Proceedings of the 2005 IEEE
International Conference on Robotics and Automation
Barcelona, Spain, April 2005

Person Tracking with a Mobile Robot using Two Uncalibrated Independently Moving Cameras*

Hyukseong Kwon, Youngrock Yoon, Jae Byung Park and Avinash C. Kak
Robot Vision Laboratory
Purdue University
West Lafayette, IN. 47907, U.S.A.
{*hyukseon,yoony,jbpark,kak*}*@purdue.edu*

***Abstract*— This paper presents an efficient person tracking algorithm for a vision-based mobile robot using two independently moving cameras each of which is mounted on its own pan/tilt unit. Without calibrating these cameras, the goal of our proposed method is to estimate the distance to a target appearing in the image sequences captured by the cameras. The main contributions of our approach include: 1) establishing the correspondence between the control inputs to the pan/tilt units and the pixel displacement in the image plane without using the intrinsic parameters of the cameras; and 2) derivation of the distance information from the correspondence between the centers of masses of the segmented color-blobs from the left and the right images without stereo camera calibration. Our proposed approach has been successfully tested on a mobile robot for the task of person following in real environments.**

Index Terms*— *mobile robot, person tracking, person following, 3-D depth estimation, camera calibration.

I. Introduction

Person pursuit with a robot is an important research topic in the machine vision area. It is believed that mobile robots of the future, especially those operating in public places, will be expected to have this skill. Person following obviously requires real-time ability to respond to the changing position of the person being followed.

In this paper, we propose a new person tracking algorithm that enables a vision based mobile robot to pursue a single person in an indoor environment. Our mobile robot employs two independently moving cameras each of which has been equipped with its own motors and controller for a *Pan/Tilt Unit* (*PTU*). There are several advantages to using such a pair of cameras. First of all, it allows the cameras, working together in a stereo mode, to cover a wider angular range of view. Secondly, by continually turning the cameras so that they stay aimed independently at the person being followed, it also minimizes the robot motion which is relatively slower than the camera motion. Thirdly, each of these two independently moving cameras can serve to track two different attributes of the person being followed. On the other hand, using two independently moving cameras makes the calibration process – either dynamic or self-calibration – very difficult. The camera parameters of both cameras would need to be acquired with precision – a difficult task to fulfill in practice.

*This work is supported by the Ford Motor Company

To get around the need for calibration, our tracking system initially builds a Look-Up-Table (LUT) that records the pixel displacement in the image plane induced by a step-wise movement of the PTU associated with each of the cameras. For the next step, using color information of the person's appearance, it calculates centers of masses of the segmented color-blobs in each of the two images that form the conjugate pair of images in a tracking sequence. The current viewing direction of each camera is adjusted so that the center of mass becomes the center of the image frame. Detailed explanation of this procedure is presented in Section IV. Without *a priori* camera calibration step – *i.e.*, without a known focal length f, or the size of the CCD chips, the idea is to estimate the distance information using simple geometry and trigonometry between the binocular camera system and the target person. The feasibility of the proposed method has been tested on our mobile robot. The experimental results of person following in a hallway, discussed in detail in Section VI, attest to the robustness of our approach.

II. Related Works

There are three kinds of papers related to the work reported here. In the first category are papers that deal with the problem of color-histogram-based person tracking in video imagery, without regard to any robot motion or camera control. The second category of papers deals with using a mono camera on a mobile robot for person following. The third category of papers deals with the issue of controlling cameras that are trying to stay aimed in some fashion on the object being tracked.

In the first category is the Pfinder contribution which uses a multi-class statistical blob model for the human figure in which the head and the hands are represented by color distributions [1] for person tracking in video imagery. There is also the contribution by Darrell *et al.* [2] that integrates stereo vision, color, and face detection for the purpose of tracking people in crowded environments.

In the second category, we have the work of Kleinhagenbrock *et al.* [3] in which a single camera and a laser range imaging sensor are used for person following. The laser sensor provides the mobile robot with the distance-to-target information. Our work is different from that of [3] in the sense that we use two cameras that yield the distance-to-target information.

The third category includes the work of Sidenbladh *et al.* [4]. and that of Schlegel *et al.* [5]. Sidenbladh *et al*. have demonstrated a mobile robot engaging in people following behavior. This work uses a color-histogram for representing a person blob in the images. This system requires the camera mounted on the robot to be calibrated. The work of [5] also uses, like our work, two independently moving cameras mounted on a mobile robot. The person-following algorithm used in this work also requires camera calibration parameters. In contrast with both these contributions, our work has no such need.

There are also other noteworthy contributions in the third category that deal specifically with the problem of how to control the two independently moving cameras as the position of the person being followed changes. Marjanovic *et al.* [6] have proposed a hill-climbing algorithm for the learning of the saccadic control inputs to a camera pan/tilt control unit in response to the changing position of the object being tracked. Another contribution is by Lim *et al.* [7] that derives projective rotations corresponding to the different orientations of the camera and thus deduces the control inputs needed for the pan/tilt unit in order to keep the cameras aimed at a target.

With regard to the control of the pointing angle of one or more cameras in response to a moving object, our work presents an alternative to the above mentioned methods reported in [6] and [7]. We believe that our algorithms are simpler and computationally more efficient. We do not require any intricate probabilistic learning scheme for the camera control parameters. Yes, we do "learn" a lookup table at the outset that stores the information between the lateral position of a target blob and the camera inputs needed to keep that blob centered in the image frame. But the learning required, if at all it could be called learning, is very straightforward, as the reader will see from the rest of this paper.

III. Image-based *PTU* Control

In this section, we present an algorithm that establishes correspondences between pixel displacements and camera pan/tilt angles. More specifically, the algorithm will construct a 2-D table (LUT) that shows the degree of pan/tilt angle needed to move an image pixel to the center of the camera image as depicted in Fig. 1.

Recall that our goal is to place the center of a target image blob at the desired position in the image space – the center of the camera image in our case – by moving the camera head. The established LUT can be used to acquire the PTU control information that is needed to move the camera head instantly to the desired position. For example, suppose the image coordinates of the center of the target image blob are acquired. Then, these image coordinates can be used to index into the LUT to read the camera control information that is needed to place the center of the blob at the center of the image. Moving the camera in accordance with the acquired camera control information should instantly move the center of the image blob to the center of the image, or very close to it if the LUT includes errors.

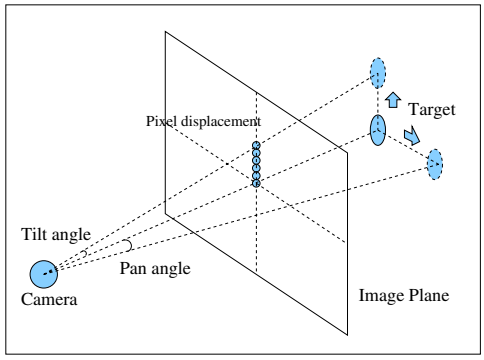

Fig. 1. Establishing the correspondence between the camera pan/tilt angles and the pixel displacement in the image plane

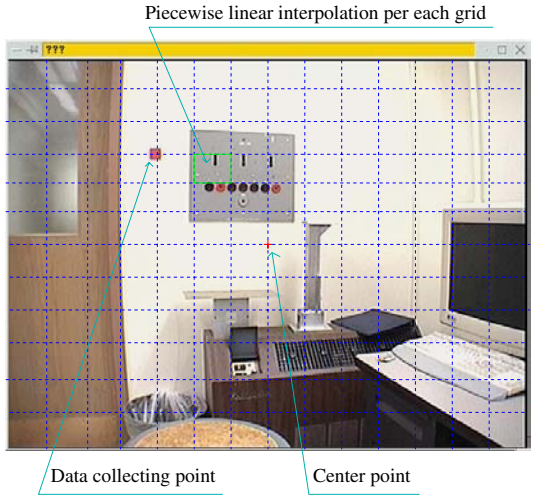

Fig. 2. Data collecting points on an input image

One may think that moving the image blob to the center of the image frame can be done simply by figuring out the proportional relationship between the pan/tilt angles and the pixel displacements. This is not true because the amount of pixel displacement that is induced by a unit pan/tilt motion at the center of the image plane is obviously different at the image border as shown in Fig. 2. Furthermore, because of various lens distortions, the relationship between the pan/tilt motions needed and the horizontal/vertical pixel displacements are different in different parts of an image. We get around these problems by establishing a one-to-one mapping between the displacement at every pixel in the image plane and the needed pan/tilt angle for that displacement. Hence our algorithm can handle any sort of camera lens distortion.

We use an image-based approach to build the 2-D LUT. That means the algorithm establishes the mapping between the pan/tilt angles and pixel displacements only by reading the amount of pixel displacements that result from actual pan/tilt motions of the camera. Since our algorithm is purely image-based, it does not need any camera-specific

information – such as focal length or the size of the CCD chip – which can be acquired only by a precise camera calibration. Since the values of LUT are related only to pan/tilt angles and pixel displacements of the image plane, which means that ours is a purely image-based approach, all the components in LUT are determined irrespective of the distance between the camera and the target. The way to measure the distance-to-target using pan angles through LUT will be discussed in Section IV.

In the rest of this section, we will describe in detail the procedure that is used to build the LUT's. Since each camera in the stereo pair uses the same algorithm to establish its own LUT, we will only present the procedure for one of them.

At first, an image blob is selected for the construction of the LUT. Any single-colored object that has sufficient color-contrast with the background in the scene can be used for this purpose. Compared to a conventional camera calibration procedure that needs a complex calibration pattern and sophisticated pattern recognition algorithm, our proposed algorithm is much more convenient.

After the target blob is selected, we track this target using a color-histogram-based algorithm – detailed description of this algorithm is given in Section V – that constantly gives the image coordinates of the center of mass of the blob.

As the next step, we move the camera so that the center of mass moves to the center of the image and we then read the current pan/tilt angle of the camera for future reference.

Ideally, if the color tracking algorithm gives the exact position of the center of mass of the target blob, we can just incrementally move the PTU and read the pan/tilt angle at each time the center of mass moves one pixel in each direction. However, the center of mass of the target blob may suffer noise-induced displacements that have nothing to do with object motion, it is often difficult to record the correct position of the center of mass. To get around this problem, we define a grid of points for data collection in the image space, as depicted in Fig. 2, and then read the pan/tilt angle at each data collecting point. For each data collecting point, we move the camera until the pixel displacement between the data collecting point and the mean of the center of mass of the blob in 10 image frame interval is less than an empirically chosen threshold, then read the pan/tilt angle of the camera for the point. The difference of this pan/tilt angle to the stored pan/tilt angle at the image center point is stored at the corresponding bin in the LUT. This step iterates until the pan/tilt angles for all data collecting points are acquired. The values for intermediate bins among the data collecting points in the LUT are linearly interpolated from the values of data collecting points in the vicinity. [1] After the whole procedure for one camera is finished, we make the LUT for the other camera using the same procedure. An overview of our proposed algorithm is depicted in Fig. 3.

[1]In our method, 10 pixel interval for both pan and tilt directions is used to define the data collecting grid. The size of this interval was empirically chosen considering the error of the color-histogram-based tracking.

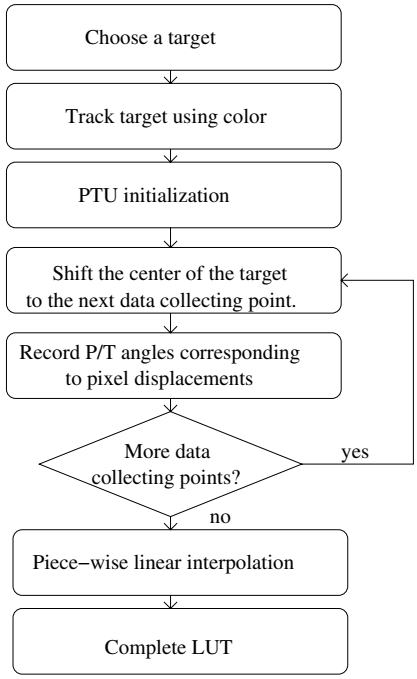

Fig. 3. An overview of our proposed image-based PTU control algorithm

Each LUT obtained in this manner is specific to the camera for which it was acquired. The LUT being specific to a camera frees us from the constraint of always having to use identical cameras on the mobile robot.

IV. Distance Estimation using Two Independently Moving Cameras

The LUTs established in Section III are used for estimating the distance from a single target person to the bisecting point of the baseline - a line connecting the rotating axes of the two cameras. Simultaneously, they determine the next pan/tilt angles for cameras to track a target person in the center of an image plane as shown in Fig. 4(a).

In our method, estimating the distance from the target person to the bisecting point of the baseline is expressed as the modification of the laws of sine. The center of a target (T) and the two rotating axes of stereo cameras (L, R) represent a triangle as depicted in Fig. 4(b). From the triangle – which is represented as $\triangle LRT$ in Fig. 4(b) – $\angle L$ represents the summation of the current pan angle of the left camera and the pan angle that the left camera should turn through in order to move the center of the target to the center of the left image plane. $\angle R$ is defined in the same manner for the right camera. Using the laws of sine, the distances l and r which are the distances from the target to the left and right cameras respectively, and the height h of the $\triangle LRT$ can be calculated as follows:

$$\begin{aligned} l &= w \cdot \frac{\sin R}{\sin(\pi - L - R)} \\ r &= w \cdot \frac{\sin L}{\sin(\pi - L - R)} \\ h &= l \cdot \sin L = r \cdot \sin R \end{aligned} \tag{1}$$

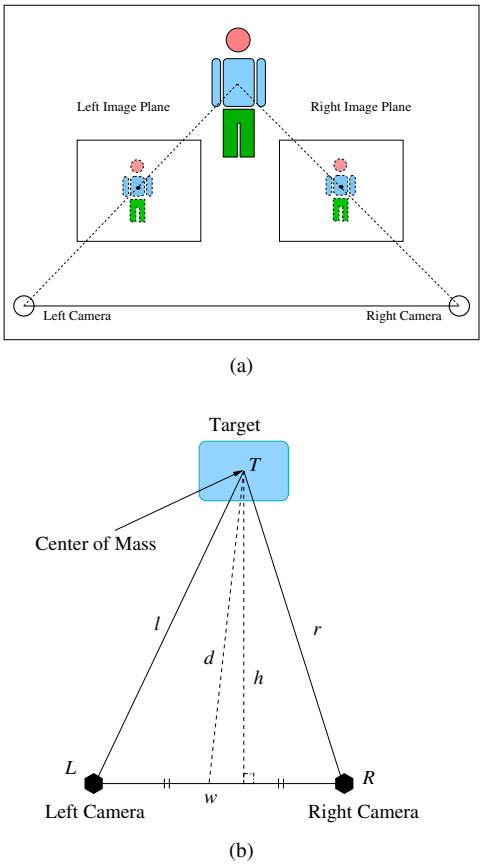

Fig. 4. Distance Estimation (a) The geometry of the stereo cameras, and (b) Calculating the distance from trigonometry in two independently moving cameras (vertices L and R represent the rotating axes of two cameras).

The equation for estimating the distance d from the target to the bisecting point of the baseline is given as follows:

$$\begin{aligned} d &= \sqrt{\left(\tfrac{w}{2} - l \cdot cosL\right)^2 + h^2} \\ &= \sqrt{\left(\tfrac{w}{2} - r \cdot \cos R\right)^2 + h^2} \end{aligned} \quad (2)$$

The reason that we don't use the tilt angles for estimating a distance is that the distance should be expressed as a length parallel to the floor plane because the motion of a wheeled mobile robot has only two degrees of freedom on the floor plane. Therefore, the triangle for estimating a distance can be projected on the floor plane. Since we use independently moving cameras, the tilt and the pan angles become independent, with the additional consequence that the pan and the tilt angles of the two cameras are completely but independently fixed by the the distance-to-target as projected on the horizontal plane (Fig. 5). Obviously, the tilt angle also depends on the vertical location of the center of mass of the target area to be tracked. Basing distance calculations on the horizontal-floor projections of the imaging geometry has other advantages as well. For example, when a robot tracks a person, it has the same estimated projected distance regardless of where we place the tracking window on the person — it can be on the face, on the torso, on the legs, etc. We can even have the two

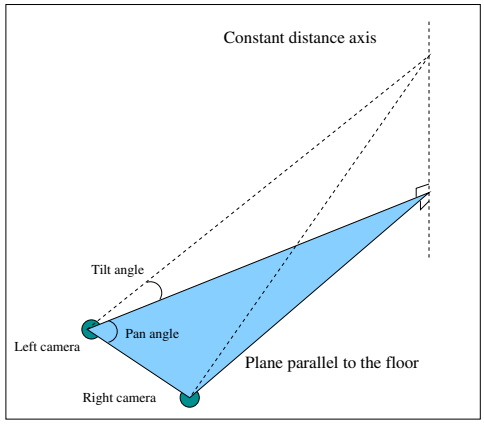

Fig. 5. Triangle projected onto the floor

cameras track different target windows as long as they are on the same vertical axis on the person being tracked.

As a person is being followed with the tracking algorithm, the center of the target is continually displaced to the centers of the image planes by turning the camera heads using appropriate control inputs to the pan/tilt motors as derived from the learned LUT's. Since we always maintain the center of a target at the centers of the two image planes, this also ensures that the target will always stay within the field of view of the cameras.

V. Person Tracking using Color

Object tracking is accomplished with a simple color-histogram based object tracking algorithm. Using color histograms of the normalized R and normalized G components of the normalized RGB space, our tracking system first learns the color distribution of the torso of a person when an initial input image is given. Then, the system segments the blob of the torso in an input image using the learned color distribution and calculates the center of mass of the blob. This center of mass is considered as the position of the person in the input image.

In the learning phase of the tracking algorithm, the operator places a learning window on the person's torso in the initial input image to learn its color distribution. Ideally, this learning window should not include any background pixels. For all the pixels in the learning window, the RGB values are normalized. Subsequently, the tracking system constructs histograms of the pixels from the normalized R and and the normalized G. The range of values of the color component in a histogram that contribute to the color of the target – the person's torso in this case – is chosen by a simple thresholding method. Bins in the histogram that have more than 1% of the learning window size are considered to contribute to the color of the target. This thresholding scheme is reasonable if care is exercised in placing the learning window on the target so that the background pixels are excluded. It is possible that more than one interval of the range in the histogram can correspond to the color values of the target if the person wears multi-colored clothes. For each of the normalized R

and G, the tracking system constructs a corresponding table – we call it a color-look-up-table (CLUT) – whose indices of entries correspond to the value of the color component. Entries of CLUT are set to 1 if their corresponding values of the color component are in the range of that of the target, otherwise 0. CLUTs are used in the tracking phase of our tracking algorithm to test whether a pixel in the input image is that of the target or not. The procedure is described in Fig. 6.

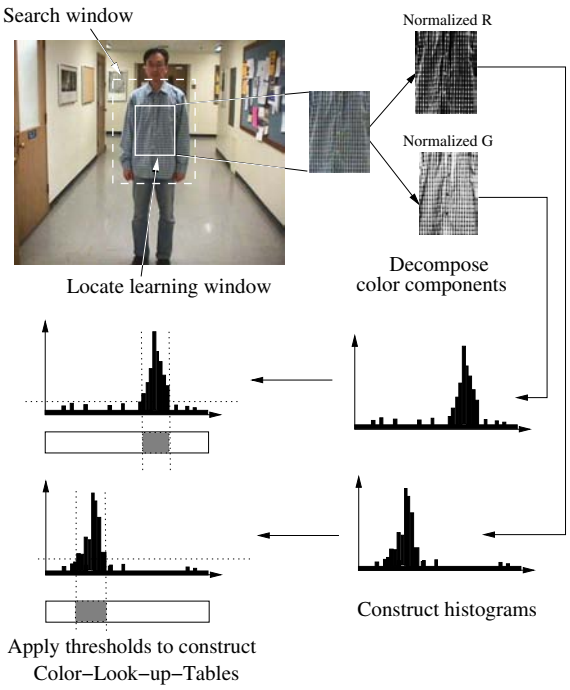

Fig. 6. Learning color distribution of a target person

In the tracking phase, the system uses a search window that is 4 times larger than the learning window – 2 times in each horizontal and vertical direction. For every pixel in the search window, RGB values are normalized and the values of these normalized color components are looked up in the corresponding CLUTs. If all the entries of the CLUTs are 1s, then the pixel is considered to be a pixel of target blob, otherwise it is considered as a background pixel. After all the pixels of the search window are tested, the center of mass of the chosen pixels is calculated. This center of mass is used as the new center of the target blob. After the center of the target blob is calculated, then the location of the search window is moved in a way that its center matches the center of the blob. The tracking system iterates to the next image in the image sequence with the search window placed in the new location.

The tracking algorithm is based on the assumption that the indoor lighting remains substantially constant during the tracking exercise. An additional assumption incorporated in our work is that the color distribution of the clothing worn by the person being followed remains substantially the same as the person twists and turns and presents different aspects of himself/herself to the cameras on the robot.

VI. Experimental Results

The proposed person following algorithm was tested with a mobile robot consisting of custom-built navigational and computing hardware and software installed on a K2A base from Cybermation. The robot is equipped with 450 Mhz Pentium II processor with 784 Mb system memory. Two Sony EVI-D100 cameras (each camera is equipped with its own built-in PTU) are used for person tracking. The two cameras are mounted on the mobile robot platform with 30 cm baseline distance.

We will now report the results of two different experiments conducted on the mobile robot. The first experiment, presented in the next subsection, demonstrates the accuracy of distance-to-target estimation with our approach. The second experiment, presented in Section VI-B shows how effectively the robot can engage in person following behavior.

A. Assessing the Accuracy of Distance-to-Target Estimation

As depicted in Fig. 7(a), we place the mobile robot in a fixed position in the hallway outside our laboratory and have the system determine the distance from the robot to a 30 cm $\times$ 30 cm blue paper sheet target that is placed in front of the robot in a direction perpendicular to the camera baseline. The paper sheet is placed at 7 different distances: 1.0, 1.5, 2.0, 2.5, 3.0 3.5 and 4.0 meters. At each location of the sheet, the system localizes the target using the same color-histogram based approach that is used for object tracking. Subsequently, the system calculates the distance to the target according to the method presented in Section IV and averages the result obtained over 50 image frames. The standard deviation associated with this averaging is also computed. TABLE I presents the statistics thus calculated for the different positions of the target *vis-a-vis* the robot. As shown in the table, the mean of

(a) (b)

Fig. 7. Experimental results (a) Distance accuracy measurement and (b) A mobile robot follows a person in the hallway

the distance error acquires a maximum value of 2.5 cm at the 1.0 m to 2.5 m range for the distance between the robot and the target. As our experiment in the next subsection shows, this is a reasonable error bound for

TABLE I

STATISTICS OF DISTANCE ESTIMATION

Distance to target (ground truth, m)	Statistics of estimated distance		
	Mean (m)	Standard deviation	
		(m)	(%)
1.0	1.0228	0.0067	0.65
1.5	1.4935	0.0125	0.83
2.0	2.0250	0.0216	1.06
2.5	2.5037	0.0302	1.20
3.0	2.9771	0.0334	1.12
3.5	3.4754	0.0643	1.85
4.0	3.9493	0.1052	2.66

person following experiments. The error and the standard deviation get larger with distance from the robot. This, however, is to be expected. We believe this error can be reduced by introducing a longer baseline distance between the cameras.

B. Person Following Experiments

For person following, our goal generally is for the robot to maintain a fixed distance of 1.5 m from the person as he/she is being followed around. A visual perspective on the relative position of the target *vis-a-vis* the robot at this distance between the two is shown in Fig. 7(b).[2] Since it is difficult to measure the actual distance between the robot and a moving person, we recorded video sequences of the robot following a person and visually measured the distance between the two from the videos. The hallway flooring in lab area consists of 22.86 cm × 22.86 cm (9 in × 9 in) tiles. As shown in Fig. 8, this flooring was marked with numbered sheets of paper placed at intervals of five tiles. By examining the positions of these floor marks, we can determine the actual distance between the robot and a moving person. Using these floor marks, we

Fig. 8. Experimental environment for person following

visually identified the locations of the feet of the person being followed and the wheels of the robot. Fig. 9 shows the estimated trace of a target person and the robot. The arrow in this figure represents the direction of motion.

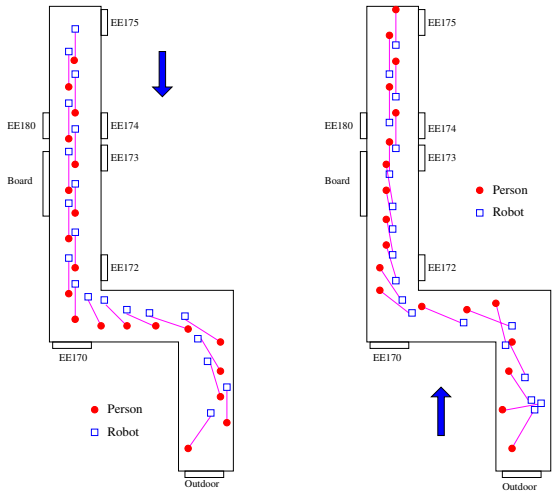

Fig. 9. Trace of the target person and the robot (a) Forward (b) Backward

Against the experimental goal of maintaining a distance of 1.5 m between the robot and the person, the average of the recorded such distances is 166.6920 cm and the standard deviation is 13.0456 cm. The measured distances are depicted in Fig. 10 as a function of the distance traveled down the hall.

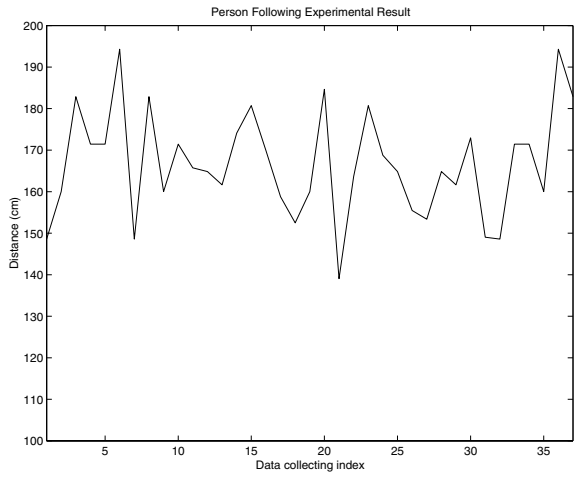

Fig. 10. Estimated distance to the target person while the robot follows the person

With regard to the result depicted in the TABLE I, note that the estimated distance is larger than 1.5 m most of the time during the person following sequence. This is expected because the robot command to follow the person always lags behind the actual distance estimation. The same lag effect also introduces a dependence of the distance estimation error on the speed of movement by the person.

[2]Demo movies of person following are available at the Purdue Robot Vision Laboratory web-site: *http://rvl1.ecn.purdue.edu/RVL/PersonFollowing.*

VII. Discussion

In this paper, we have presented an efficient person following algorithm for a mobile robot using two independently moving cameras. In order to control camera pan/tilt motions, we have presented an image-based PTU control algorithm using a lookup table that stores the correspondences between the camera pan/tilt angles required to keep a target in the center of the image frame and the pixel displacements produced by the target in the image plane. Our approach does not require any information about the cameras such as the focal length and the size of CCD chips. Our work also demonstrates how the aiming angles for the cameras can be used for estimating the distance to the target.

Our experimental results show that the estimation of distance using our method is quite accurate for the purpose at hand and that it is possible to carry out real-time person following in indoor environments.

With regard to the issues that still remain to be resolved, a change in illumination can induce shifts in the center of mass of the blob being tracked in the two camera images. To solve this problem, in the future we will experiment with a new approach to the representation of color that is reported in [8]. This is also the issue of the mobile robot inertia whose effects become particularly pronounced if the person being followed decides to move fast.

References

[1] A. Azarbayejani, C. Wren, and A. Pentland, "Pfinder: Real-time Tracking of the Human Body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780-785, 1997.

[2] T. Darrell, G. Gordon, M. Harville, and J. Woodfill, "Integrated Person Tracking using Stereo, color, and Pattern Detection," *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 601-609, 1998.

[3] M. Kleinehagenbrock, S. Lang, J. Fritsch, F. Lmker, G. A. Fink, and G. Sagerer, "Person Tracking with a Mobile Robot based on Multi-Modal Anchoring," *Proceedings of IEEE International. Workshop on Robot and Human Interactive Communacation*, pp.423-429, 2002

[4] H. Sidenbladh, D. Kragic, and H. I. Christensen, "A Person Following Behaviour for a Mobile Robot," *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 670-675, 1999

[5] C. Schlegel, J. Illmann, and H. Jaberg, "Vision Based Person Tracking with a Mobile Robot," *Proceedings of the 9th British Machine Vision Conference, Southampton*, pp. 418-427, 1998.

[6] Matthew Marjanovic, Brian Scassellati, and Matthew Williamson, "Self-Taught Visually-Guided Pointing for a Humanoid Robot," *4th International Conference on Simulation of Adaptive Behavior*, pp. 35-44, 1996.

[7] Ser-Nam Lim, Ahmed Elgammal and Larry S. Davis, "Image-based Pan-tilt Camera Control in a Multi-Camera Surveillance Environment," *IEEE International Conference on Multimedia and Expo 2003, Special Session on Visual Surveillance*, Jul 6-9, 2003.

[8] Jae Byung Park, "Efficient Color Representation for Image Segmentation under Non-white Illumination," *Proceedings of the International Symposium on Photonics (Technologies for Robotics, Automation and Manufacturing)*, Oct. 2003.