CENTROID BASED TRACKING OF THE OBJECTS EXTRACTED FROM THE IMAGES RECEIVED FROM SONAR

Raj Kumar Goswami and G Sasibhushana Rao
Department of ECE
Andhra University College of Engineering (A), Andhra University, Visakhapatnam-530003
rajkumargoswami@gmail.com

Abstract:

Image processing forms an integral part of the design of every Autonomous Underwater Vehicle (AUV) where the vehicle is required to be controlled in the constrained environment. The objects detected in the images provided by the imaging sonar fitted on board AUV are tracked and then their trajectory is determined. These are then correlated with the speed and direction of the AUV and then by using the triangulation method the collision course can be calculated and finally the collision is avoided by giving the appropriate command to the AUV. In this paper a tracking method based on the centroids of the objects has been presented and the trajectories are then determined by using the Kalman filter. The implementation has been undertaken in the MATLAB.

Keywords: Image Processing, Kalman Filtering Centroid based Tracking

1. Introduction:

There are various methods which have been used for tracking the objects in the image sequences received from the equipment fitted on the vehicle. These methods of tracking the objects are primarily based on the applications and their usage. But if the collision avoidance is the only aim then the time required for calculation becomes the important criteria as the action has to be taken before the collision occurs. Towards achieving this, following steps are required to be undertaken:-

a. Pre-processing [1] on the images to remove the noise / blurr from the images. (In present day applications this step is generally performed by the Sonar)

b. Identification of the objects by undertaking the blob analysis [2].

c. Calculation of the centroids for all the detected blobs.

d. The steps 1 to 3 are performed then on all subsequent images

e. Identification of the moving and stationary objects.

f. Determination of the association of the moving objects based on the maximum speed criterion.

g. Tracking of the moving objects using Kalman Filter [3]

h. Calculation of the trajectory.

i. Calculation of Collision course by taking the own speed and direction into consideration.

j. Finally giving the maneuvering commands to AUV

In this paper, we have proposed a simple method of tracking the objects based on their centroids. It has been assumed that the images those are received are noise free (pre processed) and the blobs have been identified. Therefore only the calculation of the centroid, tracking of the blobs and the calculation of the trajectories has been presented in this paper. The complete simulation has been undertaken in the MATLAB environment.

2. Kalman filter:

Kalman filter is one of the well known and often used tools for stochastic estimation of variables of interest from noisy sensor measurements. Kalman filter is simply an optimal recursive data preprocessing algorithm and is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in such a way that it minimizes the mean of the squared error (optimal). It applies to stationary as well as nonstationary environments. The solution is recursive it means that in each updated estimate, the state is computed from the previous estimate and the new input data, so only the previous estimate requires storage. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown. A linear, discrete-time dynamical system is described by the block diagram shown in Fig 1.

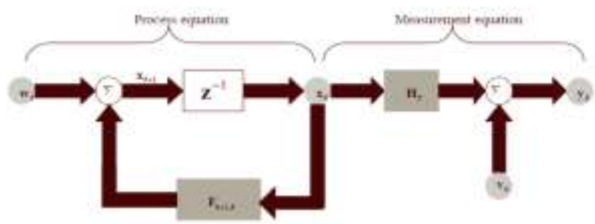


Fig. 1

The concept of *state* is fundamental to this description.
The *state vector* or simply state, denoted by x_k, is defined as the minimal set of data that is sufficient to uniquely describe the unforced dynamical behavior of the system; the subscript *k* denotes discrete time. In other words, the state is the least amount of data on the past behavior of the system that is needed to predict its future behavior. Typically, the state x_k is unknown. To estimate it, we use a set of observed data, denoted by the vector y_k.

$$x_k = Ax_{k-1} + w_{k-1} \qquad (1)$$

where w_k is random process noise and Eq. (1) is called process model.

$$y_{k-1} = Hx_{k-1} + v_{k-1} \qquad (2)$$

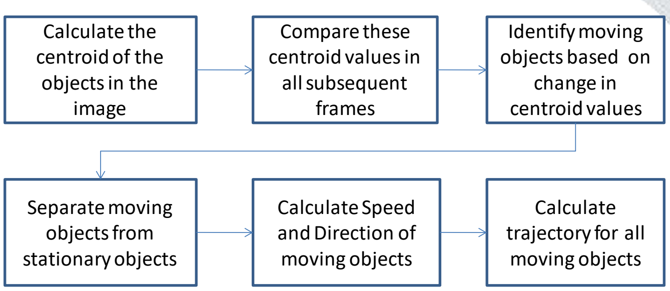
where v_k is random measurement noise and Eq.(2) is called measurement model.

$$p(w) \sim N(0,Q)$$
$$p(v) \sim N(0,R)$$

In practice, the *process noise* covariance Q and *measurement noise* covariance R matrices might change with each time step or measurement; however in our analysis we have assumed them to be constant.

3. Implementation of Tracking Algorithm:

Following block diagram explains the steps, those have been implemented to track the objects in the images received from the imaging Sonar of AUV.



The heart of the implementation of Tracking Algorithm is Kalman filter, which estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. Therefore the equations for the Kalman filter can be categorized into two sets:

(a) *Time update* equations

(b) *Measurement update* equations

The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the *a priori* ($\hat{x}_k^-$) estimates for the next time step. The measurement update equations are responsible for the feedback i.e. for incorporating a new measurement into the *a priori* estimate to obtain an improved *a posteriori* estimate ($\hat{x}_k$). The time update equations are called as *predictor* equations, while the measurements update equations as *corrector* equations. The final estimation algorithm is nothing but a *predictor-corrector* algorithm. The following are the steps for this algorithm.

Step 1: Project the state ahead: $\hat{x}_k^- = A\hat{x}_{k-1} + w_k$

Step 2: Project the error covariance ahead:

$$P_k^- = AP_{k-1}A^T + Q$$

Step 3: Compute the Kalman gain:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

Step 4: Update estimation with measurements:

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - H\hat{x}_k^-)$$

Step 5: Update the error covariance: $P_k = (I - K_k H)P_k^-$

Step 6: Repeat and go to Step 1.

The matrix *K* in (step 3) is chosen to be the *gain* or *blending factor* that minimizes the *posteriori* error covariance.

The state x_k contains the position (*x, y*) of the object in the instant *k* and also the speed of the object in both x ($\dot{x}$) and y ($\dot{y}$) directions.

The state equation is then in this case is defined as

$$x_k = Ax_{k-1} + w_{k-1}$$

$$\begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{bmatrix} + w_k$$

where 't' represents the time interval between any two time instants.

And the process measurement equation is written as

$$y_k = Hx_{k-1} + v_{k-1}$$

$$\begin{bmatrix} y_{1k} \\ y_{2k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{bmatrix} + v_k$$

4. Data Association:

Data Association plays a very important role in all tracking applications, especially in the environment which is heterogeneous and having frequent occlusion conditions. This is aptly applicable to the undersea environment where AUV operates.

To check that the objects those have appeared in the subsequent images actually belong to the same target following two conditions have been implemented in our algorithm:-

(a) The maximum pixels by which the centroid values will change in subsequent frames, if the objects are moving with the maximum speed.

(b) The variation in the position of the centroid due to the movement of water body / occlusion.

For calculating the maximum pixels following inputs are required:-

(a) Maximum speed that any object can have is X Km per Hour

(b) Frame rate is assumed to be n frame per second.

(c) The row and column pixels are assumed to be k by k (Image size).

(d) The Range of the Sonar (R) in terms of meters

Based on these above inputs the pixels are calculated which will be covered by the object who is moving with the maximum speed. The pixel values can be calculated as follows:-

Maximum distance that will be covered by the objects between consecutive frames is

[(X*1000/3600)*1/f] Meters

The resolution of the image is

R/ k meters per pixels in row and column both

Therefore the number of maximum pixels those will be covered will be

[(X*1000/3600)*1/f] *[R/k] (3)

In our simulation we have taken following values

X = 18 Km Per Hr

Frame Rate (f) is 1frame per second

Image size is taken as 300 by 300

The range of the Sonar is 300 meters

By substituting the above values in the equation 3, we can calculate the number of pixels that will be covered by the object

[(18*1000/3600)*1/1] *[300/300] = 5 Pixels

Therefore if the centroid in the consecutive images has changed by less than equal to the 5 pixels then it is assumed to be from the same object i.e. they are said to be associated. The speed of the object is then calculated accordingly. The tracking algorithm then invoked for these objects and the trajectory calculated. Similarly if the centroid remains within the specified values (depends on the water body movement) in all subsequent frames then it is assumed to be stationary.

5. Result:

In our simulation the noise has been assumed to be white Gaussian with zero mean. Fig. 2 shows the mean squared error of filtered and the smoothed estimate of the object tracking using Kalman filter.

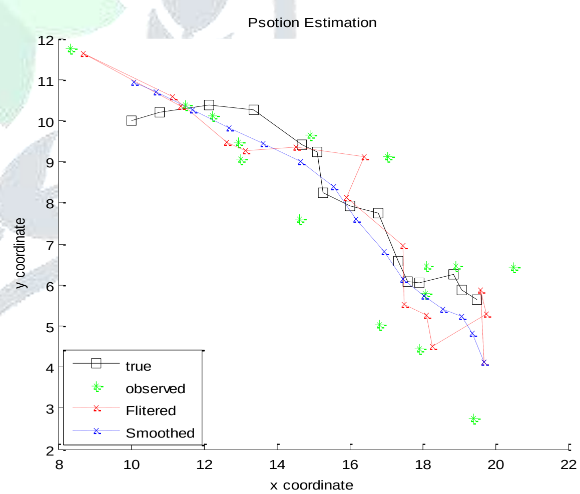


Fig 2

From the results it is observed that the mean square error of the filtered estimate is 5.0578; for the smoothed estimate it is 3.2473.

Fig 3 also indicates the measured value as well as the predicted values of all the three objects. These objects were simulated with various speeds and the directions. Total twenty image frames were taken for analysis. The trajectories those were estimated for all three objects using the Kalman Filter were as anticipated. It can be seen that the real positions those have been found in the images are close to the values predicted by the Kalman Filter. In our algorithm the objects have been associated by using the Data Association algorithm as mentioned above.

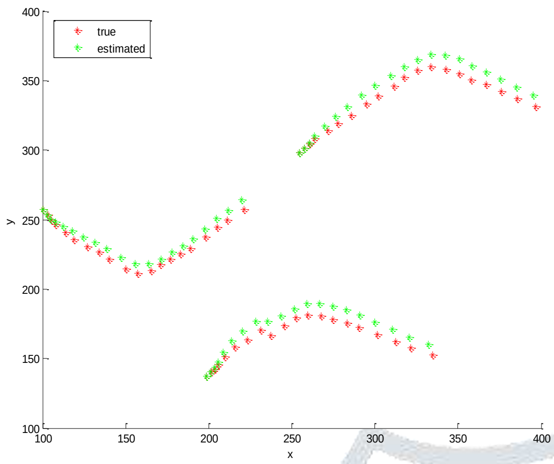


Fig 3

6. Conclusion and Future work:

It is concluded from the results that the smoothed estimate using the Kalman filter is better with smaller uncertainties and it was also concluded that the Kalman filters works fast when the vector state is small. In our simulation it has been confirmed by keeping the state vector small i.e. it had only two variables namely position, and the speed and the acceleration has been included as the part of noise. The future work should take the acceleration as the part of the state for having the best estimation of the trajectory.
In the future work it is proposed that the collision course be determined based on the trajectory calculated using this result.

7. References

[1] Xu, L.; Landabaso, J. L.; Lei, B.; "Segmentation and tracking of multiple moving objects for intelligent video analysis", BT Technology Journal, Vol 22, No 3, July 2004

[2] Umair Soori a, Nadira Nordinb & Mohd Rizal Arshad "Group Motion Tracking andiIts inherent difficulties Using Image Based Approach" The 2nd National Intelligent systems And Information Technology Symposium (ISITS'07), Oct 30-31, 2007,

[3] T.H. Lee, W.-S. Ra, T.S. Yoon and J.B. Park, "Robust Kalman filtering via Krein space estimation", IEE Proc.-Control Theory Appl., Vol. 151, No. 1, January 2004.

[4] Yifeng Zhu and Ali Shareef, "Comparisons of Three Kalman Filter Tracking Algorithms in Sensor Network", this work was supported by National Science Foundation grant 0538457 and an UMaine Startup Fund.

[5] Matthies L, Kanade T, and Szelisky R, "Kalman filter based algorithms for estimating depth from image sequences", Internat.J.Comput. Vision 3, 1989, 209-236.

[6] Ruiz Tena I, Petillot Y, Lane D and Bell J, "Tracking objects in underwater multibeam sonar images", Motion Analysis and Tracking (REF. No. 1999/103), IEE Colloquim on 10th May 1999, 11/1-11/7.

[7] I. Tena Ruiz, Y. Petillot, D. M. Lane, C. Salson "Feature Extraction and Data Association for AUV Concurrent Mapping and Localisation" Proceedings of the 2001 IEEE International Conference on Robotics & Automation Seoul, Korea • May 21-26, 2001

[8] Cina Motamed 2005, Motion detection and tracking using belief indicators for an automatic visual-surveillance system, Science direct Image and Vision Computing 24 (2006) 1192–1201.

[9] Gabriel J. Brostow and Roberto Cipolla 2006. Unsupervised Bayesian Detection of Independent Motion in Crowds. In Proceedings of the Computer Vision and Pattern Recognition, 2006 IEEE Computer Society, Volume: 1, 594-601: IEEE CVPR, June 2006, New York, NY.

[10] Stephen J. McKenna, Sumer Jabri, Zoran Duric, HarryWechsler and Azriel Rosenfeld 2000. Tracking Groups of People, Computer Vision and Image Understanding, no. 80, pp. 42--56, 2000, Academic Press Inc.

[11] C. Beleznai, T. Schlögl, B. Wachmann, H. Bischof and W. Kropatsch 2002. Tracking multiple objects in complex scenes, Vision with non-traditional sensors, Proc. of 26th Workshop of the Austrian Association for Pattern Recognition, F. Leberl and F. Fraundorfer (eds), vol. 160, pp. 175 - 182, Austrian Computer Society, 2002

[12] Tao Zhao and Ram Nevatia 2004. Tracking Multiple Humans in Crowded Environment. Proceedings of the IEEE Computer Society Conference on Computer