Identifying the Neuroretinal Rim Boundary Using Dynamic Contours

D.T. Morris, C. Donnison
Department of Computation
UMIST
PO Box 88
Manchester
M60 1QD
t.morris@co.umist.ac.uk

Abstract.

The neuroretinal rim forms the outer boundary of the optic nerve head: that region of the retina where blood vessels and nerve fibres pass out of the eye. It is normally a circular structure, but is known to change shape due to nerve damage in glaucoma. Its shape can therefore be used in the diagnosis and assessment of the treatment of this disease. Automatically finding the boundary would be useful as it would allow reliable quantitative shape measurements to be made. However, it is a difficult problem as the boundary is ill defined and partially obscured by blood vessels. In this paper we present an algorithm that successfully identifies the boundary using dynamic contours (snakes). The success of the algorithm is very dependent on preprocessing the image to enhance the contrast between the retina and the optic nerve head. We therefore describe the preprocessing in some detail. The algorithm has been tested on numerous images and found to be successful, as judged by an optometrist, in every case.

Keywords.

Retinal image processing, glaucoma, dynamic boundary, snake.

Introduction.

The blood vessels and nerve fibres serving the retina are located on the eye's inner surface. They pass through the eye at the optic nerve head (the "blind spot"). This structure is sensitive to increases in intraocular pressure (IOP) such as are associated with glaucoma: as the IOP increases, nerve cells at the optic nerve head are killed and the patient eventually notices a loss of sight, unless the IOP is reduced surgically or by medication.

At present there is no means of knowing what specific level of IOP the eye can tolerate without further damage. Clinicians treating glaucoma therefore rely on using drugs to reduce the IOP whilst examining the patient regularly. If the symptoms become worse, the dosage is increased. This regime can limit the loss of vision but cannot guarantee correct treatment. Further, since the treatment requires constant re-establishing of baseline IOP and visual function, progressive loss of vision is almost inevitable.

A more accurate means of assessing the treatment would be to assess the optic nerve head directly. This is the most sensitive and definite indicator of glaucoma in its early stages and can also give a quantification of the treatment's results as the disease progresses. The importance of quantifying the nerve head's shape is recognised in the numerous attempts that have been made at automating the measurement, the most accurate and popular at present being those using the Scanning Laser Ophthalmoscope (SLO) (1). Investigations also continue into using simpler instrumentation (2, 3). The SLO generates a high resolution volume image of reflectivities within the retina. From this the optic nerve head boundary can be traced with considerable accuracy. Although SLO approaches are attractive in this respect, they have the drawback that an SLO is an extremely expensive piece of equipment. The approaches using simpler instrumentation, whilst appearing attractive, have not yet demonstrated success, but since an ophthalmologist is able to recognise glaucoma from an image of the optic nerve, it should be possible to do so automatically. We believe that it is these approaches that will be more widely adopted, once successful algorithms have been developed.

Whatever data is used, the measurements must be made with no human intervention since inter-observer variability is comparable to inter-class differences (4), an observation that lends weight to the received wisdom that three experts will give three different diagnoses.

Some of the difficulties experienced in delineating the structure of the optic nerve head may be appreciated from figure one which shows a white light image of a healthy retina; the data collection methodology has been described elsewhere (5). Regions within the nerve head and on the retina may be readily identified, localising the boundary between them is harder since it is not sharp and, more importantly, it is obscured in many places by the blood vessels that supply the retina. Whereas it is an apparently simple task for an expert to trace the boundary, interpolating where necessary, traditional boundary-following algorithms have not succeeded in this as they do not make full use of the edge smoothness and continuity properties that the expert uses.

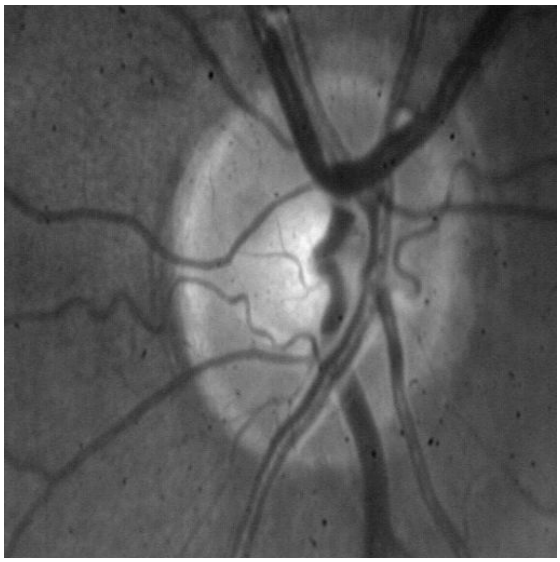

Figure 1. Sample Optic Nerve Head Image.

Kass et al (6) suggested the dynamic contour (snake) approach to implement these properties. Conceptually the dynamic contour attempts to shrink a boundary to image features. The algorithm's major advantage is that it is able to bridge discontinuities in the image feature being located: it is this property that makes the dynamic contour attractive in the present application. Since the original publication, the formulation of the algorithm has been made more efficient and robust (7, 8, 9).

The contour is shrunk with length and stiffness constraints, i.e. the final contour should be as short and straight as possible. Shrinking is opposed by constraints derived from the image data which are dependant on the type of boundary being located. For example, if a local maximum (watershed) was being traced, the constraint could be derived as a weighted sum of the local grey values; if a contour was being followed, the constraint would be derived from the image gradient. The algorithm allows the relative influences of these three factors (length, stiffness, image feature) to be varied and thus allows the user to tune the snake to a particular shape of boundary.

As mentioned above, many attempts have been made to locate the nerve head boundary using other algorithms, but only Lee and Brady (10) have used the snake. For some reason, they omitted the stiffness criterion from their formulation, consequently the snake they derived bulged inwards where the boundary was obscured. In an attempt to reduce this effect, they expended considerable effort in attempting to remove blood vessels from their data, but have not as yet reported success. In our implementation, we have included this factor, we have also minimised the energy function differently.

Whilst locating discontinuous boundaries using snakes is not novel, determining what preprocessing is required to make a robust algorithm is. It is the purpose of this paper to present the preprocessing we have used in this particular application, to discuss our implementation of the dynamic boundary location technique and to present sample results of analysing clinical data.

Pre Processing.

Capturing images of the optic nerve head was not a simple task. Although the details of how this was done have been described elsewhere, they will be summarised here for completeness. The data capture equipment consisted of a video camera attached to a Zeiss fundus camera. The output of the video

camera could be digitised using a Data Translation board driven by a PC. The major problem in capturing data was in illuminating the retina adequately. To do this, the patient's pupil was dilated and when a satisfactory image was seen through the fundus camera, the patient's eye was illuminated by a photographic flash and the video image captured. This usually resulted in satisfactory images, although the image brightness was variable and on occasion, the patient reacted to the flash fast enough that the optic nerve head was not captured. It was essentially the problem of variable and poor illumination that had to be overcome by preprocessing. Without preprocessing it was observed that the dynamic contour shrank too far into the interior of the nerve head.

It was the aim of the preprocessing stage to yield a high contrast image, with grey level gradients enhanced. The snake will be shrunk from the outer regions of this image onto the enhanced outer edges of the required region where they exist and will interpolate where they do not. Preprocessing consisted of three steps. Firstly, the image was equalised by histogram equalisation, giving results such as figure 2: the histogram equalised version of figure 1. Equalisation was found to be a useful method of enhancing the difference between the bright nerve head region and the darker surrounding retinal region of the image. Secondly, images were thresholded to remove pixels that are definitely not part of the nerve head, the image's median was used as the threshold: grey values below it were set to zero since it was assumed that they corresponded to non optic nerve head pixels. The threshold was set deliberately low to ensure that all of the nerve head pixels were selected, at the cost of including some retinal pixels. Although thresholding added an extra dimension of complexity, it also removed any superfluous image detail that could cause the snake to become attached to erroneous features; this step should therefore been thought of as a precaution.

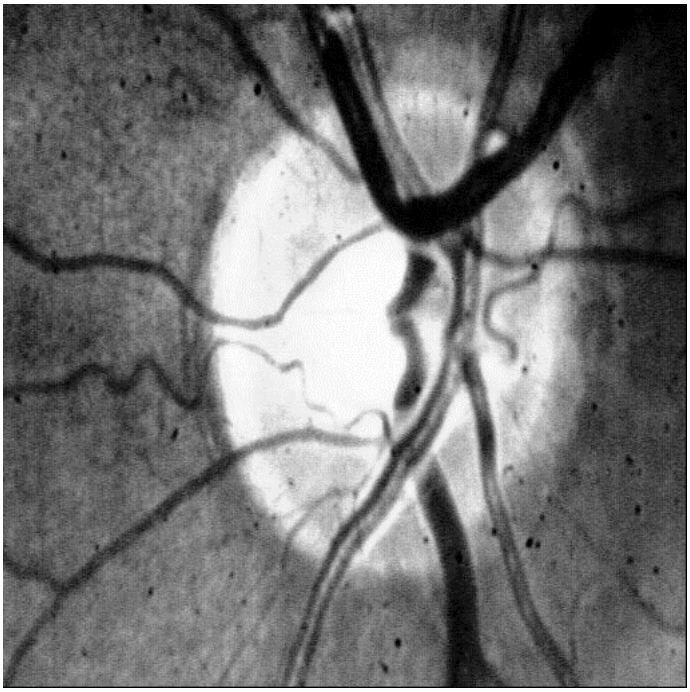

Figure 2. Histogram Equalised Version of Figure 1.

Our goal in preprocessing was to enhance the edge detail in the images. This was achieved by applying a pyramid edge detector to the equalised (contrast enhanced) image. Pixels in a layer of the pyramid data structure were computed as the average of groups of four pixels in the layer below. Pixel averaging tended to prevent future examination of the spurious edge-points arising from noise in the equalised image, it was therefore preferable to enhance the image's edges in this way rather than use straightforward edge enhancement. The optimum pyramid size was determined by balancing minimum processing time (achieved using large pyramids) with maximum accuracy (obtained from small pyramids); bearing in mind that a minimum image size is required by edge detection algorithms. Empirical testing suggested that creating more than five levels did not lead to an obvious reduction in processing time whilst using more highly condensed data led to portions of edge data being missed. A pyramid of five layers (in addition to the base layer) was chosen to provide a compromise between speed and accuracy; this fifth level contained 16x16 pixels. Figure three shows the pyramid generated using figure two as the base. Although the diagram shows all the layers it is possible to compute, only the five lowest are used.

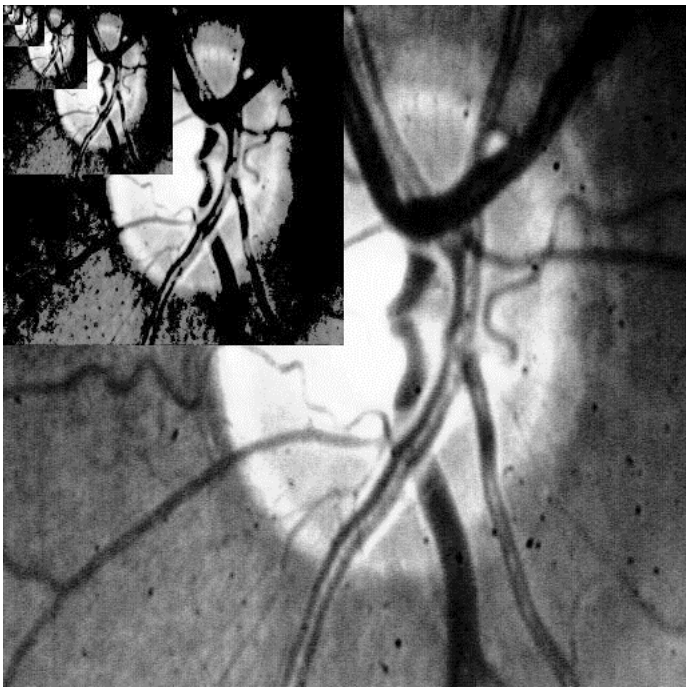

Figure 3. Pyramid Representation of Image Data.

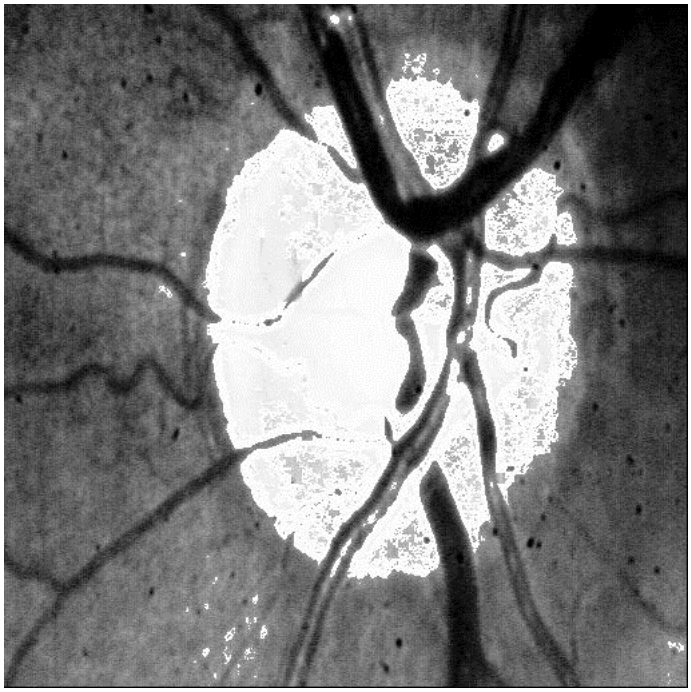

Figure 4. Preprocessed Image Data.

Comparison of edge detecting templates did not reveal any significant differences between the results given by any of the simple 2x2 and 3x3 operators. The Sobel operator was therefore implemented. The pyramid was examined using this operator in the normal fashion: starting at the highest level, if the edge detector gave a result greater than some threshold, the equivalent pixels in the layer below were examined until either the base layer was reached or the edge magnitudes fell below the threshold. The choice of threshold could have a significant effect on the final result. A small threshold was necessary to guarantee that the optic nerve was detected; however this considerably increased the processor time required as more candidate edges were generated. Conversely, a large threshold minimised the time required to detect the boundary but reduced the amount that the edge contour was enhanced. Figure 4 is typical of the results obtained.

Dynamic Boundary Location.

The dynamic boundary can be represented as a parametric curve, C, by:

$$C(s) = (x(s), y(s)) \qquad 0 \bullet t \bullet 1 \qquad \text{equation 1}$$

It moves under three influences: its length and curvature are to be minimised whilst an image property summed along its length is maximised. Since we are fitting the snake to an object's boundary, we have used edge strength, f(x,y), as the image property. The edge strength summed along the snake may be written as:

$$I(s) = \int_{t=0}^{t=1} f(x(s), y(s))\, dt \qquad \text{equation 2}$$

Requiring that the snake's length is minimised is equivalent to minimising the rate of change of distance as s varies. This is equivalent to minimising the expression:

$$L(s) = \int_{s=0}^{s=1} \left| \left(\frac{dx}{ds} \right)^2 + \left(\frac{dy}{ds} \right)^2 \right| ds \qquad \text{equation 3}$$

Finally, we require the snake's curvature to be minimised. This gives the snake stiffness properties. Without it, the snake will contract like an elastic band: where the boundary is obscured, the snake will interpolate a straight line, or even bulge inwards. With this property the snake will exhibit continuity of gradient (c1 continuity) across the boundary. In essence, this property minimises the rate of change of tangents:

$$B(s) = \int_{s=0}^{s=1} \left| \left(\frac{d^2x}{ds^2} \right)^2 + \left(\frac{d^2y}{ds^2} \right)^2 \right| ds \qquad \text{equation 4}$$

Equations 2, 3 and 4 describe the energy terms of the snake. They may be grouped into a single functional to give:

$$E(snake) = \mathrm{a}(s) B(s) + \mathrm{b}(s) L(s) - I(s) \qquad \text{equation 5}$$

The terms α(s) and β(s) are weighting factors that allow the stiffness and elasticity of the snake to be altered relative to the importance of the image information. Whilst they have been written as functions of s, we, and most other investigators, have kept them constant.

Welsh, Searby and Waite (11) showed that equation 5 is equivalent to finding a curve $(\hat{x}, \hat{y})$ satisfying:

$$-\frac{d}{ds^2}\left\{ \alpha(s) \frac{d^2\hat{x}}{ds^2} \right\} + \frac{d}{ds}\left\{ \beta(s) \frac{d\hat{x}}{ds} \right\} + \frac{1}{2} \frac{\partial f}{\partial x}\bigg|_{(\hat{x},\hat{y})} = 0 \qquad \text{equation 6}$$

$$-\frac{d}{ds^2}\left\{ \alpha(s) \frac{d^2\hat{y}}{ds^2} \right\} + \frac{d}{ds}\left\{ \beta(s) \frac{d\hat{y}}{ds} \right\} + \frac{1}{2} \frac{\partial f}{\partial y}\bigg|_{(\hat{x},\hat{y})} = 0 \qquad \text{equation 7}$$

and some specific boundary conditions.

Kass et al (6) and Waite and Welsh (12) present methods of solving these equations. Following Waite and Welsh, we have used finite differences in which Taylor's theorem was used to derive approximations to the differential and double differential of a function in terms of finite differences. Substituting these in equations 6 and 7 gave a pair of systems of equations for finding the x and y coordinates that minimised the energy function of the snake. The minimisation equations were, in turn solved by grouping like terms and rewriting them in matrix form:

$$(B+\frac{1}{\gamma}I)\underline{x}_{n+1}^{(k+1)} = -A\underline{x}_n^k + \frac{1}{\gamma}\underline{x}_n^{(k+1)} + f(x_n, y_n) \qquad \text{equation 8}$$

$$(B+\frac{1}{\gamma}I)\underline{y}_{n+1}^{(k+1)} = -A\underline{y}_n^k + \frac{1}{\gamma}\underline{y}_n^{(k+1)} + f(x_n, y_n) \qquad \text{equation 9}$$

where $\left(B+\frac{1}{\gamma}I\right)$ is a band matrix that may be expressed as a product of Cholesky factors LL^T by which the coordinates of points on the snake may be iterated by first solving equation 10 followed by equation 11 for the x coordinates (and a similar pair for the y values):

$$L\underline{\tilde{x}}_{n+1}^{(k+1)} = -A\underline{\tilde{x}}_{n+1}^{(k)} + \frac{1}{\gamma}\underline{\tilde{x}}_n + \underline{f}(\underline{x}_n, \underline{y}_n) \qquad \text{equation 10}$$

$$L^T\underline{x}_{n+1}^{(k+1)} = -\underline{\tilde{x}}_{n+1}^{(k+1)} \qquad \text{equation 11}$$

The Cholesky algorithm is numerically very stable and has the additional advantage that the matrix decomposition need only be performed once per snake initialisation: it therefore leads to a rapid solution.

The snake was initialised as a spline curve. In the current, experimental, system, the control points can be interactively edited: adding, removing or moving points as required. The only requirement of the spline was that it lay entirely outside the optic nerve head region; in a future, automatic, version, we shall achieve this by initialising the snake to coincide with the image boundary. Two versions of the boundary-locating software have been implemented: for images of complete nerve heads and images in which the nerve head overlaps the image border.

Implementation and Results.

The software has been implemented on a PC using the WATCOM C compiler. This was used in preference to Borland's or the Microsoft compilers as it provides support for the IBM 8514 graphics driver. This was used in preference to the standard VGA drivers as it allows higher resolution images to be displayed, and the display to be updated rapidly without flicker. The user interface allows the spline to be placed and the image to be preprocessed interactively, the snake parameters (α and β) to be altered and the various intermediate images to be displayed or not.

Once the image has been preprocessed satisfactorily and the snake initialised, the snake was allowed to iterate to its final location. Typically, ten to twenty iterations were required. Once the image has been preprocessed and the initial matrix decomposition performed, each iteration may be performed in less than one second using a 486DX computer.

Figure 5 shows the results of processing the original image of figure 1. (Values of 40 and 180 were used for the stiffness and elasticity parameters.) It is particularly important to note that the snake's location coincides with the boundary indicated by an optometrist (any differences are insignificant compared to the inter-optometrist differences, cf. Cox and Wood (4)). Whilst this may not be too important an observation in the light of the comments made about different optometrist's interpretations of the boundary, it does lend support to the claim that the algorithm gives a reasonably accurate boundary. Further, it will do this consistently, unlike the optometrists! Also worthy of note is the detected

boundary's continuity (in the sense that the boundary has no gaps and its curvature is continuous) where the true boundary is obscured.

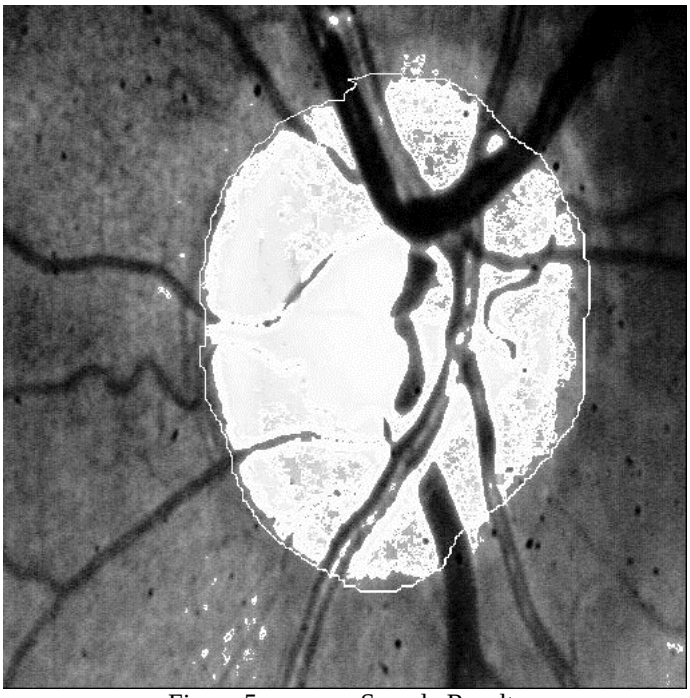

Figure 5. Sample Result.

Figure 6. Sample of a Failure.

Figure 6 shows the results of applying the algorithm with the same parameter values as before to a poorly exoposed image, not typical of the images we would normally accept for processing The snake has failed to locate the boundary in the upper right quadrant; the result is included to demonstrate the sensitivity of the snake algorithm to the preprocessing applied to the image. The preprocessing is the subject of further research.

Conclusions and Discussion.

We have implemented an experimental system to demonstrate the usefulness of the dynamic boundary location method in finding the outline of the optic nerve head. Two initial conclusions may be drawn:

• the power of the algorithm in locating unclear and possibly discontinuous boundaries is illustrated,

• measurements of the optic nerve head's dimensions may in future be made independently of any expert, thus reducing the variability of measurements.

We have shown the importance of preprocessing the image to enhance the features that are used in defining the boundary.

In the near future we plan to modify the software to perform the boundary detection automatically. Now that the algorithm for doing this has been determined, this simply requires altering the snake's initialisation. Cox and Wood (5) have suggested that the lateral (side) boundaries of the nerve head change shape as glaucoma progresses. The snake accurately locates the boundary in this region, we are hopeful that this approach will ultimately be successful in automatically quantifying the effects of glaucoma.

There are further structures within the optic nerve head that require localising, in particular the cup boundary, since the structure's shape is significant in glaucoma. We plan to use the same methods to detect this as we have used in detecting the disk boundary, but will initialise the snake to the disk instead of the image boundary.

We finally intend to investigate the usefulness or otherwise of allowing the values of α and β to vary along the length of the snake. Possibilities here could be to make them vary according to the snake's local radius of curvature or local image properties.

In parallel with these activities, we are testing the accuracy and usefulness of the algorithm(s) using a set of retinal images collected by Cox (4).

Acknowledgement.

We would like to record the support given by the International Glaucoma Association whilst the data used here was being collected.

References.

1. J.E. Nasemann, R.O.W. Burk. Scanning Laser Ophthalmology and Tomography, 1990, pp 161-264.

2. Y. Mo, D. Xiao. "Recognizing the glaucoma from ocular fundus image by image analysis." Proceedings of the 12th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Philadelphia, Nov, 1990.

3. T. Morris, I. Wood. "The Automatic Extraction of the Optic Nerve Head." American Academy of Optometrists, biennial European meeting in Amsterdam, May 1994.

4. M.J. Cox, I.C.J. Wood. "Inter- and intra-image variability in computer-assisted optic nerve head assessment." Ophthal. Physiol. Opt. vol. 11 1991, pp 36-43.

5. M.J. Cox, I.C.J. Wood. "Computer-assisted optic nerve head assessment." Ophthal. Physiol. Opt. vol. 11 1991, pp 27-35.

6. M. Kass, A. Witkin, D. Terzopoulos. "Snakes: Active Contour Models." International Journal of Computer Vision, Vol. 1 no 4, 1987, pp 321-331

7. A.A. Amini, S. Tehrani, T.E. Weymouth."Using Dynamic Programming for Minimising the Energy of Active Contours in the Presence of Hard Constraints." Proceedings of the Second International Conference on Computer Vision, pp 95-99, 1988.

8. L.D. Cohen.“Note on active contour models and balloons.” CVGIP:Image Understanding vol. 11 no 2 pp 211-218. 1991.

9. D. Terzopoulos. “Regularisation of Inverse Visual Problems Involving Discontinuities.” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No 4, July, 1986.

10. S. Lee, J.M. Brady. “Integrating stereo and photometric stereo to monitor the development of glaucoma.” Proceedings of the British Machine Vision Conference, 1990, pp 193-198.

11. W.J. Welsh, S. Searby, J.B. Waite. “Model-based image coding.” British Telecom Technology Journal Vol. 8, No. 3, July, 1990, pp 94-106

12. J.B. Waite, W.J. Welsh. “Head boundary location using snakes.” British Telecom Technology Journal Vol. 8, No. 3, July, 1990, pp 127-136