Locating the Eyes in CT Brain Scan Data

Kostis Kaggelides, Peter J. Elliott
IBM UK Scientific Centre, Winchester

Robert B. Fisher
Artificial Intelligence Dept, University of Edinburgh

Abstract

In this document, a technique for locating the eyes in Computed Tomography brain scan data, is described. The objective is to automatically localise the eyes for protection during radiotherapy planning. The image feature that is exploited is the circularity of the eyes. After data preprocessing to remove parts of the CT machinery, signature analysis is performed to locate areas of interest. By applying Canny's edge detector to these areas, data is further reduced to the significant edge fragments. The Hough Transform is then applied. The Converging Squares algorithm is used as an efficient and robust method to search the parameter space. The results are processed by the hypothesis generation stage which clusters them according with the x, y, z coordinates of the suggested centres. The ISODATA algorithm is used for clustering. The hypotheses are assessed and sorted. The most valid hypothesis is selected and refined using a second Hough Transform, this time considering spheres. After the rejection of the invalid members of the hypothesis cluster, an ellipsoid is fitted to the new cluster centre and the results are drawn on the data. The method is fast and robust. The method was tested using five different data sets and it performed well on all of them.

1 Motivation and Problem Description

In radiotherapy the objective is to plan a treatment which will send a high dose of radiation to the tumour tissue and at the same time restrict exposure to all other parts under limits specific to their individual sensitivity, so as to avoid undesirable side effects. The eyes, of course, are high in the list of sensitive (and crucial) organs which certainly have to be taken into consideration during the treatment planning stage. This project designed and implemented an algorithm which automatically identifies and locates the eyes in a Computed Tomography (CT) brain scan data set. The specialised human operator is thus relieved from the tedious and time-consuming process of manual segmentation, and the whole process of therapy planning is accelerated.

The system consists of a sequence of individual processes. The major feature exploited for the location of the eye regions throughout the system is the approximate circularity of the eye sections in every slice[1].The system layout can be seen in Figure 1.

We discuss each of these stages separately in the following sections.

2 Background Removal

In the real data sets parts of the CT machinery are visible as well as the patient's head (*e.g.* the head supporting pads and parts of the machine). The objective of this stage is to process the data set, so that these objects are removed.

The algorithm removes most of the undesired objects, making general assumptions about the object geometry or location. The decision made is based upon the following object features:

[1]This follows the assumption that the eye is approximately spherical and consequently any planar section is a circle

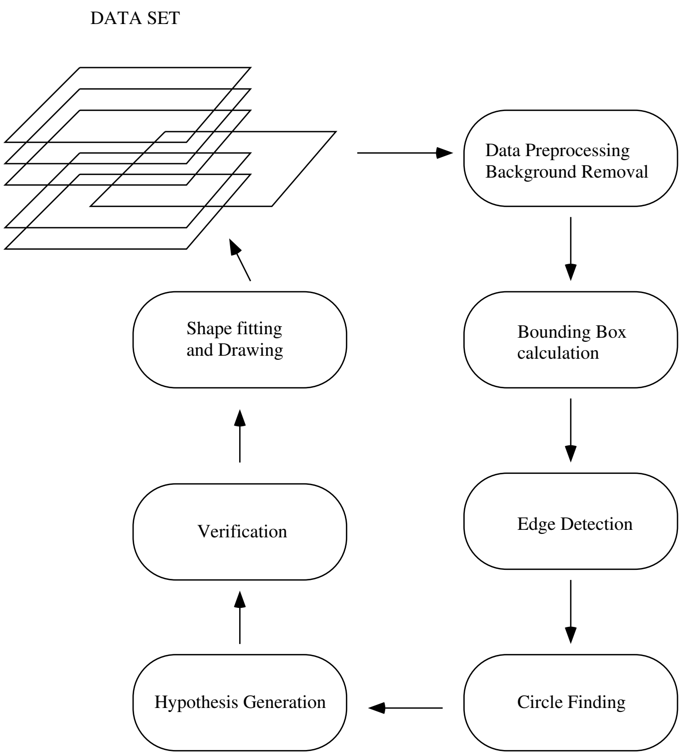

Figure 1: System Layout

1. The objects are not connected to the head [2]
2. The objects are the only features unchanged in shape and location throughout all slices[3]
3. The objects are constructed from a radio-opaque material so they have the maximum intensity value (255).

Based on these features, the algorithm developed creates a buffer frame with every pixel set at 255 and iteratively AND the first N frames so to keep only the pixels having exactly the same intensity (255) in all N frames. The result is an image consisting of parts of the wanted objects (exploiting the fact that the objects are invariant in respect with z). At each slice, all pixels connected with pixels existing in the buffer image are recursively deleted (exploiting the fact that head and objects are not connected, at least at the slices of interest).

The final result is satisfactory and the process is efficient and sufficiently robust. It fails to remove the head pads in the top of the head scan because they change their position significantly, and in this way assumption 2 is violated. Again, the algorithm fails with the resulting destruction of important parts of the image, when assumption 1, concerning connectivity is violated. The application of the algorithm in the 5 data sets gave some errors in particular cases, but in general the performance is satisfactory.

3 Finding the Bounding Box and Locating Areas of Interest

Speed is an important requirement for this system. A human operator can outline the eyes manually within few minutes. To compete with this, the system must utilise all the high level

[2] This feature excludes a part of the head supporting pads (*i.e.* top of the head).
[3] Here again we have the exception of part of the head supporting pad

knowledge as soon as possible. In the case of the CT images only a mere $\frac{1}{3}$ of the pixels contains valuable information and the rest are just background. We also can certainly assume that one and only one object of significant dimensions is present *i.e.* the patient's head. The problem consequently is to calculate the bounding box of the head. Having done this, we can locate within a small region where we expect the eyes to be.

In [3] there is a method to derive properties of an object based on the analysis of its signatures. The signature of an image is a one dimensional projection over one direction. More specifically, the horizontal signature of an image I of size d is a set S_h whose members are the values of a mapping function $f_h(c)$, where $c \in [0, d]$. In this way we map image $I \longrightarrow S_h$. The actual function value of $f_h(c)$ is the number of pixels set in column c. Note that $f_h(c)$ is not dependent on r, so $\forall r \in [0, d] f_h(c) = constant$. Respectively, for the vertical signature S_v the mapping function $f_v(r)$ is constant for all values of column c.

An efficient algorithm was developed which, using "hardcoded" knowledge about the object's approximate size, calculates the bounding box. The application of the algorithm on an arbitrary slice is displayed in parts (a),(b),(c) of Figure 2 . The algorithm traces the signatures searching

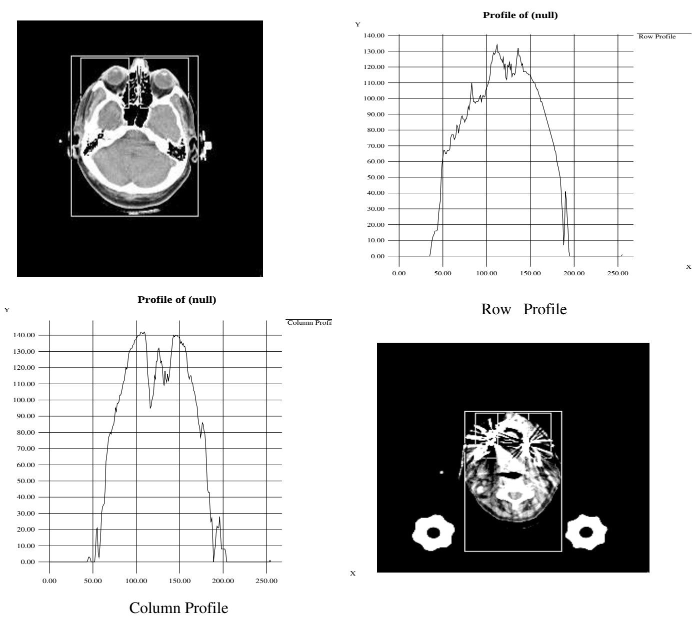

Figure 2: Finding the bounding box in image (a) using the row (b) and column (c) profiles. The failure to find the correct bounding box in (d) is because of the two unexpected objects

for a given threshold value, such that it guarantees that the row/column which corresponds to this value belongs to the object we want to locate (*i.e.* patient's head). Since we assume that no other objects are present in the image, any comparatively large value would guarantee this. To locate the minimum we trace the curve backwards up to a threshold value (*e.g.* 10 pixels) from which we stop if we detect either zero value or an increasing trend to the data. The same is done for locating the maximum[4], only this time we start tracing from the other far right side of the signature.

The algorithm, is efficient and robust. In many cases it is surprisingly successful in locating

[4]Minimum and maximum refer to the elements of the S_h and S_v sets, so actually are the minimum and maximum r or c where a local minimum exists

the correct bounding box, even though small objects exist around the patient's head. The algorithm is unsuccessful in images like that displayed in (d) of Figure 2. The cause of failure is again the violation of the basic assumption that no other big objects are present. These parts of CT machinery were not removed by the removal process because they appear only in this slice and not before or after.

Before closing this section we should discuss some practical considerations. Signature analysis is a tool applying to binary images. The CT images are not binary images. As again can be seen in Figure 2 there are areas within the patient's head with 0 intensity value. As a result, the horizontal signature does not take the highest value at the position of the nose, as it would be normally expected. Extracting further information is therefore risky. A procedure which would fill the empty regions within the object and thus solving the problem is not considered worthy of the computation time. Using just the bounding box information and a very sound assumption (the patient's head has standard orientation upwards, the eyes are expected to be always in the top side of the bounding box), we can correctly and efficiently locate the area containing the eyes within two 50×50 images[5] (which are drawn as squares in Figure 2).

4 Edge Detection

Although the size of the data to be processed has been significantly reduced, it is still too large to manipulate. With edge detection the objective is to reduce the amount of data to the significant edge fragments *i.e.* intensity changes between the different image regions. The Canny operator [2] is used for its optimised performance with noise and localisation. Additionally, the Canny operator gives a unique response to a single step.

The performance of the edge detection stage is important for the localisation of the eye by the circle finding stage. Unfortunately edge detector performance is data dependent. For this reason, there was not any special emphasis given to a very detailed adjustment of the Canny parameters (sigma value and hysterisis thresholds). The rest of the system must be designed to be compliant with edge detection performance. Actually, the CT data is a quite difficult test since the intensity values are quantised, and soft tissue boundaries are not easily detectable. A sigma value of 1.0 was selected for the Canny smoothing parameter, giving the desired level of detail in the area around the eye.

5 Circle finding

As mentioned above, the feature exploited for the localisation of the eyes is the approximate circularity of the eye sections. For this, the Hough Transform (HT) is used. This selection is based on the HT performance with incomplete or corrupted data. In general there are several inherent problems in HT applications, namely storage requirements and time complexity. Localising the feature by finding the bounding box loosens the space and time constraints. Other problems to overcome are:

- The eye sections are only approximately circular. In the first or the last slices that contain eye sections, even this approximate circularity is lost.
- There is context interference *i.e.* a dense mesh of non-circular curves around the feature. Since the HT deals with each pixel separately, it is vulnerable to this interference[6].
- There are many other approximately circular features present in the edge image. Semi-circular bones give a non fragmented edge when the Canny operator is applied and con-

[5] The exact size of the subframes depends on the pixel dimension of the data set. The program partitions the frame using millimetre dimensions. The correspondence of these dimensions in pixels is calculated using the pixel dimension

[6] Sometimes termed in relevant literature as the "connectivity" problem

sequently sharp peaks in the parameter space of the HT. Also, the eye boundary is fragmented and in parts erroneous.

These problems are addressed in the design of an HT procedure. Design decisions have to be made on the two major components of the HT: the voting scheme and the search of the parameter space. The standard theoretic framework is to view the HT as imaging using signal processing terminology [1]. Here, we view the HT as a search problem and the voting as an heuristic evaluation of the model existence in a specific location. The informedness of the selected heuristic (voting scheme) is important in this framework, and can describe more accurately the quality of the voting scheme than the Signal to Noise figure [5].

5.1 Voting Scheme

The standard HT with line voting and the use of directional information was chosen:

1. Using a standard HT with a 3 dimensional accumulator is more appropriate for the complicated images since the voting "encodes" more information in the parameter space (compared to the 2-1 method [8]).
2. Line voting using directional information (every point votes in the parameter space across the normal to the tangent) increases the informedness of our voting and many spurious votes are saved. At the same time we could characterise it as a cost efficient approximation of more elaborate schemes (Bayesian approach, template voting etc). Drawing a line is much more efficient than drawing a surface.

The geometry of the line is displayed in Figure 3. In part (a) of this figure, a point $O(X, Y, Z)$ on a curve votes in the parameter space along the line defined by the points $(X_1, Y_1, Z_1), (X_2, Y_2, Z_2)$. The calculation of their coordinates is as follows:

$$X_1 = X + Rad_{min} \cdot cos(\theta), Y_1 = Y + Rad_{min} \cdot sin(\theta), Z_1 = Rad_{min}$$
$$X_2 = X + Rad_{max} \cdot cos(\theta), Y_2 = Y + Rad_{max} \cdot sin(\theta), Z_2 = Rad_{max}$$

where θ is the orientation of the tangent at point $O(X, Y, Z)$ (measured as an offset from the vertical direction, as shown in Figure 3), and Rad_{min}, Rad_{max} are the range of radii that we are looking for. The Bresenham algorithm is used for drawing the line in the 3 dimensional parameter space. The parameter space over the whole image can be seen in Figure 4. The 3 dimensional space is projected onto a plane and the histogram of the image is equalised so even weak peaks can be seen. The existence of many peaks is the most conspicuous characteristic. From the histogram in the same Figure we select two threshold values and display the parameter space after thresholding with these values. The results (Figure 4 (d) and (e)) do not give only the desired two peaks for the eyes. Threshold selection also cannot be based just on absolute values because the eye peaks are smoothed in other slices where the eye segment is not very circular. The peaks though are always locally distinct as dense voting areas. The HT represents approximate circularity with these dense areas in the parameter space. Thresholding regards only absolute peak values and thus loses important information. In fact the search procedure is an important element of the system since it can be used to compensate for error and inaccuracies of the voting stage.

5.2 Searching the Parameter Space: the Converging Squares Algorithm

Searching the parameter space is the important last part of the circle locating stage. As we noted above, the HT has evaluated heuristically the "circularity" of various segments in the image. This evaluation is not only related to the absolute peak values but also to the density of votes in the area. As noted above, one method to incorporate local area information is to smooth the HT. O'Gorman and Sanderson in [6] present a method for locating peaks in many

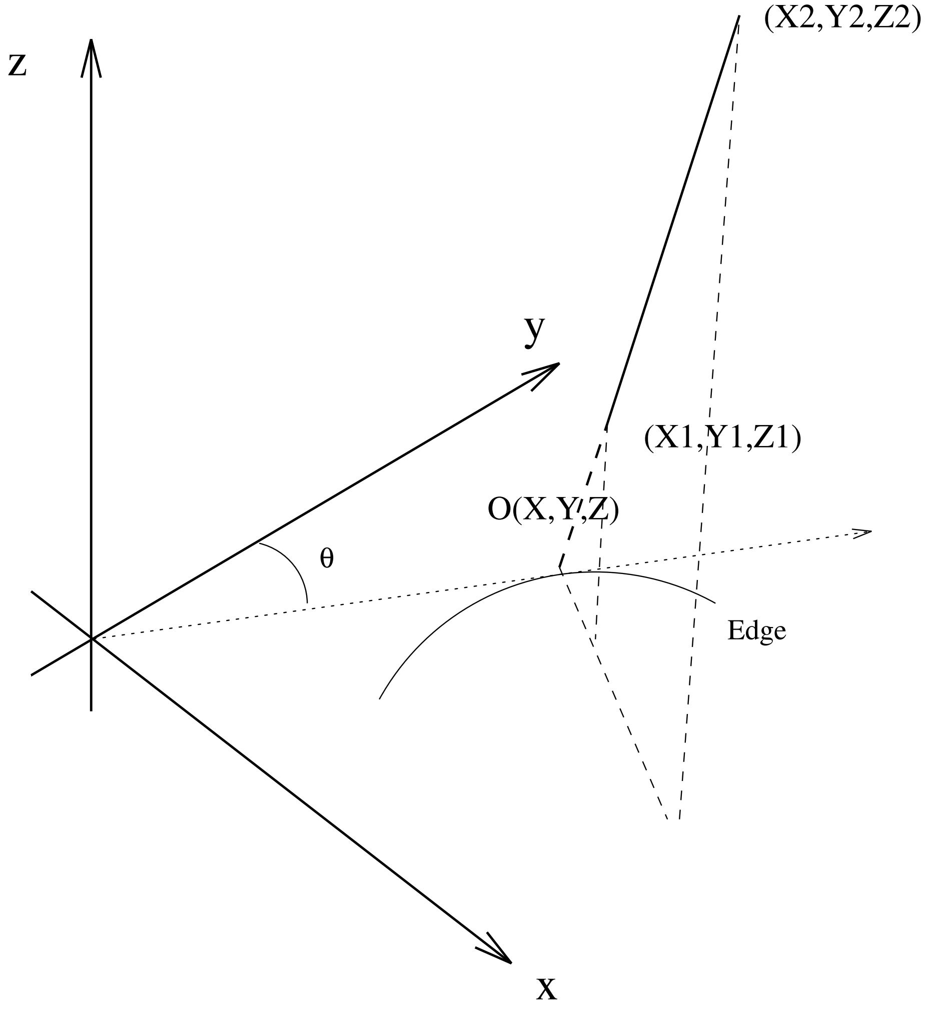

Figure 3: Calculation of line vote parameters

dimensions called the converging squares algorithm. The method (originally used over pixel intensities for object location) is cited as a method of searching the parameter space in [4].

The purpose of the algorithm is to find the maximum density region's peak with a computationally efficient method. There are no constants defined so the algorithm is equally valid to all slices[7]. The algorithm is general and can be extended to N dimensions. O'Gorman and Sanderson calculate the computational complexity of the algorithm and report the method to be faster than smoothing and search for maxima. Finally, the most important reason for selecting the converging squares algorithm is that, since it searches for density as well as for peak value, it is a more reliable search. It interprets correctly the heuristic evaluation of possible model locations, performed by the HT.

The converging squares views the 2 dimensional image as iteratively decomposable to smaller size squares. The decomposition of the image continues until the most dense 3×3 square has been located. Then the biggest value in this square is returned as the peak of the region.

The HT procedure implemented locates the eyes successfully in the majority of the slices where the eye segment is circular. The algorithm performs adequately in all 5 data sets. The processing time for a slice is a fraction of a second. Most importantly, the method is more

[7] If a threshold value was determined the procedure becomes data-dependent since the actual HT results are not appropriate for thresholding as noted in previous paragraph.

successful in the correct peak localisation than the maximum value method. Comparative results of the converging squares algorithm and the maximum value search are in Figure 5.

It is clear from Figure 5 that the converging squares is enhancing the HT robustness in circle detection. The maximum value search performs well only in the case of a privileged slice (*e.g.* slice #7 in Figure 5). In slices #8,#9 sharper peaks lead maximum value to be incorrect, where the converging squares still locates the feature.

6 Hypothesis Generation

Circle finding is successful in locating the feature when it exists, but at the same time "unsuccessful" to give a reliable evidence of the feature's absence from the image. In the data there are also present minor circular features other than the eyes. These features are going to be picked out together with the eyes. The number of votes with which a circular feature is supported at each particular slice is a measure of the feature's validity. It is not enough however to decide the existence of eyes, based on just that. The segments we want to locate have low contrast and the edge detection stage is not expected to give very good boundary description. The HT manages to locate these segments with the use of the converging squares algorithm, but records the result only with the peak value. Therefore the number of votes suggesting a circle somewhere in the image is not an appropriate criterion to perform thresholding. Also, even if this is feasible, it would introduce a heavily data dependent parameter, the threshold value.

We can view this as a three dimensional segmentation problem. The new "image" to be segmented consists of a series of triplets $(x, y, slicenumber)$. These triplets define points in a 3 dimensional space. We want to locate the parts of this 3D cluster which imply the existence of spherical/cylindrical/quadric surface in the 3D data. In other words find the approximately linear cluster. The new data to be processed is visualised in Figure 6. The data plotted here is the result of the circle finding stage for the left eye in data set 1. The z axis represents the slice number and the x, y the location of the centre given by the circle finding stage.

In part (b) of the same figure the range of z containing the slices 6 to 10 is magnified. In these slices the eye is present and the circle finding stage gives valid results. As a consequence there is this small linear cluster.

Since this classification task is a high level decision, the issues of robustness and generality are here of even greater importance. The solution must conform with the following :

- No assumptions for the nature of the data. That is, more than one spherical feature might be present (and actually **are** present).
- No assumptions for the correctness of the circle finding stage can be made. A method that would track this cluster along the z axis trying to find linear segments (something like corner detection) would overestimate the reliability of the circle detection stage. Our solution should permit the circle detection stage to fail in one or two slices where it would be normally expected to find the eye.
- Verify in 3D the results and **improve** as far as possible the accuracy of the circle finding method.

The first two restrictions are so limiting that they reject any solution that would discard data. What is suggested to be done instead is hypothesise the eye location. The only criterion we can securely apply to the data is its relation with the context (*e.g.* circle finding results in adjacent slices). Thus our problem is turned into a pattern recognition problem.

6.1 ISODATA clustering

For the reasons noted above the selected method for clustering is the ISODATA unsupervised clustering algorithm[7]. The circle finding stage will result in a set of vectors of the following format:

$$< \text{x , y , slice number z, radius r, number of votes n} >$$

where x,y refer to centre location of a circle with radius r found at slice z by the converging squares with n votes. From this information we select only the <x,y,z> part to use as input to the ISODATA module. What we want the ISODATA to perform with the clustering is the *generation of hypotheses* about the existence of spherical/quadric surfaces in the data. The Euclidian space in which the ISODATA[8] clusters the results is appropriate for centre location hypothesis generation. For this, the radius is excluded from the ISODATA input. The number of votes n, is excluded because it is not very reliable information and possible errors are amplified. This is used later, for hypothesis assessment and to check the correctness of circle finding and clustering.

The ISODATA requires several parameters to supervise the clustering ($M, N_D, \sigma^2_{axis}, d_{max}$). In more detail, the M parameter defines the minimum population of a cluster. The desired number of clusters expressed by N_D is used by the algorithm to decide whether to do cluster splitting or cluster merging in order to increase or decrease the number of clusters respectively. The σ^2_{axis} parameters are given in order to determine when and where (*i.e.* on which axis should the two new centres have different coordinates) a cluster should be split. Clusters whose centres have distance less than d_{max} are merged into one new cluster. Descriptions of the algorithm can be found in pattern recognition textbooks (*e.g.* page 218 in [7]).

In the data sets provided by the circle finding stage the ISODATA converges after 3 to 7 iterations. This is of course related to the nature of the data. The z axis value increases monotonically at all circumstances since all measurements belong to different slices. Theoretically, the ISODATA does not converge always. A maximum iterations constant in the implementation code makes ensures termination.

7 Hypothesis Refinement

To state the new problem briefly, the clustering stage gives valid circles that include the eye, but also gathers erroneous results from adjacent slices where the eye does not exist. There are several reasons why this happens:

- The clustering task of ISODATA searches among different clustering combinations that minimises the total dispersion (sum of each cluster dispersion). This is not automatically equivalent with picking only the desired circles. The algorithm, in order to converge, may classify incorrect points (vectors produced by the circle finding stage) together with the correct results.

- No matter how we may set the parameters of the ISODATA that describe the desired solution, it is impossible to exclude the case where the erroneous vectors conform to our statistics. The difference with the previous case is that the points are classified together with the correct results, not because this is comparatively the best solution, but because our description of the solution is not detailed enough to exclude them.

The results can be filtered using our high level knowledge for the desired solution. The HT can be used for a second time here, as a filter set to remove erroneous data. In this case the input is $< x, y, z >$ vectors and the output is the sphere centre coordinates X_c, Y_c, Z_c that best fits these points. We consider the range of sphere radii which the eye could have. The HT procedure can be simplified if we consider the x, y as known. The x, y coordinates accuracy is satisfactory. For this reason we can just have a two dimensional parameter space Z, R.

We can get the Z coordinates of the possible spheres containing the cross section data x, y, z, r from:

$$Z = z \pm \frac{\sqrt{(R \cdot f_{slice})^2 - (r \cdot f_{pixel})^2}}{f_{slice}}$$

[8] Although ISODATA could be possibly modified in order to use some other norm than the Euclidian distance

where f_{slice}, f_{pixel} are factors to translate distances between slices or pixels to millimetres. Using this simple formula the new HT casts votes to a two dimensional accumulator. The number of votes each point casts is the actual number of votes with which this point was suggested by the circle finding procedure minus the distance of this point from the original centre of the cluster. After this, a search for maximum value will give the Z coordinate of the sphere best fitting the data. Using this sphere we delete all the points of the cluster that are not inside the sphere.

As it is clear, the HT in this case is not used for the actual estimation of the model's parameters. It is used only as a criterion for rejecting points. The method performs well with all five data sets. In cases that the clustering is correct, the big size of the sphere makes sure that all the cluster members are included and nothing is rejected. Otherwise members that do not fit with the rest are rejected.

8 Shape Fitting

Finally at this stage we know accurately the centre coordinates of the eye. For drawing the results (and possibly for modelling the eye as well) it is possible to use an ellipsoid model. This will ensure that the eye is included in a predefined volume. For this reason we assume an egg shaped ellipsoid surface surrounding the eye (elongated along the z axis). The ellipsoid model is closer to the natural shape of the eyes[9]. We want each intersection of our ellipsoid with planes parallel to the x, y plane to be circles so the ellipsoid's equation becomes:

$$\frac{x^2}{R^2} + \frac{y^2}{R^2} + \frac{z^2}{c^2} = 1$$

$$x^2 + y^2 = R^2 \cdot (1 - \frac{z^2}{c^2})$$

The c parameter is a security parameter that ensures that we do not omit parts of the eye (not detected at all by the circle finding because there was not any circular feature) *e.g.* we know that the eye is going to be present in 6-8 (24mm-32mm) slices so we set c as 4 to cover all cases. The R parameter could be taken to be the biggest radius of the cluster members (and probably multiplied by a factor to ensure that the eye is included).The results in Figure 7 show circles of radius $R^2 \cdot (1 - \frac{z^2}{c^2})$ superimposed on the raw data.

9 Results and Conclusion

The system outlined above was implemented in C++. The system locates the eyes with satisfactory accuracy in all five data sets in less than 1.5 min running on a SUN Sparcstation 1+ (including I/O). The results of all five data sets can be seen in Figure 7. The circles drawn there are the cross-section of the estimated ellipsoid with each slice. The x, y axis localisation of the eyes is very accurate. The z dimension of the ellipsoid fitted is such that the whole eye is included in it. The primary goal is to protect the eyes from radiation, so to ensure that the eye is completely contained within the ellipsoid found is important. There are no data dependent parameters determining the operation of the system. All the parameters hardcoded in the programme are model dependent. This is the system's knowledge about the eyes as a general geometric shape. Minimal assumptions are also made about the nature of the data (*e.g.* one major object present, patient's head orientated towards the top side *etc.*). These assumptions are used for the significant relaxation of the computational task to be performed.

The system conforms with all problem constraints mentioned in the specifications section, being especially robust and quick. It is therefore a reliable alternative solution to manual or interactive segmentation, and it can save valuable time for the human operator.

[9] Although more demanding in computational terms and for this note used so far

10 Acknowledgement

The CT images were provided by the German Cancer Research Centre, Heidelberg, as part of COVIRA (COmputer VIsion in RAdiology) project A2003 of the AIM (Advanced Informatics in Medicine) programme of the European Commission.
Participants in the COVIRA consortium are:
Philips; Medical Systems, Best (NL) and Madrid (E); Corporate Research, Hamburg (D) (prime contractor)
Siemens AG, Erlangen (D) and Munich (D)
IBM UK Scientific Centre, Winchester (UK)
Gregorio Maranon General Hospital, Madrid (E)
University of Tuebingen, Neuroradiology and Theoretical Astrophysics (D)
German Cancer Research Centre, Heidelberg (D)
University of Leuven, Neurosurgery, Radiology and Electrical Engineering (B)
University of Utrecht, Neurosurgery and Computer Vision (NL)
Royal Marsden Hospital/Institute of Cancer Research, Sutton (UK)
National Hospital for Neurology and Neurosurgery, London (UK)
Foundation of Research and Technology, Crete (GR)
University of Sheffield (UK)
University of Genoa (I)
University of Aachen (D)
University of Hamburg (D)
Federal Institute of Technology, Zurich (CH)

References

[1] C. M. Brown. Inherent bias and noise in the Hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(5):493–505, September 1983.

[2] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:679–698, 1986.

[3] R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision*, pages 80–83. Addison-Wesley, 1992.

[4] J. Illingworth and J. Kittler. A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing*, 44:87–116, 1988.

[5] K. Kaggelides. Locating the eyes in CT brain scan data. Master's thesis, Department of Artificial Intelligence, Edinburgh University, 1992.

[6] L. O'Gorman and A.C. Sanderson. The converging squares algorithm: An efficient method for locating peaks in multidimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(3):280–288, May 1984.

[7] C.W. Therrien. *Decision Estimation and Classification : An Introduction to Pattern Recognition and Related Topics*. John Wiley and Sons, 1989.

[8] H.K. Yuen, J. Princen, J. Illingworth, and J. Kittler. A comparative study of Hough transform methods for circle finding. In *Proc. 4th Alvey Vision Conf.*, pages 169–174, 1989.

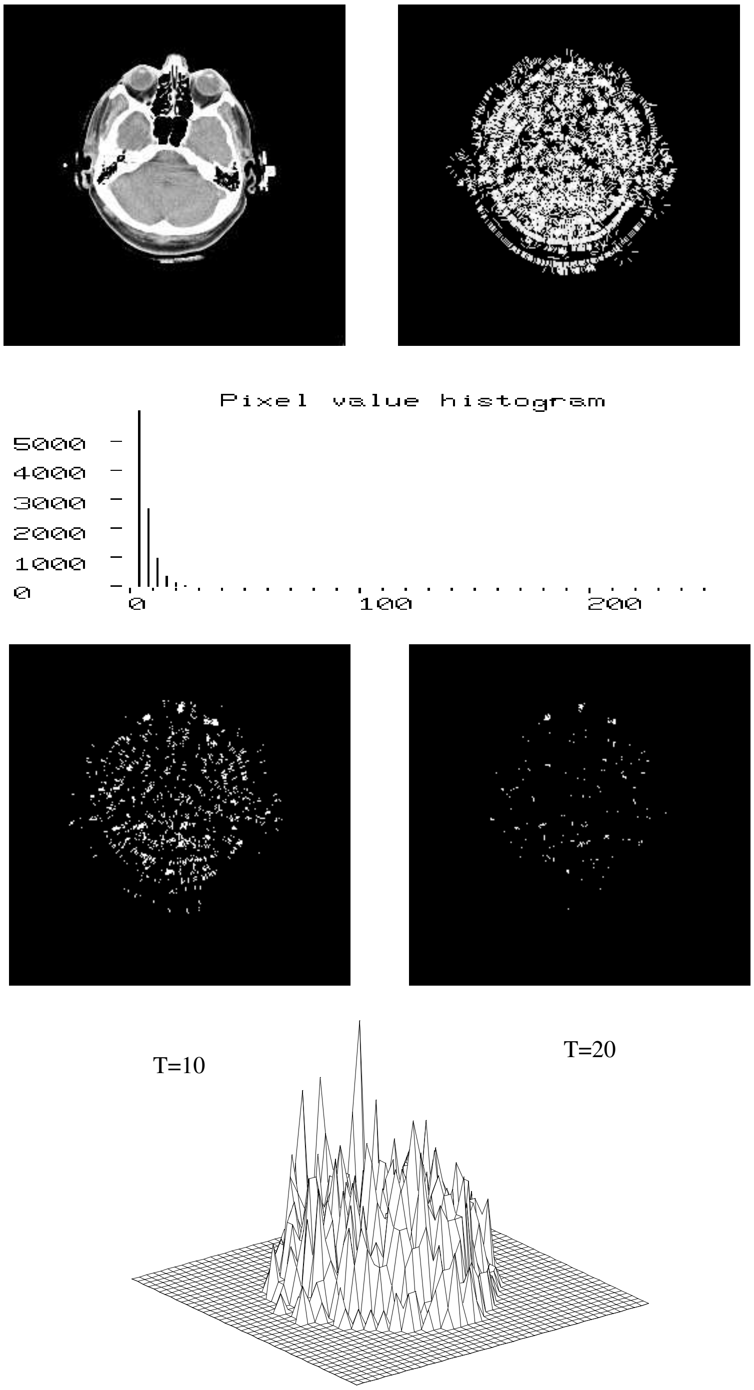

Figure 4: (a) Input Slice (b) Parameter Space projected onto one plane (c) Histogram of b (d) b Thresholded at T=10 and (e) T=20 and (f) a 3 Dimensional Plot of b

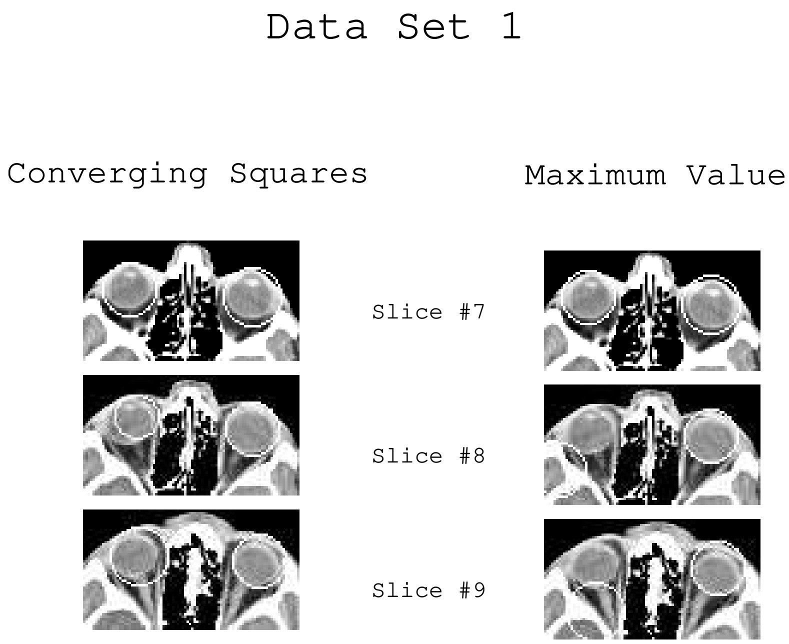

Figure 5: Results of the HT with two different search methods

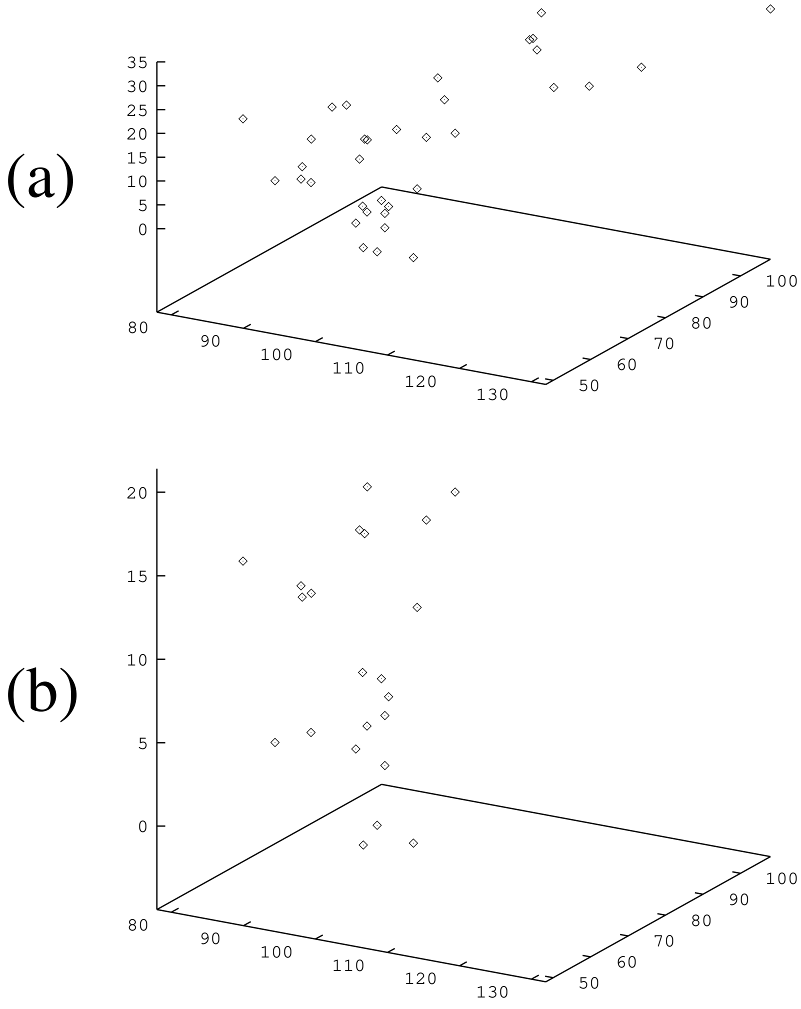

Figure 6: (a) Plot of x, y, z results for data set 1 and the left eye and (b) Magnification of (a) at z axis containing valid results within z=6 to 10

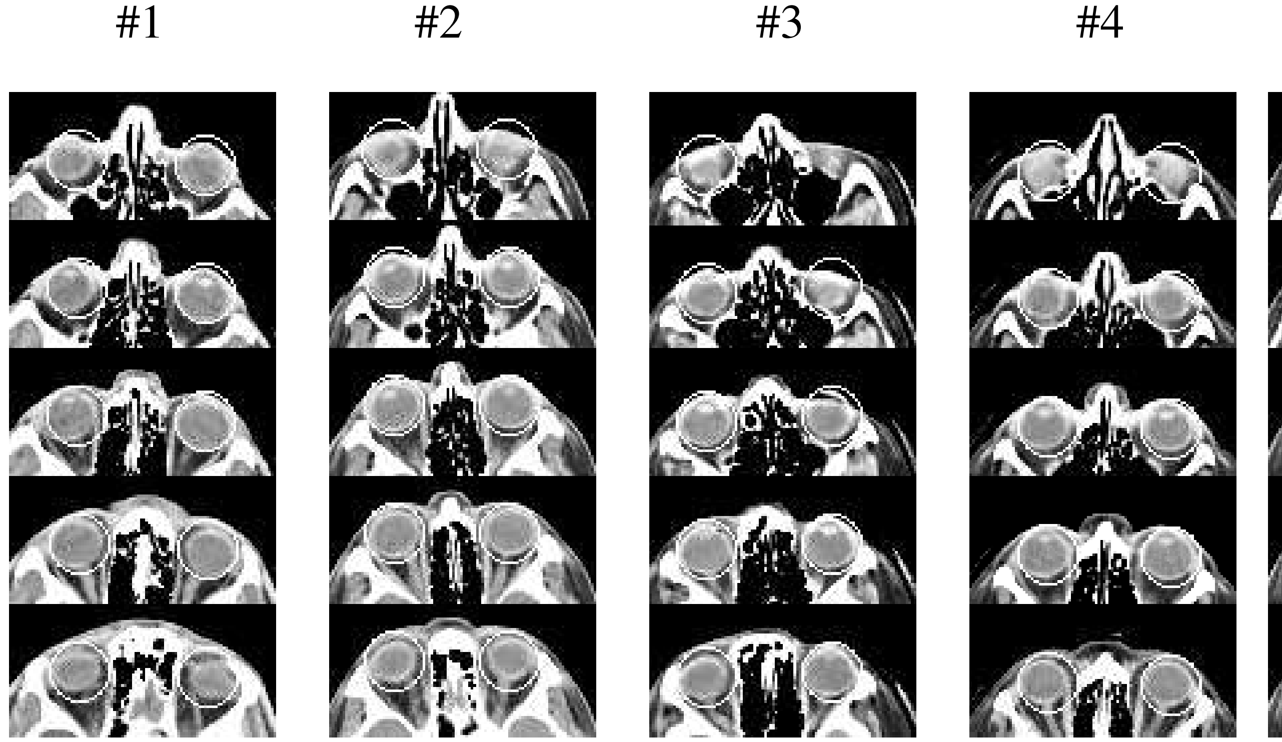

Figure 7: Results of the system applied to the first data set