Single View Based Motion Analysis in High Speed Imaging Applications: A Theoretical Framework

B.V.Sreekanth[1] Rohin Koduri[2] Dr. G.V. Maha Lakshmi[3]
[1]P.G. Student [2]Scientist [3]Professor
[1,3]Department of Electronics and Communications Engineering
[1,3]Sreenidhi Institute of Science and Technology, Hyderabad, India.
[2]Defence Research & Development Laboratory, Ministry of Defence, Hyderabad, India

Abstract— A simple methodology for estimating the trajectory and motion parameters of a high speed projectile is presented. The target being tracked is considered to have four markers pasted at known spacing. From a single viewed image, the basic principles of Perspective Projective Transform are used to obtain the real time 3D location of the target. An extremely efficient tool, the Kalman Filter is then applied on the obtained 3D positions. Based on the specified filter parameters, the tuned Kalman Filter corrects for errors in the measurements of target location. Thereafter, the Kalman Filter computes an estimate of the future trajectory of the target. For this purpose, two decoupled Kalman Filters are employed. Along with the theoretical aspects, a practical illustration is provided by taking sample information of single images from a high speed video.

Key words: Estimation, Kalman filter, motion analysis, object tracking, perspective projective transform, single view.

I. INTRODUCTION

Reliably investigating the images for various features of interest is an ongoing research in Computer Vision domain. The proliferation of high powered computers, the availability of superior quality and inexpensive video cameras, and the increasing need for automated video analysis has developed a great interest in Object Tracking algorithms. Currently, the high speed cameras come with a default user interface for video analysis [16].

In general, Motion tracking of High Speed projectiles is precisely carried out by heavy and expensive on ground radar equipment [1]. Quite often, High Speed video cameras are used for archival of the launch. In military applications, the video is used to observe for any visible malfunctioning of missiles during different launch phases. However, under certain circumstances, the video footage can provide more quantitative information about the motion of the object in the scene. When shot from a properly calibrated camera, the High Speed video can reasonably provide position and velocity measurements of the projectile. In exploring this, a theoretical methodology is provided for acquiring the real time measurements from the video. The simplest case of a Non-maneuvering, close range projectile is considered for illustration of the procedure. The simulation is performed in MATLABTM software.

The basic geometric principles of image formation and Photogrammetry may be revised from [7].There are many algorithms in Image processing literature developed for studying the motion of the object in the video sequence. One example is Optical Flow [2], which uses brightness variation from image to image to determine motion. Before proceeding further, an extensive study of the various techniques related to the basic concepts of Object Detection and Object Tracking is required. To aid in understanding these, a highly recommended guide is the survey paper, [3]. Based on the problem at hand, a suitable detection and tracking strategy may be applied to the object of interest.

Post detection, the 3Dimensional scene can be reconstructed either by taking two stereo images of the same scene or by making certain assumptions. The second technique of conditioning is used here by considering only a single image of the scene. The problem addressed in this paper is, to estimate the location of a high speed projectile in 3Dimensional coordinates from its single viewed 2Dimensional image alone. Thereby, its future course may be predicted along with its velocity profile. A simple and general case is considered to provide a solution. The experimental setup would be a single high speed camera fixed to the wing of the parent aircraft from which a missile is launched. Its traversal is recorded and the video footage is analyzed offline for determining its motion parameters. It is important to note that the solution provided is only valid under the following conditions:

1) The high speed camera acquires close range video and is properly calibrated prior to the launch.
2) The projectile is non-manoeuvring, i.e., it obeys the basic projectile motion equations. The projectile motion is assumed to be planar.
3) The object under track has four explicitly visible markers for an initial duration in the video footage. The markers are pasted linearly on the projectile.

Given that the conditions are satisfied, the flow of the process is depicted in Fig. 1.

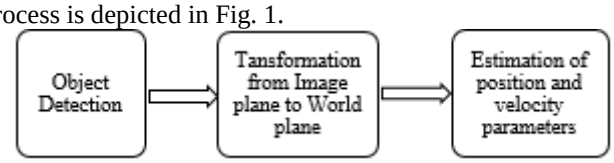


Fig. 1: Flow chart

The object detection is performed by using a trained classifier which employs Viola Jones algorithm [4] for recognition of markers on the object. After recognizing the pixel locations in the image, they are transformed into World coordinates by using the Orientation matrix [5] of the camera and solving linear equations. Finally, the filtering and prediction of known and unknown positions respectively, are carried out by tuning a decoupled linear Kalman filter [6]. Reading further, all the processes in the flowchart are debriefed as separate sections. For a single image in the video sequence, an algorithmic view of the procedure is as follows:

1) Acquire the single frame image.
2) Identify the marker locations by using the trained classifier.
3) Transform the pixel locations to Camera coordinate system.

4) Transform the Camera coordinates to World coordinate system.
5) Filter the solutions of 3D positions by using Kalman filters.
6) Predict other measurements and the future trajectory by using Kalman filters.

II. OBJECT DETECTION

The process of object detection involves the following steps:
Step 1: Define a suitable representation of the object.
Step 2: Select suitable Image features to be used as an input to the tracker.
Step 3: Choose an Object detection strategy.

In this paper, major focus is laid on the determination of real time motion parameters from the images, given the objects are detected. A template image of the feature of interest can be located in the image by using different object detection algorithms. The template matching algorithm can be as simple as convolution sum or as complex as statistical learning model. The best choice may be determined by trial and error for that particular application. From the wide range of available object detection techniques, as enumerated in [3], for the purpose of illustration, the point descriptors (centroids) and an Adaptive boosting technique (Viola Jones [4]) are used for object representation and object detection respectively.

An example Template image is shown in Fig. 2. Markers of this kind are commonly used on dummies for vehicle's crash test diagnostics. So, the marker is the object to be detected. The work to follow is based on pasting four of these markers in a line at known spacing with known dimensions. The exact centroids location of the markers in the video sequence is determined by the object detection algorithm employed. Applying [4] to determine the marker locations in MATLAB™ involves, initially training the statistical model with true (positive) images and false (negative) images. This trained model is then able to differentiate and locate the template in a given scene. It is obvious that all the four markers must be clearly visible for the detections to be made. After an initial duration, in case of lost markers in the video due to motion of the missile to out of field of view of camera or into a nonparallel motion plane, then the missile location is estimated by using a Kalman Filter as described in section 4.

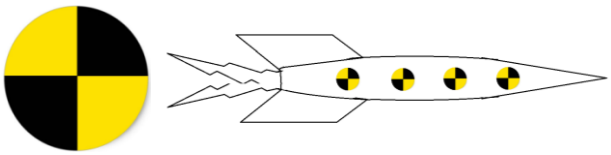


Fig. 2: Template image and its arrangement on a missile

III. TRANSFORMING PIXELS TO 3D WORLD

Three independent but relative coordinate systems must be taken into account. A right handed Cartesian coordinate system is considered as shown in Fig. 3. This section deals with the matrix transformation of pixel locations (u, v) to world coordinates (U, V, W). It explains all the intricate details from image formation to solution of perspective projections. The orientation of the camera with respect to the world coordinate system is arbitrary. However, its location C is known and fixed throughout the shoot. The image plane is parallel to the camera at a distance f (focal length), oriented according to rows and columns of MATLAB™.

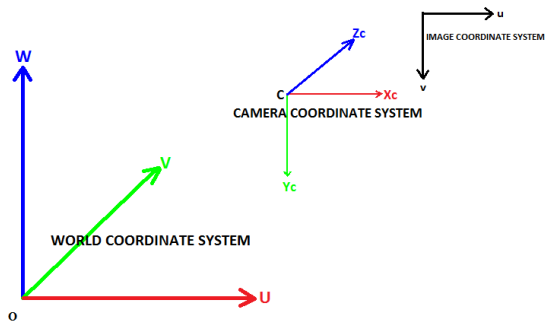


Fig. 3: Relative coordinate systems involved in the problem

A. *Perspective Projective Transform:*

The process of reconstructing 3D scene from its pixel coordinates is termed as Backward Projection. Backward projection involves three transformations as shown in Fig. 4. The first operation is an Affine transformation from pixel coordinates to sensor coordinates. Affine transformation is a linear mapping which involves operations like rotation, translation, scaling, reflection and shearing. The second and critical transformation is the Perspective projection from two dimensional sensor coordinates to three dimensional Camera coordinates. This conversion is made by intrinsic orientation matrix. The space between the sensor and the object is defined as Projective space. It obeys the rules of Projective geometry. A common mathematical model of the projective plane is conceptualized by Homogeneous coordinates. A brief introduction to projective geometry and homogeneous coordinates is provided in [8] and [9] respectively. The third transformation is a rigid rotation and translation of camera coordinated to transform to world coordinates. This transformation is known as Exterior orientation matrix.

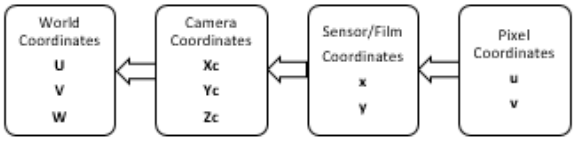


Fig. 4: Backward Projection

For the sake of simplicity the compound lens arrangement of the camera is neglected and a pinhole camera model is considered. In this model, the fundamental characteristic is that the perspective centre, image point and its corresponding object point, all lie on a line in 3D space. Considering four rays towards the perspective centre for four markers on the object, the perspective forward projected image formation is shown in Fig. 5.

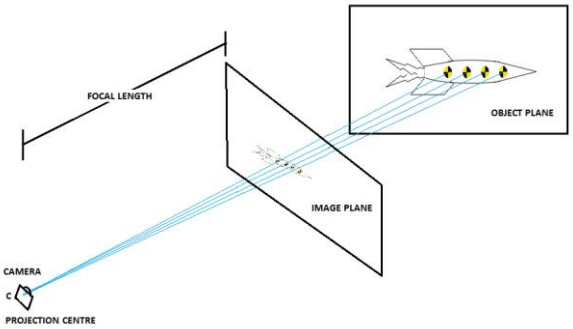


Fig. 5: Perspective projection of the object scene

B. Camera Calibration:

Calibrating a camera implies, determining the projective transformation from World coordinates to Pixel coordinates ad its inverse. Calibration of aerial cameras can be done by huge instruments like Multicollimator or Goniometer. However, for self-calibration of close range cameras there are many software techniques [7]. Among them, the simplest method is using geometric calibration. Even though they give relatively less precision, they are best suited for computer vision applications. Usually, the camera calibration matrix is a 3×4 matrix which is a product of two matrices called Intrinsic orientation matrix and Extrinsic orientation matrix, defining the characteristics of the camera relative to the sensor and the world respectively.

Perspective projections can be mathematically modelled using Homogeneous coordinates. According to [9], homogeneous coordinates are equivalence classes of the relation,

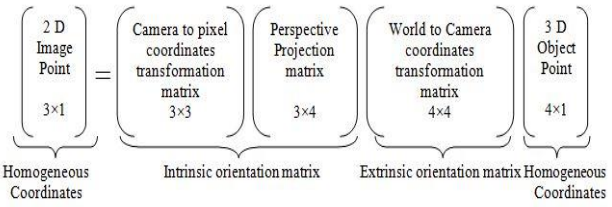
(X, Y, Z) ~ (U, V, W) if (X, Y, Z) = r (U, V, W) for $r \neq 0$.

The equivalence class of (x, y, z) is the set

[(X, Y, Z)] = {r (X, Y, Z) | r ϵ R and $r \neq 0$.

For example in Fig. 4, consider any point (u, v) in image plane. This point is the projection of the line it follows towards the corresponding object point (U, V, W). This line is in Projective plane which obeys ~ equivalence relation. Any point on this line will have the same projection (u, v) in the image plane. Hence a specific point is denoted by using scalar r. Now, suppose (a, b, c) are the homogeneous coordinates (in Projective plane) of (u, v), then, $u = \frac{a}{c}$ and $v = \frac{b}{c}$ (in Cartesian plane).

The camera calibration matrix in Projective plane is,



C. Sensor to Camera Coordinates: Interior Orientation Parameters:

In general, for video cameras with mounted CCD (Charge Coupled Device) sensors, the sensor coordinates are equal to the image coordinates, i.e., (x, y) = (u, v). Intrinsic parameters are internal to the camera and are fixed depending on the experimental setup. They transform the sensor coordinates into camera coordinates using Interior orientation matrix. These parameters include the position of the projection centre, focal length, scale factor and distortion parameters involved in the imaging process. A commonly used technique is [10]. One such procedure involves, imaging different poses of a black and white chequer board pattern at different distances from the fixed camera. The alignment in of the squares is understood by the software in the form of control points, taken as inputs along with additional information. As shown in Fig. 6, the program [11] takes a set of calibration images and manually entered control points. Here, the control points are corners of the chequer board. The additional information would be the number of squares and their dimensions.

Mathematically,

Let (a, b, r) be the homogeneous representation of (u, v) pixel locations. Hence, (u = a/ r) and (v = b/ r)

From the basic projection equations (law of similar triangles) as shown in Fig. 7,

$$x = f\frac{X}{Z} \text{ and } y = f\frac{Y}{Z},$$

where f is the focal length of the camera.

Let (x', y', z') be the homogeneous coordinates of (x, y).Then, in matrix form

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} Xc \\ Yc \\ Zc \\ 1 \end{bmatrix}$$

Fig. 6: Calibration pattern images and manual selection of control points.

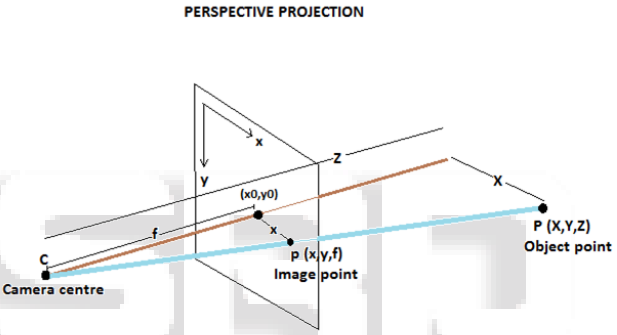


Fig. 7: Basic Perspective projection of a point P

Since the optical centre is (x0, y0), the image coordinates and the pixel locations are related as,

$$\begin{bmatrix} a \\ b \\ r \end{bmatrix} = \begin{bmatrix} \alpha x & s & x0 \\ 0 & \alpha y & y0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

$$\begin{bmatrix} a \\ b \\ r \end{bmatrix} = \begin{bmatrix} \alpha x & s & x0 \\ 0 & \alpha y & y0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} Xc \\ Yc \\ Zc \\ 1 \end{bmatrix}$$

where αx, αy are the pixel scaling factors and s is the slant or skew factor

$$r\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fx & s & x0 \\ 0 & fy & y0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} Xc \\ Yc \\ Zc \\ 1 \end{bmatrix},$$

where fx = (αx)(f) and fy = (αy)(f)

D. Determining a Unique solution set of Points:

The only invariant in perspective geometry is the "Cross ratio". It is a dimensionless ratio of ratios, which remains constant in the transformation from Cartesian plane to Projective plane. For four image points (l, m, n, q) and their corresponding Object points (L, M, N, Q) it is defined as:

In Cartesian plane, $R = \frac{\left(\frac{LN}{LQ}\right)}{\left(\frac{MN}{MQ}\right)}$

In projective plane, $r = \frac{\left(\frac{ln}{lq}\right)}{\left(\frac{mn}{mq}\right)}$

From perspective geometry, R = r. This relationship is extremely useful, to locate an unknown coordinate in a plane, when its location in the other plane is given. It is also needed to verify perspectivity of four pairs of points.

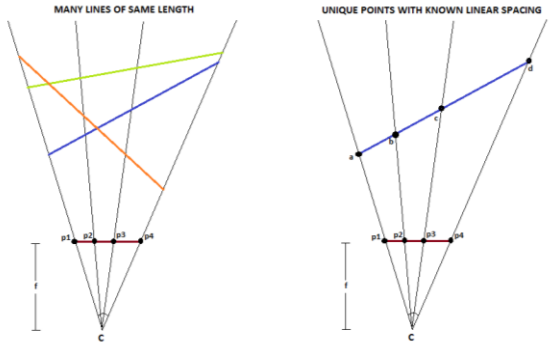


Fig. 8 (A): Many possibilities for line projection (B). Unique solution for point's projectio

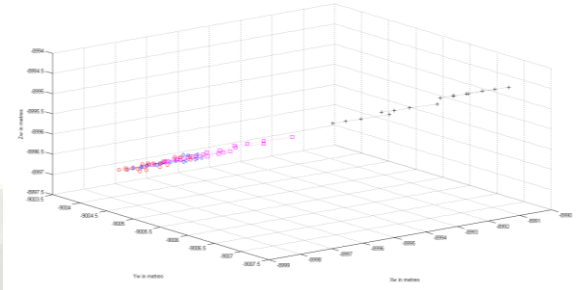


Fig. 9: Sample Point Locations in World coordinates

The points obtained in the previous section are the image coordinates in a plane parallel to the frame, with respect to the Camera Coordinate system. To obtain the real time point locations from a single image, the real time conditions must be imposed. If the length of the missile alone is taken as a condition then there are many possible orientations of the object which will result in the same projection. This is the reason why specifically four markers are chosen. These four points at known spacing can be uniquely found out from their projections, as shown in Fig. (8). The first condition is that the real time distance between individual markers is known and the second is that they are linearly placed on the missile. Four constraint equations are used to solve for these four locations, relative to camera coordinate system.

As shown in Fig. (8)B, consider the known points locations, in image plane as p1(x1,y1,z1), p2(x2,y2,z2), p3(x3,y3,z3) and p4(x4,y4,z4). Label their corresponding unknown point locations in object plane as a(xa,ya,za), b(xb,yb,zb), c(xc,yc,zc) and d(xd,yd,zd). Since, the problem is solved with respect to camera coordinate system, the camera centre, C (0, 0, 0). Suppose, the distances between four markers (points) on the missile (line) are,

$|ab| = 446.25$ mm, $|bc| = 446.25$ mm, $|cd| = 892.5$ mm

In vector form, the 3D line equation passing through, say, (x0, y0, z0) with <d, e, f> as a parallel vector is,

$$< x, y, z > = < x_0, y_0, z_0 > + t < d, e, f >$$

$$< x, y, z > = < (x_0 + td), (y_0 + te), (z_0 + tf) >,$$

where t is a positive scalar.

Since all four lines converge at the origin,

<x0, y0, z0> = (0, 0, 0),

$$< x_a, y_a, z_a > = t < x_1, y_1, z_1 > \quad \text{.....(1.1)}$$

$$< x_b, y_b, z_b > = u < x_2, y_2, z_2 > \quad \text{.....(1.2)}$$

$$< x_c, y_c, z_c > = v < x_3, y_3, z_3 > \quad \text{.....(1.3)}$$

$$< x_d, y_d, z_d > = w < x_4, y_4, z_4 > \quad \text{.....(1.4)}$$

In Fig. 8B, the position vector, $\overrightarrow{Ca} + \overrightarrow{ab} = \overrightarrow{Cb}$

that is,

$$t < x_1, y_1, z_1 > + < (x_b - x_a), (y_b - y_a), (z_b - z_a) > = u < x_2, y_2, z_2 >$$

rearranging terms,

$$t < x_1, y_1, z_1 > - u < x_2, y_2, z_2 > = < (x_b - x_a), (y_b - y_a), (z_b - z_a) >$$

by taking magnitude of the vectors and squaring on both sides,

$$\| t < x_1, y_1, z_1 > - u < x_2, y_2, z_2 > \|^2 = |ab|^2$$

In metres,

$$(tx_1 - ux_2)^2 + (ty_1 - uy_2)^2 + (tz_1 - uz_2)^2 = (0.44625)^2 \quad \text{.....(1.5)}$$

Similarly,

$$(ux_2 - vx_3)^2 + (uy_2 - vy_3)^2 + (uz_2 - vz_3)^2 = (0.44625)^2 \quad \text{.....(1.6)}$$

$$(vx_3 - wx_4)^2 + (vy_3 - wy_4)^2 + (vz_3 - wz_4)^2 = (0.8925)^2 \quad \text{.....(1.7)}$$

Now, the angle between vectors $\overrightarrow{ac}$ and $\overrightarrow{ad}$ in real time is zero, which yields,

$$(vx_3 - tx_1)(wx_4 - tx_1) + (vy_3 - ty_1)(wy_4 - ty_1) + (vz_3 - tz)(wz_4 - tz_1) = (0.8925)(1.785) \quad \text{.....(1.8)}$$

The above four equations are solved to deduce the values of t, u, v, and w. By substituting into equations (1.1), (1.2), (1.3) and (1.4), the object coordinates with respect to the camera coordinate system are determined.

E. Camera to World coordinates: Exterior Orientation Parameters:

The exterior orientation parameters describe the relationship between Camera coordinate system and World coordinate system. The matrix [12] involves two operations of rotation and translation. The single transformation matrix from Camera coordinate system to World coordinate system is defined as,

$$M_w^c = \begin{pmatrix} R_w^c & \vdots & T_w^c \\ 0 \;\; 0 \;\; 0 & \vdots & 1 \end{pmatrix}_{4\times4}$$

The inverse transformation matrix from World coordinate system to Camera coordinate system is,

$$M_c^w = \begin{pmatrix} (R_w^c)' & \vdots & -(R_w^c)'T_w^c \\ 0 \;\; 0 \;\; 0 & \vdots & 1 \end{pmatrix}_{4\times4},$$

where $'$ indicates transpose.

Here, T_w^c is the translation vector which indicates the position of the camera with respect to the world coordinate system. It moves the camera centre to the world origin. R_w^c is the rotation matrix from Camera coordinate system to World coordinate system. It is a product of rotations about all three axes, taken in the reverse order of operation. Consider three sequential rotations, first α° in clock wise direction (−α° if rotation is anti-clock wise) about X axis, second β° in clockwise direction about Y axis and third γ° in clockwise direction about Z axis. Then the total rotation matrix is,

$$R_w^c = R_\gamma R_\beta R_\alpha$$

$$= \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix}$$

For example, in Fig. 3, the sequence of rotations to convert Camera coordinate system to World coordinate system, arbitrarily is, -90˚(anti clockwise) around X axis. This rotation would align the Camera axes with World axes. Suppose, the camera is at location (9000, 9000, 9000) metres with respect to World coordinate system.

$$R_w^c = R_\alpha = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-90°) & \sin(-90°) \\ 0 & -\sin(-90°) & \cos(-90°) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \text{ and } T_w^c = \begin{bmatrix} 9000 \\ 9000 \\ 9000 \end{bmatrix} \text{ metres}$$

$$\text{then, } M_w^c = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 9000 \\ 0 & 0 & -1 & 9000 \\ 0 & 1 & 0 & 9000 \\ 0 & 0 & 0 & 1 \end{array}\right)$$

Hence, the transformed coordinates in 3D World,

$$\text{in metres, } \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix} = M_w^c \begin{bmatrix} Xcc \\ Ycc \\ Zcc \\ 1 \end{bmatrix}$$

where, (U, V, W) are the required locations of the marker (point) in real time 3Dimensional world coordinates and (Xcc, Ycc, Zcc) are the constrained locations in Camera coordinate system as obtained in the previous section. Using sample data, the point location are illustrated in Fig. 9.

IV. FILTERING AND PREDICTION

After the missile locations are located in real time, filtering the measurements and determining its velocity track remains. This task is elegantly [14] accomplished by a statistical estimation tool called the Kalman Filter.

A. The Kalman Filter:

The Kalman filter is a mathematical tool which recursively solves estimation and tracking problems. It is derived [13] as a stochastic solution to optimal Bayesian filtering algorithm. The mathematical description of discrete time Kalman filter is given in Table 1.

Object dynamic model	$x_k = \varphi_{k-1}x_{k-1} + \Gamma w_{k-1}$	$w_k \sim N(0, Q_k)$
Measurement model	$z_k = H_k x_k + v_k$	$v_k \sim N(0, R_k)$
Initial conditions	$E\{x_0\} = \hat{x}_0$ $E\{x_0 x_0'\} = P_0$	
State estimation extrapolation	$\hat{x}_k(-) = \varphi_{k-1}\hat{x}_{k-1}(+)$	
Error covariance extrapolation	$P_k(-) = \varphi_{k-1}P_{k-1}(+)\varphi_{k-1}' + \Gamma Q_{k-1}\Gamma'$	
State estimate observational update	$\hat{x}_k(+) = \hat{x}_k(-) + K_k[z_k - H_k\hat{x}_k(-)]$	
Error covariance update	$P_k(+) = [I - K_k H_k]P_k(-)[I - K_k H_k]' + K_k R_k K_k'$	
Kalman gain matrix	$K_k = P_k(-)H_k'[H_k P_k(-)H_k' + R_k]^{-1}$	

Table 1: Discrete Time Kalman Filter equations [14]

where,

$\hat{x}_k(+)$ is the state estimate after being updated by the measurement at k, known as 'aposteriori' value of the estimate. In contrast, $\hat{x}_k(-)$ is the state estimate before updating with current measurement, known as 'apriori' estimate. Similarly, $P_k(-)$ is the apriori covariance and $P_k(+)$ is the aposteriori covariance of estimation uncertainty. The other parameters are described in table (2).

In object tracking, Kalman filter can be applied to single non maneuvering target, by assuming that the object dynamic model and measurement model are linear with additive white uncorrelated Gaussian noises of zero mean and variances $Q_k$and R_k respectively, along with Gaussian posterior density of object state [13].

A second order kinematic model is used whenever the object is travelling with uniform velocity and practical random disturbances add white noise. This leads to a nonzero acceleration, hence the name White noise Acceleration model. A third order kinematic model is used whenever the object is travelling with uniform acceleration and practical random disturbances add white noise. This leads to non zero jerk, hence the name White noise Jerk model. For the problem at hand, the missile is travelling with uniform velocity in X and Y directions. However, a negative acceleration due to gravity (9.8 m/s2), is acting in the Z direction. Therefore two independent, decoupled Kalman filters are modeled as defined in Table 2. It is to be noted that if the measurements are not available, i.e., when the missile markers (all four) are not visible anymore, then the measurement at interval k, z_k is intuitively the state estimate at interval (k-1), $\hat{x}_{k-1}$.

The efficiency of the Kalman filter is dependent on the proper choice of design parameters based on observability of the system. An illustration on sample date to filter or predict the object locations and to track its velocity is shown in Fig. 10 A and B respectively. The simulation is performed for 10000 iterations, with T=0.001, σ_w^2=0.000001, σ_v^2=0.05, zero noise gain and zero initial states in MATLABTM.

	Continuous White Noise Acceleration Model for X and Y directions	Continuous White Noise Jerk Model for Z direction
State vector	$x_k = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \text{position in x} \\ \text{velocity in x} \\ \text{position in y} \\ \text{velocity in y} \end{bmatrix}$	$x_k = \begin{bmatrix} z \\ \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} \text{position in z} \\ \text{velocity in z} \\ \text{acceleration in z} \end{bmatrix}$
St ate Transitio n matrix	$\varphi_k = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\varphi_k = \begin{bmatrix} 1 & T & -\frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}$
Noise gain in Object dynamic equation	$\Gamma = \begin{bmatrix} \frac{1}{2}T^2 \\ T \\ \frac{1}{2}T^2 \\ T \end{bmatrix}$	$\Gamma = \begin{bmatrix} \frac{1}{2}T^2 \\ T \\ 1 \end{bmatrix}$
Measure ment sensitivit y matrix	$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	$H_k = [1 \quad 0 \quad 0]$

Process noise Covariance matrix	$Q_k = \begin{bmatrix} \frac{1}{4}T^4 & \frac{1}{2}T^3 & \frac{1}{4}T^4 & \frac{1}{2} \\ \frac{1}{2}T^3 & T^2 & \frac{1}{2}T^3 & \\ \frac{1}{4}T^4 & \frac{1}{2}T^3 & \frac{1}{4}T^4 & \frac{1}{2} \\ \frac{1}{2}T^3 & T^2 & \frac{1}{2}T^3 & \end{bmatrix}$	$Q_k = \begin{bmatrix} \frac{1}{4}T^4 & \frac{1}{2}T^3 & \frac{1}{2}T \\ \frac{1}{2}T^3 & T^2 & T \\ \frac{1}{2}T^2 & T & 1 \end{bmatrix}$
Measurement noise Covariance matrix	$R_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \sigma_v^2$	$R_k = \sigma_v^2$
where T = sampling period (inversely proportional to the frame rate) and k=1 to N (number of iterations)		

Table 2: Kinematic state models [15]

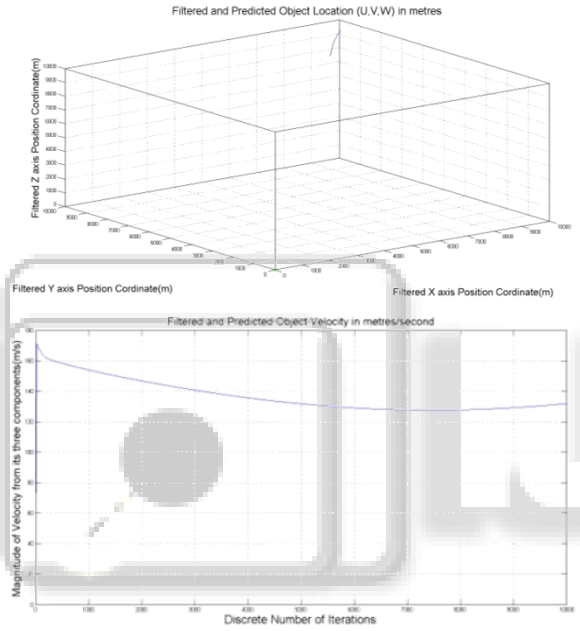


Fig. 10 (A). Filtered And Predicted Object Locations (B). Object Velocity Track

V. CONCLUSION

An experimental framework for the problem of non manoeuvring object tracking and analysis, from a single view image is presented. It can be a supplement tool for high speed imaging analysis. These methods are also useful to analyse image based collision avoidance, military testing involving high speed tiny particles and other applications, where radar is not feasible. The algorithms applied are fundamental. More complex situations of manoeuvring objects can be tackled by adaptive estimation algorithms [15].

REFERENCES

[1] Eyung W.Kang, Radar System Analysis, Design, and Simulation. Artech House, Inc., 2008.
[2] Berthold K.P.Horn, and Brian G.Schunck, "Determining optical flow", Artificial Intelligence, Vol. 17, no. 1-3, pp. 185-203, Aug. 1981.
[3] Alper Yilmaz, Omar Javed, and Mubarak Shah, "Object Tracking: A Survey", ACM Computing Surveys, Vol. 38, no. 4, Article 13, Dec. 2006.
[4] P. Viola, and M. Jones, "Rapid object detection using a boosted cascade of simple features", IEEE Computer Society Conference on CVPR, Vol. 1, pp. I-511-I-518, 2001.
[5] Lecture12,13 from http://www.cse.psu.edu/~rtc12/CSE486/
[6] R. E. Kalman, "A new Approach to Linear Filtering and Prediction Problems", ASME Journal of Basic Eng., no. 82(series D), pp. 35-45, 1960.
[7] Edward M. Mikhail, James S. Bethel, and J. Chris McGlone, Introduction to Modern Photogrammetry. John Wiley & Sons, 2001.
[8] Appendix-Projective Geometry for Machine Vision, from https://cs.uwaterloo.ca/~mannr/cs498-w01/zisser-mundy.pdf
[9] Homogeneous Coordinates from http://web.cs.iastate.edu/~cs577/
[10] Z. Zhang, "Flexible Camera Calibration by Viewing a Plane from Unknown Orientations", IEEE Int. Conf. On Computer Vision, Vol. 1, pp. 666-673, 1999.
[11] Camera Calibration Toolbox, from https://www.vision.caltech.edu/bouguetj/calib_doc/
[12] Robotics: Perception, from https://www.coursera.org/learn/robotics-perception
[13] Challa, Sudha, Fundamentals of Object Tracking. New York: Cambridge University Press, 2 011.
[14] Grewal, Mohinder S., and Angus P. Andrews. Kalman Filtering: Theory and practice using MATLAB. New Jersey: Wiley, 2008.
[15] Shalon, Yaakov, Xiao, and Thiagalingam Kirubarajan. Estimation with applications to tracking and navigation. New York: Wiley, 2001.
[16] Phantom Camera Control (PCC) software from http://www.phantomhighspeed.com/Portals/0/Files/Documentation/FE_WEB-Measurements.pdf?ver=2016-02-09-124106-550

