IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) San Diego, California, October 2007

Person Following with a Mobile Robot Using Binocular Feature-Based Tracking

Zhichao Chen and Stanley T. Birchfield
Electrical and Computer Engineering Department
Clemson University
Clemson, South Carolina 29634
Email: {zhichac,stb}@clemson.edu

***Abstract*— We present the Binocular Sparse Feature Segmentation (BSFS) algorithm for vision-based person following with a mobile robot. BSFS uses Lucas-Kanade feature detection and matching in order to determine the location of the person in the image and thereby control the robot. Matching is performed between two images of a stereo pair, as well as between successive video frames. We use the Random Sample Consensus (RANSAC) scheme for segmenting the sparse disparity map and estimating the motion models of the person and background. By fusing motion and stereo information, BSFS handles difficult situations such as dynamic backgrounds, out-of-plane rotation, and similar disparity and/or motion between the person and background. Unlike color-based approaches, the person is not required to wear clothing with a different color from the environment. Our system is able to reliably follow a person in complex dynamic, cluttered environments in real time.**

I. Introduction

The ability to automatically follow a person is a key enabling technology for mobile robots to effectively interact with the surrounding world. Numerous applications would benefit from such a capability, including security robots that detect and follow intruders, interactive robots, and service robots that must follow a person to provide continual assistance [5], [7], [4]. In our lab, we are particularly interested in developing personal digital assistants for medical personnel in hospital environments, providing physicians with ready access to charts, supplies, and patient data. Another related application is that of automating time-and-motion studies for increasing the clinical efficiency in hospitals [1].

Existing approaches to vision-based person following can be classified into three categories. First, the most popular approach is to utilize appearance properties that distinguish the target person from the surrounding environment. For example, Sidenbladh et al. [7] segment the image using binary skin color classification to determine the pixels belonging to the face. Similarly, Tarokh and Ferrari [4] use the clothing color to segment the image, applying statistical tests to the resulting blobs to find the person. Schlegel et al. [6] combine color histograms with an edge-based module to improve robustness at the expense of greater computation. More recently, Kwon et al. [3] use color histograms to locate the person in two images, then triangulate to yield the distance. One limitation of these methods is the requirement that the person wear clothes that have a different color from the background or that the person always face the camera. In addition, lighting changes tend to cause serious problems for color-based techniques.

Other researchers have applied optical flow to the problem. An example of this approach is that of Piaggio et al. [5], in which the optical flow is thresholded to segment the person from the background by assuming that the person moves differently from the background. Chivilò et al. [2] use the optical flow in the center of the image to extract velocity information, which is viewed as a disturbance to be minimized by regulation. These techniques are subject to drift as the person moves about the environment, particularly with out-of-plane rotation, and are therefore limited to short paths.

As a third approach, Beymer and Konolige [10] use dense stereo matching to reconstruct a 2D plan view of the objects in the environment. Odometry information is applied to estimate the motion of the background relative to the robot, which is then used to perform background subtraction in the plan view. The person is detected as the object that remains after the segmentation, and a Kalman filter is applied to maintain the location of the person. One of the complications arising from background subtraction is the difficulty of predicting the movement of the robot due to uneven surfaces, slippage in the wheels, and the lack of synchronization between encoders and cameras.

In this paper we present an approach based upon matching sparse Lucas-Kanade features [13], [14] in a binocular stereo system. The algorithm, which we call Binocular Sparse Feature Segmentation (BSFS), involves detecting and matching feature points both between the stereo pair of images and between successive images of the video sequence. Random Sample Consensus (RANSAC) [15] is applied to the matched points in order to estimate the motion model of the static background. Stereo and motion information are fused in a novel manner in order to segment the independently moving objects from the static background. The person from other moving objects by assuming continuity of depth and motion from the previous frame. The underlying assumption of the BSFS algorithm is modest, namely, that the disparity of the features on the person should not change drastically between successive frames.

Because the entire technique uses only gray-level information and does not attempt to reconstruct a geometric

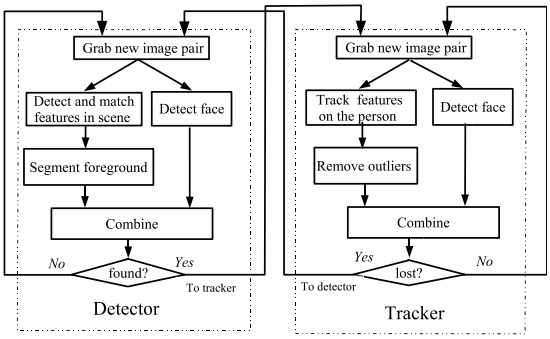

Fig. 1. Overview of the algorithm.

Fig. 2. Left and right images with features overlaid. The size of each square indicates the horizontal disparity of the feature. Since disparity is inversely proportional to depth, smaller squares are farther from the robot.

model of the environment, it does not require the person to wear a distinct color from the background, and it is robust to having other moving objects in the scene. Another advantage of using sparse features is that the stereo system does not need to be calibrated, either internally or externally. The algorithm has been tested in cluttered environments in difficult scenarios such as out-of-plane rotation, multiple moving objects, and similar disparity and motion between the person and the background.

II. Sparse Binocular Feature Algorithm

A. System overview

The system consists of a pair of forward-facing stereo cameras on a mobile robot. The algorithm for processing the binocular video, shown in Figure 1, consists of two modes. In detection mode, sparse features are matched between the two images to yield disparities, from which the segmentation of foreground and background is performed. Once the person is detected the system enters tracking mode, in which the existing features on the person are tracked from frame to frame. Features not deemed to belong to the person are discarded, and once the person has been lost the system returns to detection mode. In both modes the results of the feature algorithm are combined with the output of a face detector, which provides additional robustness when the person is facing the camera.

B. Computing disparity between feature points

Feature points are automatically matched, both between the two stereo images and in one sequence over time, using the Lucas-Kanade approach [12], [13], [14]. In detection mode, the features are matched between the stereo images to compute the disparity between them. At time t, features are selected in the left image I_t^L and matched in the right image I_t^R, after which the resulting features in the right image are matched again in the left image. This left-right consistency check [18] discards features whose initial and final locations are not within a tolerance ϵ_t, thus improving robustness by removing unreliable features. (We set $\epsilon_t = 2$ in our experiments.) The horizontal disparities of the remaining features are stored for the later processing stages. (Since the cameras are approximately aligned, the vertical disparities are nearly zero and therefore ignored.) The result on a pair of images is shown in Figure 2.

C. Segmenting the foreground

Once the features have been reliably matched, those belonging to the person are segmented from other features using both disparity and motion cues. A three-step procedure removes features that are not likely to belong to the person, based upon (1) the known disparity of the person in the previous image frame, (2) the estimated motion of the background, and (3) the computed motion of the person. These three steps are now described.

Let $\mathbf{f}_t = (x_t, y_t, d_t)$ be the image coordinates (x_t, y_t) of a feature along with its disparity d_t at time t. The first step simply discards features for which $|d_t - \tilde{d}_t| > \epsilon_d$, where $\tilde{d}_t$ is the current estimate of the disparity of the person. This estimate is initially obtained from the face detector, as explained below, and is maintained thereafter by the tracker.

The second step estimates the motion of the background by computing a 4×4 projective transformation matrix H between two image frames at times t and $t+1$:

$$\begin{bmatrix} \mathbf{f}_t^i \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{t+1}^i \\ 1 \end{bmatrix} H. \tag{1}$$

At least three points are required to solve for H:

$$H = \left[F_{t+1}^T F_{t+1}\right]^{-1} F_{t+1}^T F_t, \tag{2}$$

where F_t and F_{t+1} are the $4 \times N$ matrices consisting of N features:

$$F_t = \begin{bmatrix} \mathbf{f}_t^1 & \mathbf{f}_t^2 & \cdots & \mathbf{f}_t^N \\ 1 & 1 & \cdots & 1 \end{bmatrix}_{4XN}. \tag{3}$$

Features that fit the resulting motion are determined by

$$\left\|\mathbf{f}_{t+1}^i H - \mathbf{f}_t^i\right\| \leq \epsilon_h, \tag{4}$$

where $\epsilon_h = 1.5$ in our experiments. Due to the possible distraction caused by other moving objects in the scene, along with errors from the tracker and approximation errors in the motion model, the background motion cannot be estimated by simply fitting a model to all the features. Even a robust fitting that discards outliers will not be reliable, because the number of outliers may exceed the number of inliers.

Instead we apply the random sample consensus (RANSAC) algorithm [15] to find small groups of features (containing at least five features) with consistent

motion. We repeatedly select five random features from among the background features (determined by disparity), enforcing a minimum distance between the features to ensure that they are well spaced in the image. From these features we use Equation (2) to calculate the background motion H_b, which is then applied to all the background features to record the number of inliers. This process is repeated several times, and the motion model with the largest number of inliers is taken to be the background motion. Once the background motion has been estimated, the foreground features that do not match this motion model are discarded using Equation (4).

To remove independently moving objects, the third step calculates the motion model H_p of the person using the remaining features. RANSAC is applied, as before, to yield the dominant motion among these features. A second motion model is then determined by applying RANSAC to the features that do not fit the dominant motion model. Two cues are used to distinguish the person from another moving object, namely the size of the group and the proximity to the previous position of the person. Thus, the group that maximizes $\phi_n(s_i) - \phi_m(s_i)$, $i = 1, 2$, where $m(s_i)$ is the mean squared error M_i between the centroid of the ith group and the previous person position, taken in the horizontal direction: $m(s_i) = M_i/(M_1 + M_2)$; and $n(s_i)$ is the relative number of features N_i of the ith group: $n(s_i) = N_i/(N_1 + N_2)$. The remaining features are then projected onto the horizontal axis, and the largest connected component is retained. The bottom right image of Figure 3 shows the final result of the detector.

After these steps, the features that remain are assumed to belong to the person. If the number of features exceeds a threshold, then the person is detected, and the system enters tracking mode. An example of the three steps applied to a portion of the video sequence is shown in Figure 3. The person is typically detected after just two image frames.

An example showing the results of the algorithm when the background has a similar disparity as the person is shown in Figure 4. In this case the disparity test (Step 1) finds almost no features on the background, thus rendering the first two steps ineffective at discarding background features. Step 3, however, uses the motion of the person to correctly discard the features on the background, resulting in features that lie almost entirely on the person. Figure 4 shows that the algorithm does not assume that the person is the closest object. As shown in Figure 5, this procedure is insufficient when the person is not moving, because Step 2 of the algorithm incorrectly discards the features on the person due to the similarity between the motion of the person and background. To solve this problem, the robot simply waits until a sufficient number of features are detected before moving.

D. Face detection

In both the detection and tracking modules, the Viola-Jones frontal face detector [16] is applied at constant intervals. The face detector uses integral images to compute features resembling Haar wavelets, and a cascade architecture is used to enable the algorithm to efficiently evaluate all image locations. This detector is used both to initialize the system and to enhance robustness when the person is facing the camera. In both modes, the face detector is combined with the results of the detection or tracking algorithms by discarding features that lie outside a body bounding box just below the face detection. Note, however, that our system does not require the person to face the camera.

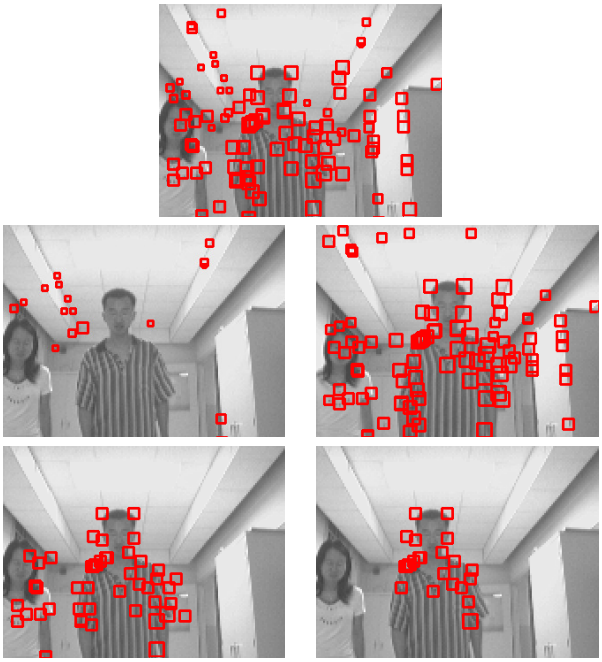

Fig. 3. Step-by-step results of the person detection algorithm. TOP: All the features that pass the consistency check. MIDDLE: The background (left) and foreground (right) features after the disparity test (Step 1). BOTTOM LEFT: The features that remain after removing those that fit the background motion (Step 2). BOTTOM RIGHT: The features that remain after removing those that do not fit the person motion (Step 3).

E. Tracking and camera control

Once the person has been detected, control passes to the tracking module. The person is tracked using the same features by applying Lucas-Kanade from frame to frame. Over time features are lost due to several reasons. Some features are discarded automatically by the Lucas-Kanade algorithm because a sufficient match is not found in the next frame. More commonly, features drift from the person to the background, in particular when the person self-occludes by rotating. To detect this event, features are discarded when their disparity differs from the disparity of the person by more than the threshold ϵ_d. When a significant number of the original features have been lost, the tracker gives up, and control returns to the detection module.

A simple proportional control scheme is applied to the robot motors, i.e., the driving speed σ_d is set to be proportional to the inverse of the disparity d to the person, while the turning speed σ_t is set to be proportional to the product of the horizontal position p of the person in the image and

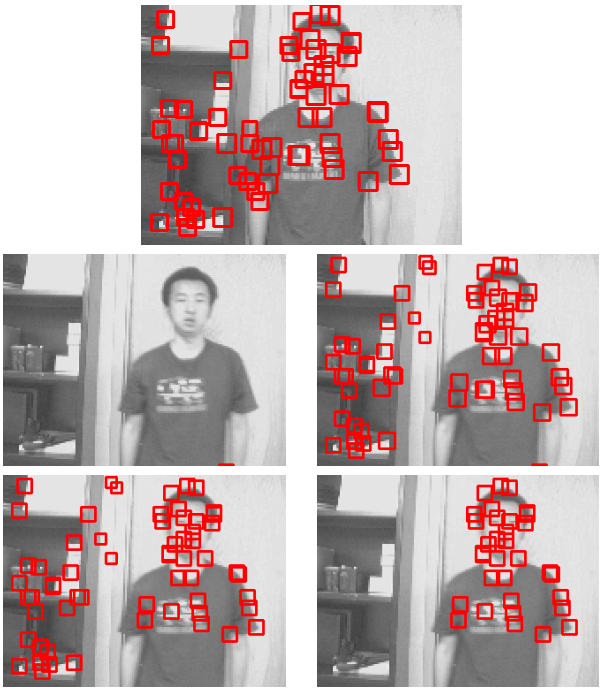

Fig. 4. Step-by-step results when the person and background have similar disparity. The images follow the same order as in Figure 3. The algorithm does not assume that the person is the closest object.

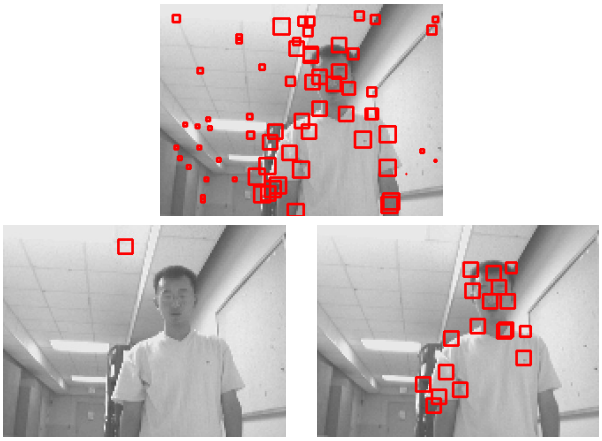

Fig. 5. Two additional examples. TOP: All the features that pass the consistency check. BOTTOM: The final result of the person detection algorithm when the person is still (left) and moving (right).

the disparity, as follows:

$$\sigma_d = K_1 d, \;\; \sigma_t = K_2 p d, \tag{5}$$

where K_1, K_2 are constants (80 and 0.01, respectively). We have found this simple control scheme to be sufficient for our purposes.

III. EXPERIMENTAL RESULTS

The proposed algorithm was implemented in Visual C++ on a Dell Inspiron 700m laptop (1.6 GHz) controlling an ActivMedia Pioneer P3-AT mobile robot. Mounted on a tripod on the robot were two ImagingSource DFK21F04 Firewire cameras with 8.0 mm F1.2 lenses spacing approximately 5.5 cm apart yielding images of size 320×240. Intel's OpenCV library [17] was used for the feature detection, feature tracking, and face detection. The maximum driving speed of the robot was 0.75 meters per second, while the maximum turning speed was 30 degrees per second. The entire system operates at an average of 16 Hz.

The algorithm has been tested extensively in indoor environments and moderately in outdoor environments. Figure 6 shows a typical run of the system, with the robot traveling over 100 meters. To initialize, the person faced the robot, after which the person walked freely at approximately 0.8 m/s, sometimes facing the robot and other times turning away. The environment was challenging because it contained a textured background, other people walking around, some lighting changes, and image saturation due to a bright ceiling light. Some example images from two experiments are shown in Figure 7 to demonstrate the ability of the system to handle an untextured shirt that is the same color as the background, as well as moving objects in the scene.

To further test the robustness of the algorithm, five people were asked to walk around the environment in a serpentine path for approximately five minutes. Two experiments were captured for each person, one at a speed of approximately 0.5 m/s and another at approximately 0.8 m/s. The shirts of the people were white, white, yellow, blue, and textured. Some of the people always faced the robot, some rarely faced the robot, and other faced the robot occasionally. Of the ten trials, the robot succeeded nine times (90% success rate). The only failure was caused by the person walking quickly away from the camera with an untextured shirt. In other experiments, the robot has successfully tracked the person for more than 20 minutes without an error.

We compared our algorithm with a popular color histogram-based algorithm, namely the Camshift technique in OpenCV. The latter was found to be much more likely to be distracted by background or lighting changes than ours. Figure 8 shows a typical run, in which the robot lost the person when he turned around the corner, whereas our algorithm successively followed the person to the end. In this experiment the person intentionally walked in a fairly straight path, and the person wore a shirt whose color was distinct from the background, in order to make the task easier for the color-based algorithm. Some images from a different experiment comparing the two algorithms are shown in Figure 9. It should be noted that more advanced color-based tracking algorithms will also fail whenever the target has a similar appearance to the background.

IV. CONCLUSION AND FUTURE WORK

We have presented a novel algorithm, called Binocular Sparse Feature Segmentation (BSFS), for vision-based mobile robot person following. The algorithm detects and matches feature points between a stereo pair of images and between successive images in the sequence in order to track 3D points in time. Segmentation of the features is accomplished through a RANSAC-based procedure to estimate the motion of each region, coupled with a disparity

Fig. 7. TOP TWO ROWS: Images taken by the left camera of the robot during an experiment in which the person wore a shirt with a similar color to the background. BOTTOM TWO ROWS: Images from an experiment in which another person walked by the camera.

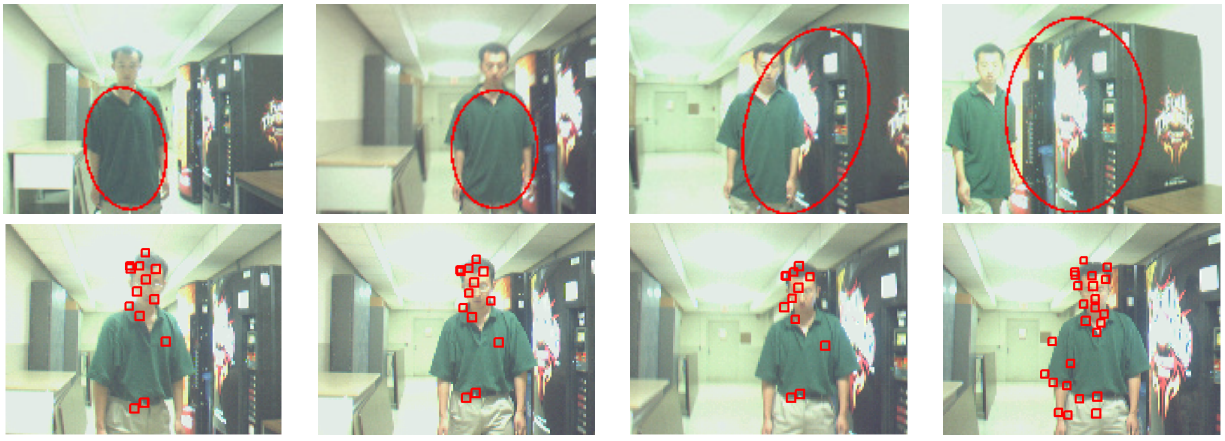

Fig. 9. The results of the color-based algorithm (top) and our algorithm (bottom) in an environment with a textured background. The former loses the person, while the latter succeeds.

test to determine the similarity with the target being tracked. The BSFS algorithm is augmented with the Viola-Jones face detector for initialization and periodic feature pruning.

This system does not require the person to wear a different color from the background, and it can reliably track a person in an office environment, even through doorways, with clutter, and in the presence of other moving objects. However, relying only upon sparse features makes the system subject to distraction by other objects with similar motion and disparity to the person being tracked. More robust performance could be achieved by fusing the information used by this algorithm with additional appearance-based

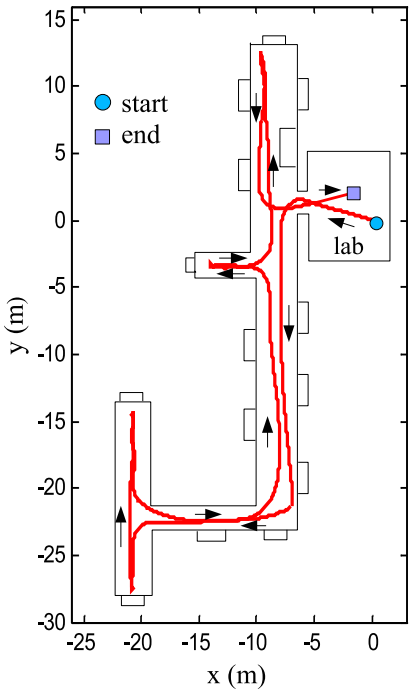

Fig. 6. The robot path as it followed the person through the hallways of our laboratory in an experiment.

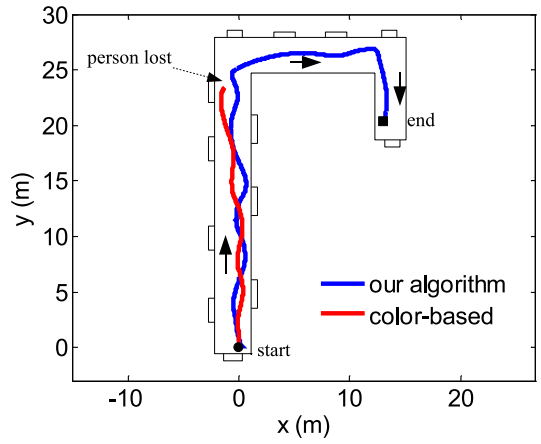

Fig. 8. Comparison with color-histogram-based tracking. The color-based algorithm lost the person when he turned down the hallway, whereas our algorithm succeeded.

information such as a template or other measure of image intensities or colors. Another limitation of the present system is its inability to handle the situation when the person leaves the field of view of the camera, or when another object completely occludes the person, in which case the robot tends to fixate on the distracting object. A pan-tilt camera or a wider field of view would overcome this problem. Finally, future should should be aimed at incorporating the proposed technique with other sensor modalities such as color information, infrared data, or range sensors to increase robustness [19].

Acknowledgments

This work was supported by a Ph.D. fellowship from the National Institute for Medical Informatics.

References

[1] R. Barnes. *Motion and Time Study: Design and Measurement of Work*. John Wiley and Sons Inc, 7th edition, 1980.

[2] G. Chivilò, F. Mezzaro, A. Sgorbissa, and R. Zaccaria. Follow-the-leader behaviour through optical flow minimization, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[3] H. Kwon, Y. Yoon, J. B. Park, and A. C. Kak. Person tracking with a mobile robot using two uncalibrated independently moving cameras. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[4] M.Tarokh and P. Ferrari. Robotic person following using fuzzy control and image segmentation. *Journal of Robotic Systems*, 20(9), 2003.

[5] M. Piaggio, P. Fornaro, A. Piombo, L. Sanna, and R. Zaccaria. An optical-flow person following behaviour. In *Proceedings of the IEEE ISIC/CIRNISAS Joint Conference*, 1998.

[6] C. Schlegel, J. Illmann, H. Jaberg, M. Schuster, and R. Worz. Vision based person tracking with a mobile robot. In *The British Machine Vision Conference*, 1998.

[7] H. Sidenbladh, D. Kragik, and H. I. Christensen. A person following behaviour of a mobile robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, 1999.

[8] J. Shin, S. Kim, S. Kang, Seong-Won Lee, J. Paik, B. Abidi, M. Abidi, Optical flow-based real-time object tracking using non-prior training active feature mode. *ELSEVIER Real-Time Imaging*, 11: 204-218, 2005.

[9] M. Agrawal, K. Konolige and L. Iocchi. Real-time detection of independent motion using stereo. *IEEE workshop on Motion*, 2005.

[10] D. Beymer and K. Konolige. Tracking people from a mobile platform, *International Joint Conferences on Artificial Intelligence*, 2001.

[11] D. Burschka, J. Geiman, and G. Hager. Optimal landmark configuration for vision-based control of mobile robots. In *Proceedings of the IEEE Conference on Robotics and Automation*, 2003.

[12] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[13] J. Shi and C. Tomasi. Good features to track. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593-600, 1994.

[14] C. Tomasi and T. Kanade. Detection and tracking of point features, *Technical report CMU-CS-91-132, Carnegie Mellon University*, 1991.

[15] M. A. Fischler, R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24: 381-395, 1981.

[16] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.

[17] `http://sourceforge.net/projects/opencvlibrary`

[18] P. Fua, A parallel stereo algorithm that produces dense depth maps and preserves image features, *Machine Vision and Applications*, 6(1): 35–49, 1993.

[19] T. Miyashita, M. Shiomi, and H. Ishiguro. Multisensor-based human tracking behaviors with Markov chain Monte Carlo methods. *Proceedings of IEEE-RAS/RSJ International Conference on Humanoid Robots*, 2004.