A Computationally Low Cost Vision Based Tracking Algorithm for Human Following Robot

Muhammad Sarmad Hassan¹, Ali Fahim Khan², Mafaz Wali Khan, and Muhammad Uzair
Institute of Space Technology
Islamabad, Pakistan
e-mail: ¹sarmadhassan@hotmail.com, ²ali.fahim@ist.edu.pk

Abstract—Recently, there has been an increasing interest in the field of human interactive robotics. Contrary to otherwise complex and resource hungry algorithms, we in this work have presented a computationally low cost algorithm for a human following robotic application. Instead of detecting the human, the algorithm makes use of a specific colour tag placed on the human subject which is detected by a camera mounted on the robot. Sensors including range sensor, magnetometer and optical encoders are utilized in tandem to assist the human following process. The method is tested on a custom built robotic platform running Raspberry pi minicomputer. We have performed and presented the results of several experiments for the evaluation of our method.

Keywords—human computer interface; interactive robotics; computer vision

I. INTRODUCTION

Robotic technology has flourished tremendously over time. The capability of a robot to track and follow a mobile human subject is of high interest as this ability can be used in numerous applications such as assisting the disabled, amusing toddlers, photography drones, and military robots, etc. to name a few.

For a robot to interact with a human, it needs a mechanism that enables it to visualize the person and act accordingly. Often, the human detection part of the human following vision algorithm poses difficulty in terms of computational complexity hence resulting in an increase in the cost of hardware capable of running computationally intense routines. There are numerous strategies in the literature to solve the problem of human detection. Few examples include the use of facial recognition [1], laser [2], depth imaging [3], skin detection [4], infrared sensing [5] and radio frequency imaging [6] to do the task.

In this paper, we present a computationally low cost alternative solution to detect and follow a human being based on tag detection and identification using a camera. The tag can be conveniently worn by a human subject which then can be detected and tracked by an autonomously operating robot running computer vision algorithms. Our results show that the proposed method is both computationally less demanding and feasible.

II. METHODOLOGY

In this section we first describe the robotic platform followed by the proposed algorithm.

A. Robotic Platform

The robotic platform is a two wheel differential drive system with an additional rear free moving wheel. Each differential drive wheel is coupled mechanically with an incremental optical shaft encoder to provide motion feedback to the control unit. The base comprises of two acrylic sheets housing the electronic and electrical components including the Arduino Uno board (control unit), power subsystem, actuation controller and magnetometer. A raspberry pi camera along with an ultrasonic ranging module and raspberry pi board (processing unit) is placed on a smaller platform mounted on an adjustable vertical boom.

B. Identification Tag

The core idea of our image processing algorithm is to distinguish on the basis of colours rather than features detection based on shapes and orientation etc. We have exploited a recently proposed identification colour tag [7] utilized in the domain of aerial photogrammetry wherein each tag was used as a marker placed along the parameter of a geographical feature under interest such as a micro lake and land, etc. The tag consists of four different colours in equal proportions and adjacent to each other. What makes the tag unique in its surrounding is the fact that there is very low probability of same colour combination in the picture frame.

C. Frame Processing

The frame processing algorithm works as follows. In each frame of the acquired image, thresholding is carried out to extract each colour of the tag. It is then saved as a separate binary image having the same dimensions as that of original frame. These four individual images undergo morphological region growing operation using a [7×7] square kernel to dilate each thresholded colour and then logical AND operation to yield an image containing the intersection of all the four colours. The centroid of this resultant image is then found out and is taken as the center of the detected colour
tag. It is then subsequently sent to the control unit for tracking. Because, only thresholding and morphological operations form the majority of the image processing pipeline, our vision algorithm is therefore computationally low cost.

D. Human Tracking

The human tracking algorithm requires information including tag center, human-robot distance, magnetometer and optical rotary encoder data. The distance $h$ between the robot and tag is determined using the ultrasonic module which accurately works up to a range of 4m. The robot stops if this distance exceeds 3.5m or either it approaches 0.3m. The robot tries to maintain the minimum distance. The magnetometer and optical rotary encoders are used together to get the orientation and navigational information of the robot. Eq. 1 shows how the angular data obtained from the magnetometer and rotary encoder are combined together in a weighted average fashion.

$$\text{angle}_{\text{Avg}} = 0.3\cdot \text{angle}_{\text{magn}} + 0.7\cdot \text{angle}_{\text{encoder}} \quad (1)$$

The detected center points generated by the frame processing routine are in the form of cartesian coordinates. Let a point $(x_2, y_2)$ be in frame ‘n’ and a point $(x_1, y_1)$ be in frame ‘n-1’. Let the error between the two points be the Euclidean distance

$$e = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2} \quad (2)$$

where the error $e$ is in pixels. The tracking algorithm is therefore a simple closed-loop controller that minimizes $e$ and tries to maintain a minimum distance of $h$. If the robot is unable to detect the tag, it will get in the scanning mode wherein it will move 300° clockwise then 300° anticlockwise. If the target still does not get in view, the robot will start rotating about its position in a clockwise manner until the tag is detected. In the scanning mode, the rotations are made with an angular speed of 60°/s. It is assumed that the moving target has slow dynamics and will therefore move slowly. In an event of erroneous tag detection, the newly detected point will be discarded according to the following criterion:

$$e_{\text{norm}} = e/d \quad (3)$$

where the error $e$ is normalized with respect to the diagonal length $d$ of the frame since the maximum error deviation possible between the two temporally separated points in the successive frames is equal to the diagonal length of the frame. If $e_{\text{norm}} > 2$, then the newly detected point is discarded and the previous point is retained.

III. EXPERIMENTS AND RESULTS

To validate the performance of our robot, different experiments are performed. First, the processing times of different routines are found. Next, the detection algorithm of the tag is evaluated followed by the evaluation of the human following algorithm.

The image processing algorithm is run on a Raspberry Pi 2 model B (900MHz quad-core ARM Cortex-A7 CPU, 1GB RAM) that is connected serially to a camera module. The control logic and sensor interfacing are being addressed by an Arduino Uno controller board. Table I shows the breakdown of the processing times of each task. The total processing time for an image frame is about 312 ms. Our system can process about three frames per second.

<table>
<thead>
<tr>
<th>Processing Type</th>
<th>Time [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Acquisition</td>
<td>48.9</td>
</tr>
<tr>
<td>Image Processing</td>
<td>195.6</td>
</tr>
<tr>
<td>Sensor Acquisition and Processing</td>
<td>68.2</td>
</tr>
<tr>
<td>Overall Time</td>
<td>312.7</td>
</tr>
</tbody>
</table>

All the experiments were performed inside a well lit room with no direct light falling onto the camera’s lens. Moreover, the surface of the floor was kept free of obstacles and was semi smooth. Fig. 1 shows typical scene of experiments wherein the robot is being shown the tag by the human to be followed.

A. Evaluation of Tag Detection

Accurate detection of the tag is of paramount importance for human following. The tag detection can be weakened or in worst scenario, even fail due to numerous factors some of which include distance of the camera from the tag, lateral orientation of the tag, field of view of the camera and lighting conditions, etc.

![Figure 1](image1.png)  
**Figure 1.** A typical experimental scene involving human following.

![Figure 2](image2.png)  
**Figure 2.** Error in the detected center due to the introduction of lateral angular perturbations.
The identification tag is prone to lateral and longitudinal angular perturbations if it experience pitch \( \theta \) or roll \( \phi \) or both. This is very likely to occur because the tag mounted on the person can easily move with the movements of the person. In such a scenario, it is likely that there will be an error in the computed center point of the detected tag. Another source of error in the computed tag center is due to the distance between the tag and the camera.

To quantify the error under study, let us assume that the real center and the detected center of the tag are denoted by \((x_o, y_o)\) and \((x_r, y_r)\) as shown in Fig. 2. Consequently, the error is Euclidean pixel-to-pixel distance between these two points and is given by (4)

\[
error = \sqrt{(y_r - y_o)^2 + (x_r - x_o)^2} \tag{4}
\]

We now attempt to convert this error into a percentage error with respect to the dimensions of the tag to make it independent of the variations of ratio of size of the tag to the size of the frame. Let the Euclidean pixel-to-pixel distance between AB, BC, AC and BD be denoted by \(l_1, l_2, l_3\) and \(l_4\) respectively. The percentage error is then given by

\[
error(\%) = \frac{error}{a l_1 + b l_2 + c l_3 + d l_4} \times 100 \% \tag{5}
\]

where \(a, b = 1/2\sqrt{2} + 2\) and \(c, d = \sqrt{2}/2\sqrt{2} + 2\) are calculated using the distance formula by the 1:1 aspect ratio of the colour tag.

![Figure 3](image)

**Figure 3.** Evaluation of tag detection under varying angle and distance conditions. The red dot near the center is the detected tag center. Each detected dot is slightly offset from the true center of the tag.

The tag was placed at certain distances away from the camera and is given several angular perturbations both in the roll and pitch angles. Fig. 3 shows three instances wherein the calculated center is at an offset to the true tag center. Due to the large number of experimental results, we have calculated and presented error percentages in Table III. As shown in this table, the tag is placed at four increasing distances. For each distance, 36 angular perturbations are applied to the tag. Irrespective of the tag distance from the camera, percentage error increases with the increase in both roll and pitch angles. However, to quantify the variation of percentage error with the increasing distance between the camera and the tag, we have calculated the summation of all the percentage errors for each distance. As expected, the error increases with the increase in the distance. This shows that the percentage error increases with both the increase in angular disturbances and distance between the tag and camera.

**TABLE II.** **PERCENTAGE ERROR DUE TO ANGULAR PERTURBATIONS AND VARYING DISTANCE.**

<table>
<thead>
<tr>
<th>Distance: 38.1 cm</th>
<th>0</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>60</th>
<th>80</th>
<th>(\Sigma)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Error [cm]</strong></td>
<td>0.073</td>
<td>0.065</td>
<td>0.105</td>
<td>0.144</td>
<td>0.108</td>
<td>0.810</td>
<td>1.305</td>
</tr>
<tr>
<td><strong>Avg. Error [cm]</strong></td>
<td>0.590</td>
<td>0.457</td>
<td>0.536</td>
<td>0.560</td>
<td>0.551</td>
<td>0.536</td>
<td>0.536</td>
</tr>
</tbody>
</table>

**B. Evaluation of Human Following**

In this section we have accessed the quality of the human following. In these experiments, a tag was held by a person and was translated along three trajectories as shown in Fig. 4 (a-c). The line in broken black shows these trajectories whereas the solid red shows the path followed by the robot. Table III shows the average and maximum error for each case. These results show that the robot followed the given trajectories quite accurately suggesting that our algorithm may be employed in human following applications.

**TABLE III.** **ERRORS IN HUMAN FOLLOWING EXPERIMENTS**

<table>
<thead>
<tr>
<th>Trial</th>
<th>Avg. Error [cm]</th>
<th>Max. Error [cm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4.1</td>
<td>6.9</td>
</tr>
<tr>
<td>B</td>
<td>3.4</td>
<td>8.6</td>
</tr>
<tr>
<td>C</td>
<td>3.8</td>
<td>15.2</td>
</tr>
</tbody>
</table>
The code for the detection of a colour tag with application to human following robot. The proposed algorithm has been successfully implemented on a custom designed and fabricated robotic platform. In the presence of angular perturbations experienced by the detection tag and increasing distance between the tag and the camera, the results were still satisfactory. We were able to process only 3 frames per second because our code on the embedded device was not optimized. Hence, we can process far greater number of frames per second upon optimizing the code for processing the imaging data. In comparison to other approaches and algorithms described in introduction section wherein laptops and high end computing devices were used, our algorithm on the contrary is computationally lighter and can be implemented on low cost embedded system such as the raspberry pi minicomputer, hence saving room for other algorithms and routines to be used by a typical human following robotic application.

V. FUTURE WORK

There are few areas for potential future research pertaining to our work. First, the robot should be able to follow the human in varying conditions of light. We have performed our experiments only in an indoor setting and moreover, it was ensured that no direct light was incident on the camera lens. This points to a need for a robust technique that can enable the robot to function in such variations of light. For this purpose a camera lens filter can be used that will allow the robot to work in the varying light conditions. To further reduce the processing time, we suggest using an adaptive image processing technique in which rather than using the full frame, only the region of interest can be extracted to reduce the amount of processing. Since the tag represents only a small percentage of the image frame, there is no need to process the entire frame. An adaptive technique can identify and track a feasible area around the identification tag inside the image frame and process the pixels that lie inside this region only. The technique is adaptive since there will be a need to track and adapt the size of this window in each frame intelligently. Processing fewer pixels translates to reduction of the overall processing time. Moreover, with a reduced frame processing time, it will be possible to use a camera with higher image resolution hence leading to further improvement in tag detection and human following.

ACKNOWLEDGEMENT

The authors are grateful to the laboratory technicians especially Mr. Muhammad Waheed, at the Institute of Space Technology who helped in the project and provided valuable technical assistance.

REFERENCES