Signal Processing: Image Communication 27 (2012) 83–95

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

Human tracking from a mobile agent: Optical flow and Kalman filter arbitration

Yuichi Motai [n], Sumit Kumar Jha, Daniel Kruse

Virginia Commonwealth University, School of Engineering, 601 West Main Street, PO Box 843068, Richmond, VA 23284-3068, USA

a r t i c l e i n f o

Article history:
Received 5 November 2010
Accepted 23 June 2011
Available online 5 July 2011

Keywords:
Human tracking
Kalman filter
Mobile robot
Optical flow

a b s t r a c t

Tracking moving objects is one of the most important but problematic features of motion analysis and understanding. The Kalman filter (KF) has commonly been used for estimation and prediction of the target position in succeeding frames. In this paper, we propose a novel and efficient method of tracking, which performs well even when the target takes a sudden turn during its motion. The proposed method arbitrates between KF and Optical flow (OF) to improve the tracking performance. Our system utilizes a laser to measure the distance to the nearest obstacle and an infrared camera to find the target. The relative data is then fused with the Arbitrate OFKF filter to perform real-time tracking. Experimental results show our suggested approach is very effective and reliable for estimating and tracking moving objects.

1. Introduction

Human tracking using mobile robots serves a lot of attention because an automated system estimating and tracking moving objects has many potential applications in the field of surveillance and engineering [46]. The mobile robot target tracking mechanism essentially needs the motion analysis of the target behavior. A number of approaches on prediction and tracking are based on the traditional Kalman filter (KF) [10–15]. In the KF approach, it is presumed that the behavior of a moving target could be characterized by a predefined model, and the models can be represented in terms of a state vector. In reality, however, these models fail to characterize the motion of moving targets accurately. As a result of this, KF fails to track a target, especially when there are occlusions caused by other objects or sudden changes in the trajectory of the motion [13]. In this paper, we propose an efficient method of tracking the target called Arbitrate OFKF, which will overcome the abovementioned problem and work successfully even when sudden changes in trajectory occur. This proposed method arbitrates the output of OF and KF depending on the trajectory of the previous motion. Our arbitration algorithm is such that the output, during sharp turns, will be taken from OF algorithm, and, in all other instances, from KF algorithm.

There are some visual restrictions in which any methods cannot be applicable, for instance in industrial applications, like the ones examined in visual environments and applications [48], airports with crowded conditions, industrial process monitoring [49] and environmental monitoring. We have exploited an infra-camera for visual content, in which the presented technique is applicable as other types of visual contents that can potentially solve these existing issues in [48,49].

The paper is organized as follows. Section 2 introduces the related work in the tracking using a mobile robot. Section 3 explains the piecewise constant acceleration model for implementation of KF. In Section 4, a solution for improving KF based tracking using arbitration between KF and OF. Section 5 reports experimental

[n] Corresponding author. Tel.: þ 1 804 828 1281.
E-mail address: ymotai@vcu.edu (Y. Motai).

0923-5965/$ - see front matter & 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2011.06.005

results and considerations. Finally, conclusions and future work are illustrated in Section 6.

2. Related works

2.1. Mobile robot

Tracking people by the use of a mobile robot is an essential task for the coming generation of service and human-interaction robots. In the field of computer vision, the tracking of moving objects by mobile robot is a relatively hard problem. The system described in [1] uses a sample-based joint probabilistic data association filter for human tracking with a mobile robot. A fuzzy logic approach is used for navigation of the mobile robot [2] whereas the neurofuzzy-based approach for tracking described in [3–6] uses a learning algorithm based on neural network techniques to tune the parameters of membership functions. Artificial potential functions [7] and vector-field histograms [8] are also used for mobile robot navigation. Most of the tracking methods focus on tracking humans in image sequences from a single camera view. In [9], each walking subject image was bounded by a rectangular box, and the centroid of the bounding box was used as the feature to track. We have used the KF and the OF for the navigation of the mobile robot because it will predict next state from the previous state.

2.2. Kalman filter-based target tracking

Human tracking by mobile robots has been an area of intense research for years. Humans can be tracked by mobile robots using 3D or 2D data by any normal KF [10,13], or using segmentation of the main target from the background [11,12]. Other approaches are based on hierarchical KF [14] and quaternion [15]. Extended KF [16] and the interactive multiple model [17] have successfully been used for human tracking using mobile robots in the past. In [18], stereo vision and KFs are used to track and follow a single person using a mobile robot for short distances. But sudden deformation in the target motion can cause the failure of the predefined behavior model of KF [13]. In [19], a robot equipped with two laser range sensors, one pointing forward and another backward, can track several people using a combination of particle filters and joint probabilistic data association. The system described in [21] uses a classic histogram intersection to identify people together with a standard KF for laser-based tracking. In [20], a particle filter is used for the data fusion of a laser and an omnidirectional camera is used for multiple people tracking. Particle filters are the sequential of Markov chain Monte Carlo methods, and alternative to the KF with the advantage that, with sufficient samples, they approach the Bayesian optimal estimate. We also have specifically developed a new KF framework of Interactive Multiple Model-based Estimation (IMME) [43–46], and showed the best prediction performance. While the methods described above are computationally expensive, our proposed method is, by comparison, inexpensive and also produces better results in the event of sudden deformations of the target motion. The experimental sections demonstrate the performance and comparisons with respect to the particle filers [47].

2.3. Optical flow-based target tracking

Optical flow can arise from relative motion between objects and the viewer [22]. OF can give important information about the spatial arrangement of the viewed objects and the rate of change of this arrangement [26]. There are a number of methods for calculating OF. The primitive methods are Sum of Squared Difference [23–25] and the Normalized Cross-Correlation [27] [29]. The other methods are based on Laplacian derivative [28,30]. In these algorithms, a small area of the image is used as a template, searched later throughout the interested region in the next frame for finding the target location. Another template-matching method called contour matching [31–33] uses a non-rigid contour as a model, and is relatively robust to a cluttered background. All of these methods are, however, computationally expensive. The algorithm for calculating OF is made a little more efficient as in [34]. Our proposed method for calculating OF does not involve much computational cost. Since, in our setup, the camera is also moving, we have to use the concept of superposition. According to superposition, OF generated by the target is the difference of the total OF generated and the OF generated by camera motion. OF-based tracking works well only when the target is moving and its surroundings are still. Due to this property, OF gives the best tracking results in an indoor environment.

2.4. Arbitration

Arbitration is the process of selecting one action or behavior from multiple possible candidates. When two or more algorithms control the motion of the robot, in order to use mutually exclusive condition for the controller, we have to use the concept of arbitration [35]. Depending on the task, the algorithm for arbitration is decided. In the tasks of recognition and detection, the arbitration is decided by the threshold value [36,39], and sensor fusion is handled by the probability based on priority [38–42]. An arbitration operator based on Revesz's definition of arbitration must use an arbitration algorithm [37]. We have developed our own algorithm for the arbitration module, which arbitrates between KF and OF depending on the trajectory of the target motion.

3. Traditional methods: Framing the model of target tracking

First, we choose a state vector to contain information about the position and the velocity of the man. We then update the state according to our model called the constant acceleration model, in which the value of acceleration is calculated from the last few states, using a Taylor series expansion of the present expansion. The state can thus be thought of as being updated through the equations:

$$x_{k+1} = Ax_k + B\frac{dx_k}{dt_k} + w_k \qquad (1)$$

And the output equations may be written as

$$y_k = Cx_k + z_k, \quad (2)$$

where A, B and C are matrices of the required order to be multiplied with the state vectors. The terms 'w' and 'z' are noise matrices of the same order as 'x' and 'y' vectors, respectively. The noise matrices can be preset to the maximum error value possible in the measurements, and may vary as the readings get more and more accurate.

3.1. Kalman filter using piecewise constant acceleration model

If the time difference of the two readings is taken to be 'Dt', we can write the present velocity of the man as

$$v_{k+1} = v_k + u_k Dt, \quad (3)$$

where u_k is the calculated acceleration of the man at the previous instant. We assume that 'Dt' being small the acceleration of the man remains small. In reality, 'Dt' is of the order of 0.05–0.067 s. So our assumption does not become a source of error.

Let us consider two variables in our state vector– one being the position of the man (p), and the other being the velocity at that instant (v). We can write the updated equations for the position and the velocity, taking into account the noise:

$$v_{k+1} = v_k + u_k t_k + \hat{v}_k \quad (4)$$

$$p_{k+1} = p_k + t_k v_k + \tfrac{1}{2} t^2 u_k + \hat{p}_k \quad (5)$$

Thus the state matrix becomes $x_k = \begin{bmatrix} p_k \\ v_k \end{bmatrix}$.

We can thus update Eqs. (1) and (2) as follows:

$$x_{k+1} = \begin{bmatrix} 1 & Dt \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} \frac{Dt^2}{2} \\ Dt \end{bmatrix} u_k + w_k \quad (6)$$

$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + z_k \quad (7)$$

The process noise covariance may be written as

$$S_w = E(w_k w_k^T). \quad (8)$$

Similarly the measurement covariance noise may be written as

$$S_z = E(z_k z_k^T). \quad (9)$$

The superscript 'T' suggests the transpose of the matrices. Thus, we have obtained the basic matrices:

$$A = \begin{bmatrix} 1 & Dt \\ 0 & 1 \end{bmatrix} \quad (10)$$

$$B = \begin{bmatrix} Dt^2/2 \\ Dt \end{bmatrix} \quad (11)$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (12)$$

We use two matrices– one called the estimation error covariance matrix, which we shall denote by 'P', and the other called the Kalman gain matrix, which will be denoted by 'K'. Initially we set P as

$$P = S_w. \quad (13)$$

And we set our initial estimate

$$x_k = x. \quad (14)$$

We first calculate the Kalman gain matrix, which is given by

$$K = AP_k C^T (CP_k C^T + S_z)^{-1}. \quad (15)$$

We then predict the next state, which is given by

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k. \quad (16)$$

Based on the data we obtain from the sensor, i.e. the infrared camera and then the range finder, we update the estimate. We develop a correction term 'Corr', which gives us the error that has propagated in our state estimate. We thus have to correct our state estimate by that amount

$$Corr_k = y_{k+1} - Cx_k, \quad (17)$$

$$\hat{x}_{k+1} = \hat{x}_{k+1} + K_k(Corr_k). \quad (18)$$

We then update the estimation error covariance matrix:

$$P_{k+1} = AP_k A^T + S_w - AP_k C^T S_z^{-1} CP_k A^T. \quad (19)$$

The first element of the estimation vector thus gives the predicted position of the man, and the second element gives its predicted velocity. Using this data, the robot is maneuvered accordingly.

3.2. Feedback mechanism

There is a time lag between the initiation of a movement by the robot, and the rate at which frames are scanned by the infrared camera. Hence we use a feedback mechanism to overwrite the command given to the robot as shown in Fig. 1.

The rate at which commands are issued to the robot depends on the rate of the camera, which is faster compared to the robot. Thus, at each stage, while the robot is in motion, frames are grabbed by the camera, and the remaining angle left to rotate is calculated and estimated. After this has been calculated, the command in the robot buffer is over-written by the remaining angle left to rotate.

One advantage of the feedback motion is the angle, by which the robot movement is constantly updated. Hence, the robot comes to a standstill when it is centered on the man it is tracking. When the man moves again from the central

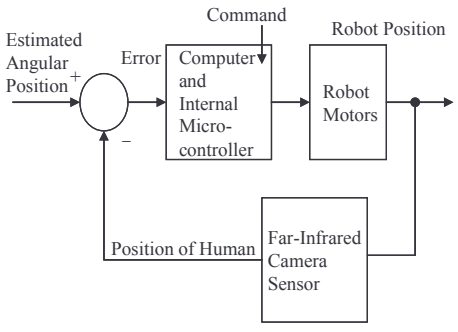

Fig. 1. Semantic diagram for feedback system of human target tracking.

position of the camera, the same process is again initiated to bring the man back to the center of the image. Thus, the camera never loses the man.

Say at (k+1)th frame, the estimated angle is γ with respect to reference, and the camera facing angle at (k)th frame is α with respect to same reference frame so that the robot should be moved for tracking the man, only by β which is equal to

$$\beta = \gamma - \alpha, \tag{20}$$

where the sign in (20) depends upon the direction of angle. If both angles are in same direction, take minus sign, otherwise plus sign will be used.

Another advantage of the feedback motion comes into play when the man suddenly changes direction while walking, or moves further away. If the feedback system had not been implemented, the man could have disappeared from the camera visibility by the time the robot rotated and got to the man's original position. With the system in place, even if the man moves further away or changes direction, the commands for robot rotation or translation will be changed immediately. We ensure that the Kalman filter is continuously updated, and new commands immediately overwrite the old commands in the robot buffer.

4. Proposed pan-tilt operation

In the target region of interest (ROI) tracking, we need to find the required pan and tilt angle for the specific camera configuration as shown in Fig. 2 so that the camera head can rotate to track the target. In other words, this is the angle vector γ required to make image center coincident with the target centroid:

$$\gamma = \begin{bmatrix} \gamma_x \\ \gamma_y \end{bmatrix}, \tag{21}$$

where the rotational angle (γ_x) with respect of the center of the image is the only angle associated with pan defined by ΔX. It is similar with (γ_y), when there is only tilt operation. We calculate the degree of these values using the geometry:

$$\gamma_x = \tan^{-1}\left[\frac{\Delta X}{l}\right], \tag{22}$$

$$\gamma_y = \tan^{-1}\left[\frac{\Delta Y}{l}\right], \tag{23}$$

where (l) is the focal length of camera, and γ_x and γ_y are unknown. We need to now calculate the rotation angles to be operated by the pan-tilt-camera. The following is our solution to this problem.

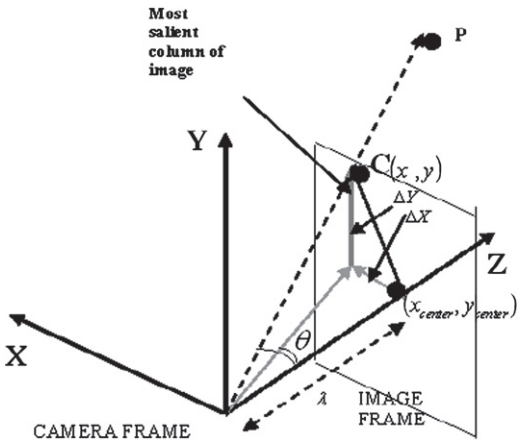

Fig. 2. Camera configuration.

Optical flow is the pattern of apparent motion of objects in a visual scene caused by the relative motion between a camera and the scene. The concept behind the Optical flow based tracking is— if the target motion is known, then the camera may be made to eliminate relative motion between it and the target, hence facilitating tracking. In particular, if the motion of the target is known well, then "perfect tracking" may be achieved, i.e. center position of the target region of interest (ROI) can always be kept at the center of image. Target motion can be estimated through the change of the image position, which is called "Optical flow" or image displacement. By calculating the image displacement or optical flow, which is induced by target motion, one can estimate the motion of the target when the camera is stationary. For a stationary object and a moving camera, the optical flow induced by the camera motion is as follows:

$$u_O = \left(\frac{xR_z}{Z} - \frac{lR_x}{Z}\right) + \frac{xyw_x}{l} - \left(\frac{l^2 + x^2}{l}\right)w_y + yw_z \tag{24}$$

$$v_O = \left(\frac{yR_z}{Z} - \frac{lR_y}{Z}\right) + \frac{l^2 + y^2}{l}w_x - \frac{xyw_y}{l} - xw_z \tag{25}$$

where u_O and v_O are obtained from the optical flow, and w_x, w_y, w_z and R_x, R_y, R_z are rotational and translational velocity of the camera, respectively, 'x' and 'y' are center of ROI in image frame, 'Z' is the Z coordinate of the target in camera frame, l is a focal length and subscript 'O' is denoted as an object. If we assume a moving object and a stationary camera instead of a moving camera and a stationary object, then we can obtain the same result as (24) and (25) except for a sign reversal [23].

In our tracking environment, both the camera and the target are non-stationary. The optical flow is thus subject to both the camera and the target movement. In order to take this into account we have to modify the optical flow equation such that the optical flow induced only by target motion is taken into account. According to superposition, the total optical flow is equal to the sum of the optical flow of target and that of camera. Suppose that the optical flow at the time instant k during the tracking phase is [u(k), v(k)], then the optical flows Are [23]

$$u(k) = u_O(k) + u_{ca}(k), \tag{26}$$

$$v(k) = v_O(k) + v_{ca}(k), \tag{27}$$

where u(k) and v(k) are the total optical flow, $v_O(k)$ and $u_O(k)$ are the optical flow induced by the target motion, whereas $u_{ca}(k)$ and $v_{ca}(k)$ are the optical flows induced by the tracking motion of the camera in X and Y directions, respectively. Therefore, the optical flow induced by the target motion

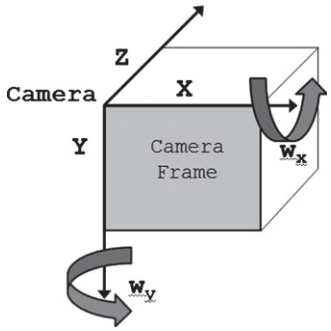

Fig. 3. Camera pan-tilt motion in 3-D plane.

during the tracking phase can be obtained by modifying (26) and (27) as follows:

$$u_O(k) = u(k) - u_{ca}(k), \tag{28}$$

$$v_O(k) = v(k) - v_{ca}(k). \tag{29}$$

Pan-directional angle w_x and tilt-directional angle w_y in Fig. 3 allow the camera to rotate only in the direction with respect to the image plane X–Y. So the motion vector of camera is

$$R = \begin{bmatrix} w_x & w_y & 0 \end{bmatrix}^T. \tag{30}$$

Usually cross-correlation and sum-of-difference method are used for calculating observable modified optical flow, but we will use our own method for finding this. By using our own motion estimation algorithm, we can estimate the target image position at any time. Assume that the center of ROI at (k)th frame is $[x(k)\ y(k)]^T$ and our camera's angular velocity is $w_x(k) = \Delta y_x/\Delta t$ and $w_y(k) = \Delta y_y/\Delta t$, where Δy_x and Δy_y are the changes in pan and tilt angles, respectively, and Δt is the time interval between two consecutive frames. Let the target move to the position $[x(k), y(k)]^T$ at the next frame, then $u(k)$, $v(k)$, $u_{ca}(k)$ and $v_{ca}(k)$ at the (k)th frame may be derived as follows using (24), (25) and (30):

$$u(k) = \frac{x(k+1) - x(k)}{\Delta t} \tag{31}$$

$$v(k) = \frac{y(k+1) - y(k)}{\Delta t}, \tag{32}$$

$$u_{ca} = \frac{x(k)y(k)}{\lambda} w_x(k+1) - \frac{\lambda^2 + x^2(k)}{\lambda} w_y(k+1), \tag{33}$$

$$v_{ca} = \frac{\lambda^2 + y^2(k)}{\lambda} w_x(k+1) - \frac{x(k)y(k)}{\lambda} w_y(k+1), \tag{34}$$

and using (28) and (29)

$$u_O(k) = \frac{x(k+1) - x(k)}{\Delta t} - \frac{x(k)y(k)}{\lambda} w_x(k+1) + \frac{\lambda^2 + x^2(k)}{\lambda} w_y(k+1) \tag{35}$$

$$v_O(k) = \frac{y(k+1) - y(k)}{\Delta t} - \frac{\lambda^2 + y^2(k)}{\lambda} w_x(k+1) + \frac{x(k)y(k)}{\lambda} w_y(k+1) \tag{36}$$

We can estimate the motion using the above modified optical flow equation. The motion estimation algorithm is based on the estimation of rotation velocity w_x and w_y. Our algorithm can estimate the motion of the target without knowing the actual depth information. We write equations (24) and (25) in discrete time expression as

$$\begin{bmatrix} u_O(k) \\ v_O(k) \end{bmatrix} = J(x(k), y(k), z(k)) \begin{bmatrix} R(k) \\ w(k) \end{bmatrix}, \tag{37}$$

where

$$J(x(k), y(k), z(k)) = \begin{bmatrix} \frac{1}{z(k)} & 0 & \frac{x(k)}{z(k)} & \frac{x(k)y(k)}{\lambda} & \frac{\lambda^2 + x^2(k)}{\lambda} & y(k) \\ 0 & \frac{1}{z(k)} & \frac{y(k)}{z(k)} & \frac{\lambda^2 + y^2(k)}{\lambda} & \frac{x(k)y(k)}{\lambda} & -x(k) \end{bmatrix} \tag{38}$$

The matrix $J(x(k), y(k), z(k))$ is called the Image Jacobian and $R = [R_x, R_y, R_z]^T \in R^3$ and $w = [w_x, w_y, w_z]^T \in R^3$ are the translational and rotational velocities of camera, respectively.

In order to keep tracking, the ROI centroid in image frame should pass through the center of the image. According to the content above, we use $(x(k), y(k))$ to represent the center of ROI at (k-1)th frame. Using (30) and (37) and perspective projection model [35] gives

$$\begin{bmatrix} u_O(k) \\ v_O(k) \end{bmatrix} = \begin{bmatrix} \frac{x(k)y(k)}{\lambda} & \frac{\lambda^2 + x^2(k)}{\lambda} \\ \frac{\lambda^2 + y^2(k)}{\lambda} & \frac{x(k)y(k)}{\lambda} \end{bmatrix} \begin{bmatrix} o_x(k) \\ o_y(k) \end{bmatrix}. \tag{39}$$

Let

$$P = \begin{bmatrix} \frac{x(k)y(k)}{\lambda} & \frac{\lambda^2 + x^2(k)}{\lambda} \\ \frac{\lambda^2 + y^2(k)}{\lambda} & \frac{x(k)y(k)}{\lambda} \end{bmatrix}, \quad V = \begin{bmatrix} o_x(k) \\ o_y(k) \end{bmatrix} \text{ and } C = \begin{bmatrix} u_O(k) \\ v_O(k) \end{bmatrix}$$

Rewrite (39) as

$$C = P\,V, \tag{40}$$

where C is the optical flow induced by target motion, P is composed of feature's image coordinate and focal length and V consists of rotational velocity of camera with respect to the camera frame. For non-singular matrix P, (40) can be easily solved as follows:

$$V = P^{-1}C. \tag{41}$$

Thus, we can calculate the unknown values for w_x and w_y using (41).

The next step of our tracking algorithm is the prediction. The purpose herein is to predict where the target image location will 'move to' in the next (k)th frame predicted. Thus, the prediction $(x_{predict}(k), y_{predict}(k))$ can be obtained as follows:

$$x_{predict}(k) = x(k) + u_O(k)\Delta t, \tag{42}$$

$$y_{predict}(k) = y(k) + v_O(k)\Delta t. \tag{43}$$

The value (x, y) is fed to the tracking module to generate the desired camera angular motion $y_x(k)$ and

$y_y(k)$ so that we can calculate the camera parameters for tracking ROI at the (k)th frame:

$$y_x(k) = \tan^{-1}\frac{x_{predict}(k) - x_{center}(k-1)}{l}, \quad (44)$$

$$y_y(k) = \tan^{-1}\frac{y_{predict}(k) - y_{center}(k-1)}{l}, \quad (45)$$

where $(x_{center}(k-1), y_{center}(k-1))$ is the center of the image, and not the center of ROI, obtained from the pan-tilt camera position using the triangulation shown in Fig. 2 at $(k-1)$th frame. Note that Eqs. (44) and (45) correspond to Eqs. (21), (22) and (23) to calculate the desired camera pan-tilt-parameters shown in Fig. 3.

5. Arbitration of Optical flow and Kalman filter

We have applied the Optical flow as shown in the flow chart in Fig. 4 for the tracking process. We further propose the Arbitrate OF and KF algorithm to calculate the estimated value of the pan angle, which will be used for our prediction. In our proposed method, both KF and OF will predict the value of the pan angle but, depending on the situation, we will decide which value should be fed to the robot for tracking the target. If the man walks smoothly, then the pan angle value predicted by KF will be fed to the robot. Otherwise, the

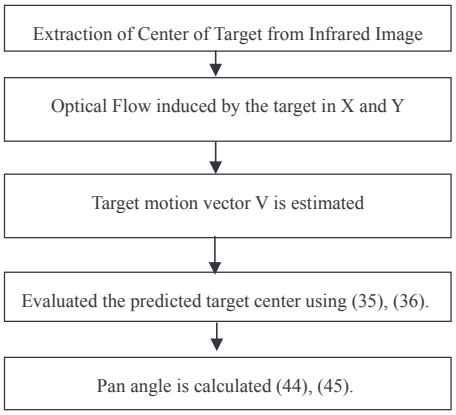

Fig. 4. Flow chart of optical-flow based target tracking.

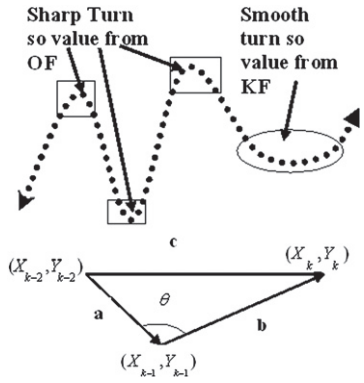

Fig. 5. Previous trajectory of man's motion by which the required pan angle is decided.

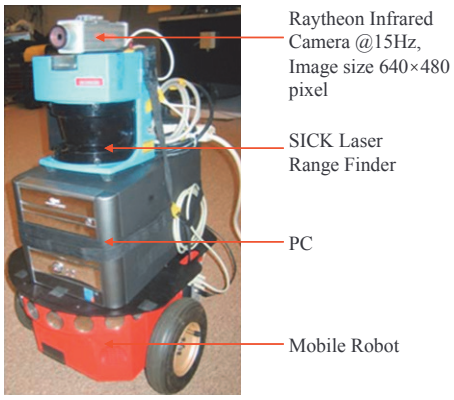

Fig. 6. Pioneer mobile robot platform.

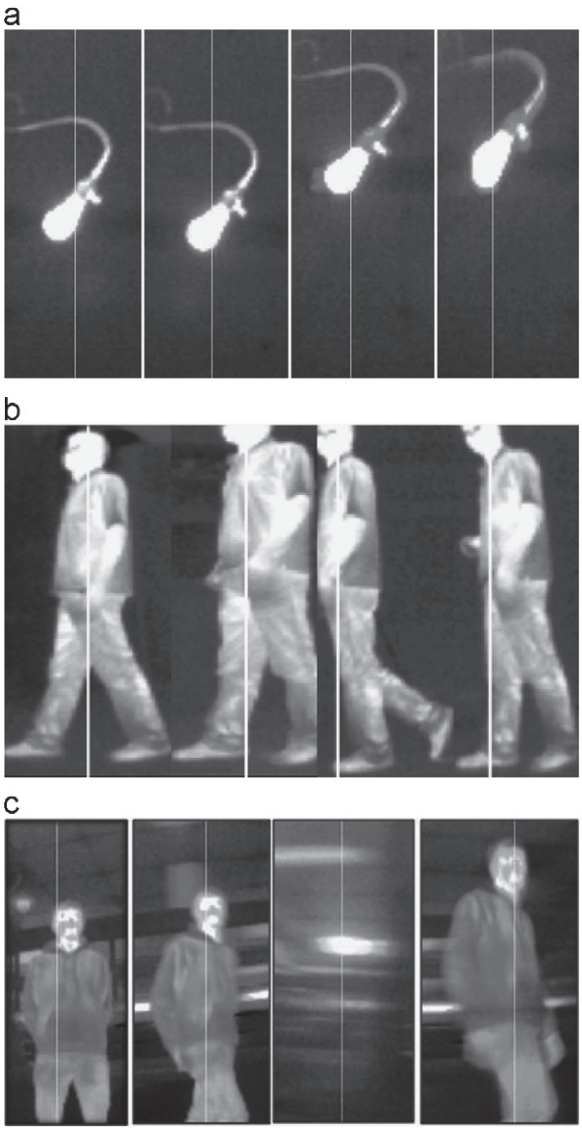

Fig. 7. Sample images from (a) Ground Proof, (b) Human Target and (c) Human Target with a noisy background.

pan angle value predicted by OF will be fed to the robot. The proposed algorithm for arbitration is as follows:

Step 1: Make a triangle using the center of the target in (k)th, (k−1)th and (k−2)th as vertices as shown in Fig. 5. Say (X_k, Y_k), (X_{k-1}, Y_{k-1}) and (X_{k-2}, Y_{k-2}) are the centers of the target in (k)th, (k−1)th and (k−2)th frames, respectively. Find out the length of each side:

$$a = \sqrt{(X_k - X_{k-1})^2 + (Y_k - Y_{k-1})^2} \quad (46)$$

$$b = \sqrt{(X_k - X_{k-2})^2 + (Y_k - Y_{k-2})^2} \quad (47)$$

$$c = \sqrt{(X_k - X_{k-2})^2 + (Y_k - Y_{k-2})^2} \quad (48)$$

Step 2: Using a, b and c from (46), (47) and (48), the angle y between side of length a and b equals to

$$y = \cos^{-1}\frac{a^2 + b^2 - c^2}{2ab} \quad (49)$$

Step 3: From (49) y is an angle of the triangle, so its value will be less than 180. If $160 < y < 180$, then we are assuming that the man is walking in a straight line. The pan angle value will in this case be estimated by KF. In all other cases, the pan angle value is estimated by the OF. As one example, the pan angle will be estimated by OF for the trajectories shown by rectangular boxes in Fig. 5, otherwise KF will estimate the pan angle.

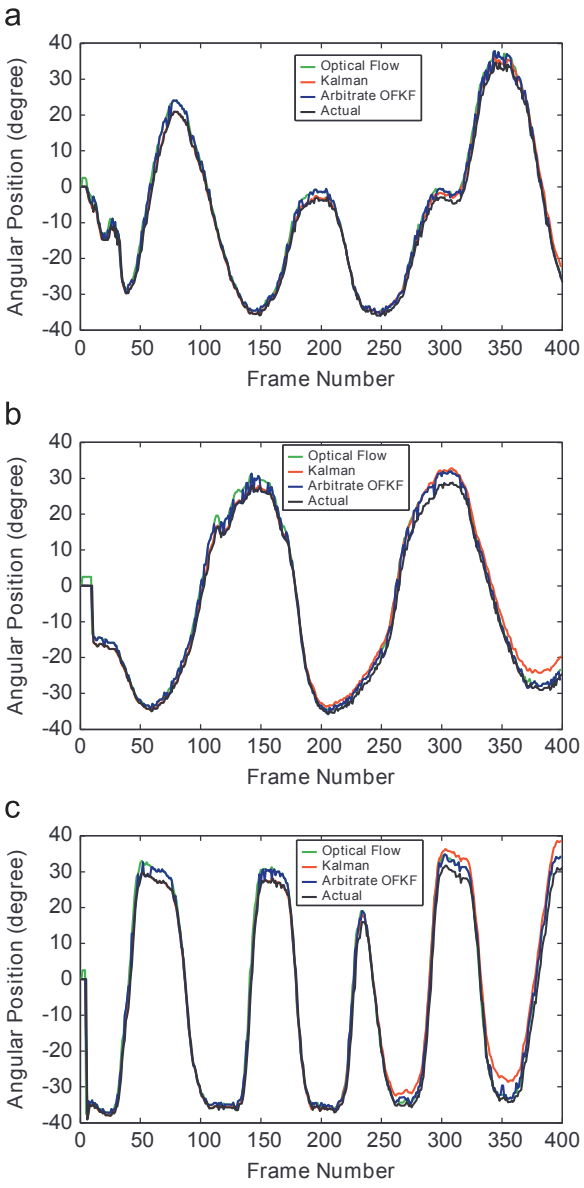

Fig. 8. Angular position of Ground Proof for (a) slow mode, (b) moderate mode and (c) fast mode.

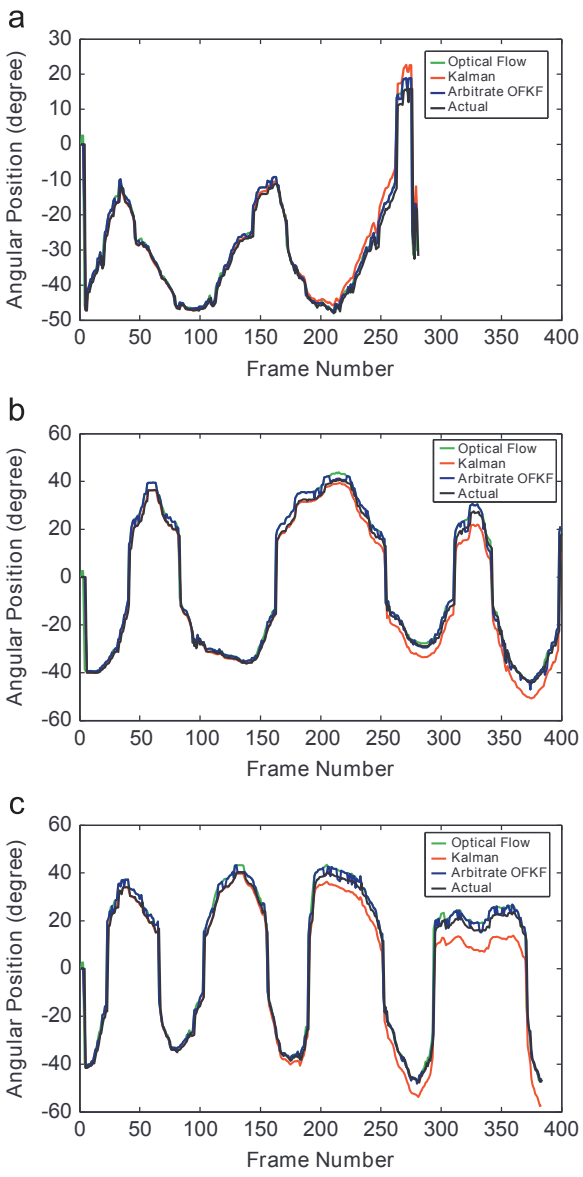

Fig. 9. Angular position of Human Target versus time for (a) slow walking mode, (b) moderate moving mode and (c) fast walking mode.

6. Experimental results

This section compares the results obtained from algorithms using only KF, only OF, and finally the Arbitrate OFKF model developed in this paper. Since we have tracked a single point in the KF, we have used only one feature point in the OF for comparisons. A final comparison is made between Arbitrate OFKF and the particle filter at the end of the section. In the series of the experiments, we have specifically evaluated pan angle values.

6.1. Input datasets

Online data was collected using a Raytheon Infrared Camera operated at 15 Hz frame rate and image size of

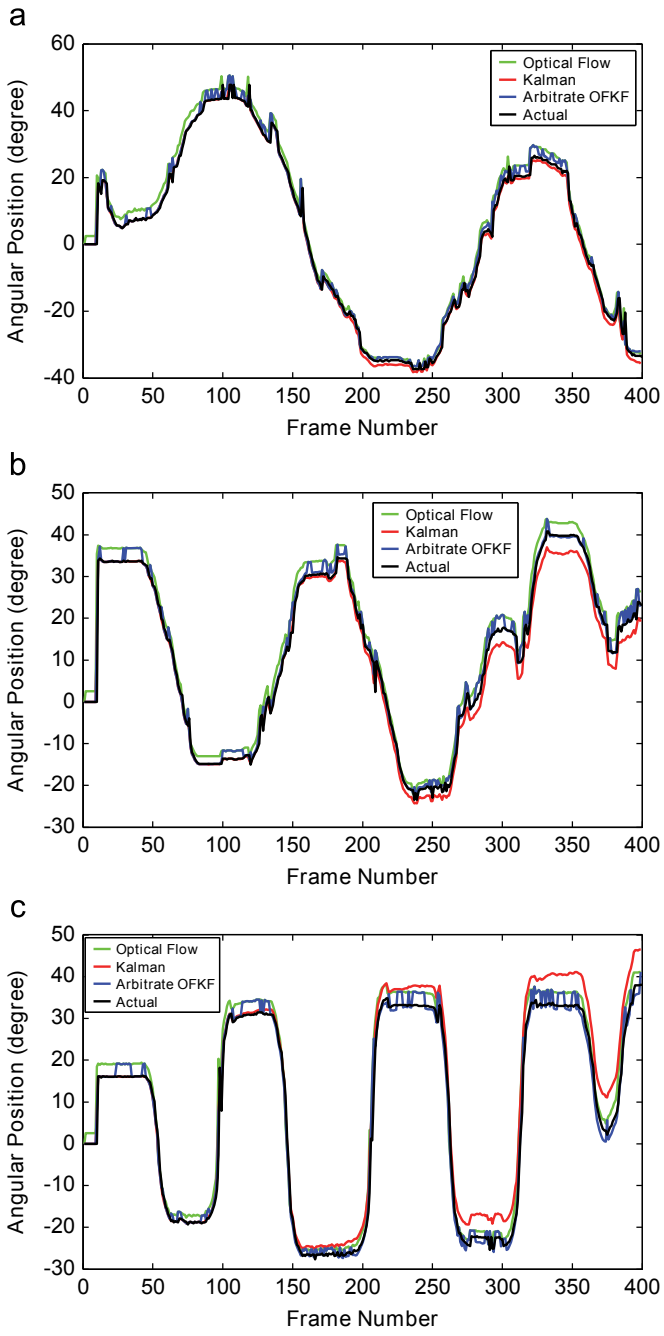

Fig. 10. Angular position of Human Target with a noisy background versus time for (a) slow walking mode, (b) moderate walking mode and (c) fast walking mode.

640 × 480 pixels. The data collection procedure consisted of a target moving in front of an infrared camera, which was mounted on the Active Media robot shown in Fig. 6. Data was collected for the three modes of walking: slow, moderate and fast. In slow walking mode, the target moved slowly in front of the camera without taking any sudden turns. In the moderate walking mode, the target moved with moderate speed in front of the camera taking smooth turns. In the fast walking mode, the target moved with high velocity taking sudden turns during the course of its motion.

Fig. 7a–c is collections of images that show the various target types during their motion. Each of these images is 30 frames apart in the time reference. The white line, as shown in Fig. 7, is the most salient image column, which is our measured tracking location through our algorithms. Instead of showing specific Human Target tracking results, we have evaluated three different target cases. Fig. 7a is a ground proof experiment, which involves a simple target in a noiseless background, Fig. 7b involves a human target walking in a low noise environment and Fig. 7c shows a human target walking in a very noisy environment. The ground proof was examined using a simple hot spot shown in Fig. 7a to make sure whether the experimental condition was under control or not. The noise was simulated using numerous light and heat sources in the area of the experiment to attract the thermal cameras.

In any Kalman filter implementation, the next state prediction depends on the previous state vector. So without correct initialization the predicted values become erroneous. The piecewise constant acceleration model carries the assumption that the initial state vector is equal to zero, i.e. the initial angular position of the target is zero (with respect to Sick) and the initial velocity of the target is zero. Mathematically, $x_0 = \begin{bmatrix} p_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

In optical flow, we don't have to initialize any variable because we are not using an initial state vector.

6.2. Prediction accuracy of OF, KF and arbitrate OFKF

In fact, moderate and slow walking modes, both the algorithms track well when the target moves in one direction, i.e. along a straight line. However, during a

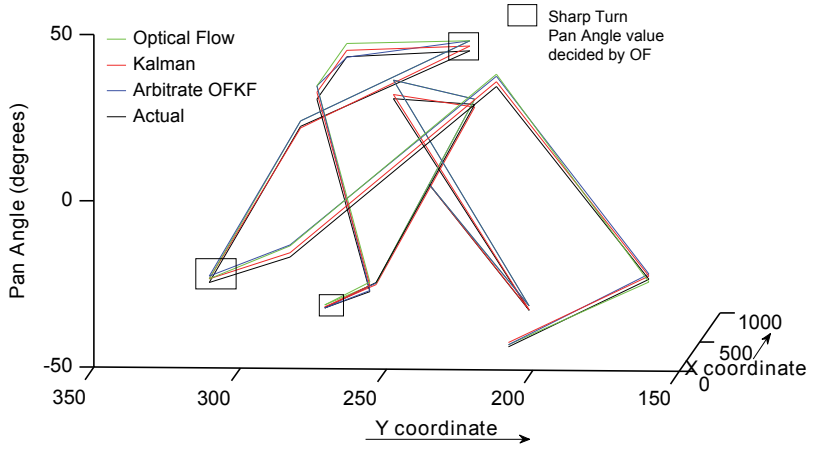

Fig. 11. Pan angle versus (X, Y) coordinate of Human Target in image plane for KF, OF, arbitrate OFKF and Actual.

Table 1
Error analysis of Arbitrate OFKF for (A) Ground Proof, (B) Human Target and (C) Human Target with noisy background.

Filter	Slow walking				Moderate walking				Fast walking			
	Avg. err. (°)	OS (%)	Avg. (°)	Max. (°)	Avg. err. (°)	OS (%)	Avg. (°)	Max. (°)	Avg. err. (°)	OS (%)	Avg. (°)	Max. (°)
(A)												
OF	2.11 ± 1.36	30.30	2.13	5.85	2.11 ± 1.43	33.57	2.09	13.22	2.59 ± 2.75	31.76	2.58	37.58
KF	1.90 ± 1.82	34.78	2.07	6.62	2.63 ± 2.35	37.02	2.99	8.12	2.66 ± 2.71	28.81	3.06	19.57
Arbitrate OFKF	1.69 ± 1.30	26.37	1.76	5.85	2.06 ± 1.69	29.30	2.23	7.80	2.11 ± 1.88	21.23	2.20	19.57
(B)												
OF	2.04 ± 4.15	8.88	1.89	45.98	2.86 ± 4.57	12.77	2.73	41.25	3.31 ± 4.96	15.02	3.33	44.33
KF	1.76 ± 2.44	6.66	1.71	23.80	2.69 ± 3.36	14.07	3.29	20.05	3.64 ± 3.68	20.18	4.50	24.16
Arbitrate OFKF	1.74 ± 3.43	7.98	1.68	40.44	2.34 ± 3.73	11.96	2.52	41.25	2.68 ± 4.24	14.11	3.06	39.42
(C)												
OF	2.53 ± 1.94	19.53	2.49	20.58	2.86 ± 2.15	45.58	2.81	35.50	2.88 ± 2.39	27.30	2.86	23.06
KF	1.00 ± 1.02	13.98	1.40	9.16	1.80 ± 1.71	36.32	2.30	16.89	3.59 ± 3.08	37.50	4.53	11.77
Arbitrate OFKF	1.37 ± 1.63	14.34	1.80	11.10	1.30 ± 1.63	34.66	1.79	16.89	1.70 ± 1.67	24.74	2.14	15.01

walk if the target takes a sudden turn, OF gives a better estimate of angular position of the target as compared to the KF as shown in Figs. 8–10.

The main problem lies in the fact that in KF, the prediction of present state is based on the previous state vector. So when the target takes a sudden turn, the KF has no way to predict using only the previous data. On the contrary, the KF would predict a motion along the previous direction. The OF uses positional coordinates of the target in the previous frame to predict the pan angle. Figs. 8–10 show that optical flow works better than the Kalman filter in regions of sharp turns and zigzag motion.

As shown in Figs. 8–10 the error in angular position at turning points is the highest in the fast walking mode and the lowest in the slow walking mode. When the target walks slowly, the acceleration change on a turn is very small when compared to the same value for a fast walk. When we compared the targets variations among Figs. 8–10, the prediction performance of ground proof object was the best, and the background noise was the worst, although the difference between them was the modest.

The arbitration of either KF and OF will provide a good estimation of the angular position of the target. Depending on the situation we will decide which value should be fed to the robot for efficient tracking. We have used (41) for estimating the required pan angle. The Arbitrate OFKF gives relatively less error in estimation of angular position for all walking mode as shown in Figs. 8–10.

A 3-D plot of pan angle corresponding to the (x, y) coordinate of target in the image plane is shown in Fig. 11. Fig. 11 shows that the pan angle estimated by Arbitrate OFKF closely matches the actual value of the pan angle. We have chosen OF prediction at sharp turning points (shown by rectangular boxes in Fig. 11) and for the rest of the motion we have followed the KF prediction. It is thus evident that the arbitrate OFKF works better than either the KF or the OF individually.

6.3. Prediction error with overshoot

This section presents statistical analysis of the errors in the prediction of pan angle by OF and KF. The error in angular position shows how precisely the algorithm estimates the angular position of the target. A comprehensive table for average error and Overshoot (OS) for OF and KF is shown in Table 1.

We defined the percentage overshoot (OS) as the number of times the error in the estimation of the pan angle exceeded a certain threshold value. Since a large error generally occurs during points where the target turns, the OS can indirectly give us information about turning point prediction accuracy. From Table 1, it is evident that the KF, in general, works better than the OF for all three walking modes. Around turning points, however, the OS of the error is relatively less in the OF than KF.

The percentage OS of the error is drastically improved by the Arbitrate OFKF as shown in Fig. 12. As shown in Table 1, it is clear that Arbitrate OFKF predicts the angular position more accurately than OF and KF for all three modes of walking. The percentage OS of the error is improved by the Arbitrate OFKF as shown in Fig. 12. As shown in Table 1, it is clear that Arbitrate OFKF predicts the angular position much more accurately than OF and KF especially in the moderate and fast modes of walking.

The outcomes of Table 1 Error Analysis shows (A) Ground Proof has reached the best prediction performance, although (C) Human Target with Noise background has not degenerated as expected, compared to (B) Human Target case.

6.4. Execution time

The execution time per iteration was observed for the KF, the OF and the arbitrate OFKF. Since the number of mathematical operations involved per iteration in the KF

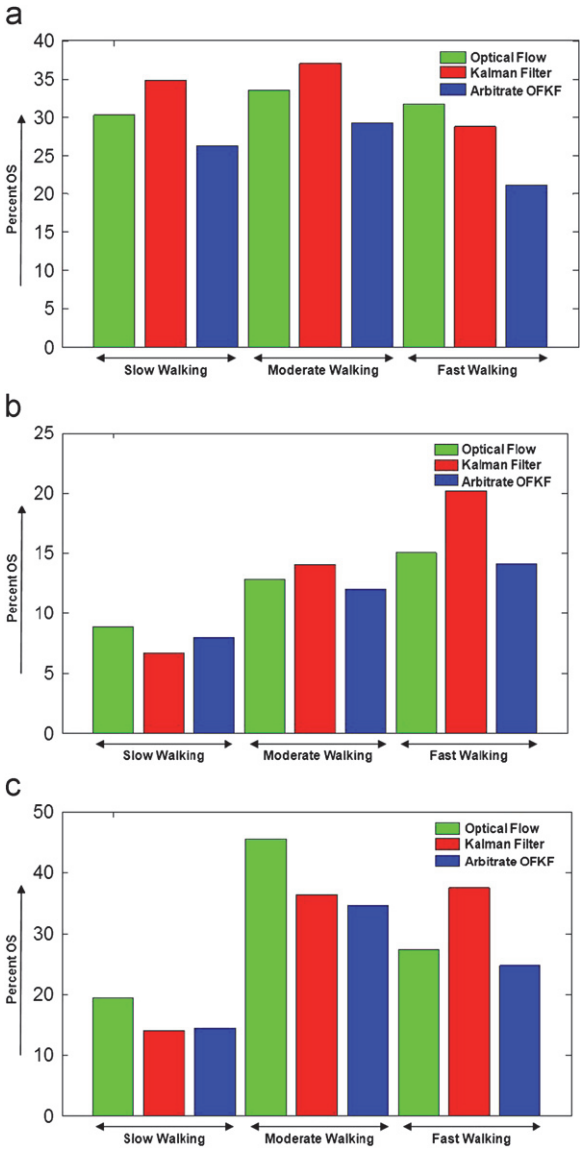

Fig. 12. Comparison of average overshoot angle for the turning periods for the three walking modes for (a) Ground Proof, (b) Human Target and (c) Human Target with noisy background.

algorithm is higher than that required in the OF algorithms, the time required for the OF was experimentally confirmed to be lower than that required for the KF computations. In fact, the OF algorithms took only 58% of the time taken by the KF algorithm. The data was collected for the 6.5 s of motion of the man in front of the infrared camera operated at 15 frames per second, and the corresponding plot of the time required per iteration is shown in Fig. 13.

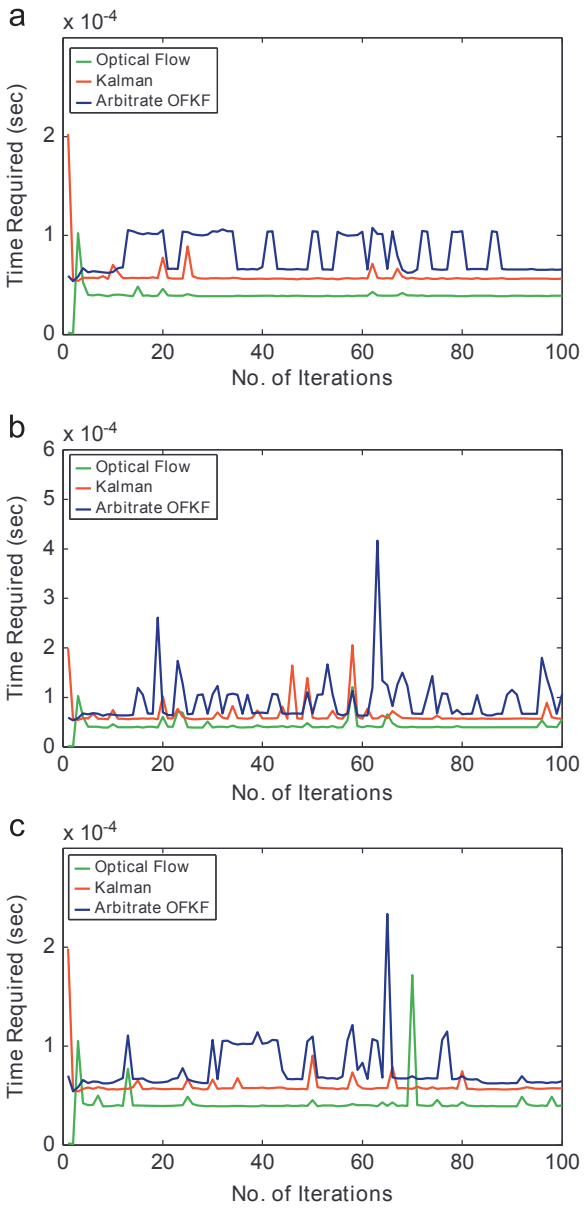

Fig. 13. Computation time required for each iteration versus no. of iteration for (a) Ground Proof, (b) Human Target and (c) Human Target with noisy background.

Regarding time complexity, the arbitrate OFKF will require, on average, 2.5 10^{-5} s more than the KF. It is much less, however, in comparison to the time interval between two consecutive frames, which is equal to 0.067 s, so the time factor won't matter much. However, we gain in accuracy when we use the arbitrate OFKF.

6.5. Performance analysis of arbitrate OFKF versus particle filter

Finally in Table 2, we compared the overall performance of arbitrate OFKF versus particle filter [47] using Human Target in the different speed settings as shown. The particle filter was initialized with 50 particles and its likelihood was measured by a Gaussian weight (Fig. 14).

In all walking modes, the filters had similar performances. The particle filter had less error on average from the measured track; however it also had a higher percent overshoot. Most notably, the particle filter required much more computation time per frame than the arbitrate OFKF algorithm.

7. Conclusion

In this paper, a novel tracking method is proposed, which is able to predict a target position very efficiently even if the target object turns suddenly during its motion. The proposed method is based on an arbitration between OF and KF. It takes into consideration the trajectory of the target motion, and gives a much better result of tracking than individual OF or KF filters. In this paper, attention has been drawn to different scenarios where either the KF works better or the OF does. A comparison has also been made with the particle filter and been shown to have a similar performance with a great decrease in computation time. Our algorithm for the arbitrate OFKF has been successfully tested on our mobile robot in real-time tracking of a man in an indoor lab environment. As a part of future work, we will address the design of an autonomous mobile robot, which can track two or more targets simultaneously. We also feel our system might be made even more robust and efficient by insertion of other sensors resulting in asynchronous and heterogeneous multiple sensory fusions.

Table 2
Comparison of Arbitrate OFKF versus Particle Filter for Human Target.

Filter	Slow walking			Moderate walking			Fast walking		
	Error (°)	OS (%)	Computation time (ms)	Error (°)	OS (%)	Computation time (ms)	Error (°)	OS (%)	Computation time (ms)
Arbitrate OFKF	1.21 ± 3.45	5.56	0.08	1.59 ± 3.14	9.24	0.08	1.83 ± 3.56	11.91	0.09
Particle Filter	1.14 ± 4.40	4.43	1.17	1.39 ± 4.24	11.44	1.16	1.62 ± 4.37	15.67	1.18

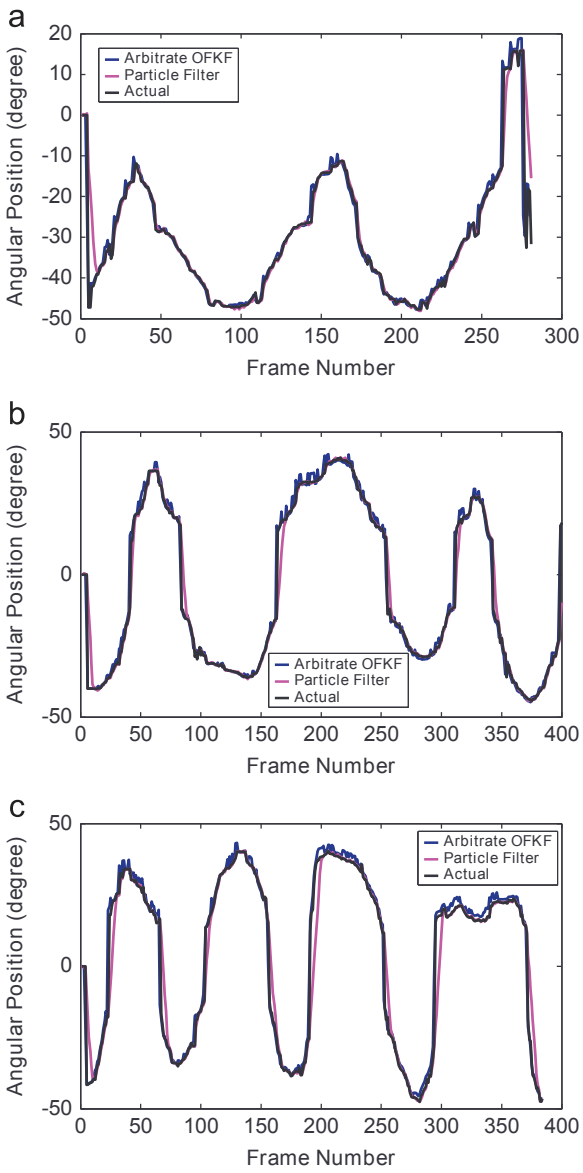

Fig. 14. Comparison of arbitrate OFKF versus a Particle Filter tracking Human Target in (a) slow walking mode, (b) moderate walking mode and (c) fast walking mode.

Acknowledgment

This study was supported in part by the School of Engineering at Virginia Commonwealth University (VCU) through the internship program between Indian Institute of Technology, Kharagpur and VCU. The first author also acknowledges the support of NSF Division of Electrical, Communications and Cyber Systems, CAREER Award #1054333.

References

[1] W. Schulz, D. Burgard, Fox, A.B. Cremers, People tracking with mobile robots using sample-based joint probabilistic data association filters, International Journal of Robotics Research 22 (2003) 99–116.
[2] E. Aguirre, A. Gonzalez, Fuzzy behaviors for mobile robot navigation: design, coordination and fusion, International Journal of Approximate Reasoning 25 (2000) 255–289.
[3] A.M. Zhu, S.X. Yang, Neurofuzzy-based approach to mobile robot navigation in unknown environments, IEEE Transactions on Systems Man and Cybernetics Part C– Applications and Reviews 37 (2007) 610–621.
[4] S.J. Kim, W.K. Choi, H.T. Jeon, Intelligent robot control with personal digital assistants using fuzzy logic and neural network, in: Knowledge-Based Intelligent Information and Engineering Systems, Pt 3, Proceedings, vol. 3215, 2004, pp. 589–595.
[5] J.B. Mbede, P. Ele, C.M. Mveh-Abia, Y. Toure, V. Graefe, S.G. Ma, Intelligent mobile manipulator navigation using adaptive neuro-fuzzy systems, Information Sciences 171 (2005) 447–474.
[6] J.B. Mbede, W. Wei, Q.S. Zhang, Fuzzy and recurrent neural network motion control among dynamic obstacles for robot manipulators, Journal of Intelligent & Robotic Systems 30 (2001) 155–177.
[7] E. Rimon, D.E. Koditschek, Exact robot navigation using artificial potential functions, IEEE Transactions on Robotics and Automation 8 (1992) 501–518.
[8] J. Borenstein, Y. Koren, The vector field histogram– fast obstacle aviodance for mobile robots, IEEE Transactions on Robotics and Automation 7 (1991) 278–288.
[9] R. Polana, R. Nelson, Low level recognition of human motion (or how to get your man without finding his body parts), in: Proceedings of the IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated Objects, Austin, TX, 1994, pp. 77–82.
[10] M. Mikic, E. Trivedi, Hunter, P. Cosman, Human body model acquisition and tracking using voxel data, International Journal of Computer Vision 53 (2003) 199–223.
[11] A.K. Rastogi, B.N. Chatterji, A.K. Ray, Design of a real-time tracking system for fast-moving objects, IETE Journal of Research 43 (1997) 359–369.
[12] X.D. Sun, J. Foote, D. Kimber, B.S. Manjunath, Region of interest extraction and virtual camera control based on panoramic video capturing, IEEE Transactions on Multimedia 7 (2005) 981–990.
[13] D.S. Jang, S.W. Jang, H.I. Choi, 2D human body tracking with structural Kalman filter, Pattern Recognition 35 (2002) 2041–2049.
[14] S. Jung, K. Wohn, Tracking and motion estimation of the articulated object: a hierarchical Kalman filter approach, Real-Time Imaging 3 (1997) 415–432.
[15] X.P. Yun, E.R. Bachmann, Design, implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking, IEEE Transactions on Robotics 22 (2006) 1216–1227.
[16] R. Rosales, S. Sclaroff, A framework for heading-guided recognition of human activity, Computer Vision and Image Understanding 91 (2003) 335–367.
[17] S.G. Wu, L. Hong, Hand tracking in a natural conversational environment by the interacting multiple model and probabilistic data association (IMM-PDA) algorithm, Pattern Recognition 38 (2005) 2143–2158.
[18] Beymer,K. Konolige, Tracking people from a mobile platform, in: IJCAI-2001 Workshop on Reasoning with Uncertainty in Robotics, Seattle, WA, USA, 2001.
[19] D. Schulz, W. Burgard, D. Fox, A.B. Cremers, People tracking with mobile robots using sample-based joint probabilistic data association filters, International Journal of Robotics Research 22 (2) (2003) 99–116.
[20] P. Chakravarty, R. Jarvis, Panoramic vision and laser range finder fusion for multiple person tracking, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Beijing, China, 2006, pp. 2949–2954.
[21] M. Bennewitz, G. Cielniak, W. Burgard, Utilizing learned motion patterns to robustly track persons, in: Proceedings of the Joint IEEE Interntional Workshop on VS-PETS, Nice, France, 2003, pp. 102–109.
[22] Gibson. 1.1., The Perception of the Visual World, Riverside Press, Cambridge, 1950.
[23] N.P. Papanikolopoulos, P.K. Khosla, Adaptive robotic visual tracking: theory and experiments, IEEE Transactions on Automatic Control 38 (3) (1993) 429–445.
[24] S. Nassif, D. Capson, Real-time template matching using cooperative windows, IEEE Canadian Conference on Electrical and Computer Engineering 2 (1997) 391–394.
[25] L. Matthies, T. Kanade, R. Szeliski, Kalman filter-based algorithms for estimating depth from image sequences, International Journal of Computer Vision 3 (1989) 209–236.
[26] J.J. Gibson, On the analysis of change in the optic array, Scandinavian I. Psychology 18 (1977) 161–163.

[27] J. Schilling Robert, Fundamentals of Robotics Analysis and Control, in: B. Horn, B. Schunck (Eds.), "Determining Optical Flow". Artificial Intelligence, vol. 17, Prentice-Hall, Englewood Cliffs, N.J., 1990, pp. 185–203in: B. Horn, B. Schunck (Eds.), "Determining Optical Flow". Artificial Intelligence, vol. 17, Prentice-Hall, Englewood Cliffs, N.J., 1981, pp. 185–203.
[28] B. Horn, B. Schunck, Determining optical flow, Artificial Intelligence 17 (1981) 185–203.
[29] N. Sawasaki, T. Morita, T. Uchiyama, Design and implementation of high-speed visual tracking systems for real-time motion analysis, IEEE International Conference on Pattern Recognition 3 (1996) 478–483.
[30] Kazutoshi Koga, Hidetoshi Miike, Determining optical flow from sequential images, Systems and Computers in Japan 19 (8) (1988) 77–86.
[31] G.D. Hager, P.N. Belhumeur, efficient region tracking with parametric models of geometry and illumination, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (10) (1998) 1025–1039.
[32] A. Blake, R. Curwen, A. Zisserman, A framework for spatiotemporal control in the tracking of visual contours, International Journal of Computer Vision 11 (2) (1993) 127–145.
[33] J.S. Park, J.H. Han, Contour matching: a curvature-based approach, Image and Vision Computing 16 (1998) 181–189.
[34] W.G. Yau, Design and implementation of visual servoing system for realistic air target tracking, Master's thesis, Dept. Elec. Eng., National Taiwan University, 2000.
[35] J. Mataric Maja, The Robotics Primer, MIT Press, Cambridge, 2007, p. 300.
[36] H.A. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 23–38.
[37] P. Liberatore, M. Schaerf, Arbitration (or how to merge knowledge bases), IEEE Transactions on Knowledge and Data Engineering 10 (1998) 76–90.
[38] R.O. Atienza, M.H. Ang, A flexible control architecture for mobile robots: an application for a walking robot, Journal of Intelligent & Robotic Systems 30 (2001) 29–48.
[39] K. Jung, Neural network-based text location in color images, Pattern Recognition Letters 22 (2001) 1503–1515.
[40] T. Kampke, Functional arbitration for multipurpose senso-motory systems, Robotics and Autonomous Systems 27 (1999) 129–150.
[41] M. Kam, X.X. Zhu, P. Kalata, Sensor fusion for mobile robot navigation, Proceedings of the IEEE 85 (1997) 108–119.
[42] R.T. Pack, D. Mitchell Wilkes, K. Kawamura, A software architecture for integrated service robot development, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, vol. 4, 1997, pp. 3774–3779.
[43] H. Himberg, Y. Motai, Head orientation prediction: delta quaternion versus quaternion, IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics 39 (6) (2009) 1382–1392.
[44] H. Himberg, Y. Motai, C. Barrios, R-Adaptive Kalman filtering approach to estimate head orientation for driving simulator, Proceedings of International IEEE Conference on Intelligent Transportation Systems (2006) 851–857.
[45] C. Barrios, H. Himberg, Y. Motai, A. Sadek, Multiple model framework of adaptive extended Kalman filtering for predicting vehicle location, Proceedings of International IEEE Conference on Intelligent Transportation Systems (2006) 1053–1059.
[46] F. Tafazzoli, R. Safabakhsh, Model-based human gait recognition using leg and arm movements, Engineering Applications of Artificial Intelligence 23 (2006) 1237–1246.
[47] M. Isard, A. Blake, CONDENSATION– conditional density propagation for visual tracking, International Journal of Computer Vision 29 (1) (1998) 5–28.
[48] A. Doulamis, N. Matsatsinis, Visual understanding industrial workflows under uncertainty on distributed service oriented architectures, Future Generation Computer Systems, in press.
[49] Y. Motai, Visual-based human–robotic interaction for extracting salient features of an industrial object for an automated assembly system, Special Issue on Machine Vision, International Journal of Computers in Industry 56 (8–9) (2005) 943–957.