Image-Centroid Tracking using Filtering Algorithms-A brief Review with Some Results

Parimala, P.1, J. R. Raol2
1Asst. Prof., Dept. of Telecommunication Engg, MSRIT, and Research Scholar, Jain University, Bangalore.
2Prof. Emeritus, MSRIT, Bangalore.

ABSTRACT
The image-based detection and tracking of moving objects are extremely important for many aerospace and aviation applications, including satellite-based imagery. In this paper, we briefly review various approaches related to image-centroid and/or target tracking. We emphasize square root algorithms that are supposed to be more efficient than the basic implementation of KF. Then, we present some comparative results of image-centroid tracking based on Kalman filtering and U-D factorization square root type algorithm implemented in MATLAB.

KEYWORDS: Target/image-centroid tracking, centroid features, Kalman filtering, square root type algorithms, U-D filter.

I. INTRODUCTION
There is always considerable interest in many military and civilian applications to deploy an automated system for video-based observation/surveillance, and the target-image tracking is one of the very important aspects of locating moving objects in real-time using some online and appropriate filtering algorithm. Such an algorithm utilizes each image-frame that arrives in a sequence, and outputs the location of moving target/s, and it has two basic steps: a) detection of moving objects in each frame, and b) filtering/tracking of thus detected object/s in each consecutive frame. There are several algorithms that could be utilized for target/image tracking, but, we concentrate on square root type algorithms for the reasons that the latter are computationally more efficient, accurate and stable compared to the conventional filtering algorithms, and especially these aspects are very important for online/real-time applications.

In several such situations, the acquired image would be often cluttered, dim, spurious and/or noisy, and at times, this might be due to the fact that the distance to the target from the sensing sites/centres is very relatively great. The target-image tracking problem involves processing of measurements from (deployed sensors/radars for) a target of interest and producing at each time step, an estimate of the target’s current position, velocity, and even acceleration states. The uncertainties that are mostly present, are usually modelled as additive random noise in the measured values; as well as the corresponding uncertainties in the target states. Also, there would be additional uncertainty regarding the origin of the received data, which may or may not include actual measurements from the targets; and in most cases these might be due to some random clutter (e.g. false alarms); this latter aspect leads to the need for data association, i.e. the question: which measurement originated from which target? As a result, detection and tracking of moving object is a reasonably difficult problem in forward-looking infrared (FLIR) image sequences, and more because of: i) low signal-to-noise (SNR) ratio, ii) low (intensity) contrast, iii) presence of background clutter/false alarms, and/or iv) a partial occlusion of target; hence, we need efficient, accurate, and numerically stable filtering algorithms for image-centroid tracking.

The literature on the target/image tracking approaches and methods is relatively rich [1]-[11]: a) correlation trackers can be used with structured targets, even in highly cluttered background conditions, b) automatic detection and tracking of targets from a sequence of images can be carried out for defence applications using cubic spline model and KF, c) image centroid tracking can be done using least square (LS) linear regression method for weld pool application, d) decimation method can be used for tracking large targets in real time with sizes in excess of some defined one (31x31 pixel), e) availability of square root algorithms to reduce the storage and computation involved with estimation of certain classes of large scale interconnected systems, f) for cooperative tracking approach, the use of a square root sigma point information filter (SRSPIF), h) tracking of uninhabited aerial vehicles (UAVs) with (vision-) camera based sensors, i) ANFIS (adaptive neuro fuzzy inference system) based KF for object tracking purposes, and j) many image-template matching applications. However, for such purpose efficient and numerically stable centroid tracking algorithms are highly preferred.
In the present paper, we briefly review some of the preceding/foregoing approaches/aspects and specifically study the ones based on square-root filtering algorithm for target-image centroid tracking. Although, several studies on UDF (square root type factorization filtering) algorithm for state estimation and target tracking have been done, however, so far there has been no concrete comparative study for the problem of image centroid tracking, beyond [12]. Hence, we present the performance results of image-centroid tracking using KF, and UDF (U-D factorization filter) algorithms implemented in MATLAB.

II. CENTROID FEATURES

For an object/image that is wide, and deep, and spread across an area or several pixels, the object is not just a point mass; and hence, one must assign a correct meaning to the term coordinates or position of the considered image [13]. For this, to determine the coordinate of the object/image, the center of area (COA) is chosen as the representative position; and the COA is estimated by the center of mass (COM) or the so called centroid of the object. For example, determining the position of a star by just taking the position of its maximum intensity would, at best, give the precision to one pixel of the measurement; and hence, it is highly preferable to determine the COM or the centroid; and hence, the concept of estimation/filtering in image processing relates to the evaluation of image parameters like the centroid. This is considered to be relevant to the characterization of the objects in the image; and the image analysis would involve measurements of certain characteristics of the image: i) intensity, ii) geometric features, and iii) centroid. The geometric features are: i) Length, L of a line in a discrete image is the distance between the centers of the pixels \( L = d - 1 \), here, d is the number of pixels the line covers, for if an object occupies one pixel, its length is zero; L=1-1=0; ii) Perimeter, P is equal to the sum of the side lengths; iii) Area, A is equal to the sum of all the pixels covered by the object, i.e. area of an object in a digital image is the number of points in the object, thus one can compute the area of the object by \( A = \text{total number of pixels} \). If an object is larger than one pixel, better the area measurement, thus, for better centroid estimation the image should be spread over 2 or more pixels.

The two basic methods for determining the centroid of a star object are: i) the moment analysis, and ii) profile fitting or point spread function (PSF) fitting. In the first approach, when a set of values has a tendency to cluster around some particular value, then it would be useful to characterize this set by a few numbers that are related to its moments, i.e., the sums of integer powers of the values themselves; that is if an object in an image is defined by the function \( I(x,y) \), then the moments generated by this function give interesting features of the object; and for digital images the \( (n+m) \) th order is defined as

\[
I_{nm} = \sum_{x} \sum_{y} x^n y^m I(x, y) \tag{1}
\]

The moments’ values would depend on the intensity or grey level; and image moments include center of mass, variance and orientation. For, n=m=0, we get the \( I(.,.) \) as the total intensity of the image as can be seen from (1). If one considers the intensity of grey level \( I(x,y) \) at each point \( (x,y) \) of the given image, \( I \), as the mass \( \mu \) of the object, then one can define the centroid, the COM and other moments of \( I \). In the 2-D case, the COM is given as \( (I_{00}, I_{00}) \); the normalized values of which are given as

\[
I_{00} = \sum_{x} \sum_{y} I(x, y) / I_{00}; \quad I_{00} = \sum_{x} \sum_{y} y I(x, y) / I_{00} \tag{2}
\]

The variance is given as

\[
\sigma_x^2 = I_{20} - I_{00}^2; \quad \sigma_y^2 = I_{02} - I_{00}^2 \tag{3}
\]

The variances in (3) characterize the spread or extension of the object-image in x- and y-directions. The orientation is defined as the angle of axis of the least moment of inertia (MOI)

\[
\tan(2\theta) = \frac{2I_{11}}{I_{20} - I_{02}}; \quad \text{for } I_{11} \neq 0; \text{ and } I_{20} \neq I_{02} \tag{4}
\]

The centroid of a cluster (in the normalized way) can also be determined using non-convolution method as

\[
(x_c, y_c) = \frac{1}{\sum_{i} \sum_{j} I(i,j)} \left\{ \sum_{i} \sum_{j} I(i,j) \right\} \tag{5}
\]

In (5), \( I_{ij} \) is the intensity of the pixel and \( n, m \) are the dimensions of the cluster. One can use the ‘regionprops’ in MATLAB.
III. TRACKING OF TARGETS AND IMAGE-CENTROIDS

Several modern tracking systems tend to integrate the signal processing unit/s for sensor signals and the data processing unit/s for target tracking; requiring real time signal/data processing capability; in turn these systems need target-image-centroid tracking algorithms with lower computational cost in filtering and efficient data association schemes. We in the sequel, consider a brief review of such schemes/algorithms.

An image tracking algorithm [1] combines template matching and PSNF-m (probabilistic strongest neighbour filter for m-validated measurements) to estimate the states of a tracked target; the template matching method uses correlation as a similarity index in tracking the target in the given sequence of images; this method is similar to strongest neighbour (data association) filter (SNF) that regards the measurements with the highest signal intensity as the target originated measurements (compared to other observations). The SNF assumes that the strongest neighbour in the validation gate (VG) originates from the target of interest and SNF utilizes the SN in the update step of a standard KF. The similarity index used is defined as [1]

\[
S(T,I) = \frac{\sigma_T + C}{\sigma_T + C}
\]

In (6), \(\mu_T\) and \(\sigma_T\) correspond to the mean and standard deviation of template intensity respectively, and \(\mu_T\) and \(\sigma_T\) correspond to that of for the overlapped area in the input frame respectively; \(C\) is a constant included to avoid instability, when the denominator is close to zero; \(S(T,I)\) is a correlation value. \(-1 \leq S(T,I) \leq 1\); as the \(S(T,I)\) value goes closer to 1, the overlapped area in the input frame is very similar to the template; and hence, the position with the maximum value of \(S(T,I)\) is regarded as the target position. In image tracking using template matching scheme, constant false alarm rate (CFAR) processors are used for detecting targets in background for which all parameters in the statistical distribution are not known and may not be stationary [1]. The threshold in a CFAR detector is set on a cell-by-cell basis using estimated noise power by processing a group of reference cells that surround the cell under observation. The greatest of constant average (GOCA) CFAR is applied as it reduces the number of measurements until it is under the value \(m\). The algorithm is used starting with 6400 measurements until it reduces to 13 or less; PSNF accounts for the probability of the feasible events that is correlated with the data association method using the SN measurement/s. The idea of PSNF-m algorithm is to improve the tracking performance by regarding the cpdf (conditional probability density functions) of SN measurement as a function of the \(m\)-validated measurements in the VG. The PSNF-m filter is initiated by: i) the template matching that is performed to the entire search space of the image by using the given template; then as a result of matching, VG is allocated, whose centre is equal to the area’s centre that has the largest correlation value; ii) in the next frame, matching is performed only to the inside of VG and select the SN measurement; then the filtering of the state variables and parameters is initiated using two measurements from the first and second frame and the track score \(\mu_k\) is computed, iii) the track score \(\mu_k\) for every sequential input frame is computed, and if \(\mu_k\) remains smaller than \(\mu_x\) before the fixed frame, it’s track is terminated and a new target track is generated.

Several other methods of image tracking are compared in [2]. Correlation trackers can be used with structured targets, even in highly cluttered background conditions; and the most prevalent algorithm is a centroid tracker that determines a target aim point by computing the intensity or geometric centroid of the object based on a segmentation method: it could be either intensity or binary centroid tracker. In segmentation, the image in the target window is partitioned into two regions: i) target, and ii) background. In the optimal threshold method, the pdfs of the target and the background are assumed known. Assuming the pdfs are Gaussian, image pixels having grey values below \(T_n\) are considered as background and those above the \(T_n\) are taken as target points. The optimal threshold is given by

\[
T_n = \frac{\mu_1 + \mu_2}{2} + \sigma^2 \frac{\ln \left( \frac{P_1}{P_2} \right)}{\mu_1 + \mu_2}
\]

In (7), \(\mu_1\) and \(\mu_2\) are the mean values of the target and background respectively; \(\sigma\) is the standard deviation of each class, and \(P_1, P_2\) are a priori probabilities of the two classes. In the statistical segmentation algorithm,
intensity histograms of the background and the target are used to identify the target; as usual histograms are used to estimate a pdf. The classification rule is given as

\[ B(l)_k = \begin{cases} 1 & \text{if } \text{Hist}_l(l)_k > \gamma \text{Hist}_b(l)_k \\ 0 & \text{otherwise} \end{cases} \]  

\[ \text{Hist}_l(l)_k = (1 - \alpha_1) \text{Hist}_l(l)_{k-2} + \alpha_1 \left[ \frac{N_l(l)_{k-1}}{\text{Total}_l} \right] \]  

\[ \text{Hist}_b(l)_k = (1 - \alpha_2) \text{Hist}_b(l)_{k-2} + \alpha_2 \left[ \frac{N_b(l)_{k-1}}{\text{Total}_b} \right] \]  

In (8), Hist\(_l\)(l)\(_k\) and Hist\(_b\)(l)\(_k\) are the normalised target, and background histograms at the frame number k; N(l)\(_k\) are the numbers of occurrences belonging to target, background having a specified grey level ‘l’ in k\(^{th}\) frame; Total\(_l\) and Total\(_b\) are total number of exceedance (B(l)\(_k\) = 1) and (B(l)\(_k\) = 0); and \(\alpha_1\) and \(\alpha_2\) are parameters that govern the adaptation rates of target and background histograms respectively.

In optimal layering method, the image is divided into several layers of grey level intensities, and each layer is signified by its upper and lower grey level bounds. Based on the a priori information about the grey level intensities of the target layer, using the Bayesian risk calculation, optimal target layer boundary could be obtained

\[ B(x(i), y(i)) = \begin{cases} 1 & \text{if } I(x(i), y(i)) \leq I_N \\ 0 & \text{otherwise} \end{cases} \]  

In (11), x(i), y(i) classify pixel position for the i-th pixel and I(x(i), y(i)) signifies the grey level intensity value in that pixel position. These conventional histogram-based or fixed value-thresholdings are deficient in segmenting targets due to the poor contrast between target and background or due to the change of illumination irrespective of low computation time and simplicity of implementation. Feature based clustering segmentation is another approach to extract the moving target in cluttered image sequences; for real time applications one should select proper features values that can distinctly identify the background from the target.

In [3], the image processing techniques are employed to analyse the features of weld-pool and its surroundings; the centroid of the weld-pool image is extracted as a parameter for seam tracking; and therefore, the deviations between the arc-tip and seam centreline can be estimated by this centroid. Least squares (LS) method is utilized to correlate the relationship between the centroid and the deviations under different welding conditions; and to remove the noise of weld-pool images, a median filter is applied; this would bring down contrast of weld-pool and weld-seam. An intensity transformation algorithm is applied to get an enhanced image. If, g(i,j) is the grey scale value related to the pixel (i,j), and \(x_i\) and \(y_j\) the x-axis and y-axis co-ordinates of the pixel (i,j); then the measured centroid position \((x_c, y_c)\) is given by

\[
\begin{bmatrix}
  x_c \\
  y_c
\end{bmatrix} = \begin{bmatrix}
  \sum_{j=1}^{M} \sum_{i=1}^{N} x_i g(i,j) \\
  \sum_{j=1}^{M} \sum_{i=1}^{N} y_i g(i,j)
\end{bmatrix} / \begin{bmatrix}
  \sum_{j=1}^{M} \sum_{i=1}^{N} g(i,j) \\
  \sum_{j=1}^{M} \sum_{i=1}^{N} g(i,j)
\end{bmatrix}
\]  

In (12), M and N are pixel numbers of one row and one column of the specified region of the image respectively. The deviations between the arc-tip and seam centreline can be computed by measuring the position and displacement of the centroid. The weld pool images are captured by CCD camera arranged ahead of a welding torch [3]. Image processing techniques include image acquisition (board), industrial computer, motion control board, and stepping motors are employed to analyse the features of weld pool and its surroundings. This experiment is carried out in the GTAW (Gas Tungsten arc welding) process, in which the camera captures the images of welding area in real time. The deviation between the weld seam centreline and arc tip can be
eliminated by driving the torch position according to the position of weld pool image centroid, which can be computed rapidly in real time. According to the computed centroid position and deviations between the arc tip and seam centreline, the least LS method is used to establish a simple linear regression model using 500 data pairs for each experiment under different conditions. The simple linear regression model can be expressed as

\[ f(x) = b_0 + b_1 x \]  

(13)

In (13), \( x \) is the deviation of centroid position, \( f(x) \) is the deviation between the arc tip and seam centreline; \( b_0 \) and \( b_1 \) are unknown coefficients; that are obtained by minimizing the sum of squared residuals using LS method

\[ J = \sum_{i=1}^{N} [f(x_i) - z_i]^2 \]  

(14)

In (14), \( x_i \) is the deviation of centroid position, \( f(x_i) \) is the estimated deviation between the arc tip and the seam centreline, \( z_i \) is the actual measured deviation between the arc and seam centreline. Four linear regression models under different experimental conditions are described for analysis

\[ f(x) = 0.208x + 0.392, I = 70A \]
\[ f(x) = 0.162x - 0.359, I = 73A \]
\[ f(x) = 0.126x + 0.117, I = 75A \]
\[ f(x) = 0.194x + 0.131, I = 80A \]  

(15)

In linear regression models, the value of the total sum squares (TSS) is the sum of squares of the difference among the observed variables and its arithmetic can be written as follows

\[ TSS = \sum_{i=1}^{N} (z_i - \bar{z})^2 \]  

(16)

In (16), \( z_i \) is the observed variable and the second term is the arithmetic mean of \( z_i \). The TSS can also be subdivided into two components as follows

\[ TSS = \sum_{i=1}^{N} RSS + \sum_{i=1}^{N} RESS; RSS = \sum_{i=1}^{N} (\hat{z}_i - \bar{z})^2; RESS = \sum_{i=1}^{N} (z_i - \hat{z}_i)^2 \]  

(17)

In (17), RSS is regression sum of squares which represents the effects caused by independent variable \( x_i \) and RESS is the residual error sum of square which unfolds the effects caused by random errors. The significance of the linear regression model is the indicator of the linear correlation between \( f(x) \) and \( x \). The F-test method is applied to test the significance of the deduced linear regression model

\[ F = \frac{RSS / v_{RSS}}{RESS / v_{RESS}} \]  

(18)

In (18), \( v_{RSS} \) is the degrees of freedom (DOF) of RSS corresponding to the number of independent variable and \( v_{RESS} \) is the DOF’s of RESS, \( V_{RESS}=N-1-V_{RSS} \).

The squareroot information filter (SRIF) can dramatically reduce the storage and computation involved with estimation of certain classes of large scale interconnected systems [4]; such systems appear in the context of economic and power systems, and especially in certain orbit tracking-cum-determination applications. These systems constitute n-subsystems: \( S_1, S_2, S_3, \ldots, S_n \) each having dimension \( n_x \). It is assumed that all these subsystems are linked to a reference subsystem \( S_0 \) assuming it is noise free; and the subsystems are assumed dynamically decoupled from one another, but are coupled observationally to the reference subsystem; e.g. the multi-station satellite tracking. Here, the reference subsystem could be satellite ephemeris, subsystem \( j \) representing the parameters (station location co-ordinate corrections, range, oscillator drifts, etc.) associated with station \( j \). The SRIF algorithm is a direct application of matrix partitioning to the optimal filtering algorithms. It employs an
initial information array $[\hat{R}, \hat{y}]$ related to the KF covariance and state estimate $\hat{P}, \hat{x}$; here, $R$ is the square root of the information matrix (inverse of $P$), and $y$ is the associated state vector (in information domain). The information array variables have an extremely useful interpretation as a data equation

$$\hat{P} = \hat{R}^{-1} \hat{R}^T; \hat{x} = \hat{R}^{-1} \hat{y}$$

$$\hat{y} = \hat{R} \hat{x} + \hat{v}; \hat{v} \in N(0, I)$$

(19)

In (8), the second equation is called the data equation because, we have $z=Hx+v$, as the measurement equation in the usual KF. The SRIF can be also utilized for image-centroid tracking in a similar manner as KF can be utilized.

Multiple dissimilar sensor-multi-target tracking (MSMT) and identification algorithm using SRIF (square root information filter) is proposed in [5]. The CASE-ATTI simulation test bed offers a modular, portable and flexible environment to test advanced multisensory tracking algorithms in a systematic fashion. The use of information filters in the MSMT environment substantially improves the performance of track-to-track fusion (TTF) process. A cumulative approach to determining track to track association provides a more reliable means of determining which tracks should be fused. Thus, a coordinated approach to the usage of Kinematic and non-kinematic information, is the potential to improve the surveillance knowledge/picture provided by the tracking function. TTF association is almost exclusively performed in a simple pair-wise manner by constructing a threshold on the statistical distance measure between two sensor-tracks. This is accomplished in two steps: i) gating the tracks with tracks across the sensors, and ii) using compatible constraints to construct exclusive and all the admissible partitions. In kinematic gating, the SRIF with a modified Householder type sequential approach is employed to propagate an inverse covariance matrix factor in an efficient manner. The TTF is performed in a linear LS manner (also known as static fusion)

$$\hat{x}_G(k+1/k) = [\sum_{i=1}^{m} P_i^{-1}(k+1/k)]^{-1} \sum_{i=1}^{m} [P_i^{-1}(k+1/k) \hat{x}_i(k+1/k)]$$

(20)

Here, $\hat{x}_G$ is the global estimate formed by linear combination of $m$-sensor level estimates and it requires that the estimation errors of the local tracks be orthogonal, and $P(.)$ the state-error covariance matrix. In the information domain, static fusion of the sensor level pseudo estimates can also be easily carried out by utilizing the relations in (19). At the individual sensor level after feedback, the global tracks replace local ones that contributed to the formation of that particular global track; this feedback has the added merit of improving the quality (of) tracks that are used locally for measurement to track association.

Co-operative tracking for uninhabited/unmanned aerial vehicles (UAVs) with camera based sensors is developed and verified with the flight data in [6]; the approach utilizes a square root sigma point information filter (SRSPIF) which takes (improved): i) numerical accuracy by using square roots, ii) tracking accuracy by using sigma points, and iii) fusion ability of the information filter, see (19). The essential ingredients of automatic target recognition (ATR) process [7] involve: i) target acquisition, ii) identification, and iii) tracking by processing a sequence of images; the two aspects involved here are: a) location and identity estimation (LIE) of the target by fusing infrared and acoustic data, and b) tracking of a centroid for target state estimation using IR sensor data. The IR sensors detect all the targets in its field of view (FOV) and generate images. This provides the information of both location and identity of the target. Acoustic sensor provides data about the direction of the target; these sensor outputs/data cannot be fused directly if the features captured are different. Tracking is carried by centroid tracking algorithm (CTA), which in the presence of clutter is achieved using nearest neighbour Kalman filter (NNKF) and probability data association filter PDAF. In general, the gating and data association enable tracking in multi-sensor multi-target (MSMT) scenario. The NNKF or PDAF is necessary for centroid tracking application because in the neighbourhood of the predicted location of the target there could be several centroids found due to splitting of the target clusters or due to noise clusters. The constant velocity kinematic model is used for generating the data which determines the position of the target in each scan.

A system to detect and track moving objects from an airborne platform [8] given a global map, e.g. a satellite image, the target in geo-coordinates (longitude and latitude) is obtained from geo-registration. Here, a motion model in geo-coordinates is considered physically more meaningful than that in image coordinates. A two-step geo-registration approach to stitch images acquired by satellite and UAV cameras has been followed, and mutual information is used to find correspondences between these two different types of modalities. After motion segmentation and geo-registration, the tracking is carried out in a hierarchical manner: i) at the
temporally local level, moving image blobs extracted by motion segmentation are associated into track-lets, i.e. sub-tracks, and ii) at the global level, track-lets are linked by their appearance and the spatio-temporal consistency on the global map. To achieve efficient time performance, graphics processing unit (GPU) techniques are applied in the geo-registration and motion detection modules, which seem a bottleneck of the entire system; however, the experiments have shown that the method can efficiently deal with long term occlusion and segmented tracks even when targets fall out of the field of view [8]. Automatic detection and tracking of intended targets from a sequence of images (for defence related applications) is addressed in [9]. Image registration algorithm suitable for real time tracking application, cubic spline model and a KF model for prediction have been proposed. A real time image tracking algorithm having many computer vision applications such as video surveillance, human-machine interaction systems and object based video compression is discussed in [10]. It gives small weights to pixels farther from the object centre and uses the quantized image grey scales as a template. It derives the target location by mean shift method. The algorithm sets up the binary image model and arrives at the target’s scale using image feature recognition. A decimation method is proposed to track large targets in real time whose sizes is in excess of some defined size (31x31 pixel). Spatial localization includes histogram based target model, the mean shift iterative process and the decimation method. In histogram-based target model, a target is represented by an ellipsoidal region, whose centre is the central position of the target \((x_c, y_c)\). All the pixels in the ellipsoidal region \(X_i\) \((i=1,2,\ldots n)\) are normalised to a unit circle

\[
X_i^* = \left(\frac{x_i}{L_x}, \frac{y_i}{L_y}\right)
\]

(21)

In (21), \(L_x\) and \(L_y\) are the horizontal axis and the vertical axis of the ellipse, respectively. While tracking the target, the peripheral pixels are the least reliable, being often affected by occlusions or interference from the background. In order to increase the robustness, the pixels \(X_i\) are assigned with different weights, according to the distance between the pixels and the object centre:

\[
d_i = \sqrt{\left(\frac{x_i - x_c}{L_x}\right)^2 + \left(\frac{y_i - y_c}{L_y}\right)^2}
\]

(22)

The weighted grey scale histograms of the target are computed as

\[
q_u = \frac{1}{C_q} \sum_{i=1}^{n} w_u(\chi_i) \delta[b(\chi_i) - u]
\]

(23)

Here, \(C_q = \sum_{i=1}^{n} w_u(\chi_i)\), \(\sum_{u=1}^{m} q_u = 1\); \(n\) is the number of the pixels located in the ellipsoidal region, and \(\delta\) is the Kronecker delta function.

The performance of multi sensor system can be improved by a hybrid fusion algorithm based on adaptive square root Cubature KF (SRCKF) [11]. This algorithm was applied to the vessel dynamic positioning system simulation; and it introduces filter gain correction for the case of measurement malfunctions, and proposes a switching criterion between an optimal filter selected from adaptive and conventional SRCKF, based on the measurement quality; and a single scale factor would be required. The system state and error covariance matrices are predicted with the a priori fusion and both are integrated with the predicted and estimated states and covariance matrices of subsystems to update the fusion estimation. A triple redundancy position measurement system of vessel dynamic positioning is regarded as the research objective to evaluate the performance of proposed algorithm.
We now discuss two important centroid tracking filtering algorithms (CTA). As we know, the determination of a moving object’s position and velocity from a noisy time series of images captured by image sensors constitutes a statistical estimation problem, which leads to a CTA; for which conventionally covariance based KF is often used. A suitable state space model for centroid representation is given by

\[
x(k+1) = \phi x(k) + Gw(k) \tag{24}
\]

\[
z(k+1) = Hx(k) + v(k) \tag{25}
\]

In (24), \(x\) is a state vector that contains the image-centroid coordinates of a target, \(z\) is the vector of observables, and \(w(.)\), and \(v(.)\) are process and measurement noises with zero means and covariance matrices \(Q\), and \(R\) respectively.

4.1 The discrete KF

The filtering algorithm is given as

**State propagation**

- State estimate \(\hat{x}(k+1) = \phi \hat{x}(k)\) \(\tag{26}\)
- Covariance (a priori) \(\hat{P}(k+1) = \phi \hat{P}(k)\phi^T + GQG^T\) \(\tag{27}\)

**Measurement Update**

- Residuals/innovations \(e(k+1) = z(k+1) - H \hat{x}(k+1)\) \(\tag{28}\)
- Kalman Gain \(K = \hat{P}H^T (H\hat{P}H^T + R)^{-1}\) \(\tag{29}\)
- Filtered estimate \(\hat{x}(k+1) = \hat{x}(k+1) + Ke(k+1)\) \(\tag{30}\)
- Covariance (a posteriori) \(\hat{P} = (I - KH)\hat{P}\) \(\tag{31}\)

4.2 UD factorization filter

At times, implementation of KF on a finite word length computing machine could pose a problem; and the effects are greatly reduced by implementing it in a factorized form; such factorization implicitly preserves the symmetry and ensures the non-negativity of the covariance matrix \(P\)\(^{[14]}\). One such widely used from is the UD factorization filtering algorithm; here, \(U\) and \(D\), are matrix factors of the covariance matrix \(P\) of the KF, where \(U\) is a unit upper triangular matrix (with 1’s on diagonal elements) and \(D\) is a diagonal matrix. The major advantage from UDF comes from the fact that the square-root type algorithm processes square roots of the covariance matrices and hence, they essentially use half the word length normally required by the conventional KFs; in the UDF, the covariance update formulae of the conventional KF and the estimation recursion are reformulated. Specifically, we use recursions for \(U\) and \(D\) factors of covariance matrix \(P = UDUT\). The U-D filtering algorithm is given in two parts like the KF.

**Time Propagation**

We have for the covariance matrix propagation from KF recursion, (27)

\[
\hat{P}(k+1|k) = \phi \hat{P}(k)\phi^T + GQG^T \tag{32}
\]

Given \(\hat{P} = \hat{UD}\hat{U}^T\) (a priori factors are assumed known), and \(Q\) as the process noise covariance matrix, the
time propagated factors $\tilde{U}$, $\tilde{D}$ are obtained by utilizing the modified Gram-Schmidt orthogonalization process: we define $V = [\varphi \tilde{U} | \varphi \tilde{G}]$ and $\tilde{D} = \text{diag}(\tilde{D}, Q)$. and $V^T = [v_1, v_2, \ldots, v_n]$. then $P$ is reformulated as $\tilde{P} = \tilde{V} \tilde{D} \tilde{V}^T$. The $U$ and $D$ factors of $\tilde{V} \tilde{D} \tilde{V}^T$ are computed by the following recursions: For $j = 1, \ldots, n$; we evaluate [14]

$$
\tilde{D}_j = \langle v_j, v_j \rangle_D
$$

(33)

$$
\tilde{U}_{ij} = (\tilde{U} \tilde{D}_j) \langle v_i, v_j \rangle_D, \quad i = 1, \ldots, j-1
$$

(34)

$$
v_i = v_i - \tilde{U}_{ij} v_j
$$

(35)

In (33), $\langle v_i, v_j \rangle_D = v_i^T \tilde{D} v_j$ is the weighted inner product between $v_i$ and $v_j$; The time propagation algorithm directly and efficiently produces the required $U, D$ factors, taking the effect of previous $U, D$ factors and the process noise, and also preserves the symmetry of the $P$ matrix.

**Measurements data Update**

The measurement-data update process in KF combines a priori estimates $\tilde{x}$ and error covariance $\tilde{P}$ (obtained from the time evolution/propagation) with a scalar observation $z = c x + v$ to construct an updated estimate and covariance as

$$
K = \tilde{P} c^T / s
$$

(36)

$$
\hat{x} = \tilde{x} + K (z - c \tilde{x})
$$

(37)

$$
s = c \hat{P} c^T + r
$$

(38)

$$
\hat{P} = \tilde{P} - K c \tilde{P}
$$

(39)

In (36), $\tilde{P} = \tilde{U} \tilde{D} \tilde{U}^T$; $c$ is the (scalar) measurement matrix, $r$ is the measurement noise variance, and $z$ is the vector of noisy measurements. Kalman gain $K$, and updated covariance factors $\hat{U}$ and $\hat{D}$ can be obtained from the following equations [14]

$$
g = \tilde{U}^T c^T; \quad g^T = (g_1, \ldots, g_n); \quad w = \tilde{D} g
$$

(40)

$$
\hat{d}_1 = \tilde{d}_1 R / s_1; \quad s_1 = R + w_1 g_1
$$

(41)

For $j = 2, \ldots, n$; evaluate/compute the following

$$
s_j = s_{j-1} + w_j g_j \quad \hat{d}_j = d_{j} s_{j-1} / s_j
$$

(42)
\[
\hat{u}_j = \hat{u}_j + \lambda_j K_j; \quad \lambda_j = -g_j / s_{j-1} \quad (43)
\]

\[
K_{j+1} = K_j + w_j \hat{u}_j; \quad \hat{U} = [\hat{u}_1, \ldots, \hat{u}_n] \quad (44)
\]

Then, the Kalman gain is given by

\[
K = K_{n+1} / s_n \quad (45)
\]

In (42), \( \hat{d} \) is the predicted diagonal element, and \( \hat{d}_j \) is the updated diagonal element of the \( D \) matrix. The time propagation and measurement update for the state vector \( x \) are just similar to KF.

V. RESULT OF EVALUATION

A set of image-frames that is used in centroid tracking is generated synthetically; 50 frames of the images are generated which represent target environment and is given as input for further processing. For centroid computation (5) is used. In the present scenario we have: the image is of dimension 64 x 64; the target size is fixed with a dimension of 9 x 9; the image consists of object and its surrounding along with noise that is uniformly distributed; the image would have intensity in the range 0 to 255; the target intensity value and its background have a certain mean and variance. A 2-D array of pixels is considered where each pixel is represented by a single index \( i=1, \ldots, m \) and the intensity of pixel is given by \( I_i = s_i + n_i \); wherein, \( s_i \) is the target/background intensity and \( n_i \) is the noise intensity in pixel ‘\( i \)’, this noise is assumed to be Gaussian with zero mean and covariance \( \sigma^2 \). The input parameters for the tracking algorithms are

i) Measurement model/matrix: \( H = [1 \ 0 \ 0 \ 0; 0 \ 1 \ 0 \ 0] \)

ii) State transition matrix (eqn. (24); ‘phi’: \( [1 \ T \ 0 \ 0; 0 \ 0 \ 1 \ T; 0 \ 0 \ 0 \ 1] \)

iii) Measurement noise variance: \( R=0.5 \)

iv) Process noise coefficient matrix: \( G = [T^2/2 \ 0; \ T \ 0; 0 \ T^2/2; 0 \ T] \)

v) Process noise variance: \( Q = 0.00001 \)

Other image related parameters are

(i) Target image mean and STD: \( (100, 10) \)

(ii) Background mean and STD: \( (50, 50) \)

(iii) Track scan (sampling interval/period, T): 1 sec.

(iv) The initial states \( x(0), y(0)=(10,10) \), with constant initial velocity of 1 m/s in both the coordinates.

The results are generated by using CTKF, and CTUDF algorithms implemented in MATLAB. Table 1 gives the performance metrics for different target image noise STDs (STDTGN) for both the filters; it is seen from Table I and Figure 1, that there is not much trend of the performance metrics wrt the STDS, however, it is clearly seen, from the forth row of the Table I, and Figure 1, that the CTUDF performs better than CTKF in this centroid tracking task. Figure 2 shows the time histories of the state-errors with their theoretical bounds for CTKF and CTUDF algorithms for STDGN=3. From the presented performance metrics and plots we can see that the centroid tracking algorithms perform very well.

<table>
<thead>
<tr>
<th>Target noise standard deviation=1</th>
<th>Target noise standard deviation=3</th>
<th>Target noise standard deviation=5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PFEX</td>
<td>PFEY</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td><strong>KF</strong></td>
<td>0.5893</td>
<td>0.5467</td>
</tr>
<tr>
<td><strong>UDF</strong></td>
<td>0.2639</td>
<td>0.3715</td>
</tr>
</tbody>
</table>
FIGURE 1 PERFORMANCE METRIC TRENDS

FIGURE 2 STATE ERRORS FOR CTKF (LEFT) AND CTUDF (RIGHT) WITH THEIR THEORETICAL BOUNDS.

CONCLUDING REMARKS

We have considered the image-centroid tracking and some related literature for brief review of the tracking approaches/algorithms. We also specifically considered Kalman filter, and U-D factorization filtering as CTAs, and evaluated their performances with a synthetic image generated using MATALB. Based on the performance metrics and plots, it is found that the CTUDF performs better than the CTKF, and it is expected that the benefits of using UDF for image-centroid tracking would be still more if the CTUDF is implemented using higher order models, and used in multi-sensor multi-image (MSMI) tracking and fusion scenarios.

REFERENCES


