Unscented Kalman Filter for Visual Curve Tracking

Peihua Li Tianwen Zhang

Dept. of Computer Science and Engineering 360, Harbin Institute of Technology, Harbin, Hei Long Jiang Province, China,150001

Email: peihualj@hotmail.com

Abstract

Visual contour tracking in complex background is a difficult task. The measurement model is often nonlinear due to clutter in images. Traditional visual tracker based on Kalman filter employs simple linear measurement model, and often collapses in tracking process. The paper presents a new contour tracker based on Unscented Kalman filter that is superior to extended Kalman filter both in theory and in many practical situations. The new tracker employs more accurate nonlinear measurement model, without the computation of Jacobian matrix. Thanks to a set of appropriately chosen sample points, multiple hypothesises can be maintained and the best observation can be made during one time step. The algorithm can obtain more exact estimate of the state of the system, and has the same order of complexity as that of Extend Kalman Filter. The experiments show that the new algorithm is superior to those based on Kalman filter.

1. Introduction

In the last ten years, visual contour tracking has been broadly used in many research areas, such as object-based video coding, video surveillance, human-machine interface, biomedical image analysis etc. In this contour tracking field, methods based on probability and statistics have long been active research topics. Earlier work mainly focuses on Kalman Filtering[2,4,5,6], in which state equations and measurement equations are both constructed as linear ones. However, in real-world environment, due to the existence of the visual clutter, the measurement equations are most often nonlinear. This often cause the collapse of the linear Kalman Filter[9]. To address this problem, a new sequential Monte Carlo sampling algorithm, under the name of Condensation, has been developed[9]. The algorithm can deal with non-Gaussian, nonlinear contour tracking tasks in a unified way. But it often demands a large number of discrete samples, typically, several hundreds to thousands.

Julier et al recently proposed a new extension of the Kalman Filter, named unscented Kalman Filter, to address nonlinear filtering problems[15,16]. The algorithm is able to predict the state and covariance of the system more accurately and is easier to implement. In the algorithm, a small number of carefully chosen sample points are propagated in each estimation step, which provide compact parameterisation of the underlying distribution. It is in contrast to random sampling methods, such as Condensation which requires a larger number of sample points, and is therefore computationally expensive.

In the visual contour tracking paradigm, we present a tracking algorithm based on unscented Kalman filter. Compared with the previous work based on Kalman filtering, it has the following advantages:

- More accurate nonlinear measurement model can be employed, without computing its Jacobians.
- Due to carefully chosen sample points, we can have multiple measurement hypothesises and obtain the best observation in each time step, according to the measurement probability distribution function (*p.d.f*), which is in contrast with single hypothesis in Kalman filtering.
- More exact estimate of the state mean and covariance of the system can be achieved.
- It is more robust than tracking algorithm based on Kalman filter.

Compared with Condensation, our algorithm has the following advantage and limitation:

Advantage:

- It needs only a small number of deterministic sample points, and is therefore computationally efficient.

Limitation

- It assumes unimodal probability distribution essentially, and can not deal with multimodal distribution.

The structure of the paper is as follows. Section 2 presents a brief literature survey of visual contour tracking. In section 3, we introduce the system dynamics and measurement models, and then describe the tracking algorithm in detail. Experiments with real image sequence are demonstrated in section 4. The concluding remarks are given at last.

2. Previous Work

The probabilistic visual contour tracking is first investigates by Terzopoulos and Szeliski[2]. They employ the motion model of snake as a system model to construct a Kalman filter. The high computational cost of updating the filter, resulted from the representation of contours with discrete points, is reduced by allowing only the diagonal terms of the covariance of the estimate to vary with time. Blake et al used B-spline curve to represent the shape of the object, and employed the constant velocity model to describe the motion of the contour[5]. On the one hand, they develop a mechanism for incorporating an affine invariant shape template into the tracker. As a result, the visual tracker is selective for specific shape and therefore is able to ignore background clutter to some extent. On the other hand, the statistical insight into the tracker leads to automatically determining of spatio-temporal scale of feature search. Peterfreud proposes a velocity snake model, in which a control term is introduced that measures the difference between the contour velocity and the apparent velocity of the image [8]. Peterfreud constructs two Kalman snakes by incorporating the velocity snake dynamics into Kalman filter:1) the batch-mode Kalman snake model, which uses additional independent estimates of image velocity at contour position as an input control, 2) the real-time Kalman snake model, which uses optical-flow measurements, along with measurements of image-gradients, as the system observations [4]. If the motion model can be learned from training image sequences, the performance of the tracker will be enhanced to ignore background distractions. The problem of learning motion model and the corresponding Kalman filter was thoroughly investigated in[6].

The Condensation algorithm is proposed by Isard and Blake to estimate nonlinear probability density function of the state of the tracked contour[9]. The basic idea of Condensation is that the posterior density is approximated by a set of discrete samples with associated weights. At every time step the algorithm mainly involves three stages, including sampling stage—drawing discrete samples from the transition prior, updating stage—computing the new weights, and resampling stage—resampling to ensure a uniform weight distribution. Because the algorithm does not take advantage of the most current observation in sampling stage, it demands a large number of samples[10]. Much effort has been devoted to address the problem. Isard et al proposed an importance sampling method, in which another tracker —blob tracker is incorporated into Condensation for hand tracking[11]. In many situations, however, the auxiliary tracker can not be obtained. Peihua et al combine the newest measurements into the sampling stage by presenting a sub-optimal importance proposal density, to decrease the number of samples in the algorithm[12]. MacCormick et al developed partitioned sampling which, nevertheless, required that the state space can be sliced[13]. Sullivan et al proposed layer sampling using multi-scale processing of images[14]. How to decrease the computational load of Condensation is an active research topic at present.

3. Unscented Kalman Filter for Visual Curve Tracking

Unscented Kalman filter was first proposed by Julier et al to address nonlinear state estimation in the context of control theory[15]. The algorithm uses a set of carefully chosen sample points to capture mean and covariance of the system. The samples are propagated through true nonlinear equations, the linearization of which is unnecessary at all. They can capture the states of the state up to 2nd order, and has the same order of computation complexity with Extended Kalman filter. It is superior to EKF both in theory and in many practical situations [15,16,17]. In the section, we implement the Unscented Kalman filter in the visual contour tracking framework. After introducing the motion model and measurement model briefly, we present the tracking algorithm in detail.

3.1 System Dynamics

The system dynamics modelling follows that described in[5,6]. The tracked objects are modelled as B-spline curves

$$r(s,t)=\begin{bmatrix} B(s)^T & 0 \\ 0 & B(s)^T \end{bmatrix}\begin{bmatrix} Q^x(t) \\ Q^y(t) \end{bmatrix} \quad \text{for} \quad 0 \le s \le L \quad (1)$$

where $B(s)=\begin{bmatrix} b_0(s) & \cdots & b_{q-1}(s) \end{bmatrix}^T$, $b_i(s)\,(0 \le i \le q-1)$ is the ith B-spline basis function, Q^x and Q^y are vectors of B-spline control point co-ordinates and L is the number of spans. The configuration of the spline is restricted to a shape-space of vectors χ defined by

$$\begin{bmatrix} Q^x \\ Q^y \end{bmatrix} = W\chi + \begin{bmatrix} \overline{Q}^x \\ \overline{Q}^y \end{bmatrix} \quad (2)$$

where W is a shape matrix whose rank is less than $2q$. Typically the shape-space may allow affine deformation of the template shape $\overline{Q}$. The object dynamics is described by AR(2) process

$$X_k - \overline{X} = A(X_{k-1} - \overline{X}) + w_k \quad (3)$$

where $X_k = [\chi_k{}^T \quad \chi_{k-1}{}^T]^T$, whose dimension is n, $\overline{X}$ is mean shape vector, w_k is process noise with $E(w_k)=0$, $E[w_{k_1} w_{k_2} T] = \delta(k_1 - k_2) R^w$, where $\delta(\bullet)$ is Delta function. A and R^w are learned from example training sequences[6].

3.2 Measurement Equation

The measurement proceeds as follows: for each sampled point $r(s_j)(j=1, \cdots m)$ on the curve, search along the normal to the curve $r(s_j)$; generally more than one features $z_k^{(j)}$ $(k=1, \cdots, l_j)$ will be detected, due to clutter, and the one with the maximum contrast is selected as the true feature $\tilde{z}_j$. The 1D measurement density along the normal line to $r(s_j)$ can be described as

$$p_j(z \mid v=\{v(j)\})=$$
$$const \times (q_{01}+\frac{q_{11}}{\lambda}\sum_{k=1}^{l_j}\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{1}{2\sigma^2}(z_k^{(j)}-v(j))^2) \quad (4)$$

where l_j is the number of features detected along the normal, q_{01} is probability that no feature is detected, $q_{11}=1-q_{01}$, λ is Possion constant, $v(j)$ is the search scale on each side of $r(s_j)$. Because feature outputs on distinct normal lines are statistically independent, the overall measurement density becomes

$$p(Y \mid X)=\prod_{j=1}^{m} p_j(z \mid v=\{v(j)\}) \quad (5)$$

Details about the Possion likelihood can be found in [18].

Note that each detected feature $\tilde{z}_j$ is a point in the image, which could be described by a x_j and y_j coordinate. We rearrange the measurement in the form of vector as

$$Y=\begin{bmatrix}x_1 & y_1 & x_2 & y_2 & \cdots & x_m & y_m\end{bmatrix}^T \quad (6)$$

and the measurement noise can be reasonably described as

$$R^v = diag(\sigma^2 \quad \cdots \quad \sigma^2) \quad (7)$$

3.3 Tracking Algorithm

The tracking algorithm is as follows.

Tracking algorithm using Unscented Kalman filter

1. Initialisation with $\hat{X}(0 \mid 0)$, $\hat{P}(0 \mid 0)$, k=1
2. Deterministically compute $2n+1$ sample points and the weights, { $X_i(k-1 \mid k-1)$, $W_i^{(m)}$, $W_i^{(c)}$ }, $i=0, 1, \cdots, 2n$, n is the dimension of the state.

$$X_0(k-1 \mid k-1)=\hat{X}(k-1 \mid k-1)$$
$$X_i(k-1 \mid k-1)=\hat{X}(k-1 \mid k-1)+(\sqrt{(n+\lambda)\hat{P}(k-1 \mid k-1)})_i \quad , \quad i=1, 2, \cdots, n$$
$$X_i(k-1 \mid k-1)=\hat{X}(k-1 \mid k-1)-(\sqrt{(n+\lambda)\hat{P}(k-1 \mid k-1)})_{i-n} , \quad i=n+1, n+2, \cdots, 2n$$
$$W_0^{(m)}=\lambda(n+\lambda) \quad W_0^{(c)}=W_0^{(m)}+(1-\alpha^2+\beta) \quad W_i^{(m)}=W_i^{(c)}=\frac{1}{2(n+\lambda)} , \quad i=1, \cdots, 2n$$

where $\lambda=\alpha^2(n+\kappa)-n$ is a scaling parameter. The constant α determines the spread of the sample points around $\hat{X}(k-1 \mid k-1)$. The constant κ is a secondary scaling parameter that is usually set to $3-n$, and β is used to incorporate prior knowledge of the distribution of state and is usually set to 2. $(\sqrt{(n+\lambda)\hat{P}(k-1 \mid k-1)})_i$ is the ith column of the matrix square root.

3. Prediction

Propagate the samples in terms of the state equation (3)

$$X_i(k \mid k-1)=AX_i(k-1 \mid k-1)+(I-A)\overline{X} \quad i=0, \cdots, 2n$$

Compute the predicted state mean

$$\hat{X}(k \mid k-1)=\sum_{i=0}^{2n} W_i^{(m)} X_i(k \mid k-1)$$

Make observation to obtain $Y_i(k \mid k-1)$ ($i=0, \cdots, 2n$) and compute its point density

$$p_i(k \mid k-1)=p(Y \mid X_i(k \mid k-1)) ,$$

according to (5), then compute the predicted observation mean

$$\hat{Y}(k \mid k-1)=\sum_{i=0}^{2n} W_i^{(m)} Y_i(k \mid k-1)$$

Compute the predicted covariance

$$\hat{P}(k \mid k-1) = \sum_{i=0}^{2n} W_i^{(c)} [X_i(k \mid k-1) - \hat{X}(k \mid k-1)]\,[X_i(k \mid k-1) - \hat{X}(k \mid k-1)]^T + R^w$$

4. Measurement Update

Compute the auto-correlation of measurement and the cross-correlation of state and measurement as

$$\hat{P}_{Y_k Y_k} = \sum_{i=0}^{2n} W_i^{(c)} [Y_i(k \mid k-1) - \hat{Y}(k \mid k-1)]\,[Y_i(k \mid k-1) - \hat{Y}(k \mid k-1)]^T + R^v$$

$$\hat{P}_{X_k Y_k} = \sum_{i=0}^{2n} W_i^{(c)} [X_i(k \mid k-1) - \hat{X}(k \mid k-1)]\,[Y_i(k \mid k-1) - \hat{Y}(k \mid k-1)]^T$$

Determine the true measurement as $Y(k) = Y_{i^*}(k \mid k-1)$ for which $p_{i^*}(k) = \max_{i=0,\cdots,2n} p_i(k \mid k-1)$, then update the mean and covariance

$$\hat{K}(k) = \hat{P}_{X_k Y_k} \hat{P}_{Y_k Y_k}^{-1}$$

$$\hat{X}(k \mid k) = \hat{X}(k \mid k-1) + \hat{K}(k)(Y(k) - \hat{Y}(k \mid k-1))$$

$$\hat{P}(k \mid k) = \hat{P}(k \mid k-1) + \hat{K}(k) P_{Y_k Y_k} \hat{K}(k)^T$$

5. $k = k+1$. Go to step 2.

The tracking algorithm proceeds as follows. First, a small set of sample points are deterministically computed according to the mean and the covariance of the states. These sample points accurately capture the distribution of the state of the system. Then the sample points are propagated in terms of the motion model (3) and the observations are made with regard to each sample. During the process the linearization of nonlinear equations is unnecessary . In accordance with the measurement *p.d.f,* the true measurement is determined as the one with the largest point density. At last the mean and covariance of the states are updated using the predictions and measurements of these points.

4. Experiments

The “pointing” gesture has been broadly used in human-machine interaction. For example, Cipolla et al visually tracked pointing gesture, then instruct the robot to grasp the object pointed by the hand[7]. In computer vision, pointing gesture is also used in virtual Touch-screen (VT) applications or used as virtual mouse to manipulate computer[3]. In our experiments, to demonstrate the performance of the new algorithm, we collect a test image sequence involving 370 fields(field interval is 20ms), in which a moving hand with commonly used pointing gesture moves fast and randomly in front of a computer. In the images, the boundary of the computer and the contents on the screen constitute the clutter. A training sequence is also recorded, in which the background is clutter-free and the motion of the moving hand is slow. A fine-tuned default Kalman filter is used to track the hand in the training sequence, from which the motion model is learned [6]. The shape space is 2D affine $(n = 12)$ and the template is hand-drawn in the first field. The algorithm is implemented using Visual C++ on a computer with Pentium II 350MHZ CPU.

We compare our algorithm with the typical algorithm based on Kalman filter, as described in [6]. Some typical results using the two algorithms are presented in Figure 1 and Figure 2 respectively. In Figure 2, the Unscented Kalman filter can successfully track the moving hand throughout the sequence, notwithstanding the clutter in the images. In Figure 2, the tracking results as well as the deterministic samples are drawn, as shown with the thick solid curve and thin dashed curves respectively. The samples represent multiple hypothesises concerning the object in the current field. The Kalman filter can successfully track up to the 130th field (2.6s). Then it is attracted by clutter and little by little loses lock. Driven by motion model and affected by clutter, it gradually locks on the upper left corner of the computer and never moves away. The centroid of the object in the image plane at each time-step is plotted in Figure 3, to demonstrate the behaviours of the two trackers. From this graph, we can see clearly that the tracker based on Kalman filter begins to diavate from the object in the 130th field, and from then on, loses lock gradually. The mean time consumed by each field is approaximately 6ms and 30ms for Kalman filter and Unscented Kalman filter respectively.

5. Conclusions

In the framework of visual curve tracking, we implement the Unscented Kalman filter. The algorithm employs a nonlinear measurement model, linearization of which is no more needed. The small set of appropriately chosen weighted samples can maintain multiple hypothesises in one time-step, and the best true observation is obtained according to measurement probability density function. The limitation of the algorithm is that it is essentially unimodal and can not apply to multimodal distribution. The experiment shows that the new algorithm is more robust than those based on KF or EKF.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments and suggestions towards improving the manuscript.

Reference

[1] H. Park, T. Schoepflin, and Y. Kim , "Active Contour Model with Gradient Directional Information: Directional Snake", *IEEE Trans. Circuits Syst. Video Technol.*,2001, 11(2),pp. 252-256.

[2] D. Terzopoulos, and R. Szeliski, "Tracking with Kalman Snakes", *Active Vision*, 1992,3-20.

[3] R. Cipolla, and A. Pentland, *Computer Vision for Human-machine Interaction*, Cambridge University Press, 1998.

[4] N. Deterfreund, "Robust Tracking of Position and Velocity with Kalman Snakes", *IEEE Trans. on PAMI*, 1999, 21(6),pp. 564-569.

[5] A. Blake, R. Curwen, and A. Zisserman, "A Framework for Spatio-temporal Control in the Tracking of Visual Contours", *Int. J. Computer Vision*, 1993, 11(2), pp. 127-145.

[6] A. Blake, M. Isard, and D. Reynard, "Learning to Track the Visual Motion of Contours", *J. Artificial Intelligence*, 1995, 78, pp. 101-134.

[7] R. Cipolla, and N.J. Hollinghurst, "Human-Robot Interface by Pointing with Uncalibrated Stereo Vision", *J. Image and Vision Computing*, 1996, 14(3), pp. 171-178.

[8] N. Peterfreud., "The Velocity Snake: Deformable Contour for Tracking in Spatio-Velocity Space", *Computer Vision and Image Understanding*, 1998, 73(3), pp. 346-356.

[9] M. Isard, and A. Blake, "Condensation-Conditional Density Propagation for Visual Tracking", *Int. J. Computer Vision*, 1998, 29(1), pp. 5-28.

[10] R. Merwe, A. Doucet, N. Freitas, and E. Wan, "The Unscented Particle Filter", *Technical Report CUED/F-INFENG/TR380*, Cambridge University, Engineering Department, August 2000.

[11] M. Isard , and A. Blake, "ICondensation: Unifying Low-level and High–level Tracking in a Stochastic Framework", *Proc. 5th European Conf. Computer Vision*, 1998, pp.893-908.

[12] L. Peihua, and Z. Tianwen, "Visual Contour Tracking Based on Sequential Importance Sampling/Resampling Algorithm", *Proc. 16th Int. Conf. Pattern Recognition*, Quebec, Canada, 2002, in Press.

[13] J. MacCormick, and A. Blake, "Partitioned Sampling, Articulated Objects and Interface-Quality Hand Tracking", *Proc. 7th European Conf. Computer Vision*, 2000.

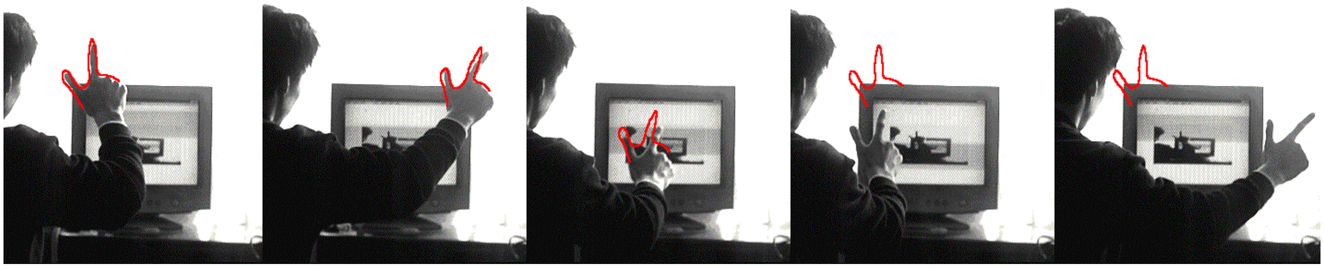
[14] J. Sullivan, A. Blake A, M. Isard, and J. MacCormick, "Object Localization by Bayesian Correlation", *Proc. 7th Int. Conf. on Computer Vision*, 1999, 2, pp. 1068-1075.

[15] S.J. Julier, and J.K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems", *Proc. Aerosense: The 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls*, 1997, pp. 182-193.

[16] S.J. Julier, and J.K. Uhlmann, H.F Durrant-Whyte, "A New Approach for The Nonlinear Transformation of Means and Covariances", *IEEE Trans. on Automatic Control*, 1996.

[17] R. Merse, and E. Wan, "The Unscented Kalman Filter for Nonlinear Estimation", *Proc. of IEEE Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control (AS-SPCC)*, IEEE, Lake Louise, Alberta, Canada, Oct, 2000

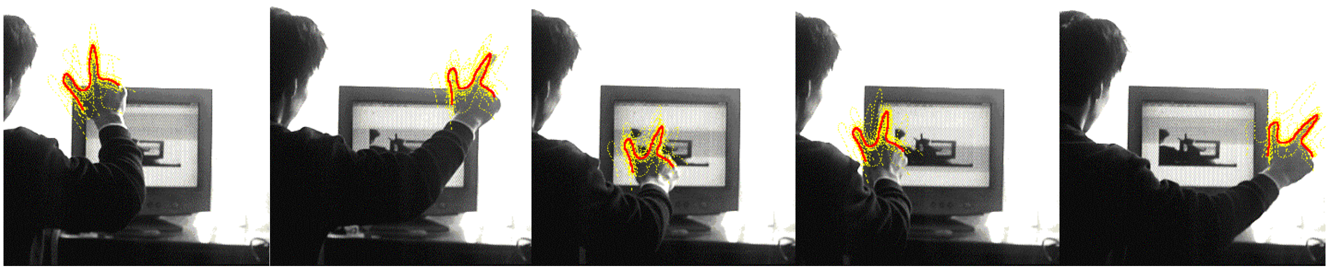
[18] J. MacCormick, and A. Blake, "Spatial Dependence in the Observation of Visual Contours", *Proc. 5th European Conf. Computer Vision*, 1998, pp. 765-781.



0.2s 1.3s 2.6s 5.5s 7.1s

Figure 1. Tracking results with Kalman filter

Affected by clutter in images, the tracker begins to deviate from the object in the 130th field(2.6s), and gradually loses lock. It evolves randomly until locks on the upper left corner of the computer and never moves away.



0.2s 1.3s 2.6s 5.5s 7.1s

Figure 2. Tracking results with Unscented Kalman filter

The Unscented Kalman filter can successfully track the object throughout the video-stream. The thick solid curve represents the object and the thin dashed curves denote the deterministic samples, which represent multiple hypothesises concerning the target in the current field.

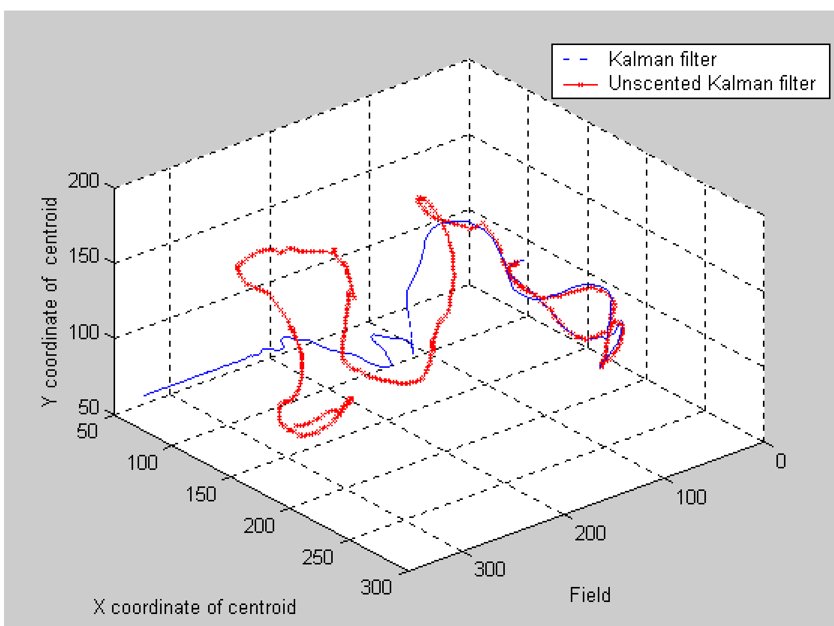


Figure 3. The centroid of contour in each field

The tracking behaviour of Kalman filter can be clearly seen that gradually diverges and loses lock.