates the value Wl. This can be efficiently computed by studying the histogram of a small rectangular window along the pattern limb as shown in the left image of Figure 3.4. Let $h[i]_0^L$ be the regional histogram, where L is the number of intensity levels in the image. Then we compute the reference threshold T_{ref} as follows:

$$T_{ref} = \max_t \ni \sum_{i=t}^{L} h[i] > wl \tag{3.3}$$

i.e. we select a threshold so that the summation of the histogram from $i = L$ to $i = T_{ref}$ is equal to or greater than the estimated area of the pattern of correct

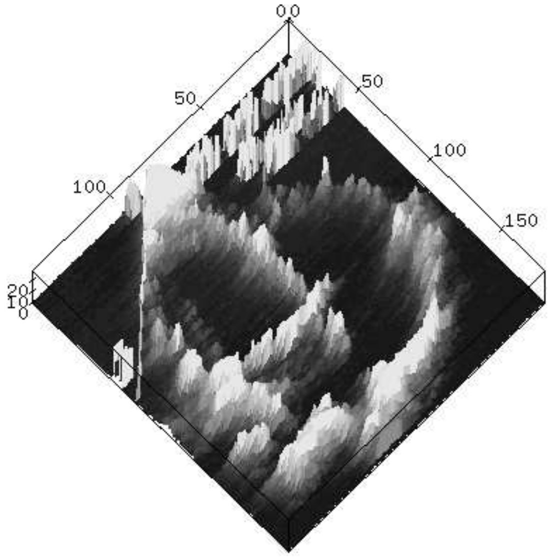


Figure 3.2: Three dimensional rendering of intensity map of a long-axis two dimensional echocardiogram.

The higher thresholds yield narrower patterns and missing regions, and the lower thresholds yield thicker patterns with extraneous regions. On the other hand to *extract* the object, the selected threshold should be such that the pattern is of proper thickness, and complete without misclassified regions. The proper selection of the threshold for extraction is usually domain dependent. For example, in line images, line extraction requires a threshold that extracts a thin strokes, and for ultrasound heart images, it is required that the threshold yield correct wall thickness.

3.2.1 Supervised selection of a local threshold

Although it is difficult to find a global threshold that meets the above requirements, it is often possible to use either a supervisory input or *a priori* knowledge of the domain to precisely select a regional threshold. For example, for line images or text images, it is often advantageous to extract thin line segments or strokes. Therefore,

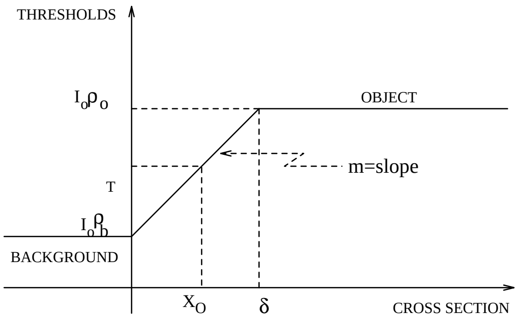


Figure 3.1: A schematic diagram of the ramp transition between object and background levels. I_o is the local illumination, ρ_o and ρ_b are the object and background reflectances respectively.

array [84]. In Figure 3.2 we show a three dimensional rendering of a two dimensional echocardiogram. Ideally, blood to wall transitions should have been abrupt, but the imaging process has smoothed the transitions. Therefore the thresholded pattern thickness increases when the selected threshold is decreased, as shown in Figure 3.3.

This model of the data enables us to propagate a selected threshold at one location to the rest of the image by selecting the thresholds proportional to the local slopes of the object-to-background transition.

3.2 Selection of a regional threshold

The developed image intensity distribution map implies that in images of a binary scene, the object and background are separated by a linear ramp transition with a slope proportional to the illumination. To *detect* the object pattern in these images, it is sufficient if we select a threshold anywhere on the ramp of transition between the object and the background. The threshold level controls the pattern size. Figure 3.3 shows that the width of the generated pattern is sensitive to the selected threshold.

brightness observed at a location does not change with change in the direction of observation[83]. Therefore the intensity at an image point can be approximated by:

$$I(x,y) = N_o + \int_{\Delta a_{(x,y)}} (\vec{I_o}(x',y') \cdot \vec{N}(x',y'))\rho(x',y')dx'dy' \tag{3.1}$$

where $\vec{I_o}(.)$ is the illumination over the corresponding incremental area $\Delta a_{(x,y)}$, with unit surface normal $\vec{N}(.)$, and N_o is the noise component, (x,y) and (x',y') are the corresponding coordinates in the image space and the object space. ρ is the object surface reflectance.

For document images, owing to relatively large source distances, and flat object surfaces, we approximate the above model by:

$$I(x,y) = N_o + \alpha I_o \int_{\Delta a_{(x,y)}} \rho(x',y')dx'dy' \tag{3.2}$$

where α is a constant, and I_o is the intensity magnitude.

The area $\Delta a_{(x,y)}$ is finite in practice, thereby linearly smoothing the changes in the reflectance ρ. In a binary scene, reflectance ρ_o of the object region differs from the reflectance of the background ρ_b. Ideally, we should observe a contrast or grey level step transition of $I_o(\rho_o - \rho_b)$ at all the pattern edges. From equation 3.2, we observe that the imaging process linearly smoothes this change in reflectance to yield linearly smoothed image intensity gradients with slopes proportional to the ratio of local illumination to the resolution, $(I_o/\sqrt{\Delta a})$. If we approximate the incremental area Δa by a square of side δ, then the projection of the ramp in the image plane is δ (see Figure 3.1). A similar linear approximation model can obtained for B-scan ultrasound image data, where sound beam of finite area of cross section scans an internal organ, and reflected sound energy is registered in a two dimensional can provide width information by indicating a section of the pattern. This method is useful where the geometric precision of the binarized patterns is important, and where the object regions in an image form elongated patterns.

In section 3.1 we develop a data model, and show that in an imaging system, a step transition is degraded to a ramp transition, and the gradient of the ramp is proportional to the local illumination (or regional time-gain-compensation for ultrasound images). Therefore, the width of the thresholded pattern is sensitive to the selected threshold, and proper positioning of the threshold is important to get the pattern of correct width. We describe computation of the correct local threshold in section 3.2. In section 3.3, we describe a method that propagates the regional threshold to the rest of the image proportional to the slope of the transition.

Thresholding often yields jagged edges and rough patterns. In section 3.4 we describe a morphological smoothing method that modifies the propagated threshold so that the edges of the generated patterns are smooth and noise specks on the background and holes on the object are suppressed. In section 3.5 we discuss the results, and conclude the chapter by summarizing the overall scheme (section 3.6).

3.1 Properties of the image data

In general, the intensity at a point in an image of an opaque surface depends on several parameters such as reflectance of the surface, brightness and directions of the light sources, distance of the surface from the image plane and the light source, and the direction of the surface normal with respect to light source and the camera aperture. If the object surface is opaque and is free from specular reflections, the

Chapter 3

Supervised Segmentation by Thresholding

Thresholding is a process of dividing the pixels in an image into two groups by comparing the intensity value of a pixel with a computed value, called the threshold. If the pixel intensity is greater than the threshold, the pixel is classified as an object pixel, otherwise it is classified as a background pixel. If the data is an image of a binary scene such as images of text or two dimensional echocardiograms, thresholding can be used to satisfactorily segment the entire image. Our goal is to generate an initial estimate of the object by thresholding, using the supervisor's specifications.

In this chapter, we describe a thresholding method that uses the pattern width at an indicated section of the image to determine the correct threshold value for that region. This regional threshold is then propagated to the rest of the image using a linear approximation of the image intensity gradients. For optical images of line diagrams or text, one can automatically select a section of the pattern of desired width. For complex images such as two dimensional echocardiograms, a supervisor and image energy terms as the constraints on the possible contour location and shape, and we select a smooth contour that satisfies the constraints. To find the optimum contour we enumerate the all the contours that satisfy the constraint, and select the one that passes through the most number of locally optimum edge points.

2.5 Summary of the review

In conventional image processing and computer vision research, researchers study the data and data domains and develop techniques to automatically solve well constrained problems with a reasonable accuracy, but little has been published in efficiently combining the expertise of both the user and the computer. In our review, we have covered the relevant automatic techniques, and also some interactive methods that are closely related to our goal of supervised image segmentation.

Several researchers have felt the need for a system that makes use of the complementary capabilities of the human expert and the machine. While psychologists and cognitive scientists are concentrating on the problem of efficient communication between man and the machine, many engineers and computer scientists have concentrated on automated segmentation, assuming that such algorithms will be installed in a library of tools. Our goal is to involve human interaction at the processing level, so that the developed tool can take advantage of the supervisor's viewpoint to improve the reliability of segmentation and segmentation refinement. Even though little has been published directly on this type of work, many earlier research shed light on the relevant issues.

and Jain [80]. A greedy algorithm for faster evaluation of the minimum energy contour is suggested by William and Shah [81].

In these methods the internal energy of the contour is defined by:

$$E_{int} = \alpha(s)|V_s(s)|^2 + \beta(s)|V_{ss}(s)|^2 \tag{2.5}$$

where $V_s(s)$ and $V_{ss}(s)$ are first and second order derivatives of the contour $V(s)$ with respect to the parameter s, and α, β determine the allowable bending and shearing at a point.

The image energy (E_{image}) is defined as a weighted combination of the three functionals:

$$E_{image} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term} \tag{2.6}$$

The weights were suitably chosen depending on the domain knowledge. The line energy term is proportional to the intensity of the point. For edge energy (E_{edge}) Kass *et al.* experimented with negative magnitudes of a simple gradient operator, and a zero crossing based operator. The termination energy (E_{term}) at a point is determined as an incremental change in intensity gradient perpendicular to the gradient direction.

The external constraint energy term E_{ext} is defined in terms of distances from the points indicated by the user. The energy magnitudes are proportional to square of the distances from the desired points, or to the negative square of the distance from an undesired point. Yuille et al. [82] have reported a similar work where *a priori* knowledge of the shape of the object is used for building deformable templates.

In our *constrained contouring* algorithm we treat the above internal, external through a graph formed by linking the edge elements according to a set of heuristic linkage rules [79]. These methods may not yield the desired boundaries because contours are often context dependent, and subjective [76; 27].

2.4.1 Deformable contouring

To generate satisfactory contours researchers have proposed providing external constraints and physical curvature restrictions to edge selection. Constraints are usually set according to *a priori* domain knowledge. Kass, Witkins and Terzopoulus [23] suggested a *deformable contouring* model to select the best set of edges from the all possible edge locations, so that a set of global constraints are satisfied. The basic premise of their work is that an approximate contour is placed near the desired boundary by some external mechanism. They use energy minimization method to refine the shape and location of the approximate contour under the constraints of connectivity, bending and domain dependent image features. Since our work closely relates to ideas of this work, we will briefly describe this work.

A contour is defined in the parametric form: $V(s) = (x(s), y(s))$, s in some interval I. The energy of a contour $E_{contour}$ is defined as a summation over the entire contour of three separate energy terms, namely, the internal energy term E_{int}, the external constraint term E_{ext} and the image energy term E_{image}.

$$E_{contour} = \int_I E_{int}(V(s)) + E_{image}(V(s)) + E_{ext}(V(s)) \tag{2.4}$$

The researchers have suggested finding a minimum energy ($E_{contour}$) contour by variational calculus technique [23]. A more efficient and robust way to achieve energy minimization by dynamic programming has been suggested by Amini, Weymouth distance measure.

2.4 Contour tracing

A contour is an outline of a region of interest, and it is useful for the analysis of shape, form, size and motion of an object. A contour is also useful for semantic interpretation of a scene. Although a contour is usually considered to pass through the points of change in brightness and color gradients, sometimes it also passes through the regions where there is no apparent change in image brightness or color, *e. g.* contours that extract the occluded objects, or the contours that extract textured regions. Furthermore, contours are sometimes subjective [76], and are influenced by the context of the image [27]. In this thesis, we describe a method to trace closed contour of a region of interest. The contour is constrained by the connectivity, smoothness, contextual and subjective constraints. The subjective constraints are provided by an approximate contour input by the supervisor or by an external process.

A conventional edge operator that detects the change in the brightness or color gradient, facilitates only a partial detection of a contour. Edge operators are regional, and it is difficult to design them to to extract edges that conform to a global shape. Therefore, an edge operator can not satisfactorily contour a desired region. However the output of a suitable edge operator can be used to reduce the search space for selecting a satisfactory contour. The early work in contour tracing involved search for connectivity in the neighborhood of edge points and refining them [77; 78]. In several other methods, a contour is viewed as a heuristically selected path

Gambotto and Monga [75] proposed a parallel hierarchical algorithm for region growing. Initially they consider each pixel as a separate region and combine the adjacent regions by comparing the mean gray level of the region with merge measure. The merge measure is iteratively updated to control the growth. They also suggest an intuitively appealing termination criterion that, "any two regions separated by a significant intensity gradient at their boundaries are less likely merge if they were uniform, than if they were noisy."

There are three main issues involved in region growing algorithms that are not readily solvable. They are:

1. *selection of representative sample of the region:* Researchers have used single pixel to large chunks of image as the starting blocks to grow the regions.

2. *selection of a distance measure:* A region is grown by combining with a neighbor that is short distance away, but it is not clear what measure, and how it should relate to the type of the data.

3. *termination condition:* The region growing terminates when all the pixels belong to one region, or when all the neighboring pairs of regions have distance measure higher than a threshold. The value of this threshold is usually determined empirically.

Over the years, researchers have suggested many heuristic and empirical methods to shed light on these three important issues in region growing. In our research, region growing is controlled by an external supervisor, hence we can develop good ways to handle the issues (1) and (3). We suggest a new set of heuristics for the age [69], hybrid linkage [70], and centroid linkage algorithms [71]. Single linkage region growing schemes regard each pixel as a node in a graph. Neighboring pixels whose properties are similar are joined by an arc, and all connected pixels form a region. In hybrid linkage region growing schemes, all non-edge pixels are assigned a property vector and neighboring pixels who have similar property vector are linked together. These methods are more noise tolerant than the single linkage methods. The centroid linkage region growing methods do not compare the neighboring pixels, but images are scanned in a predetermined order and if the pixel is similar to the mean of an already created region, it is added to that region and the mean is updated. If there are more than one region, then the pixel is added to the most similar region. If the pixel is not similar to any region, a new region is created with the pixel as its first member. Haralick and Shapiro [68] conclude from their survey that the region growing techniques can be made more powerful if the similarity measures are based on some statistical equality test.

Recently, there have been several interesting works that are not covered by these surveys. Zamperoni [72] proposed a region growing operator that simulates a local competition process between pixels. Rafaat *et al.* [73] use texture information measure together with a texture distance measure to direct the growth of various textured regions. Pavlidis and Liow [74] integrate the split and merge algorithm with edge detection for image segmentation. They devise a clever merge technique that detects the artifacts generated by the 'split' procedure, and merge the neighboring regions if the intermediate edges and artifacts are caused only by the split operation.

$S(t)$

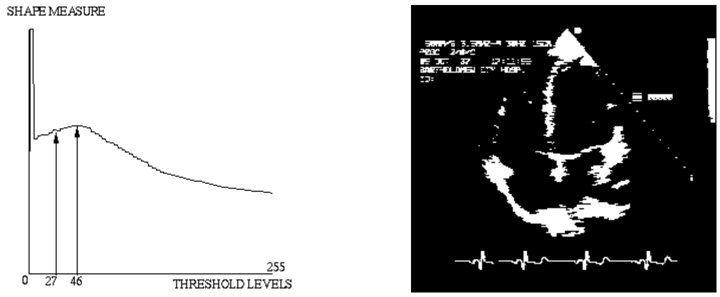


Figure 2.3: Evaluation of the shape measure for a two dimensional echocardiogram. The optimum global threshold is at 27 and optimum shape measure is at 46. The image is thresholded at threshold t=46.

measurements of the extracted patterns is of diagnostic interest, and to the fields that involve extraction of lines and strokes from digitized images of line drawings and text.

2.3 Region growing

A region in an image is characterized by a set of contiguous points with a common property. If a characteristic region is known, the bounds of the entire region can be determined by collecting all the connected points that have the same property. This method of segmenting a region is called *region growing*. Zucker [62] has surveyed the research in region growing methods. He classifies the approaches into five categories, namely: 1) Regional neighbor search [63], 2) Multiregional heuristics [64], 3) Functional approximation and merging [65], 4) Split and merge [66], and 5) Regional interpretation and semantics [67].

In another survey, Haralick and Shapiro [68] give a broader interpretation to region growing. They view that the general problem of image segmentation is solved by various forms of region growing. They classify region growing into single link-measure $S(t)$ at threshold t is given by:

$$S(t) = \frac{\sum_(x,y)Sgn(f(x,y) - \overline{f_N(x,y)})\Delta(x,y)Sgn(f(x,y) - t)}{C} \tag{2.3}$$

where,

$$Sgn(x) = \begin{cases} -1 & x < 0 \\ 1 & otherwise \end{cases}$$

and $\overline{f_N(x,y)}$ is the average grey level of the neighborhood.

The shape measure expects a threshold such that all the points with positive gradient be above the threshold, and the points with negative gradient be below the threshold. The term *shape measure* seems to be a misnomer, because it depends mainly on the assumption that regions are uniform and the grey level gradients in the image correspond to the transitions between the two regions to be separated. Therefore, not surprisingly the optimum shape measure threshold generates the image regions similar to the ones generated by the optimum uniformity measure threshold. In Figure 2.3, we show the variation of the shape measure for a typical echocardiogram and the thresholded pattern generated by the optimum shape measure global threshold. The figure also shows that the user specified optimum global threshold of Figure 2.1 has a lower shape measure.

The above study of uniformity measure and shape measure that evaluates the detection capability of a threshold, shows that these measures may not be applicable for certain classes of images. In this thesis, we describe a method that adaptively propagates a local threshold over the entire image so that the entire pattern of correct width is extracted. Our method facilitates extraction of the patterns from images whose object regions are not necessarily predominantly uniform. This method of extraction is advantageous to fields such as medical image processing where direct

$$1 - \frac{\sigma_1^2(t) + \sigma_2^2(t)}{C}$$

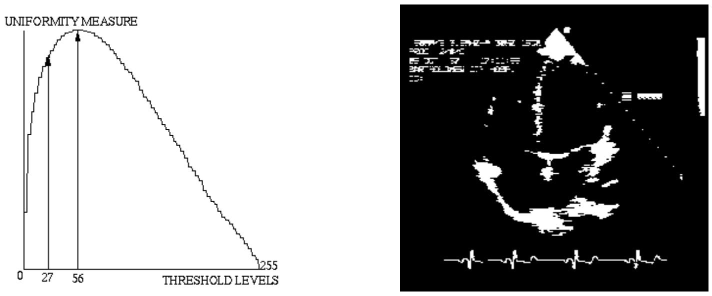


Figure 2.2: Evaluation of the uniformity measure for a two dimensional echocardiogram. The optimum global threshold is at t=27 and the optimum uniformity measure is at t=56. The thresholded image at t=56 is shown in the right.

uniform areas, but it may fail to properly evaluate the goodness of the threshold for the images with nonuniform background or object regions. In Figure 2.2, we show a curve tracing the *uniformity measure* for all possible thresholds for the image displayed in Figure 2.1. The criterion is well behaved with a unique maxima, but the threshold selected at the maxima does not yield satisfactory segmentation. Although at this threshold the object pattern is detected and the regions have maximum uniformity, the shape and size of the pattern is not be preserved. The figure also shows that the optimum user specified threshold of Figure 2.1, has a lower uniformity measure.

Shape criterion

To account for shape inaccuracies, Sahoo *et al*[24] suggest a *shape measure* for threshold evaluation. They assume a (3X3) neighborhood centered around the test pixel (x, y) with grey value $f(x, y)$, and define a general gradient $\Delta(x, y)$, as the root-mean-square vector sum of the grey level difference in all four directions. The shape

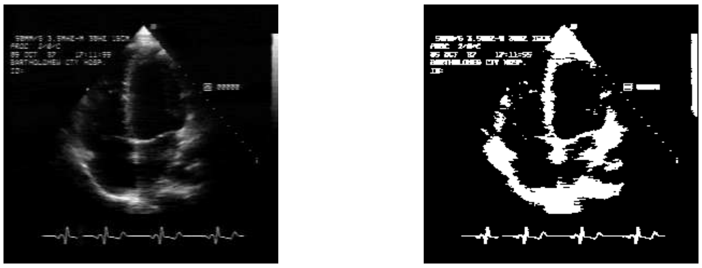


Figure 2.1: A typical two dimensional echocardiogram, and its thresholded pattern generated by user specified optimum global threshold, t=27.

chamber, long axis echocardiogram image, and the corresponding binary pattern generated by the user specified optimum global threshold. The threshold was selected interactively by observing the thickness of the detected wall pattern. Though the threshold is inaccurate at certain locations, it does preserve the shape and size of the heart wall pattern. We study the threshold evaluation methods with reference to this image and the threshold.

Uniformity criterion

Sahoo *et al*[24] define the uniformity measure as follows:

$$U(t) = 1 - \frac{\sigma_1^2(t) + \sigma_2^2(t)}{C} \tag{2.2}$$

where $\sigma_1^2(t)$ and $\sigma_2^2(t)$ are the variances of the two segmented regions generated by thresholding at t, and C is a normalization constant. This measure is adapted from the work of Levine and Nazif [61]. According to this measure, a thresholding method is *uniformly good* if it selects a threshold such that each region is as uniform as possible.

This criterion evaluates the goodness of a threshold for the images with large is proposed by Ahuja *et al.* [55]. They use the features of neighborhood gray level patterns for pixel classification. The feature vectors are computed from a known set of neighbors. In addition to the image features, they also use contextual and texture information for image segmentation [56; 57; 58]. These methods have proved to be useful for interpretation of complex images.

The surveyed thresholding methods mainly concentrate on pattern *detection* problem. In contrast, the *extraction* problem is determination of the geometry of the desired object in the image. Several applications require that the geometry of the thresholded pattern be directly proportional to the object geometry. The thresholding algorithms for such applications must address both the detection and the extraction problems. For example, in echocardiography cardiologists obtain the heart wall thickness, ventricular volume, etc. by direct measurement of the pattern extracted from a two dimensional echocardiogram [59]. Furthermore, in line images, it is often advantageous to extract a thin line pattern [60]. In this thesis we develop an adaptive threshold propagation method to *extract* the pattern from images with line or elongated patterns.

2.2.1 Evaluation of thresholding methods

Sahoo *et al*[24] provide an excellent survey of many recent thresholding methods. They also suggest two measures, *viz. Uniformity measure* and *Shape measure* to evaluate and compare various thresholding methods. We briefly study these two measures, and highlight the need for a thresholding method for *pattern extraction* in contrast to conventional *pattern detection*. In Figure 2.1, we show a typical four than a specified constant. Ogawa *et al.* [50] in their Chinese character recognition system proposed an algorithm, where a set of neighbors are chosen depending on the *a priori* line width information. They classified a pixel as an object point, if it is darker than the average gray value of the neighbors, by more than an experimentally selected constant.

The loss of information because of digitization of a line image is often aggravated by thresholding. This leads to the formation of tailed and jagged patterns during thinning of the thresholded pattern. To reduce this degradation, a method of double adaptive thresholding is proposed by Watson *et al.* [51]. One more local method that has more of global flavor is proposed by Ridler and Calvard [52], where an iterative technique is used for the selection of a threshold. Wojcik [53] proposed a new thresholding scheme, where the existing gray value of a pixel is replaced by the maximum homogeneous gray value, which is found by a rotating window technique. The method tends to blunt corners and sharp curvatures. The performance of the method can be improved by giving weights to the pixels in the window based on a proximity measure.

In dynamic thresholding schemes, the threshold at a point depends on coordinates of the point, in addition to its own gray value and the gray values of a set of neighbors. Chow and Kaneko [54] compute the threshold at a point depending on its proximity to boundary points. The threshold for these boundary points is determined by local histogram analysis.

A recent trend in image classification is to use information integration, which is how the biological vision is assumed to be working. An approach in that direction

Weszka et.al. [43] used a pure Laplacian operator to define the edge value. Since these are second derivative operators, they have a zero value on a linear ramp formed at the object/background boundaries, but high edge value on the shoulders on either side of the ramp. Thus the points having high Laplacian values will be adjacent to, but not on, the boundaries. The histogram of high edge value pixels should now have two peaks representing the shoulders of the boundary and a deep valley representing the boundary.

All the above described global methods assume that the object and the background gray levels are uniform in their respective regions. This is not true for many real images, especially for the images of hand-written characters or hand-drawn line sketches, because of uneven stress put by hand. Global thresholding of such images may lead to generation of disconnected binary patterns. Moreover, all the global methods are sensitive to noise, because they do not exploit the gray level uniformity around a pixel. To alleviate such draw backs, local thresholding schemes were proposed. In these schemes a pixel is classified by comparing its gray value with a value computed from a set of neighboring gray values. The spatial extent of this neighborhood is called a window. The size and shape of the window varies from method to method. Some methods assume minimum width of the object pattern and some are selected by experimental studies. A review of such schemes is given by Weszka [38] and Ullmann [48].

Wolfe [49] assumed that in text images, width of a limb does not exceed a known limit. He proposed that a pixel is an object pixel, if it has a gray value higher than the gray values of a predetermined pair of neighboring pixels, by more the optimum threshold using discriminant analysis [42]. The discriminant measures maximize the separability of the resultant classes of pixels. The procedure uses the 0th and the 1st order cumulative moments of the image histogram. Hence the computation time is linear function of the size of the image.

In cases of out-of-focus images, or images with object areas very small compared to the background area, bimodality of the image is not distinct. It is difficult to locate histogram valleys in such images. To overcome this difficulty, in some of the global methods, researchers study local properties of the image [43; 44]. These methods transform the image histogram so that the valley gets enhanced.

In determining how each point of the image should contribute to the transformed histograms, the rate of change of gray levels around a point along with the gray level at that point, are considered. Normally, a transition in gray levels occurs at the edges common to the object and the background. Therefore the rate of gray level transition is also called *edge value.* Edge values are computed using the edge operators such as Laplacian, Robert Cross, DIF1(maximum difference of average gray level in pairs of horizontally and vertically adjacent 2-by-2 neighborhoods), etc. [45]. The points at the interior of the object or at the background, usually have low edge values because of regional uniformity. On the other hand, the points on the object-background boundaries have high edge values. Thus, if a histogram is obtained for only low edge value pixels, histogram peaks remain the same while the valley becomes prominent [44]. On the other hand, if a histogram of pixels having only high edge values will have a peak at the valley of the image histogram [46; 47]. The modified histograms can then be searched to determine the threshold.

In general, a threshold $T(.)$ at any point can be expressed as a function of the gray level at that point, its neighbors, overall gray level distribution of the image, and the spatial location of the point under consideration.

$$T(x,y) = f(g(x,y), N(x,y), I_n, x, y) \tag{2.1}$$

where f is the thresholding function, g(x,y) is the gray level of the pixel at (x,y), N(x,y) is a set of neighbors of pixel (x,y), and I_n is the gray level distribution of the complete image. If $T(x,y)$ is a function of I_n only, then it is called a *global threshold.* The global threshold for any level is computed only once for the entire image. If $T(x,y)$ is a function of $g(x,y)$ and $N(x,y)$, then it is called a *local threshold.* If $T(x,y)$ is a function of the coordinates (x,y) of the pixel under consideration, it is called a *dynamic threshold.* Many recent methods use a combination of these techniques.

A global technique based on prior information of object area of the image is proposed by Doyle [39]. It was developed for thresholding similarity invariant patterns. Here a threshold is selected such that only a certain number of pixels equivalent to the known object area have gray level greater than the threshold. This method is obviously not applicable, if the object area is unknown or if the area varies from picture to picture.

For segmenting white blood corpuscles, Prewitt and Mendelson [40] chose the threshold at the valley of the image histogram. This technique is called the mode method. The histogram is smoothed to remove spurious modes and valleys, and threshold is selected between two prominent modes. Otsu [41] has proposed a method along this idea that can be extended for multilevel thresholding. He selects we review each of the three major components of our research, namely *thresholding, region growing and contour tracing* separately.

We use thresholding to generate an initial segmentation of the data with aid of a simple user interaction. This segmentation is refined by the user through simple hand gestures that are mapped to region growing and contour tracing algorithms. The following three sections review the published research on these methods.

2.2 Thresholding

Thresholding is a popular way to segment an image of a binary scene. The segmentation is carried out by classifying the pixels in an image into two regions by comparing their intensity with a computed value, called the threshold. Thresholding schemes can be broadly classified into conventional schemes and information integrating schemes. Conventional techniques look for either maximal regions satisfying some homogeneity criterion or edge information between the regions, whereas the information integrating techniques try to emulate the biological system approach for processing visual data. In the information integrating techniques, information from various related sources are used to put constraints on the data to be processed, to obtain an unambiguous result. Weszka [38] surveys the conventional thresholding schemes and categorizes them as follows:

1. Global Thresholding
2. Local thresholding
3. Dynamic Thresholding.

matical model of the tract of interest.

There has been a wide spread interest shown in interactive systems by cognitive scientists and man-machine communication researchers, for efficiency of communication and user friendliness[31; 32; 33; 34]. Shortlife[31] in his work on medical consultation systems, argues that doctors/expert-users want to be the sole decision makers, but are motivated to use computer assisted systems to improve accuracy, reliability, efficiency, and also to improve selection of tests and therapies. In the collection of research edited by Carroll[33], researchers have extensively studied the role of psychology in designing man-machine interface dialogs. Manhartsberger[35], Riecken[36], Rogers[37] and Norcio[34] have suggested development of a user model for friendly and efficient interfaces. All these research activities show the importance of using human involvement in automatic systems that make critical decisions.

Conventionally in interactive systems a user is provided with a set of image processing algorithms that she is expected to use to aid in her decision making[lewin-75]. These systems require that the users are familiar with various image processing algorithms and their effect on the image data. This is often a unrealistic requirement, when users are not trained to think in terms of digital images and computer algorithms.

In this thesis our intention is to develop a supervisory system that receives the input from the user as a set of natural gestures, and a computer algorithm maps these gestures to relevant image processing tools that account for both the image data and the approximations and intentions of user's gestures. Little has been published that directly relates to our proposed supervised image segmentation scheme. Therefore

2.1 Supervised image segmentation systems

Conventionally computer vision and image processing research develops automatic methods that provide satisfactory solutions to the problems involving two or three dimensional images. Pavlidis [26] did a critical survey of these methods. He reveals that *"in spite of increased understanding of the nature of images, we have been very slow in integrating the results into useful image analysis programs."* The major reason behind such a lack of progress could be attributed to the complexity of the task, and incomplete and fuzzy specifications of the problem. The image analysis tasks are further complicated because of their strong context sensitivity. Toussaint [27], in his award winning article gives an excellent description of context sensitivity of vision problems.

Owing to the complexity of the task and difficulties involved in precise specification of the task, over the years researchers have proposed using human interaction[28; 29; 30] to solve vision problems. These early systems rely heavily on user interactions. The image processing methods are provided in a library of image processing tools that could be invoked by a user [28]. Since analyzing an image may involve a sequence of such invocations, Billingsley *et al.* [28] propose a language that facilitates succinct communication. Orhaug *et al.* [29] envisage a dedicated special purpose computer so that time critical interactions are carried out with short turnaround times. In addition to providing the library of tools, they envisage user feed back at the imaging stage too. Linkens [30] proposed a system where the user visually studies the medical data to find out signal parameters such as wave-shape and frequencies of the data, and sets the corresponding parameters in the mathe-

Chapter 2

Review of Supervised Image Segmentation Research

Little has been published in supervisory control of image segmentation. Therefore we study the systems that involve human interaction, and study the image segmentation techniques separately.

In section 2.1 we study the systems that facilitate human interaction in digital image processing in general. The techniques of thresholding, region growing and contour tracing, form the important components of our research. We critically review the published methods in these three areas in sections 2.2, 2.3 and 2.4. While there are no well known methods to evaluate region growing and contour tracing techniques, researchers have proposed methods to evaluate a selected threshold [24; 25]. We critically study the evaluation method proposed by Sahoo *et al.* [24], and show that the best threshold suggested by this evaluation may not be adequate for all data domains. We summarize the chapter in section 2.5.

1.6 Outline of the dissertation

In chapter 2, we review the various published methods in supervised image segmentation in general, and threshold based segmentation methods in particular. The review also covers the relevant contouring, and region growing methods. This review will introduce the reader to various techniques proposed by the researchers, and highlight the major difference between the methods developed in this thesis and the published methods.

In chapter 3, we describe a adaptive threshold propagation method that makes use of the local *a priori* knowledge to compute the suitable threshold values for the rest of the image, and for the other image frames in the same temporal sequence.

We describe the algorithms for supervised refinement of the segmentation, in chapter 4. This chapter contains two major sections, one describing our region growing technique that detects the precise extents of a pointed region, and the other describing our constrained contouring algorithm that generates the extents of a region indicated by circling.

In chapter 5, we conclude the dissertation with a discussion on major contribution of the research, and its future directions. WE have applied the constrained contouring algorithm to Venu's [91] human face location results, and show the results in appendix-a.

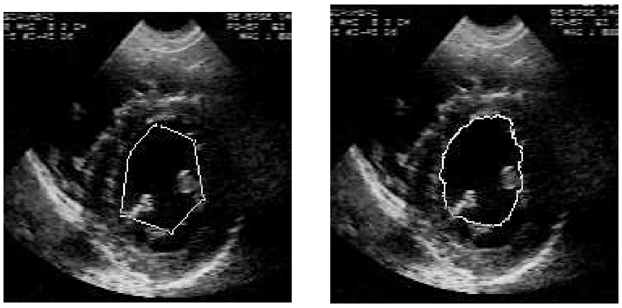


Figure 1.7: A short-axis two-dimensional echocardiogram with the circled input is on the, and the right is the corresponding refined output.

2. a set of Locally Optimum Contour Locations (LOCL) is determined in all the radial directions according to the constraints of context(shape), distance from the circled approximate contour, and the relevant image features.

3. a contour that passes through the maximum number of LOCLs, while satisfying the imposed smoothness constraint is the contour of the intended region.

We have developed a graph search based algorithm that selects the optimum contour. The detected LOCLs are represented as the nodes of a graph of all possible contours. The links of the graph are weighted so that the shortest path between a pair of nodes is a *smooth* contour that traces the maximum number of intermediate LOCLs, and the shortest cycle in the graph gives the optimum contour.

In Figure 1.7 we show processing of a short-axis two-dimensional echocardiogram. The left image shows the circled input, and the right image shows the refined boundary of the intended region.

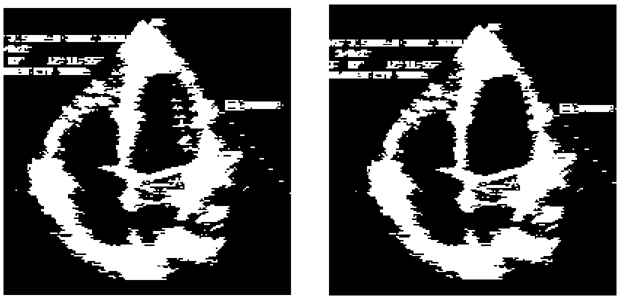


Figure 1.6: The segmentation of a two-dimensional echocardiogram on the left is refined by pointing. The refined output is shown in the right.

influenced not only by the image features such as intensity and intersecting edges, but also by the supervisor's expectancy. In Figure 1.6 we show a segmented two dimensional echocardiogram in the left, and the corresponding refined output in the right.

1.5.2 Segmentation refinement by circling

If the initial segmentation is erroneous for the large part, or if there are too many isolated misclassified regions, the supervisor may want to refine the region by circling. This gesture provides a rough approximation of the boundaries of the indicated region. Our goal now is to modify this input so that it traces the precise boundaries of the intended region. This problem relates to the problem of *active contouring* discussed by Kass *et al.*[23].

We have developed a constrained contouring method that analyses the data in polar coordinates, and traces the precise boundaries of the indicated region. Our constrained contouring method can be summarized by the following three steps:

1. data is transformed into polar coordinates.

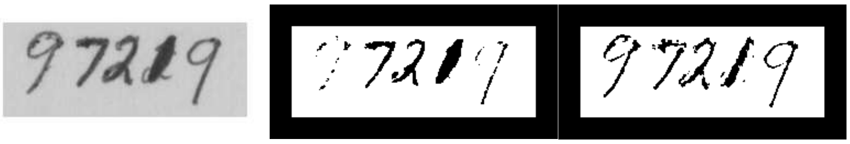


Figure 1.5: Optically scanned image of hand written zip-code (left), center image is the thresholded pattern generated by a high threshold that yields thin limbs. The right image is the output generated by our adaptively propagated threshold, which yields continuous thin limbs.

update the segmentation according to suggested corrections. For a dynamic scene, the updating is effected on all temporal frames in the sequence. To handle temporal propagation of corrections, while refining by pointing, the data is treated as a single block of three-dimensional data. On the other hand, while refining by circling, the corrections are propagated by using the refined output at one frame as the circled input for its neighboring frames.

1.5.1 Segmentation refinement by pointing

If in the initial segmentation there are only a few misclassified regions, it is natural for the supervisor to point at the regions and suggest that the regions be reclassified. But these gestures are only approximate. The goal of refinement by pointing is to find the boundaries of the pointed locations and reclassify them as suggested.

From the visual attention map we know that there is a region in the vicinity of the pointed location that characterizes the precise region to be reclassified. This type of problem is conventionally solved using *region growing* techniques. In this thesis we develop a new region growing algorithm to find the bounds of the pointed region. The algorithm uses the visual attention map for weighting the distance measure used for gathering the points in the region. Hence the region so gathered is features. The deformation of the curve away from this band of focus is partly based on both the existence of image features and distance from the initial location.

1.4 Interactive image segmentation algorithm

In general, it is hard to determine a single global measure that can segment a region of interest. On the other hand, at a small region in the image, it is possible to find a good local measure that can segment the region correctly. For example, in two dimensional echocardiograms, or for line or text images, it is hard to estimate a good global threshold, but it is possible to determine a precise threshold at a wall section or a line segment whose orientation and thickness are known. The segmentation algorithm developed in this thesis exploits available pattern thickness information to determine a local reference threshold, and propagates the reference threshold to the rest of the image, by accounting for the regional illumination changes. Figure 1.5 shows comparative results in processing a hand written text image. The left image when thresholded at a high level generates the output shown in the center image. The high threshold is chosen so that it yields the text patterns of desired thickness. The right image is the output generated by the adaptively propagated thresholding. The output of the adaptive thresholding is of the same thickness as obtained from the high threshold, but is more continuous.

1.5 Supervised segmentation refinement

If some regions in the initial segmentation are misclassified, the supervisor indicates those regions either by *pointing* or by *circling*. The supervised refinement algorithms eyes.

While the above methods studied mainly the luminance detection paradigm, Downing [8] studied the visual attention maps for luminance detection, brightness discrimination and form discrimination paradigms. Her study revealed that visual sensitivity to these tasks was maximal where a target stimulus was expected and generally decreased with distance from that location.

1.3.3 Computational adaptation of visual attention map

We have combined the spot-light and gradient models of the visual attention maps proposed in psychological and psychophysical studies in my supervised image segmentation algorithms. Figure 1.4, displays a schematic plot of the attention response curve that combines the spot-light and gradient models. A linear approximation of this response curve is used as a constraint in the supervised segmentation refinement algorithms.

For refinement by pointing, we consider the pointed location as the point of focus. All the points in the spot-light characterize the indicated region. The region growing algorithm then collects all the neighboring points that have similar characteristics, but the measure that is used to test the similarity is weighted by attention map distribution curve.

For refinement by circling, we consider that the supervisor's point of focus traces the input curve. Therefore there is a small band of region surrounding this curve within which if the curve is displaced, the supervisor will not be surprised. In other words we can deform the curve within this band of focus purely based on image

Preview of the studies in visual attention models

Posner *et al.* [14] and Hoffman *et al.* [15] experimentally showed that there is a discrete bounded region around the point of attention, called the *spot-light*, with an uniform reaction time. The angle subtended by the spot-light is about 1^o at the viewer's eyes [16]. Later Ericsen [17] and LaBerge [18] showed that the spot-light has flexible boundaries and its size is task dependent.

Hughes *et al.* [19] examined the spatial extent of the visual attention by inducing subjects to expect the target at a location and introducing occasional probe flashes at other locations throughout the visual field. The main conclusion of their study was that the directed visual attention was not a spatially limited phenomenon, but it extended throughout the visual hemifield. It was independent of both the expected target locations and the validity of precue. In addition, they supported the spot-light theory by showing that the field of attention had uniform reaction time.

A psychophysical analysis of the visual attention span is reported by Mangun *et al.* [7], where they studied the changes in the event related brain potentials (ERPs) with respect to visual stimuli as the locus of attention was shifted across the visual fields. They observed a decline in attention indicated by decline in the amplitude of the sensory evoked components of the ERPs, at increasing distances away from the attended location. Their study supported both the spot-light and the gradient models of the visual attention maps. Additional evidences to support these models are provided by Shulman *et al.* [20] and Bashinski [21]. Tsal [22] showed that the decrease in attention was linear with respect to the angle subtended at the viewer's

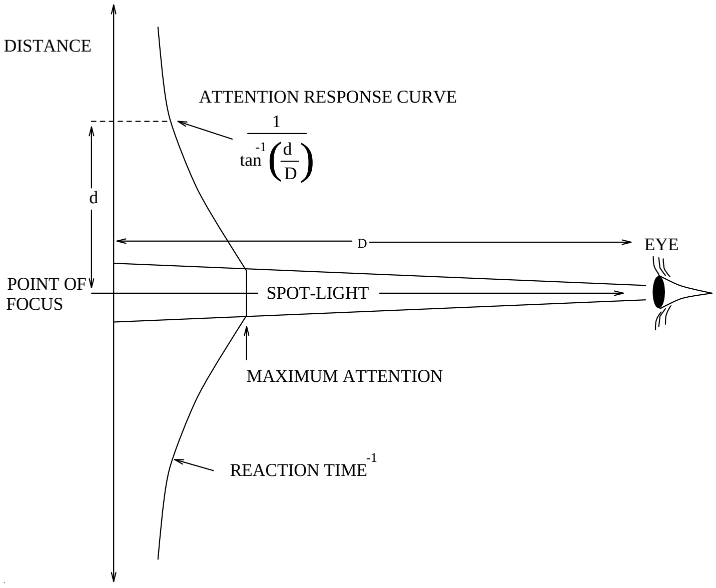


Figure 1.4: A schematic curve showing the human attention distribution.

1.3.2 Human visual attention model

An assumption central to the current view of visual expectancy effects is that the attention mechanism is limited in its spatial extent. It is usually characterized by the reaction time. A slower reaction time indicates an unexpected region and a faster reaction time indicates the attended region. Figure 1.4 displays a schematic plot of the attention response curve. The curve shows small reaction time at the point of focus, and it is constant over a small discrete region around it. This constant focus area is called the *spot light*. The reaction time increases linearly with increase in the angular distance from the point of focus. We will briefly cover the relevant readings in the following section.

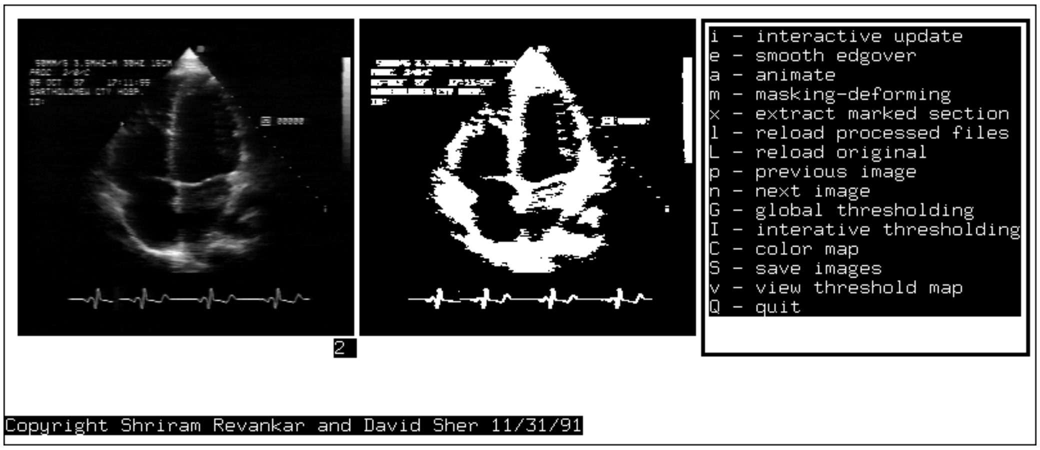


Figure 1.3: Snap-shot of the Graphics User Interface (GUI) display.

left subwindow displays the original image frames, the center window displays the corresponding segmented image and the right window provides a menu of operations that facilitate generation of segmentation, and its verification and refinement. These features satisfy the thumb-rules suggested for a good GUI [11; 12].

1.3.1 Interaction through pointing and circling

Interactions are mainly mouse-based. A mouse facilitates accurate cursor positioning. Reinhart and Marken [13] have recommended mouse as the most satisfactory general purpose pointing devise.

Circling is a special case of simple mouse dragging. Recently research has been done to recognize the selection based on simple circling, without button press [5]. In our implementation, we simulate circling by a sequence of mouse clicks.

The supervisor can iteratively use both pointing and circling to achieve the desired result. In a dynamic scene we have a temporal sequence of frames. Usually corrections are made in one of the frames, and the corresponding refinement is propagated to the rest of the frames according to the inter-frame temporal correspondence. For example, the fragments inside the chamber in the upper frame of Figure 1.2-(2) move along with the wall, hence it is necessary to establish a temporal correspondence for a successful deletion of the artifacts in all the frames.

1.3 Graphics user interface (GUI)

Although my research concentrates on developing the necessary segmentation algorithms, a significant effort is put towards developing a friendly interface for facilitating interaction through natural gestures such as pointing and circling [5; 6], and by incorporating a supervisor's visual attention model [7; 8], that accounts for the spatial extents of the gestured corrections and visual expectations. This facility makes the interface user friendly because:

- pointing and circling are easy to use and easy to learn mouse gestures [6; 5],
- the gestured input need not be precise, and
- the supervisor's expectations are fulfilled, because the results are influenced by the human attention distribution model [9; 10].

A snap-shot of the GUI display is shown in Figure 1.3. The interface is developed in the X-Window environment. The main display window contains three permanent subwindows. Several other temporary subwindows facilitate a friendly dialog. The this, we have developed a segmentation method that uses the supervisor's input to compute precise threshold values at all the image points. Initially, the supervisor indicates a section of the wall where the thickness of the wall is known *a priori*. A local threshold is computed at the interactively suggested wall section, and it is adaptively propagated to the rest of the image, and the rest of the frames of the sequence, if it is a dynamic scene. This threshold gives a good estimate of the walls for the most part as shown in the pair of corresponding frames in Figure 1.2-(2).

The upper image in the Figure 1.2-(2) pair shows a clear contiguous top-right chamber (left ventricular chamber), while showing some fragmented regions inside it. These fragments are quite often extraneous reflections caused by the overlapping papillary muscles, and the endocardium. If the supervisor is convinced that they are extraneous, he just points at those fragments using mouse-clicks as shown in Figure 1.2-(3), and a *region growing algorithm* deletes those regions and displays the refined output as shown in the upper frame of Figure 1.2-(4).

On the other hand, if the initial segmentation yields poor results, as shown in the lower frame of Figure 1.2-(2), supervisor may want to delineate the entire chamber. If so, she circles the chamber and indicates that a contiguous internal contour be obtained. (We have simulated the circling by allowing the supervisor to select a set of points by mouse.) A *Constrained Contouring* algorithm uses that circled input as the approximation of the precise contour, and generates a refined contour that closely approximates the precise contour. In the lower frame of Figure 1.2-(3) shows the approximate input contour, and Figure 1.2-(4), shows the corresponding final contour.

with the original images on a graphics display. If the segmentation is satisfactory, the supervisor uses it directly for further processing. On the other hand, if the supervisor finds the errors in segmentation, she indicates the corrections through simple hand gestures, and the refinement algorithms refine the segmentation to generate better approximations.

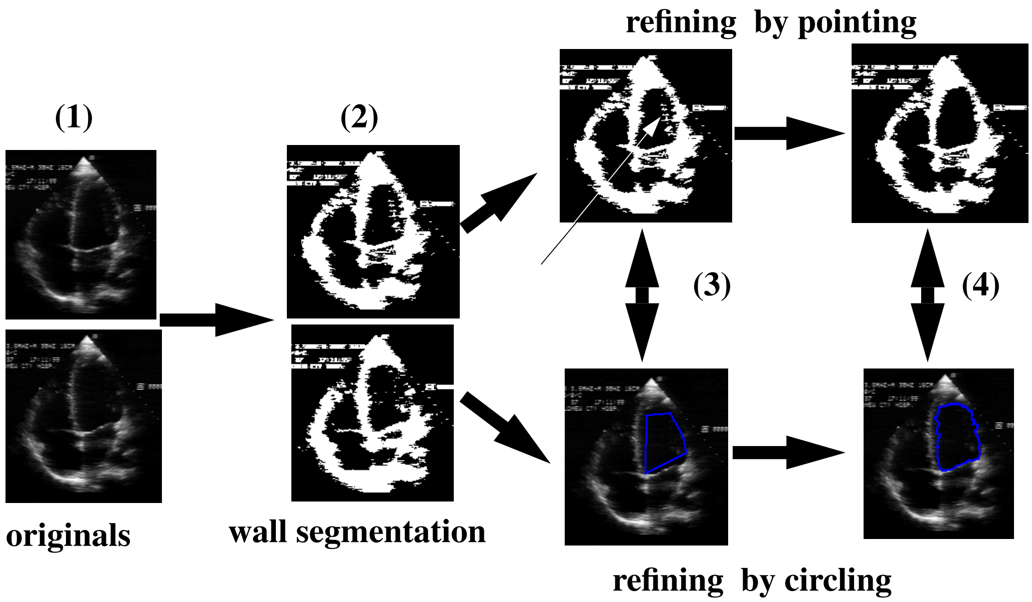


Figure 1.2: Sequence of stages in supervised segmentation of two dimensional ultrasound images of human heart.

A snapshot of the several stages of processing done on two different frames of two-dimensional ultrasound heart images is shown in Figure 1.2. Figure 1.2-(1) shows a pair of two dimensional long-axis echocardiograms. The goal is to separate heart walls from the background. Since the thickness of the extracted pattern is used in the estimation of diagnostically interesting parameters such as wall thickness, volume of a heart chamber, amount of blood transfer, etc., it is necessary that we segment the walls so that they give precise estimate of the thickness. To achieve and precise image segmentation. *In the described supervised segmentation scheme, a human supervisor with extensive knowledge of the data domain, initiates the segmentation process, and later, if required, interactively refines the computer generated segmentation, until it is satisfactory.*

1.2 Overview of the supervised segmentation scheme

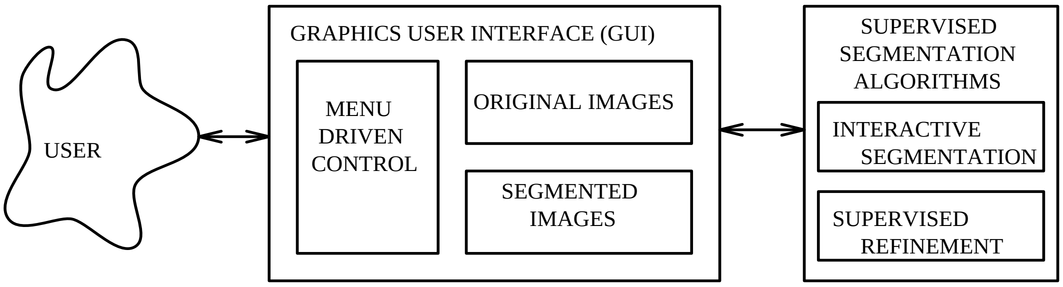


Figure 1.1: Schematic overview of the supervised image segmentation scheme

In designing a supervisory image processing system one has to give emphasis on two major issues namely,

- system design for friendly interaction, and
- developing the associated image processing algorithms to execute the interactively suggested tasks

A schematic overview of the developed supervised image segmentation scheme is shown in Figure 1.1. As mentioned, the overall system consists of two major components, *viz.* the *Graphics user Interface* (GUI) and the *supervised image segmentation* algorithms. The supervisor initiates the segmentation process by simple mouse based selection, and observes the computer generated segmentation results along

Table 1.1: **Relative capabilities of humans and machines.**

(Reproduced from Marlin Brown's book – General human-computer interface concepts.)

Humans generally better	**Machines generally better**
Sense low level stimuli	Sense stimuli outside human range
Detect stimuli in noisy background	Count or measure physical quantities
Recognize constant patterns in varying situations	store quantities of coded information accurately
Sense unusual and unexpected events	Monitor prespecified events especially infrequent ones
	Make rapid and consistent responses to input signals
Remember principles and strategies	Recall quantities of detailed information accurately
	process quantitative data in specified ways
Retrieve pertinent details without a priori connection	
Draw upon experience and adapt decisions to situation	
Select alternatives if original approach fails	
Reason inductively: generalize from observations	Reason deductively: infer from a general principle
Act in unanticipated emergencies and novel situations	Perform repetitive pre-programmed actions reliably
	Exert great, highly controlled physical force
Apply principles to solve varied problems	
Make subjective evaluations Develop new solutions	
Concentrate on important tasks when overload occurs	Perform several tasks simultaneously
	Maintain operations under heavy information load
	Maintain performance over extended periods of time
Adapt physical response to changes in situation	

accuracy of its approximation and its error rate.

Completely automated image segmentation systems have been used satisfactorily in the fields where a *specified error rate* is acceptable. For example, the applications such as automatic sorting of mail [1; 2], barcode reading in supermarkets[3] have been shown to benefit from automated decision making. On the other hand, an automatic technique with a low error rate may not be useful if the application domain demands high reliability and accuracy. For example, an automatic algorithm may not be used to extract contours of a brain tumor that is to be destroyed, or to trace heart wall contours for surgery. In critical domains where lives hang in balance such as medical and military imaging, achieving the necessary reliability and precision of segmentation requires interaction with human experts. Over the years, researchers have paid increased attention to optimizing the human involvement in such systems. In this thesis, we develop a system that facilitates expert verification and supervised refinement of image segmentation, and hence the title – *Supervised Image Segmentation.*

Humans and machines have many complementary capabilities. A table listing relative capabilities is shown in Table 1.1 [4]. The implications of these comparisons for human-computer interface design is that the best role for a human is to serve as a supervisor, i.e. to control, monitor, act as decision maker, and to respond to unexpected events. The computer is better suited to do prespecified procedures fast. Little has been published in combining these complementary capabilities for generating reliable and precise image segmentations. The main goal of my research is to combine many of these apparently complementary capabilities to obtain reliable

Chapter 1

Introduction

Image segmentation and tracking nonrigid motion have been important problems in computer vision. The supervised image segmentation scheme developed in this thesis facilitates reliable image segmentation, and efficient tracking of non-rigid motion, using interactive inputs.

1.1 Motivation for supervised segmentation

Image segmentation is a process of dividing images into regions depending on the task at hand or for their correspondence to physical objects. Tracking *non-rigid motion* involves tracing the boundaries of flexing non-rigid objects, and establishing a correspondence among the traced boundary contours over several image frames. While it is widely believed that image segmentation and non-rigid motion tracking problems are of exponential time computational complexity, researchers have developed many approximation algorithms that run in polynomial time, and provide acceptable answers. In recent years substantial progress has been made in developing these algorithms. The goodness of an algorithm is usually assessed by the

List of Figures

List of Tables

Contents

Acknowledgments

My sincerest thanks to Dr. David Sher, Chairman of my thesis committee. His advise and encouragement have played a major role in my PhD research. The innumerable late nights that we spent arguing about research directions have shaped this thesis. I also thank him for being a good friend, and for his wholehearted concern towards my career development. I am thankful to my dissertation committee members Professor Sargur Srihari and Dr. Raj Acharya for their support, and for their valuable time.

My graduate studies at SUNY Buffalo Computer Science department has been a wonderful experience. Our department secretaries: Gloria Koontz, Eloise Benzel, Sally Elder and Leslie Russo have been very helpful and supportive. I am thankful to them for all their help, and concern toward my wellbeing. Our system support staff is of the highest quality, and my work experience outside of UB has increased my respect for them. Especially, I am thankful to Ken Smith and Devon Bowen for providing a very reliable computing environment, and for providing me technical help whenever I needed it.

Any acknowledgment would be incomplete without acknowledging the help of my friends – Venu who provided me his thesis results, Sunder who motivated me to work a bit harder, and Krishna, Sarnath and Ajay who were there to make my student life more colorful. My heartfelt thanks to them.

I am thankful to my brother Dr. Vithal Revankar and my sister-in-law Smita Revankar for being there for me all the time, and to my parents whose love and sacrifice for my progress is far beyond my comprehension.

Dedicated to my dear parents whose love and sacrifices are far beyond my comprehension.

Abstract

Image segmentation is a process of dividing an image into functionally distinct regions. In this thesis we develop a scheme for supervisory verification and modification of the computer generated segmentation. Our supervised image segmentation scheme combines the reliability of a human expert with the precision of the automatic image segmentation algorithms, thereby yielding precise and reliable segmentation.

To enable active and natural interaction between a supervisor and the relevant image processing algorithms, we facilitate two common modes of natural interactive gestures, namely *pointing* and *circling*. To account for imprecision and extents of manual suggestions, we use the human visual attention distribution model obtained from the published psychological and psychophysical research. The supervisor's interactions by pointing activate our region growing algorithm, and interactions by circling activate our constrained contouring algorithm.

The important contributions of this research are:

1. Development of a threshold propagation algorithm for image segmentation that facilitates supervisory specification of width of an elongated pattern in an image.

2. Development of a graph search based constrained contouring algorithm, that refines a rough input contour to yield the boundaries of the intended object.

3. The overall supervised image segmentation scheme is significantly more sophisticated than the published interactive segmentation schemes, because the interactions are approximate natural gestures, and are used in processing stage of the algorithms. The algorithms account for both the image data constraints and the approximations in the human gestures, by incorporating the human visual attention map.

Supervised Image Segmentation

Shriram Revankar
Ph.D. dissertation

submitted to the
Department of Computer Science, and
the Faculty of the Graduate School of
the State University of New York at Buffalo
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Dissertation Committee

David B. Sher, Chairman
Sargur N. Srihari
Raj Acharya