A Vision based Robot Navigation and Human Tracking for Social Robotics

Genci Capi, Hideki Toda, and Takuya Nagasaki *Member, IEEE*

Abstract— In this paper, we introduce a new vision based method for robot navigation and human tracking. For robot navigation, we convert the captured image in a binary one, which after the partition is used as the input of the neural controller. The neural control system, which maps the visual information to motor commands, is evolved online using real robots. For human tracking, after face detection, the color of human's clothing is extracted. Then the robot starts tracking the human in the environment. The performance of proposed method is evaluated using the TateRob mobile robot. The experimental results and analysis are presented.

I. Introduction

ROBOT navigation and people tracking are important in order to develop robots that can interact with humans. Navigation in everyday life environments poses a significant challenge because of their complexity and inherent uncertainty.

Robot navigation in human environments has been previously investigated from a number of different approaches. In most of these methods, the robots utilize the distance, ultrasonic, or laser sensors to navigate in the environment. However, the main drawback of the sonar or laser sensors lies in the fact that one sensor is required for one distance measurement, that is, in order to obtain a complete picture of the environment around the robot, a number of sensors must be used. Moreover, to achieve the accuracy in detection, they will have to be placed perpendicular to the target.

Recently, vision based robot navigation has attracted many researchers. Variations have included using sensory input from stereo vision, monocular vision, and the combination of vision with other sensors. Methods also vary in how they deal with temporal in formation, from using individual frames exclusively [1] to computing optical flow fields from multiple frames. Domains include road and off-road travel [2, 3, 4, 5, 6, 7] and indoor robotic navigation [1, 8, 9, 10].

Neural networks (NNs) had also been used by some other researchers for solving the said problem. However, the performance of an NN depends on its architecture and connecting synaptic weights, optimal selection of which is a tedious job. A variety of tools based on supervised and reinforcement learning algorithms had been used by a few investigators for this purpose.

Unlike previous works, in the experiments reported here, we consider evolution of neural controllers for robot navigation in unstructured environments. In our method, we convert the captured image in a binary one, which after the partition is used as the input of the neural controller. In difference from previous works that tested their motion planning algorithms through computer simulations, we evolved the neural controllers online using TateRob platform.

Tracking people in dynamic and changing environments is very important for socially robots. Tracking people using a camera mounted on a robot is challenging because the conditions are not fixed and require adaptive methods to deal with the changes in the scene.

There has been much work in the area of people tracking using computer vision techniques. Several real-time systems have been presented that work reliably under a few established conditions that do not change over time. Color-based tracking with a static background is the basis of the widely used people-tracking system, Pfinder [11]. Several other systems have also concentrated on similar color metrics for tracking faces in real-time [12, 13, 14]. The increase in computational power has made it possible to further extend the tracking methods to allow for multiple cameras [15] and combination of other visual cues like motion and stereo [16]. This has resulted in a variety of real-time systems that have been used by thousands of people [17, 18, 19, 20] at various exhibitions. There has also been much progress in locating people in a scene using methods to look for faces [21, 22, 23]. These methods, though much more reliable than the color-tracking methods, are computationally quite expensive making them somewhat infeasible for real-time systems. These methods require higher resolution images than the color-tracking methods. Some attempts have been made to incorporate these methods with the real-time systems to aid in initialization and registration [17]. Unfortunately, most of the scenarios in which we would like these machines to operate in are very dynamic and unconstrained. In this paper, we present methods to combine color, motion, and depth cues for robust tracking of users under conditions of moving cameras, changing backgrounds, varying lighting conditions.

Authors are with the University of Toyama, Gofuku 3910, 930-8555 Japan (e-mail: capi@ eng.u-toyama.ac.jp).

978-1-4244-7148-5/10/$26.00 ©2010 IEEE

II. TATEROB PLATFORM

In the experiments presented in this paper, we use the TateRob (Fig. 1). The key performance specifications of TateRob are:

Total mass 20.5kg
Length 0.5mx0.5mx0.4m
Maximum speed 3m/s

Due to the desired tasks and the environment in which the robot has been designed to operate, vision was chosen as the primary sensor for navigation. It is also expected that in the environments with sufficient features and color information TateRob can make accurate vision-based odometry estimation. Therefore, the vision system consists of a stereo camera. By changing the lens, the peripheral and foeval vision can be realized.

In order to operate in environments with insufficient lighting conditions, the TateRob is equipped with Laser range sensor. The SICK LMS 200 used in our experiments has a field-of-view of 180 degrees and returns 181 distance readings (one per degree). The maximum error is +/-3 cm per 80 meters.

The robot is actuated by two 24V batteries. One of the batteries actuates the motors and the other one the main CPU and the sensors. A PC/104 stack running the Linux operating system provides the software interface to record and process all the sensor information in real time. The robot can communicate with the operator by wireless LAN in a maximum distance of 100m.

The robot can operate in the following different modes.

1) The robot can move in the environment controlled by a joystick:

a) Directly connected with the robot.

b) Connected with operator PC and remotely controlling the robot.

2) The operator can control the robot remotely by sending commands like move forward, rotate right or left.

3) The robot can operate autonomously by processing the sensors data in the onboard computer.

III. NEURAL CONTROLLER BASED NAVIGATION

In order to calculate the sensory input of the neural controller, the captured color image is converted to a binary image, as shown in Fig. 2. The input of the visual processing module is the image frame captured from the robot's camera. The image first is converted to a grey scale and then to a binary image. In our implementation the size of captured image is 240 by 320 pixels.

We implemented a feed-forward neural controller with 20, 4 and 2 units in the input, hidden and output layers, respectively. A set of visual neurons, arranged on a 2 by 10 grid, receive information about the grey level of the corresponding pixels in the image provided by the camera of the robot. Each input unit covers an area of 50 by 32 pixels in the image. In order to increase the image processing speed, only the half-bottom of the captured image is processed (100 by 320 pixels). The activation of input units, scaled between 0 and 1, is given by the average grey level of all pixels within the partition. The hidden and output units use sigmoid activation function:

$$y_i = \frac{1}{1+e^{-x_i}} \tag{1}$$

where the incoming activation for node *i* is:

$$x_i = \sum_j w_{ji} y_j \tag{2}$$

and *j* ranges over nodes with weights into node *i*.

The output units directly control the right and left wheel angular velocities where 0 corresponds to no motion and 1 corresponds to full-speed forward rotation. The left and right wheel angular velocities, ω_{right} and ω_{left}, are calculated as:

$$\begin{aligned} \omega_{right} &= \omega_{\max} * y_{right} \\ \omega_{left} &= \omega_{\max} * y_{left} \end{aligned} \tag{3}$$

where $\omega_{\max}$ is the maximum angular velocity and y_{right} and y_{left} are the neuron outputs. The maximum forward velocity is considered to be 0.5 m/s.

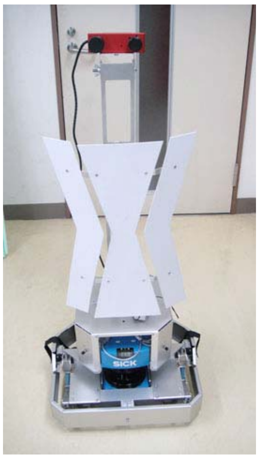


Fig. 1 TateRob.

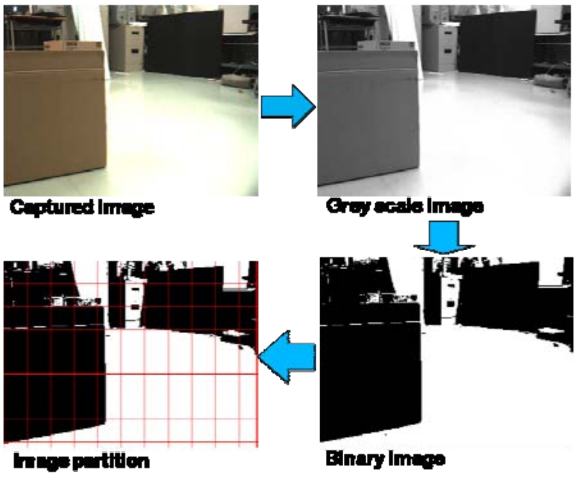


Fig. 2. Sample of image processing.

The neural controllers are evolved in real the real hardware of the TateRob platform. The fitness function selects robots for their ability to move among obstacles as long as possible for the duration of the life of the individual. Therefore, the fitness is the distance traveled by the robot. Every-time that the robot hits an obstacle (detected by the laser sensor), the robot moves backward and the remained lifetime is reduced by 100 steps.

In the first generation 60 neural controllers with randomly selected weight connections are generated. In the second generation the population size is reduced to 20, by selecting the best individuals of the first generation based on the fitness value. The evolution terminated after 9 generations.

IV. Tracking Algorithm

We designed two methods for following a person. The first! method is to have the robot always attempt to drive directly toward the person's location using only the visual information. From general observations, we suspect that this is how people most often follow other people. This method often results in the follower cutting corners and generally not following in the exact footsteps of the leader. The second method, then, is to have the robot attempt to follow the person using the visual and laser information. While this method may not be the most human-like method, we hypothesized that it may better match people's expectations for a machine-like robot. For example, if a person is leading a robot somewhere, any step in the person's path may be taken for reasons that the robot does not know, and thus following the person's exact path may be the more appropriate behavior. Using the person-tracker described above, we have implemented both of these methods.

A. *Face Detection*

In our implementation, we utilized the OpenCV library for face detection. OpenCV uses a type of face detector called a Haar Cascade classifier. Fig. 3 shows an example of OpenCV's face detector. In the image captured from the camera, the face detector examines each image location and classifies it as "Face" or "Not Face." Classification assumes a fixed scale for the face. Since faces in an image might be smaller or larger than this, the classifier runs over the image several times, to search for faces across a range of scales. This may seem an enormous amount of processing, but thanks to algorithmic tricks, classification is very fast, even when it's applied at several scales.

B. *Camera Based Tracking*

After the face detection the clothing information is extracted. Then, the human turn back and starts moving in the environment. Therefore, the human tracking will be mainly based on clothing's color. It is easier to track clothing color because of the large area available for detection. Due to the large tracking area (clothing color), it is possible to track the object under occlusion. This is under the assumption that the object cannot be totally blocked. A portion of its clothing color is exposing throughout the occlusion. In addition, we employed the labeling technique in order not to consider similar color objects that enter in the captured image, as shown in Fig. 4. After labeling, the clothing center position in the captured image and clothing size (number of pixels) are determined. Based on these two parameters the right and left wheel angular velocities are calculated. Also, tracking people without static backgrounds or with a camera mounted on a mobile robot requires us to develop methods that are robust to camera movements.

C. *Camera-LRF Based Tracking*

In order to compare the performance, we integrated the camera image with the LRF data for human tracking. The captured image is divided in three parts (Fig. 5). After we calculated the clothing center position, in order to reduce the execution time, we processed the corresponding LRF data. Based on the data of LRF, we determined the distance from the robot to the human legs, which is very accurate. The robot velocity is considered inverse proportional with the distance to the human.

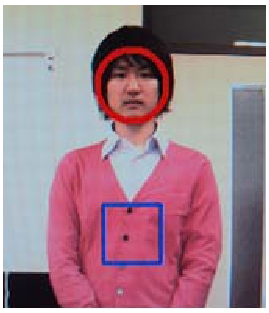


Fig. 3. Face detection.

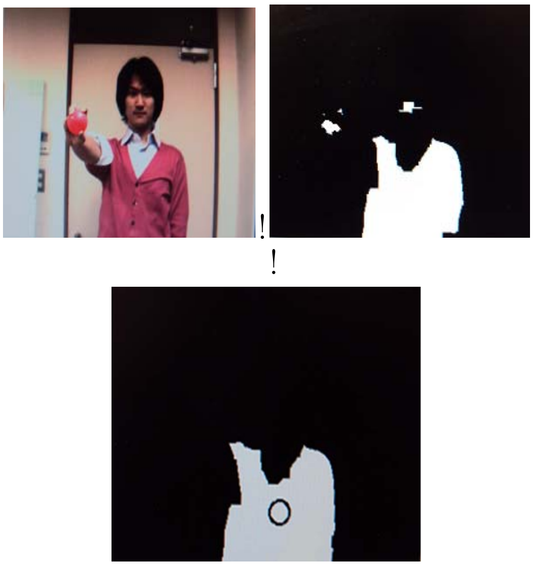


Fig. 4. Captured image after the labeling.

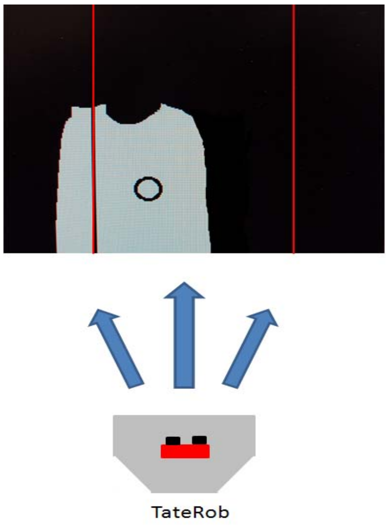


Fig. 5. Analyzing laser data based on the person position.

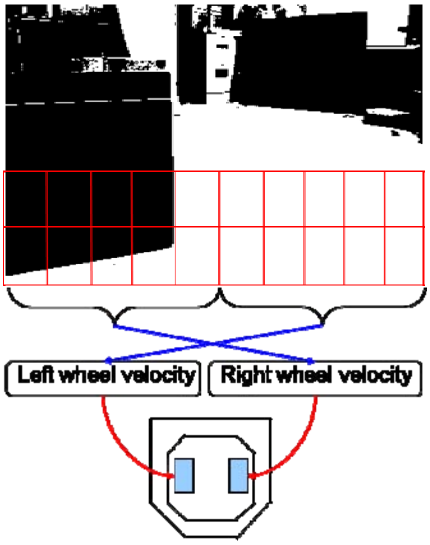


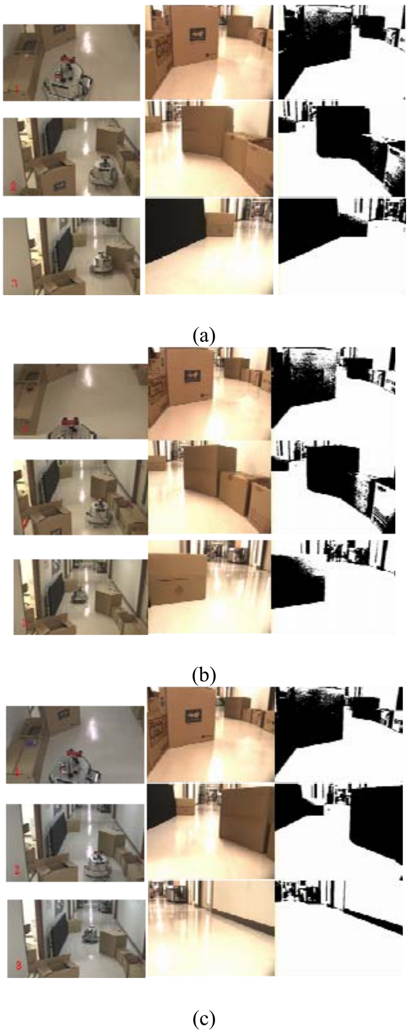
Fig. 6. Algorithmic method for robot navigation.

V. Results

A. Robot navigation

In order to evaluate the performance of evolved neural controllers, we developed an algorithmic navigation method where the right and left wheel angular velocities are calculated based on the average grey level of pixels in the left and right visual field. The angular velocity of right wheel is considered inverse proportional with the average gray level of bottom left half of visual field and vice-versa (Fig. 6).

Fig. 7(a) and Fig 7(b) show the performance of the algorithmic method for two different maximum velocities, 0.5 m/s and 0.3 m/s, respectively. At the beginning, the robot avoids collision with an obstacle and moves fast in the environment (Fig. 7(a)). Then, obstacles become visible in the right half of the visual field.


(a)

(b)

(c)

Fig. 7. Video capture of robot motion. (a) Algorithmic method with max. speed 0.5 m/s. (b) Algorithmic method with max. speed 0.3 m/s. (c) Neural Controller with max. speed 0.5 m/s.

However, due to the restriction in the bottom part of the visual field, the obstacle in the front of the robot (middle upper part of the image) is not considered. Therefore, the small number of black pixels in the bottom right corner does not have a great effect on reducing the angular speed of the left wheel, making impossible for the robot to avoid hitting the obstacle. When the robot was controlled using the algorithmic method, but the maximum velocity is reduced to 0.3 m/s, the robot was able to navigate through the obstacles, as shown in Fig. 7(b).

Due to the low speed motion, the right and left angular velocities are updated in a shorter moving distance. Therefore, the robot trajectory is different compared with the previous experiments (Fig. 8).

Fig. 7(c) shows the performance of robot controlled by the best evolved neural controller. The maximum velocity is 0.5 m/s. The robot avoids obstacles and outperformed the algorithmic method by moving smoothly among obstacles.

The Hinton diagram of the weight connections shows that input neurons positioned in the center of the visual field have positive connections with the hidden neurons. In addition, most of hidden neurons also have positive connections with output neurons that control the right and left wheel angular velocities. Therefore, the robot moves fast in the forward direction when there is no obstacle in the front. When an obstacle enters in the visual field in the left or right side, the respective wheel' angular velocity is reduced due to the negative connection weights and the robot avoids hitting the obstacles.

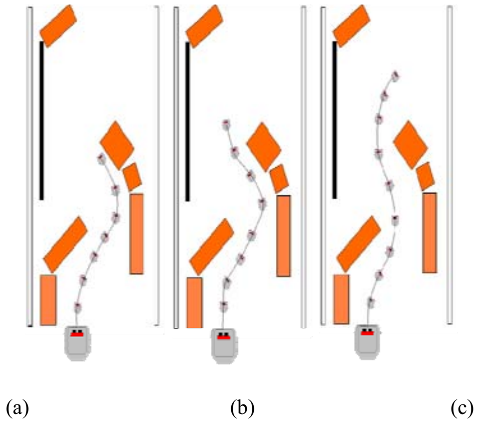

(a) (b) (c)

Fig. 8. TateRob trajectory. (a) Algorithmic method with max. speed 0.5 m/s. (b) Algorithmic method with max. speed 0.3 m/s. (c) Neural Controller with max. speed 0.5 m/s.

B. *Human tracking*

In both methods, the robot begins to follow a person as soon as someone is detected within 125cm of the robot. The robot then attempts to remain a constant distance from the tracked person. This is achieved based on the error between the current clothing's size and its desired size determined after face detection. If the robot is too far from the person and the clothing size has decreased (meaning the robot is falling further behind), then the velocity is increased based on the error; if the robot is too close and is getting closer, then the velocity is similarly decreased. The robot stops if the clothing covers the entire captured image. The robot's maximum velocity is capped at 60cm/s, due to safety concerns. Currently, the distance at which the robot tries to follow is held constant. We have not tested the person following with different distances at this time. Fig. 9 shows the performance of camera based tracking. The experimental results show that robot follows the human in the indoor environment. However, the lighting conditions affected the robot performance. Due to the error in determining the clothing center position the robot's motion deviated from the correct motion toward the person.

In order to overcome this problem we integrated the camera and the laser sensor. This method outperformed the vision based method in terms of the quality of the robot motion. The effect of the noise in the visual sensor due to the lighting conditions was reduced by utilizing the data of the LRF. Based on the position of the clothing center the appropriate LRF data are analyzed, which resulted in better distinction between human and obstacles. In addition, the processing time is reduced. Therefore, the robot is able to follow the person as shown in Fig. 10.

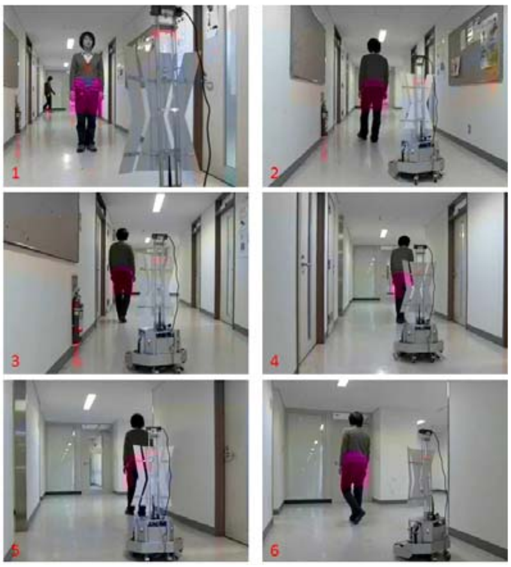


Fig. 9. Human tracking by visual sensor only.

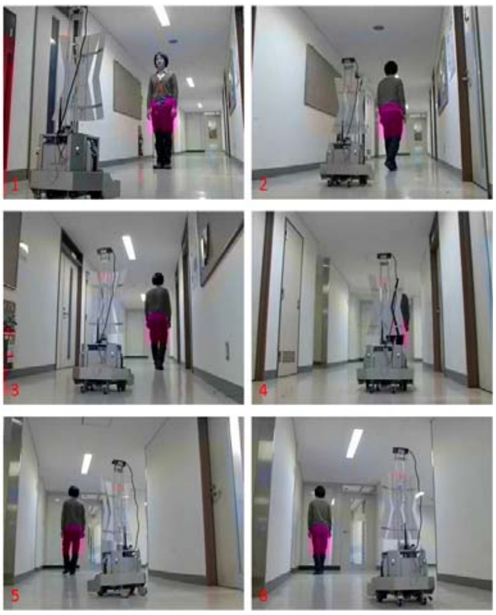


Fig. 10. Human tracking by visual sensor and LRF.

VI. Conclusion

We presented a new method for robot navigation and human tracking for social robotics. In difference from previous works, we utilized the visual sensor. The results show that the evolved neural network outperforms an algorithmic method for robot navigation. For human tracking behavior, combination of visual and LRF sensors performed better than visual sensor only. In the future, it will be interesting to investigate how to evolve neural controllers for human tracking behavior.

References

[1] I.D. Horswill, "Polly: A vision based artifcial agent", In Eleventh Natl. Conf. on AI, pp. 824-829, 1993.

[2] M. Hebert, D. Pomerleau, A. Stentz, C. Thorpe, "Computer vision for navigation: the CMU UGV project", In Proceedings of the Workshop on Vision for Robots, IEEE Computer Society Press. pp. 97-102, 1995.

[3] J. Rosenblatt, and C. Thorpe, "Combining multiple goals in a behavior-based architecture", In Proceedings of IEEE Int'l Conf. on Intelligent Robots and Systems, 1:136-141, 1995.

[4] M.A. Turk, D.G. Morgenthaler, K.D. Gremban, and M. Marra, "VITS - a vision system for autonomous land vehicle navigation", IEEE Transactions on Pattern Analysis and Machine Intelligence, 10(3):342-361, 1988.

[5] J.D. Crisman, "Color region tracking for vehicle guidance. Active Vision", MIT Press: Cambridge, MA. pp. 107-220, 1991.

[6] N. Zeng, J.D. Crisman, "Categorical color projection for robot road following", In Proceedings of IEEE Int'l Conf. on Robotics and Automation, pp. 1080-1085, 1995.

[7] E.D. Dickmanns, B. Mysliwetz, and T. Christians, An integrated spatio-temporal approach to automatic visual guidance of autonomous vehicles. IEEE Transactions on Systems, Man, and Cybernetics 20(6):1273-1284, 1990.

[8] D. Coombs, and K. Roberts, "Bee-bot": using peripheral optical flow to avoid obstacles", In Intelligent Robots and Computer Vision Boston, MA, SPIE 1825:714-721, 1992.

[9] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi, "Divergent stereo in autonomous navigation: from bees to robots", Int'l Journal of Computer Vision, pp. 159-177, 1995.

[10] J. Santos-Victor and G. Sandini, Uncalibrated obstacle detection using normal flow, 1995.

[11] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):780–785, 1997.

[12] J. Crowley and F. Berard. Multi-modal tracking of faces for video communications. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Press, 1997.

[13] T.S. Jebara and A.P. Pentland. Parametrized structure from motion for 3d adaptive feedback tracking of faces. In Proceedings of Computer Vision and Pattern Recognition Conference, pages 144–150, 1997.

[14] J. Yang and A. Waibel. Skin-color modeling and adaptation. In Proceedings of IEEE Workshop on Applications of Computer Vision, pages 142–147. IEEE Computer Society Press, 1996.

[15] A. Azarbayejani and A. Pentland. Real-time self calibrating stereo person tracking using 3-D shape estimation from blob features. In Proceedings of the International Conference on Pattern Recognition 1996, Vienna, Austria, 1996.

[16] I. Haritaoglu, D. Harwood, and L. Davis. W4s: A real time system for detecting and tracking people in 2.5 d. In Eurepean Conference on Computer Vision, 1998.

[17] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Multi-modal person detection and identification for interactive systems. In Proceeding of Computer Vision and Pattern Recognition Conference. IEEE Computer Society Press, 1998.

[18] A. F. Bobick and J.W. Davis. An apearance-based representation of action. In Proceedings of International Conference on Pattern Recognition 1996, August 1996.

[19] Pattie Maes. ALIVE: an artificial life interactive video environment. In ACM SIGGRAPH Visual Proceedings, page 189, MIT Media Laboratory, 1993.

[20] J.M. Rehg, M. Loughlin, and K. Waters. Vision for a smart kiosk. In Proceedings of Computer Vision and Pattern Recognition Conference, pages 690–696, 1997.

[21] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In Computer Vision and Pattern Recognition Conference, pages 84–91. IEEE Computer Society, 1994.

[22] H.A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In Proceedings of Computer Vision and Pattern Recognition, 1998.

[23] K. Sung and T. Poggio. Example-based learning for view-based human face detection. Technical Report 1521,MIT AI Laboratory, 1995. http://www.ai.mit.edu/.