Underwater vehicle path planning using a multi-beam forward looking sonar

Y. Petillot, I. Tena Ruiz, D. M. Lane,
Y. Wang, E. Trucco, N. Pican,
Ocean Systems Laboratory
Computing & Electrical Engineering
Heriot-Watt University
Riccarton, Edinburgh, EH14-4AS, UK http://www.cee.hw.ac.uk/oceans

Abstract— **This paper describes a new obstacle avoidance and path planning system for underwater vehicles based on a multi-beam forward looking sonar sensor. The real-time data flow (acoustic images) at the input of the system is first processed (segmentation and feature extraction) to create a representation of the workspace of the vehicle. This representation uses constructive solid geometry (CSG) to create a convex set of obstacles defining the workspace. We also take advantage of the real-time data stream to track the obstacles in the subsequent frames to obtain their dynamic characteristics. This will also allow us to optimise the preprocessing phases in segmenting only the relevant part of the images as well as to take into account obstacles which are no longer in the field of view of the sonar in the path planning phase. A well proven nonlinear search (sequential quadratic programming) is then employed, where obstacles are expressed as constraints in the search space. This approach is less affected by local minima than classical methods using potential fields. The proposed system is not only capable of obstacle avoidance but also of path planning in complex environments which include fast moving obstacles. Preliminary results obtained on real data are shown and discussed.**

Keywords: Path planning, Obstacle avoidance, Sonar, Underwater robotics

I. Introduction

We address here the general path planning and obstacle avoidance problem for an underwater vehicle using high resolution real-time sonar sensory data.

Although related problems such as 2D map building, environment modeling [1], [2] and motion estimation could be tackled in the framework of the presented system, we will focus on obstacle avoidance.

Until recently, most obstacle avoidance systems used low resolution or low frame rate sonar sensors yielding inaccurate estimations of the obstacles positions and movement. These systems were suitable for reactive obstacle avoidance ('reflex behaviour') but not for real path planning in a complex and changing environment.

With the recent development of reliable high resolution multi-beam sonars, a new range of methods have emerged which allow for a more detailed description of the environment and has broaden the spectrum of techniques that can be used [3], [4], [5], [6], [7].

A. Target application

Our aim is to develop an obstacle avoidance system for the ARAMIS (Advanced ROV Package for Automatic Mobile Investigation of Sediments) tool-skid, where ROV stands for Remotely Operated Vehicles. ARAMIS (MAST-CT97-0083) provides a geological/scientific tool-skid which will be mounted on two different ROVs, VICTOR from IFREMER (France) and ROMEO from CNR-IAN (Italy), operating at a close distance from the seabed (2 meters) at depths ranging from 50 meters to 2000 meters. The cruising speed for both ROVs is around 1 knot and the movements of the ROVs are measured by several on-board sensors feeding the obstacle avoidance system with position, speed and orientation of the vehicle in world coordinates.

B. Obstacle avoidance system overview

The system we are currently designing (see figure 1) is modular in nature. Each module performs a distinct function :

B.1 Segmentation:

Considering the very nature of multi-beam sonar images, we have decided to discard the certainty grid approach [8] often used in air ultrasonic sensor based motion planning and to focus on an object oriented description of the workspace.

B.2 Feature extraction:

Once the image has been segmented, the visible obstacles with their main features (position, moments, area, ...) are identified. These features will be used later to discard false alarms and track the obstacles and the vehicle.

B.3 Tracking:

This is one of the most important part of the system as it provides a dynamic model of the obstacles. Moreover, considering the amount of data to be processed, the tracking drives the segmentation and reduce the computational cost. It also enables us to create a world coordinates map of the obstacles surrounding the ROV including those that are no longer in the field of view of the sonar.

B.4 ROV dynamic modeling:

We have a dynamic and kinematic model of Angus 002, an ROV developed in Heriot-Watt University [9]. This model takes into account any type of sea current. It will be used until we have a model of the ROVs used in the project. In any case, the Angus model gives a realistic description of the behaviour of a typical ROV.

B.5 Workspace representation:

From the extracted obstacles and features of the current image, we can build an intra-frame workspace (frozen in time). Combining this intra-frame workspace with previous instances of intra-frame workspaces, a new dynamic workspace is built and constantly updated. It forms the basis for the path planning algorithm.

B.6 Path planning:

A nonlinear programming technique based on a constructive solid geometry representation of the obstacles is used for the path planning. Each obstacle in the workspace is represented as a constraint that has to be met in the search space (path not crossing the obstacle) while optimizing the Euclidean distance to the goal. This approach is based on our previous work [5], [6].

Each module will now be detailed and its main characteristics explained. Results will be presented for each module and the general path planner will be demonstrated in section V on real data.

II. Segmentation

Multi-beam sonar images are generally noisy and need to be filtered. A common segmentation procedure consists of median filtering followed by thresholding. The filtering part is generally very time-consuming. We have tried several techniques and found out that a good compromise between quality and speed was reached using the following scheme:

• **Filtering:** The filter used to remove the backscatter noise is a 7×7 mean filter which yields results almost as good as the median filter even on noisy images but at a reduced computational cost [10].

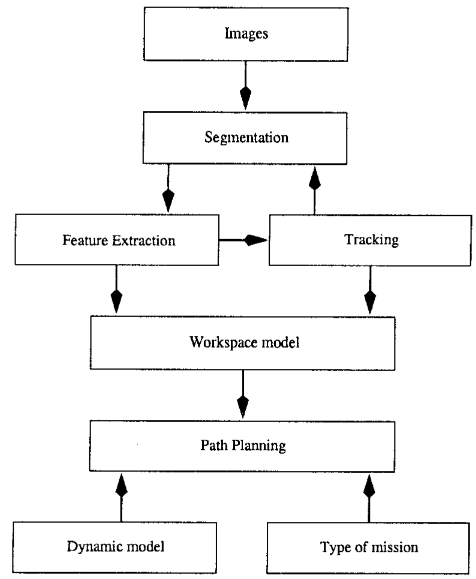

Fig. 1: Principle of sonar based real time path planning.

• **Threshold :** A single fixed threshold generally gives results which are highly dependent on the background level. We have used an adaptive thresholding technique based on the image histogram which is independent of the actual signal level. The idea is to estimate the noise probability density function assuming that the histogram of the studied image is a good estimate of it (thus assuming that obstacles are small in the image). The false alarm rate (FAR) is then fixed and used in conjunction with the histogram to derive the threshold value. Special cases where the images are mainly composed of obstacles (with high returns) or with a lot of backscatter noise from the seabed can easily be detected as they show a high variance. The process can then be adapted to these special cases. Although not as good as a real adaptive filter where the threshold value is *locally* derived with respect to the surrounding pixels, this technique gives results which match the real-time constraints. In the future, this scheme will be used on a subsampled image to detect potential obstacles in the scene while the tracking will allow the use of more elaborate techniques on selected parts of the scene.

Two examples of the same segmentation process on a clear and a noisy images are given in figure 2. It should be noted that no tuning parameter is needed and that the process is fully automatic.

III. Feature extraction and Tracking

Once segmented, the different regions representing the obstacles in the image are labeled using a standard labeling algorithm, and the major features of each obstacle are

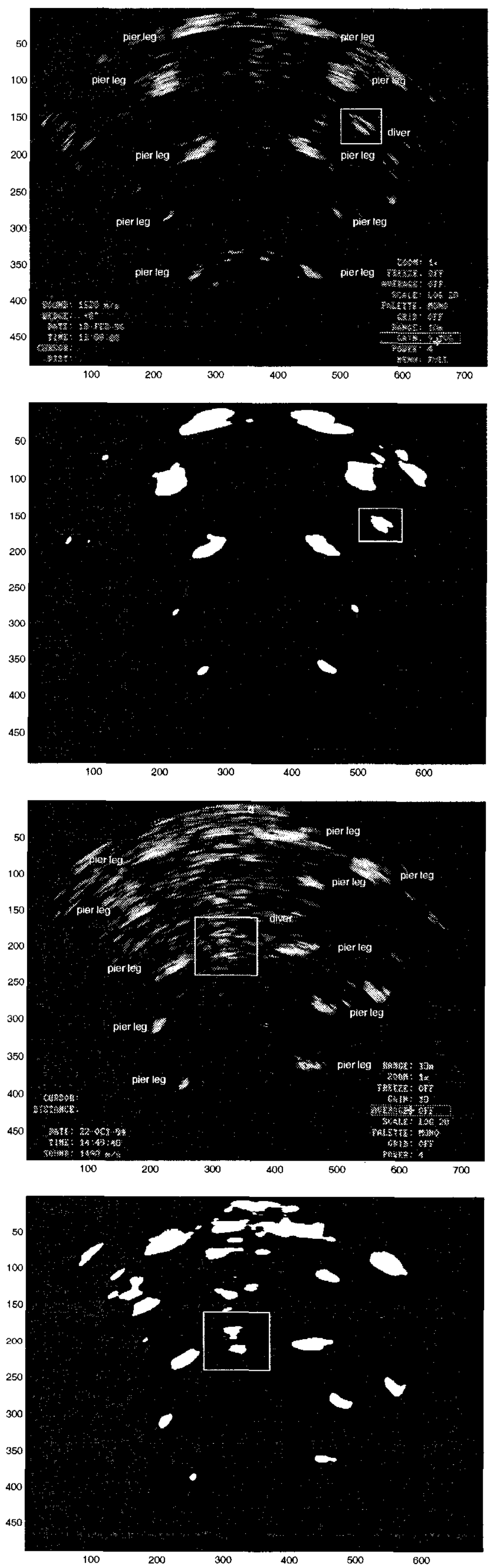

Fig. 2: Segmentation of multi-beam high-resolution sonar images containing pier legs and a diver with a clear image (top) and a noisy image (bottom). Scans were obtained from a Seabat 6012 from Reson.

extracted (area, perimeter, moments). In order to track the obstacles in a sequence of images, the first step is to associate the new observations (obstacles) of the image under study with the tracks of the previously identified obstacles. This is known as the data association phase. We employ a nearest neighbour algorithm to associate the predicted position of the tracked obstacles with the new observations. Three cases are then possible :

- There is a new observation matching the predicted position. A Kalman filter is then applied, a new state vector derived and new internal values computed.
- No new observation matches the prediction. The obstacle prediction is then updated using the Kalman filter internal values which are not updated. If no match is found between the observations and a given tracked obstacle on a predefined number of frames, the obstacle is discarded as a false alarm.
- An observation is not associated to any prediction, a new obstacle is created and its corresponding Kalman filter initialized.

This scheme has been successfully tested on simulated data but not yet on real segmented data. This will be done in the near future. For real data, the extracted features will be used as a consistency test in the data association algorithm. Obstacles will also be merged if they appear to be the result of several returns from the same obstacle, a very common situation with moving obstacles when using multi-beam sonars(see figure 2(b)). We are currently developing and testing standard Kalman filter and extended Kalman filters for tracking realistic underwater targets. More details on tracking and data association schemes can be found in [11].

IV. Workspace representation and path planning

The choice of a workspace representation is intimately linked with the path planning technique which will be applied. Most path planning algorithms assume a convex representation of the obstacles to ensure that the goal will be reached. When dealing with a changing environment which is sensed on the fly, it is better to use a reactive path planning technique which does not need a complete description of the workspace from the current position to the goal for:

- Only partial information is available (due to the limitations of the sensor).
- New obstacles can appear in the workspace at any time.
- The precision in the representation of the obstacles will change with their respective distance to the vehicle.

Global path planning techniques need a complete description of the workspace as they define the complete path from the start to the goal while local path planning ones define a partial path towards the goal given a possibly incomplete representation of the neighbouring workspace. Global path planning is therefore not advisable here as the enviroment is sensed while moving and therefore the Workspace is continuously changing.

We have chosen to use a local path planning technique based on some of our previous work [5], [6] where only the next step of the path leading towards the goal is calculated. The central idea of the method is to represent the free space of the workspace as a set of inequality constraints of a nonlinear programming problem. The goal point is designed as a unique global minimum of the objective function. The initial configuration of the vehicle is treated as the starting point of the nonlinear search. Constructive Solid Geometry (CSG) is used to represent the free space of the robot as a set of inequalities.

A. *Workspace representation using CSG*

In the following, we operate in the configuration space of the vehicle. It means that in this space, which integrates both the kinematics and the link geometry of the vehicle, the vehicle can be represented as a point.

The choice of CSG to represent obstacles is driven by the fact that classical surfaces such as spheres, cylinders, and half-spaces are CSG primitives that can be very easily combined.

Each obstacle in the workspace is given a mathematical representation. Let S be the $2D$ or $3D$ surface of an Euclidean space E representing the obstacle, and let's denote its interior points by I, its boundary points by B and its exterior points by T in a topological sense:

$$\begin{array}{l} I \cup B \cup T = E \\ I \cap B = B \cap T = I \cap T = \emptyset \end{array} \tag{1}$$

The non-negative function f on E is called a defining function (in the CSG sense) of the obstacle S if:

$$\begin{array}{ll} \forall\, p \in I, & 0 < f(p) < 1, \\ \forall\, p \in B, & f(p) = 1, \\ \forall\, p \in T, & f(p) > 1. \end{array} \tag{2}$$

As an example, the defining function of an ellipse when $E = \mathbb{R}^2$ is :

$$\forall\, p \in \mathbb{R}^2, \quad f(p) = (x/a)^2 + (y/b)^2 \tag{3}$$

where a and b are the half-axes of the ellipse and p is the point of coordinates (x, y) in the plan.

One of the major interest of CSG lies in the fact that complex objects can easily be constructed from simple canonical objects using the union and intersection operations:

$$\forall\, p \in E, \quad f^I(p) = max(f_1(p), f_2(p), ..., f_n(p)) \tag{4}$$

defines the intersection of n objects whose respective defining functions are $f_1, f_2, ..., f_n$ while

$$\forall\, p \in E, \quad f^U(p) = min(f_1(p), f_2(p), ..., f_n(p)) \tag{5}$$

defines the union of the same objects. These functions are difficult to obtain in practice and are replaced by the following approximation:

$$f^I = (f_1^m + f_2^m + ... + f_n^m)^{\frac{1}{m}} \tag{6}$$

approximates the intersection of n objects whose respective defining functions are $f_1, f_2, ..., f_n$ while

$$f^U = (f_1^{-m} + f_2^{-m} + ... + f_n^{-m})^{\frac{-1}{m}} \tag{7}$$

approximates the union of the same objects. m is any positive real number. m can be used to control the accuracy of the smooth approximation and can be used to obtain convex unions and intersections [5].

Here, for the sake of simplicity and without loss of generality, we have decided to represent the obstacles as ellipses. More general representation are possible, including polygonal ones [6]. From the real obstacles contours, their convex hull is extracted and an optimal elliptic fitting algorithm is applied to obtain the representation of a given obstacle.

B. *Path planning algorithm*

All obstacles O_i ($i \in [1, n]$) of the workspace are defined as ellipses whose defining functions g_i in a $2D$ Euclidean space are defined in equation 3. The free space of the vehicle with respect to obstacle O_i is defined as:

$$\{p \in E \mid 1 - g_i(p) < 0\} \tag{8}$$

Therefore the complete free space of the vehicle can be represented as:

$$\{p \in E \mid \forall\, i \in [1, n], \;\; 1 - g_i(p) < 0\} \tag{9}$$

Let's now define the objective function f representing the practical problem to be solved, in our case, the minimum distance from the start to the goal point in the configuration space as:

$$\forall\, p \in E, \quad f(p) = (p - p_g)^t (p - p_g); \tag{10}$$

where p_g designs the goal point and ^{t} is the transpose operation.

We now have completely defined our path planning problem:

Optimize f under the constraints $g_i,\ i \in [1, n]$.

This is a classical problem in optimisation. As in the general case the defining functions are nonlinear, we use well-proven numerical nonlinear programming techniques to solve the path planning problem. This approach generates very smooth paths compatible with feasible vehicle motion. The effect of each constraint can be clearly seen while it is often hidden in a single objective function in other optimisation techniques such as potential fields where the careless definition of the potential functions can easily lead to local minima. Finally the CSG modeling of the obstacles offers a lot of flexibility in the representation of the workspace.

V. Results

Although the tracking has been tested on simulated data, it has not been implemented yet in the general system. We have tested the combination of the segmentation, the feature extraction and the path planning modules on real sequences of sonar data. In order to test the algorithm, we used sequences composed of pier legs and a moving diver taken from a still sonar and we have simulated the movement of a "blind" ROV, driven according to the data received from the sonar. This does not alter the validity of the approach as the tracking module will enable the creation (and update) of a dynamic map of the environment in real world coordinates.

An animated MPEG version of the results displayed in figures 3 and 4 can be found on our Web page at: http://www.cee.hw.ac.uk/~aramis/resources/.

The goal is set in so that the generated path crosses the path of the moving diver. On figures 3 and 4, the left image is the original image while the right image is the segmented image showing the identified obstacles. On the segmented images, the obstacles contour are displayed in cyan while their elliptic representation is displayed in red. The planned path is also drawn on both sequences of images (in yellow).

In order to achieve faster processing time, the original images (300×700 pixels) were subsampled by a factor of 4 in both directions. Using Matlab 5.2 on a Sun Ultra-10, the whole process (segmentation, workspace representation and path planning) takes 3.8 seconds per frame. Considering a frame rate of a few images per seconds and using optimised code, a real-time system is certainly achievable using the framework we present here.

VI. Conclusion

We have presented here a general framework for performing $2D$ and $3D$ obstacle avoidance and path planning for underwater vehicles based on a multi-beam forward looking sonar sensor.

The ability of the system has been demonstrated on real sonar data. The sequence used is very noisy, thus corresponding to realistic situations, and the system still performs very well. Compared to other methods, our system generates very smooth paths, can handle complex and changing workspaces and presents no local minima as we use a convex representation for the obstacles.

Future work will include the tracking module as well as the dynamic models of the ROVs ROMEO and VICTOR. Each module will then be optimised with respect to the missions that will have to be handled.

VII. Acknowledgments:

This research has been supported by the European community under the MAST project CT97-0083.

References

[1] J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar sensing for mobile robot navigation.* The Kluwer international series in engineering and computer science, Kluwer Academic Publishers, 1992.

[2] R. Bono, M. Caccia, and G. Veruggio, "Reconstructing 2-D maps from multiple sonar scans," *Oceans Conference Record (IEEE)*, vol. 1, pp. 164–169, 1994.

[3] L. Henriksen, "Real-time underwater object detection based on electrically scanned high-resolution sonar," in *1994 Symposium on Autonomous Underwater Vehicle Technology, AUV'94*, (Cambridge, MA, USA), IEEE Oceanic Engineering Society, july 1994.

[4] M. J. Chantler and J. P. Stoner, "Automatic interpretation of sonar image sequences using temporal feature measurements," *IEEE journal of oceanic engineering*, vol. 22, pp. 47–56, january 1997.

[5] Y. Wang and D. M. Lane, "Subsea vehicle path planning using nonlinear programming and constructive solid geometry," *IEE Proceedings-Control Theory Applications*, vol. 144, pp. 143–152, 1997.

[6] Y. Wang and D. M. Lane, "Path planning for underwater vehicles using constrained optimisation," in *Oceanology International Conference 1998*, (Brighton), pp. 175–186, march 1998.

[7] D. M. Lane, M. J. Chantler, and D. Dai, "Robust tracking of multiple objects in sector-scan sonar image sequences using optical flow motion estimation," *IEEE journal of oceanic engineering*, vol. 23, pp. 31–46, january 1998.

[8] J. C. Latombe, *Robot motion planning.* Boston: Kluwer Academic Publishers, 1991.

[9] P. Bellec, "Simulation of the 6 degree of freedom motion of a remotely controlled unmanned submersible," Master's thesis, Heriot-Watt University, Edinburgh, EH14 4AS, Riccarton, 1980.

[10] R. Gonzalez and R. Woods, *Digital Image Processing.* Addison-Wesley, 1992.

[11] Y. Bar-Shalom and T. E. Fortmann, *Tracking and data association*, vol. 179 of *Mathematics in Science and Engineering.* Academic Press, 1988.

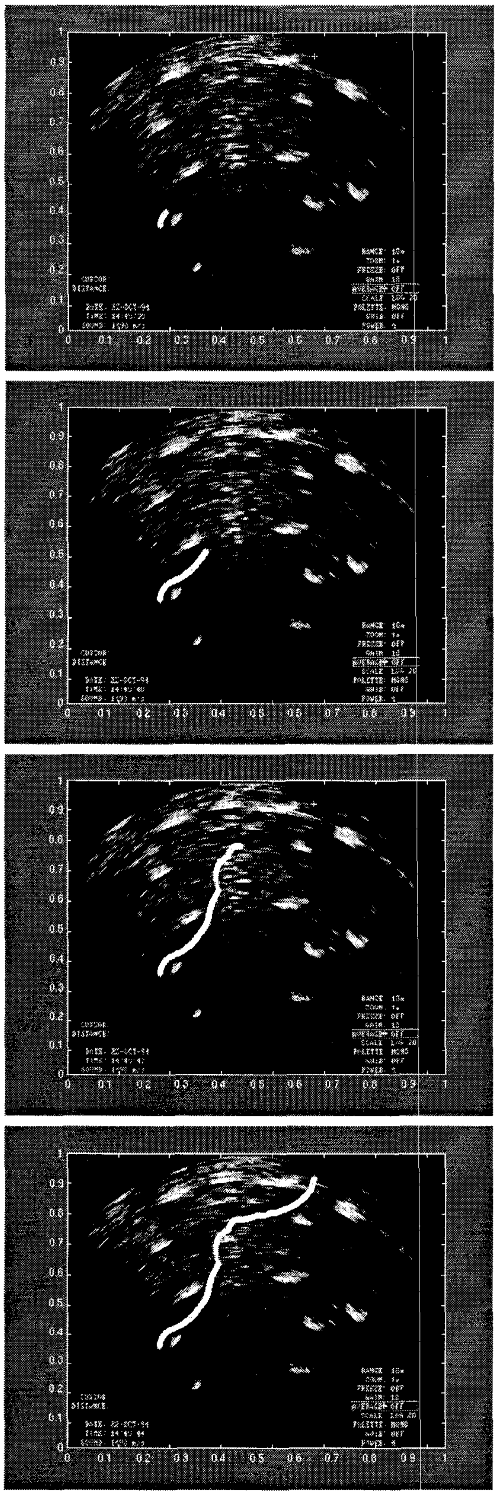

Fig. 3: Example of path planning on a sequence of real images containing the diver (in the centre) and several pier legs. The blue cross represents the goal.

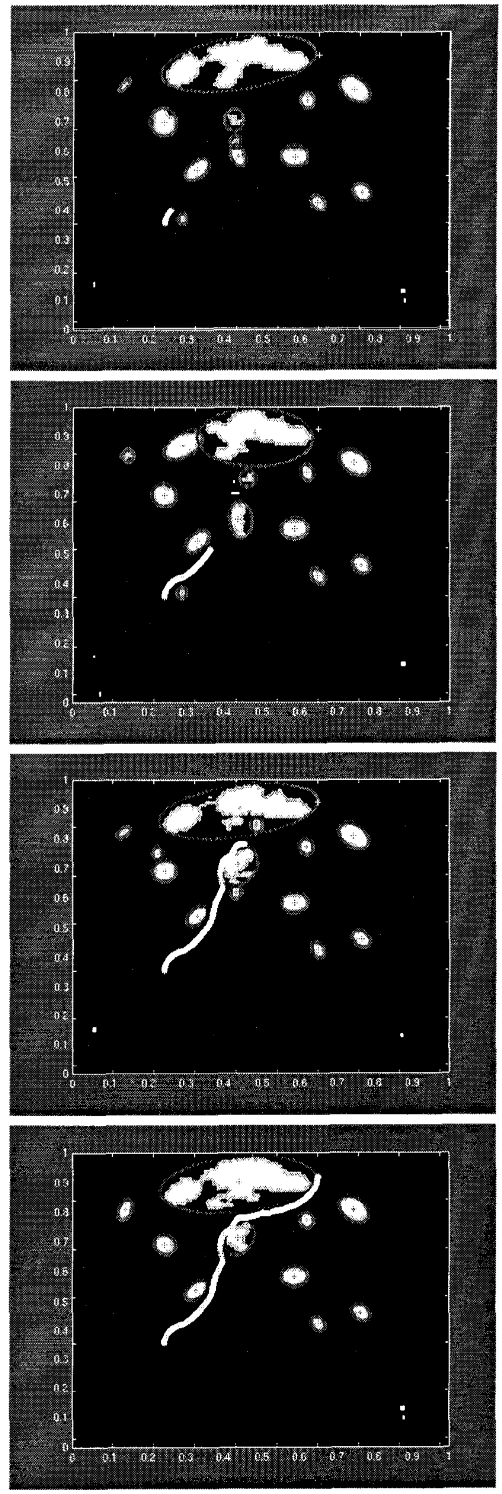

Fig. 4: Same example with the segmented images and their associated elliptic representations. The boundaries of the segmented objects are represented in cyan