A Robust Visual Human Detection Approach with UKF Based Motion Tracking for a Mobile Robot

Meenakshi Gupta, Laxmidhar Behera, Senior Member, IEEE, K. S. Venkatesh, and Mo Jamshidi, Fellow, IEEE

Abstract—Robust tracking of a human in a video sequence is an essential prerequisite to an increasing number of applications where a robot needs to interact with a human user or operates in a human inhabited environment. This paper presents a robust approach that enables a mobile robot to detect and track a human using an on-board RGB-D sensor. Such robots could be used for security, surveillance and assistive robotics applications. Our approach has real-time computation power through a unique combination of new ideas and well-established techniques. In the proposed method, background subtraction is combined with depth segmentation detector and template matching method to initialize the human tracking automatically. We introduce the novel concept of Head and Hand creation based on depth of interest to track the human silhouette in a dynamic environment, when the robot is moving. To make the algorithm robust, we utilize a series of detectors (e.g., height, size, shape) to distinguish target human from other objects. Because of the relatively high computation time of the silhouette matching based method, we define a confidence level which allows us to use the matching based method only where it is imperative. An unscented Kalman filter (UKF) is used to predict the human location in the image frame so as to maintain the continuity of the robot motion. The efficacy of the approach is demonstrated through a real experiment on a mobile robot navigating in an indoor environment.

Index Terms—Human silhouette, Projection histogram, Head and Hand creation, Distance transform, Unscented Kalman filter.

I. INTRODUCTION

Introducing visual tracking capabilities in artificial visual systems is one of the most active research challenges in mobile robotics. Visual tracking of non-rigid object such as a human, is an interesting research field in mobile robotics and has received much attention in recent years because of its potential applications such as security [1], rehabilitation in hospitals [2], guidance in museums [4], assistance in offices [5], and other military applications. In such applications, a mobile robot not only needs to detect the human, but also needs to track it continuously in a dynamic environment where the usual background subtraction could not be used. It is also necessary to be able to give motion commands to the robot at regular intervals in order to maintain a continuous and smooth motion of the robot, even when the image processing may take more time than the permitted interval. In such a case, it would be necessary to predict the human location in the image plane based on an approximate human motion model [6].

The primary sensor used for human tracking in robotic applications is a vision sensor such as a camera [7][8]. Vision is an attractive choice as it facilitates passive sensing of the environment and provides valuable information about the scene that is unavailable through other sensors. Owing to this fact, many algorithms have been developed which detect a human in color images by extracting features such as face [9], skin color [10], cloth color [11], and have been implemented on mobile robotic platforms. Although the algorithms developed using a single feature (e.g., face, skin or cloth color) are computationally effective, fail to detect the human robustly in dynamic environment. For example, the algorithm [12] which uses face detection for tracking a human, fails to detect human when implemented on a mobile robotic platform. As in practical scenarios, when a robot starts tracking a human, the face is actually not available to the robot. Therefore, researchers have started to combine the multiple visual features to make the human detection robust.

Darrell et al. [13] combine multiple visual modalities for real-time person tracking. Depth information is extracted using a dense real-time stereo technique and is used to segment user from the background. Skin color and face detection algorithm are then applied on the segmented regions. Their algorithm assumes that the user will be nearest to the stereo and human face is visible to the robot. Gaverila [14] presented a multi-cue vision system for the real-time detection and tracking of pedestrians from a moving vehicle. The algorithm integrates the consecutive modules such as stereo-based ROI generation, shape-based detection, texture-based classification and stereo-based verification. The algorithm has high computation time as stereo is used for disparity map generation. In literature, the human detection algorithm developed by Navneet et al. [15] is found to be the most robust algorithm. They have used Histograms of Oriented Gradient (HOG) descriptors and a SVM classifier to detect the human. Although the algorithm is robust, its high computation time limits its application for real-time systems. In [16], Liyuan et al. integrates multiple vision models for robust human detection and tracking. They combined the HOG based and stereo-based human detection through mean-shift tracking.

Combining the multiple vision models makes the human detection robust but simultaneously increases the computation cost of the system. To meet the real-time requirements of human tracking, most existing systems employ either a laser sensor or combine the laser sensor information with the color camera information. Woojin et al. [17] proposed the detection
and tracking schemes for human legs by using a single LRF. They derived the common attributes of legs from a large number of sample data and classified them using support vector data description scheme. The scheme proposed by them works under the assumption that the attributes derived by the leg samples does not match with other objects. In [18] Nicola et al. implemented a multisensor data fusion techniques for human tracking with a mobile robot. The approach is based on the recognition of typical leg patterns extracted from laser scans, which are shown to be very discriminative in cluttered environments. Furthermore, faces are detected using the robot’s camera, and the information is fused to the leg’s position using a sequential implementation of UKF. The state model taken by them does not consider the robot motion. Also, the state equations use real world coordinates and thus requires odometry data which is inaccurate. Control action for a robot to follow a human is not defined by the authors in [18]. In [19] Sabeti et al. fuses laser range and color data to train a robot vision system. For each pixel in the robot’s field-of-view, it has color, depth, and surface normal information, which help to extract 3D features. This technique has good detection accuracy, but the speed of the algorithm is far from real-time.

The inclusion of laser sensor increases the robustness of the algorithm but also increases the experimental cost of the system. While both sensing modalities (vision and laser) have advantages and drawbacks, their distinction may become obsolele with the availability of Microsoft Kinect that provide both RGB image and depth (range) data. Hao et al. [20] developed an algorithm for real-time human tracking using color-depth camera. They remove the ground and ceiling planes from the 3-D point cloud input to separate candidate point clusters. The concept of depth of interest (DOI) is used to identify the candidates for detection. A cascade of detectors are used to distinguish humans and objects from possible candidates. The algorithm works robustly in real time but requires the prior knowledge of ground and ceiling plane.

In spite of the advancement made in the the field of human tracking, the vision based human tracking still necessitates an algorithm which is robust, have real-time computation power and does not require any a priori knowledge about the environment. In this paper, we introduce a new, real-time human tracking algorithm which can detect and track target human in dynamic indoor environments, using a mobile robot that is equipped with a color-depth camera. Our algorithm initializes the human tracking automatically and learns the hue histogram of target humans torso and legs. For tracking the human in dynamic environment the hue histograms are back-projected. To make the complete human silhouette, all the blobs found in back-projected image are passed through a Head and Hand creation algorithm which is based on DOI. Afterwards, a series of detectors are used to distinguish between the human and other objects.

As image processing which is computationally expensive is used to detect the human, it is a good choice to use filter for predicting the motion [21]. In this work, UKF is used to predict the position of the human in the image so that the robot can be commanded continuously at regular intervals even when the image processing may take more time than the permissible limit. Unlike [18] which uses a motion model based on odometry information, we propose a model directly based on image information thereby avoiding the dependence on the noisy odometry data.

The major contributions of this paper include:

- The introduction of the new Head and Hand creation concept using color-depth images, allows us to detect and track the complete human silhouette without processing the computationally expensive 3D point cloud data.
- The use of multiple detectors and confidence level allow us to track a complete human silhouette robustly in a complex dynamic indoor environment with real time computation power.
- Unlike other UKF based human tracking techniques, we have proposed a state model which works with image space coordinates directly and obviate the use of odometry data.
- Rather than combining the vision sensor with range sensor (e.g. laser) to make the system robust, our algorithm uses a color-depth camera (Kinect). Therefore, our mobile platform is cost effective.

The concept of Head and Hand creation along with the use of multiple detectors, and confidence level based decision allow us to construct an algorithm for mobile platforms (equipped with color-depth camera), which is capable to detect and track a target human robustly in complex dynamic indoor environment.

The paper has been organized in six sections. Section II and section III describe the proposed scheme for initial detection of the human and for the subsequent tracking, respectively. Section IV describes the modified template matching method for human detection. Section V discusses the affine transformation model used to find out the scaling and translation, human silhouette had in two consecutive images in \(x\) and \(y\) direction, respectively. In Section VI, human position and velocity are modelled in state equation and UKF tracking is discussed. Experimental results are discussed in Section VII and the main conclusions are drawn in Section VIII.

II. INITIAL HUMAN DETECTION UNDER THE STATIC ENVIRONMENT

In our approach, the robot is assumed to be initially static, therefore, background subtraction [22][23] can be applied for the initial human detection. We have used Gaussian mixture model (GMM) for background [23], which gives satisfactory performance in indoor environments. The moving objects are detected by comparing each new frame with it. The resultant background subtracted image, contains background pixels as black and foreground pixels as white. If the number of white pixels in the resultant image is above a certain threshold and do not touch the image boundary, then it is said that the foreground is detected. The detected foreground regions are then morphologically closed and smoothed by a median filter. Area thresholding is then applied to remove the noisy small regions, and holes inside the remaining regions are filled. A RGB-based shadow removal technique [24] is used which
The lower and upper threshold of the aspect ratio of human size detector. The size detector rejects all the blobs which do not make a cluster in the corresponding depth image. If the depth of the 95% of the pixels of a blob lies between ±5% of the depth of the center pixel, then it is said that the blob makes a cluster in the depth image. Now for each blob, a ROI slightly bigger than its bounding rectangle is selected. The ROI is resized to the templates size and template matching algorithm is applied to know whether the blob corresponds to a human or not. Once a foreground blob which represents a human is obtained, segmentation is done along the pronounced valleys in the horizontal projection histogram (HPH) [25]. Fig.1 shows the human detection and segmentation on a test image. Once segmentation is done, the hue histogram of the human torso and the legs are learnt individually. The height of the target human is calculated by taking the difference of the real world Y coordinate of the first and last non-zero pixel of human silhouette while traversing vertically from the top. The real world coordinates (X,Y,Z) of a RGB image pixel (x,y) with respect to the camera origin can be calculated as

$$X = xZ/f, \quad Y = yZ/f, \quad f = \text{focal length of camera} \quad (1)$$

### III. HUMAN DETECTION IN SUBSEQUENT FRAMES UNDER DYNAMIC ENVIRONMENT

Once the robot is moving, the background becomes dynamic and background subtraction technique cannot be applied. To track the human in dynamic environment, hue histogram of the human torso and the legs are back-projected in the hue image of the current frame. The back-projected image contains detected foreground blobs’ pixels as white and the rest as black. After morphological operations, area thresholding is applied on binary motion image. In the binary motion image some other blobs may also get detected, which correspond to the objects having the same hue histogram as that of torso or legs. All the blobs are then passed through a size detector to reject the blobs which do not correspond to an acceptable human body size and aspect.

By back-projecting the hue histograms, entire human silhouette will not appear in the binary image. To get the complete human silhouette, the head and hand creation concept is applied on all the blobs. In this concept, the bounding rectangle of the each blob is resized (1/3rd of the height of the rectangle is added to the upper side and 1/2 of the width of the rectangle is added on both the left and right side). In the upper 2/3rd of the resized rectangle, all the black pixels of the binary image which have approximately same depth as that of the blob center are converted to the white pixels. In the bounding rectangle, all the white pixels of the binary image which do not have comparable depth as that of the blob center are converted to the black pixels. In this way, if a blob corresponds to a human being, the head and hands will appear in the binary image and the background noise presents in the blob will be removed. All the blobs are then passed through a height detector which rejects all the blobs which do not have comparable height (±0.1 m) as that of target human.

All the blobs which are possible candidates for the human being are passed through a shape based detector. Shape based detector consists of two modules, template matching and Leg analysis. If more than one blob is present, then template matching algorithm is applied otherwise leg analysis algorithm is applied. Leg analysis algorithm is faster than template

---

**Fig. 1.** Automatic human detection and segmentation. Image: (a) background (b) RGB (c) background subtracted (d) after shadow removal (e) ROI (f) resized ROI (g) edge (h) distance transformed (i) matched template (j) HPH of foreground blob (k) segmented human (in three parts) (l) hue histogram of human torso (m) hue histogram of human legs.

**Fig. 2.** Flowchart of human detection in dynamic environment.
matching algorithm but can not differentiate distinctively between a human and a non-human blob. Therefore, when multiple blobs are detected, template matching algorithm is applied to differentiate between a human and a non-human blob and when single blob is detected leg analysis algorithm is applied to make detection process faster. These algorithms are discussed in further sections. For applying a template matching algorithm, a ROI which is slightly bigger than the bounding rectangle of the blob is selected and then the ROI is resized to the size of the human template used for the database. A confidence level (CL) is defined, which depends on the occurrence of the “two legs apart pattern” (TLAP) in the blob and the width of the legs found in the blob. When the confidence level is below a certain threshold, then template matching is applied. The next section discusses the template matching process for human detection.

IV. TEMPLATE MATCHING FOR HUMAN DETECTION

This section describes a human detection method based on template matching with DT [26]. Ideally, a template database should cover all possible human shapes and poses, hence it ought to contain an infinite number of templates. That is mainly due to the shape and scale variance. However, in the present work, the human is assumed to always remain in up-right positions and robot is expected to see the human from behind. Therefore, the otherwise vaste set of possible human posture templates is reduced to the templates such as shown in Fig. 5 are considered. The scale variance can be overcome by utilizing the ROI calculated in section III. The resized ROI image is converted to an edge image using a canny edge detector. The edge image is then Distance transformed (DT). In DT

<table>
<thead>
<tr>
<th>TABLE I</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ROUTINE FOR CALCULATING THE WIDTH OF THE HUMAN LEGS</strong></td>
</tr>
<tr>
<td>1: Z distance of the human from the camera</td>
</tr>
<tr>
<td>2: f focal length of the RGB camera</td>
</tr>
<tr>
<td>3: bound bounding rectangle of the human silhouette</td>
</tr>
<tr>
<td>4: initialize flag = 0</td>
</tr>
<tr>
<td>5: initialize j = bound_y + 0.75 bound_height</td>
</tr>
<tr>
<td>6: if TLAP = true then</td>
</tr>
<tr>
<td>7: for i = bound_x to bound_x + bound_width do</td>
</tr>
<tr>
<td>8: if pix[i][j] = 255 ∧ flag=0 then</td>
</tr>
<tr>
<td>9: p1 ← i, flag=1</td>
</tr>
<tr>
<td>10: end if</td>
</tr>
<tr>
<td>11: if pix[i][j] = 0 ∧ flag=1 then</td>
</tr>
<tr>
<td>12: p2 ← i-1, flag=2</td>
</tr>
<tr>
<td>13: end if</td>
</tr>
<tr>
<td>14: if pix[i][j] = 255 ∧ flag=2 then</td>
</tr>
<tr>
<td>15: p3 ← i, flag=3</td>
</tr>
<tr>
<td>16: end if</td>
</tr>
<tr>
<td>17: if pix[i][j] = 0 ∧ flag=3 then</td>
</tr>
<tr>
<td>18: p4 ← i-1</td>
</tr>
<tr>
<td>19: width of leg1 = Z(p2 - p1)/f</td>
</tr>
<tr>
<td>20: width of leg2 = Z(p4 - p3)/f</td>
</tr>
<tr>
<td>21: break</td>
</tr>
<tr>
<td>22: end if</td>
</tr>
<tr>
<td>23: end for</td>
</tr>
<tr>
<td>24: else</td>
</tr>
<tr>
<td>25: for i = bound_x to bound_x + bound_width do</td>
</tr>
<tr>
<td>26: if pix[i][j] = 255 ∧ flag=0 then</td>
</tr>
<tr>
<td>27: p1 ← i, flag=1</td>
</tr>
<tr>
<td>28: end if</td>
</tr>
<tr>
<td>29: if pix[i][j] = 0 ∧ flag=1 then</td>
</tr>
<tr>
<td>30: p2 ← i-1</td>
</tr>
<tr>
<td>31: Combined width of legs = Z(p2 - p1)/f</td>
</tr>
<tr>
<td>32: break</td>
</tr>
<tr>
<td>33: end if</td>
</tr>
<tr>
<td>34: end for</td>
</tr>
<tr>
<td>35: end if</td>
</tr>
</tbody>
</table>

width is calculated at a height of 0.25 of the height of the bounding rectangle of human silhouette from the bottom. If TLAP is detected in the VPH of the human silhouette then the width of both the legs should lie between 10cm to 20 cm. If both the legs are detected together, then the combined width of legs should lies between 15cm to 40cm. The leg analysis algorithm is applied only when a single blob gets detected in the binary image. In a case, where multiple blobs are detected in the binary image or when the confidence level is below a certain threshold, then template matching is applied. The next section discusses the template matching process for human detection.
image each pixel value is proportional to the distance from the nearest boundary pixel. This DT image is then compared with a database of human templates for human detection. Using DT for matching rather than matching with an edge image itself has an advantage of accommodating slight local mismatches. The DT provides a gradual measure of shape dissimilarity.

Matching with DT is shown in Fig. 6. It involves two binary images, a “feature template” $T$ and a “feature image” $I$. The “on” pixels denote the presence of a feature and the “off” pixels the absence of a feature in these binary images. The feature template is given on-line for a particular application, and the feature image is derived from the image of interest by feature extraction. Matching $T$ and $I$ involves computing the DT of the feature image $I$. The template $T$ is positioned over the resulting DT image of $I$. The matching measure $D(T, I)$ is determined by the pixel values of the DT image which lie under the “on” pixels of the template. These pixel values form a distribution of distances of the template features to the nearest features in the image. The smaller these distances are, the better the match between image and template at this location. The chamfer distance (dissimilarity index) is calculated as

$$D_{chamfer}(T, I) = \frac{1}{|T|} \sum d_I(t)$$

(2)

where $|T|$ is the number of pixels in the template image and $d_I(t)$ is the distance between feature $t$ in $T$ and the closest feature in $I$. Thus, the chamfer distance consists of a correlation between $T$ and the distance image of $I$, followed by a division. A template is considered matched, if the dissimilarity index $D_{chamfer}(T, I)$ is below a user supplied threshold $T_{cham}$.

$$D_{chamfer}(T, I) < T_{cham}$$

(3)

The threshold value for the dissimilarity index $D_{chamfer}(T, I)$ should be chosen carefully as too high a threshold value increases the false positives and too low a value of threshold will not be able to absorb slight local mismatches. To get an acceptable value of threshold various human silhouettes were taken from the algorithm and the $D_{chamfer}(T, I)$ was calculated for all the templates in the database as shown in TABLE II. It is to be noted that the threshold value of 2.5 pixel is acceptable for the current implementation. The next section discusses the affine transformation model which is used to measure the scaling and the translation human had in two consecutive frame.

![Fig. 4. Human back posture templates in up-right position](image)

![Fig. 5. Human side posture templates in up-right position](image)

![Fig. 6. Matching using a Distance transform](image)
The general AT is defined by six constants. In this paper, all correspondences found by optical flow in two consecutive images and lie inside the region of human silhouette are considered and a least square method is used to find the six AT constants. It should be noted that optical flow is applied on a ROI, therefore, the time consumed in almost one-third of the time taken by applying optical flow in whole image. 

\( t = (\alpha_2 \delta_5)^T \) gives the translation \((\Delta x, \Delta y)\) along \(x\) and \(y\) direction. \( \mathbf{A} \) is the combined effect of rotation, scaling and shearing. The matrix \( \mathbf{A} \) can be decomposed as

\[
\mathbf{A} = \mathbf{R}(\theta)\mathbf{R}(\phi)\mathbf{D}(\phi) \tag{6}
\]

where \( \mathbf{R}(\alpha) \) and \( \mathbf{D} \) are a planner rotation by \( \alpha \) and a diagonal matrix, respectively:

\[
\mathbf{R}(\alpha) = \begin{pmatrix}
\cos(\alpha) & -\sin(\alpha) \\
\sin(\alpha) & \cos(\alpha)
\end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix}
\lambda_x & 0 \\
0 & \lambda_y
\end{pmatrix} \tag{7}
\]

where \( \lambda_x \) and \( \lambda_y \) are the scaling in \(x\) and \(y\) direction, respectively and can be obtained by calculating singular value decomposition of matrix \( \mathbf{A} \).

### VI. Tracking

Tracking a walking person is one of the most challenging task as human motion will scarcely be of constant velocity or acceleration. Motion with non constant velocity or acceleration is usually modelled by using white noise. In this work, prediction of human motion is desired in the image plane. The human motion is considered to be the superposition of an ideal basic motion (constant velocity) and white noise. It is possible to track all the boundary points of a human silhouette but for the reasons of computational cost, only a single point (ground touching point) is modelled with the UKF. This section describes the methodology used in our tracking system.

#### A. State and observation model

As motion prediction is applied on images, the process state vector be of 2D position and 2D velocity. In real time, the single point observation taken in image space is noisy, therefore, other observations need to be added in the model. The other observations available are the width and height of the bounding rectangle of the human silhouette. Thus, \( S_{k,x}, S_{k,y}, V_{k,x} \) and \( V_{k,y} \) are used for \(x\) and \(y\) position and velocity, respectively, and \( a_k \) and \( b_k \) are used as the width and height of the human silhouette, respectively. The state equation for the human motion (with respect to the camera placed on robotic platform), using white noise constant velocity dynamic model [28] can be written as

\[
\begin{bmatrix}
S_{k+1,x} \\
S_{k+1,y} \\
V_{k+1,x} \\
V_{k+1,y} \\
a_{k+1} \\
b_{k+1}
\end{bmatrix} =
\begin{bmatrix}
1 & \Delta t & 0 & 0 & 0 \\
0 & 1 & 0 & \Delta t & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & \lambda_x \\
0 & 0 & 0 & 0 & \lambda_y
\end{bmatrix}
\begin{bmatrix}
S_{k,x} \\
S_{k,y} \\
V_{k,x} \\
V_{k,y} \\
a_k \\
b_k
\end{bmatrix}
\]

\[
+ \begin{bmatrix}
\Delta t \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\begin{bmatrix}
\dot{u} \\
\dot{v}
\end{bmatrix} + \mathbf{w}_k \tag{8}
\]

where \( w_k \) is the process noise and \( \Delta t \) is time defined as: \( \Delta t = t_k - t_{k-1} \). \( \lambda_x \) and \( \lambda_y \) are 1 in absence of scaling obtained by AT model, \( \dot{u} \) and \( \dot{v} \) are image plane \(x\) and \(y\) velocity of a pixel \((S_{k,x}, S_{k,y})\) due to camera motion. The image feature velocity is related to the camera velocity through the interaction matrix [29] as

\[
\dot{u} = \frac{S_{k,x} - x_0}{Z} - \frac{f^2 + (S_{k,x} - x_0)^2}{f} w \tag{9}
\]

\[
\dot{v} = \frac{S_{k,y} - y_0}{Z} - \frac{(S_{k,x} - x_0)(S_{k,y} - y_0)}{f} w \tag{10}
\]

where \((x_0, y_0)\) is the focus of expansion of the image, \(v\) and \(w\) are the relative translational and rotational velocity of the robot with respect to human. It is to be noted that the mobile robot has 2-degree of freedom and it can move along the \(z\) axis and can rotate about the \(y\) axis only. In this work, a constant translational velocity (i.e. 0.1 m/s) and rotational velocity (0 rad/sec) for the human is assumed and the difference in actual and assumed velocities are taken care by white noise.

Tracking on the image, however allows to measure the position or its projection only. It makes no sense to calculate the velocity out of the difference in position. In this work, two observation models are used and shown below as

\[
\begin{bmatrix}
S_{k,x}^m \\
S_{k,y}^m \\
a_k^m \\
b_k^m
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
S_{k,x} \\
S_{k,y} \\
V_{k,x} \\
V_{k,y} \\
a_k \\
b_k
\end{bmatrix} + \mathbf{e}_1 \tag{11}
\]

\[
\begin{bmatrix}
\Delta x_{k,x}^m \\
\Delta y_{k,y}^m
\end{bmatrix} = \begin{bmatrix}
0 & 0 & \Delta t & 0 & 0 \\
0 & 0 & 0 & \Delta t & 0
\end{bmatrix}
\begin{bmatrix}
S_{k,x} \\
S_{k,y} \\
V_{k,x} \\
V_{k,y} \\
a_k \\
b_k
\end{bmatrix} + \begin{bmatrix}
\Delta t \\
0
\end{bmatrix}
\begin{bmatrix}
\dot{u} \\
\dot{v}
\end{bmatrix} + \mathbf{e}_2 \tag{12}
\]

where \( e_1 \) and \( e_2 \) are the measurement noise with covariance matrix \( \mathbf{R}_{1_k} \) and \( \mathbf{R}_{2_k} \), respectively.

#### B. Determining error covariance matrix

The Unscented Kalman filter works correctly, if

\[
E[w_k w_i^T] = \begin{cases}
Q_k & i = k \\
0 & i \neq k
\end{cases} \tag{13}
\]

\[
E[e_k e_i^T] = \begin{cases}
\mathbf{R}_1 & i = k \\
0 & i \neq k
\end{cases} \tag{14}
\]

\[
E[w_k e_i^T] = 0 \quad \forall i, k \tag{15}
\]

\[
E[e_k e_i^T] = \begin{cases}
\mathbf{R}_2 & i = k \\
0 & i \neq k
\end{cases} \tag{16}
\]

\[
E[w_k e_i^T] = 0 \quad \forall i, k \tag{17}
\]
(acceleration) is defined by a random variable which has a zero auto-correlation for two different time-steps, it is obvious that \( E[w_k w_l^T] = 0 \) for \( i \neq k \). The sequence consisting of \( e_k \) is also a white noise, and hence, two samples at different time steps are uncorrelated. So \( E[e_k e_l^T] = 0 \) for \( i \neq k \). This means a measurement at time \( t_k \) should not have any influence on a later measurement. Further, an acceleration at time \( t_k \) should not have any influence on a later measurement. Both is true, as the human will just act without thinking what acceleration he did before. So \( E[w_k e_l^T] = 0 \) for \( i \neq k \). In a similar way, it can be shown that \( E[e_2 k e_1^T] = 0 \) for \( i \neq k \) and \( E[w_k e_1^T] = 0 \) for \( i \neq k \).

1) Covariance matrix of the Process noise: In the state model, the process noise is taken as a white noise. For any filter model with translational motion of constant velocity and random acceleration (white noise), the process covariance matrix is define as [30]

\[
Q_k = \frac{a^2 \Delta t}{6} \begin{bmatrix}
2I(\Delta t)^2 & 3I\Delta t & 6I
\end{bmatrix}
\]  
(18)

where \( I \) is a \( 2 \times 2 \) identity matrix, \( a \) is the spectral amplitude of the white noise. Usually, \( a \) is defined as a constant value for a fixed interaction area. But, in this work the interaction area is not fixed so \( a \) is multiplied by a variable gain \( G \) and defined as

\[
a = G \times 1 \left( \frac{\text{pixel}}{s^2} \right), \quad G = \frac{f}{\Sigma}.
\]  
(19)

The process noise for the width and height of the bounding rectangle is considered zero, therefore, the noise covariance matrix \( Q_k \) for the real time state model can be written as

\[
Q_k = a^2 \begin{bmatrix}
\Delta x^2 & 0 & 0 & 0 & 0 & 0 \\
0 & \Delta y^2 & 0 & 0 & 0 & 0 \\
0 & 0 & \Delta t & 0 & 0 & 0 \\
0 & 0 & 0 & \Delta t & 0 & 0 \\
0 & 0 & 0 & 0 & \Delta t & 0 \\
0 & 0 & 0 & 0 & 0 & \Delta t
\end{bmatrix}
\]  
(20)

2) Covariance matrix of the Measurement noise: Let the random variables that describes the measurement errors are \( X_{e,x}(t), X_{e,y}(t), X_{e,a}(t) \) and \( X_{e,b}(t) \) for \( x, y, a \) and \( b \), respectively. As there is no dependence on an occurring error of the \( x, y \), \( a \) and \( b \) measurement, therefore, the random variables \( X_{e,x}(t), X_{e,y}(t), X_{e,a}(t) \) and \( X_{e,b}(t) \) are zero-mean and uncorrelated. Assuming that the measurement error appears to be an error of plus and minus two pixel with the same probability, the square of the standard deviation is calculated as,

\[
\sigma^2 = \frac{1}{5}((-2)^2 + (-1)^2 + (0)^2 + (1)^2 + (2)^2) = 2.
\]  
(21)

Usually, the image measurements are not always reliable. Hence, taking a fixed error measurement covariance matrices may detract the filter performance. Since the trade-off between using the prediction or the measurement is controlled by the Kalman gain \( K \), which is inversely proportional to the measurement noise covariance matrix \( R \), one can adaptively adjust the portion contributed from measurement by changing the covariance matrices as

\[
R_{1,k} = \begin{bmatrix}
\sigma_x^2 & \rho_{sx} & 0 & 0 & 0 & 0 \\
\rho_{sx} & \sigma_y^2 & \rho_{sy} & 0 & 0 & 0 \\
0 & \rho_{sy} & \sigma_a^2 & \rho_{sa} & 0 & 0 \\
0 & 0 & \rho_{sa} & \sigma_b^2 & \rho_{sb} & 0 \\
0 & 0 & 0 & \rho_{sb} & \sigma_{\Delta x}^2 & \rho_{\Delta x} \\
0 & 0 & 0 & 0 & \rho_{\Delta x} & \sigma_{\Delta y}^2
\end{bmatrix}
\]  
(22)

\[
R_{2,k} = \begin{bmatrix}
\sigma_{\Delta x}^2 & \rho_{\Delta x}^s & 0 & 0 & 0 & 0 \\
\rho_{\Delta x}^s & \sigma_{\Delta y}^2 & \rho_{\Delta y} & 0 & 0 & 0 \\
0 & \rho_{\Delta y} & \sigma_a^2 & \rho_{\Delta x} & 0 & 0 \\
0 & 0 & \rho_{\Delta x} & \sigma_b^2 & \rho_{\Delta y} & 0 \\
0 & 0 & 0 & \rho_{\Delta x} & \sigma_{\Delta x}^2 & \rho_{\Delta x} \\
0 & 0 & 0 & 0 & \rho_{\Delta y} & \sigma_{\Delta y}^2
\end{bmatrix}
\]  
(23)

where \( \rho_{sx} = \frac{a_k}{a_{k-1}} - s_{x,k} \) and \( \rho_{sy} = \frac{b_k}{b_{k-1}} - s_{y,k} \) are the dissimilarity measure of scaling obtained by image measurement \( (a, b) \) and AT model in \( x \) and \( y \) direction, respectively. The value of \( \sigma_x, \sigma_y, \sigma_a, \sigma_b, \sigma_{\Delta x}, \sigma_{\Delta y} \) are taken as 4.

C. UKF tracker

Kalman filtering is a well-known technique of estimation for robot vision [21]. Among stochastic estimations, the most popular one for nonlinear system is the Extended Kalman Filter (EKF). Although widely used, EKF suffers from the deficiencies including the requirement of sufficient differentiability of the state dynamics, the susceptibility to bias and divergence during the estimation. The Unscented Kalman Filter introduced by Julier et al. [31], provides a derivative-free way to the state parameter estimation of nonlinear systems by introducing the so called “unscented transformation” (UT), while achieving the second-order accuracy (the accuracy of EKF is first order) with the same computational complexity as that of EKF. In contrast with particle filters, the small number of points used by the UKF makes this estimator particularly appealing for real-time applications with limited computational power. Therefore, in the present work UKF is used for state estimation.

Let \( x_k = [S_{k,x}, S_{k,y}, V_{k,x}, V_{k,y}, a_k, b_k] \) be a state vector of dimension \( n_x \) and \( y_k = [S_{k,x}, S_{k,y}, a_k, b_k] \) or \( [\Delta s_{k,x}, \Delta s_{k,y}] \) be a measurement vector of dimension \( n_y \). Let \( x_k = x_0 \) at \( k=0 \). The UKF tracker can be initialized as:

\[
\hat{x}_0 = E[x_0], \quad P_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]
\]

\[
\lambda = \alpha^2 (n_x + \kappa) - n_x, \quad \gamma = \sqrt{(n_x + \lambda)}
\]

\[
W_0^{(m)} = \frac{\lambda}{(n_x + \lambda)}
\]

\[
W_0^{(c)} = \frac{\lambda}{(n_x + \lambda)} + (1 - \alpha^2 + \beta)
\]

\[
W_i^{(m)} = W_i^{(c)} = \frac{1}{2(n_x + \lambda)} \quad i = 1, 2, ..., n_x
\]

where the constant \( \alpha \) determines the spread of the sigma points around \( \hat{x} \) and is usually set to a small positive value. The constant \( \kappa \) is a secondary scaling parameter which is usually set to 0 or 3-\( n_x \) [31]. \( \beta \) is a non-negative weighting term which can be used to incorporate knowledge of the higher order moments of the distribution of \( x \), for a Gaussian distribution, and \( \beta=2 \) is optimal. \( W_i \) is the weight associated with the \( i \)th sigma point. Here, these parameters were set to \( \alpha = 0.9, \beta = 2 \) and \( \kappa = 0 \). Since, in this work two observation model are considered, the correction step of a Kalman filter is performed sequentially as shown in the flowchart 7. The states
are initialized as $x_0$ at $k=0$. The UKF predict the next states. If the measurements are available from the image processing or AT model then the UKF sequentially update the prediction otherwise it keeps on predicting the states. The mobile robot is commanded by two controller as follows,

$$v = \begin{cases} V_{\text{max}} & \text{if} \ y_c \geq y_2 \\ V_{\text{min}} & \text{else if} \ y_2 > Y_c \geq y_1 \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

$$\omega = k_{p1}(\frac{-\phi}{2} + \frac{X_c m}{M} \phi) \quad (30)$$

where $V_{\text{max}}$ and $V_{\text{min}}$ are the maximum and minimum translational speed of the robot, $y_2$ and $y_1$ are user defined threshold for image $y$ coordinate to reduce oscillation in $y$ measurement, $M$ and $N$ are width and height of the image, respectively, $\phi$ is the view angle of the RGB camera, $k_{p1}$ is the proportional gain constant for rotational speed of the robot and is kept less than one and $X_c, Y_c$ are either measured image coordinates or predicted image coordinates.

### VII. Results

To test the performance and the validity of the proposed algorithm in a real time situation, the proposed algorithm has been implemented on the Pioneer P3-DX mobile robot. Microsoft Kinect is mounted on the mobile robot through USB connection. OpenNI is used to access the images from the Kinect camera and OpenCV library is used for image processing. The robot is accessed through Player software. The image size is $480 \times 640$ and template size used for human data set is $240 \times 100$. To prove the effectiveness of the proposed detection and tracking algorithm, several experiments are conducted and are discussed later in this section. The time consumed in various process are shown in Table III. The workspace used for the experiments is of $8m \times 8m$ area, having uniformly distributed illumination.

#### A. Real-time experiments

In experiment I the performance of the proposed human detection algorithm is examined. Experiment II showed that the proposed tracking scheme works robustly even when the detection algorithm fails to detect the human. The performance of the algorithm in complex dynamic environment where multiple humans were present is shown in experiment III. Experiment IV shows the real time experiments conducted to prove the applicability of the UKF predictor.

**TABLE III.** The workspace used for the experiments is of $8m \times 8m$ area, having uniformly distributed illumination.

<table>
<thead>
<tr>
<th>Process</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background modelling</td>
<td>0.056</td>
</tr>
<tr>
<td>Background subtraction</td>
<td>0.001</td>
</tr>
<tr>
<td>Shadow removal</td>
<td>&lt; 0.02</td>
</tr>
<tr>
<td>Template matching (Distance transform)</td>
<td>0.009 - 0.144</td>
</tr>
<tr>
<td>Segmentation (HPH)</td>
<td>0.021</td>
</tr>
<tr>
<td>Back-projection</td>
<td>0.011</td>
</tr>
<tr>
<td>Shape analysis algorithm</td>
<td>0.002</td>
</tr>
<tr>
<td>Head and hand creation</td>
<td>0.037</td>
</tr>
<tr>
<td>Optical flow and &amp; Affine transformation</td>
<td>0.025</td>
</tr>
<tr>
<td>Total time for human detection in dynamic environment</td>
<td>0.240</td>
</tr>
</tbody>
</table>

1) Experiment I: In this experiment, robot is made to remain in static position and different scenarios are created to test the human detection algorithm. Fig. 8 shows the scenario where some part of the background matches with the human color. The background along with the human get detected in the back-projected image and get merged with the human blob as shown in Fig. 8(c). If the gravity center of the blob lies inside the human silhouette, then head and hands creation algorithm rejects all the background pixels detected in the blob based on the depth information and successfully reconstruct the human silhouette as shown in Fig. 8(d). If the blob gravity center lies in the background region then head and hands creation algorithm fails to reconstruct the human silhouette. In such a case, tracking algorithm tracks the human based on the predictor data.

Fig. 9 shows the scenario where two humans having same T-shirt color and different trousers color are present in the environment. Therefore, multiple foreground blobs are detected in the back-projected image. The blob other than desired human silhouette is discarded by the algorithm based on the aspect ratio of the human. Fig. 10 shows the scenario where an object having same color as that of human is present in the environment. The object satisfies the aspect ration criterion of the human, therefore, head and hands creation algorithm is applied on both the blobs. After this, the blob other than the human silhouette is discarded by the algorithm as the height of the object blob is not comparable with the height of human detected in initial detection. Fig. 11 shows the scenario where an object having same color as that of human is present in the environment. The object satisfy the aspect ration criterion of the human and have comparable height as that of human detected in initial detection. The blob other than the human silhouette is discarded by the detection algorithm based on the template matching.

Fig. 12 shows the condition where two human having the same color of T-shirt and trousers are present in the scene. As the height of both the human are comparable to the height of the human calculated in initial detection, the algorithm detects both the human. The desired human is tracked by tracking algorithm based on the predictor data.

To evaluate the performance of the human detection algorithm, two other techniques frequently used in the literature have been implemented for comparison. The first one is the Histograms of Oriented Gradient (HOG) descriptors for human detection proposed by Navneet et al [15]. The proposed
algorithm does not miss the human in the scene but false positive alarm are common. The results shown in [15], proofs that the HOG detects the human robustly but computation time of the algorithm is high which limits its applicability to real time applications. The algorithm proposed by Nicola et. al [32] detects leg patterns in laser scan and faces from the camera image and fuses the information through UKF. However, the leg detection algorithm is fast and works in real time, the face detection using Haar like feature classifier increases the computational time of the algorithm. Even the assumption of human faces the robot while being tracked by robot is impractical. Table IV shows the computational time of our approach, and the approach discussed in [15] and [32]. Fig. 13 shows the results correspond to all three algorithm on a test image.

However, the computational time of our approach is very less, the algorithm requires uniformly distributed illumination. The performance of proposed approach can be further increased by applying HOG descriptors on the region (ROI) detected by the approach. As HOG descriptors will be used for ROI, its computational time will be less as compared to applying HOG descriptors for whole image.
2) **Experiment II**: Fig. 14 shows the environment for experiment II. At $t_0$, the robot detects the human and learns the histogram of his torso and legs. The robot view at $t_0$ and segmentation result are shown in Fig. 15. To test the performance of the tracking algorithm, target human is made to walk on the red trajectory and the other human is made to walk on black trajectory.

At $t_1$, an another human having the same T-shirt and trousers color as that of target human is seen by the robot in the environment as shown in Fig. 16. The height of the other human is comparable to the height of the target human, therefore, human detection algorithm fails to differentiate between the target human and other human. Although the detection algorithm detects both the human as shown in Fig. 16(e), the tracking algorithm continues to track the target human using predictor data to as shown in Fig. 16(f).

At $t_2$, the human detection algorithm fails to detect the human as some part of the background along with the human get detected in the back-projected image as shown in Fig. 17. The robot view at $t_2$ and corresponding depth image are shown in Figs. 17(a) and 17(b), respectively. As shown by back-projected image and its corresponding reconstructed image in Figs. 17(c) and 17(d), the detection algorithm fails to find the human in the image. Fig. 17 shows, although the detection algorithm fails to detect the human but tracking algorithm successfully track the target human, using predictor data to command the robot. At $t_3$, the human is again detected by the detection algorithm and the tracking algorithm commands the robot using image processing data. The robot view at $t_3$, corresponding detection results and external camera view are shown in Fig. 18.

3) **Experiment III**: In this experiment, there were two other humans, robots, chairs and different objects present in the workspace to make a complex environment. Fig. 19 shows the 28 different frames of a test run of the algorithm in complex environment. The human to be tracked is shown in first frame. The detection results of the proposed algorithm and HOG based algorithm are shown in red color boundary and green rectangles, respectively. It is to be noted that the proposed algorithm works robustly in the complex environment. The HOG based human detection algorithm detects all the humans present in the image but the false detection increases when the images contains shadow of the human.

4) **Experiment IV**: In this experiment a human is made to walk in a circle and in a straight line. The UKF predictor predicts the position of human in each iteration and corrected by measurements in each alternative iteration. The actual and predicted trajectory of the human (in image space) while walking in a circle and in a straight line are shown in Fig. 20(a) and Fig. 20(c), respectively. It is to be noted that the trajectory estimated by UKF with adaptive $R$, approximate the actual trajectory of the human fairly. The trajectory of the robot while following the human walking in a circle and in a straight line are shown in Fig. 20(b) and Fig. 20(d), respectively.
This paper has presented a robust approach to detect and track human for intelligent service mobile robots. The proposed approach is a practical solution for real time applications of assistive robots and security robots. It is shown that initialization of human tracking can be done effectively by using background subtraction technique and template matching algorithm. The initialization process learns the hue histogram of human torso and legs and store the height of the human. The proposed approach extract the human silhouette by integrating depth information into back-projected image. A leg analysis algorithm is developed which analyze the human shape by calculating the width of human legs and find the TLAP in the VPH of the detected silhouette. The template matching algorithm used for robust detection of human is applied on a resized ROI image, hence, results in less computational cost to CPU. The UKF predictor is used to predict the position of the human in image space to command the robot, while either image processing data is not available or detection algorithm fails to find the target human. Thus makes the following robot to move in a smooth trajectory. The effectiveness of the algorithm is demonstrated by the several experiments.

VIII. CONCLUSION

REFERENCES


Meenakshi Gupta received her B.E in Electronics & Instrumentation Engineering from University of Rajasthan, India, in 2004 and M.Tech. in Mobile robotics from IIT Kanpur, India, in 2008. She is currently pursuing her Ph.D. in the Department of Electrical Engineering, IIT Kanpur, India. Her research interests include computer vision with applications in machine vision and mobile robotics.

Laxmidhar Behera (S’92-M’03-SM’03) received the B.Sc. and M.Sc. degrees in engineering from NIT Rourkela, India, in 1988 and 1990, respectively, and the Ph.D. degree from the IIT Delhi, India. He pursued his postdoctoral studies in the German National Res earch Center for Information Technology, Sank Augustin, Germany, in 2000-2001. He was an Assistant Professor with the BITS Pilani, India 1995-1999. He has also worked as a Reader with the University of Ulster, Londonderry, UK, and as a Visiting Researcher at Fraunhofer-Gesellschaft, Sankt Augustin, Bonn, and at Eidgenoessische Technische Hochschule Zurich, Switzerland. He is currently a Professor with the Department of Electrical Engineering, IIT Kanpur, India. His research interests include intelligent control, robotics, neural networks, and cognitive modeling.

Venkatesh K. Subramanian obtained his BE degree from Bangalore University, in Electronics in 1987. From 1987 to 1995, he completed his M.Tech (Communication) and PhD (Signal Processing) from IIT Kanpur. He was Assistant Professor in the Electronics and Communication Department at IIT Guwahati from 1995-1999. Since 1999, he is on the faculty of the Electrical Engineering Department at IIT Kanpur. His interests include generalizations of system theory, image/video processing, computer vision with applications in machine vision, visual robot navigation, human computer interfaces, light fields, computational photography and allied areas.

Mo M. Jamshidi (Fellow IEEE, Fellow ASME, A. Fellow-AIAA, Fellow AAS, A. Fellow TWAS, Fellow NYAS, Member HAE) received BS in EE, Oregon State University, Corvallis, OR, USA in June 1967, the MS and Ph.D. degrees in EE from University of Illinois at Urbana-Champaign, IL, USA in June 1969 and February 1971, respectively. He holds honorary doctorate degrees from University of Waterloo, Canada, 2004 and Technical University of Crete, Greece, 2004 and Odlar Youdu University, Baku, Azerbaijan, 1999. Currently, he is the Lutcher Brown Endowed Distinguished Chaired Professor at the University of Texas, San Antonio, TX, USA. He has been an advisor to NASA ( including 1st MARS Mission), USAF Research Laboratory, USDOE and EC/ECU (Brussels). He is the Founding Editor or co-founding editor or Editor-in-Chief of 5 journals including *IEEE Control Systems Magazine* and *IEEE Systems Journal*. He is currently involved in research on system of systems engineering with emphasis on cloud computing, robotics, UAVs, biological and sustainable energy systems, including smart grids and big data analytic.