Building Topological Maps by Looking at People: An Example of Cooperation between Intelligent Spaces and Robots

Guido Appenzeller, Joo-Ho Lee, Hideki Hashimoto

Institute of Industrial Science, University of Tokyo
Roppongi 7-22-1, Minato-ku, Tokyo 106, Japan
{G.Appenzeller,JH.Lee,H.Hashimoto}@IEEE.org

Abstract

Intelligent spaces [1, 2] are rooms or areas that are equipped with sensors such as microphones or cameras that enable them to perceive what is happening in them. In such spaces that have an intelligence of their own a world model no longer is something the robot has alone but a service offered by the information infrastructure of the space. In this article we show how such an intelligent space can generate a topological map for robots by looking at the movements of people in the room. We describe the stereo vision system that is capable of tracking the 3D movements of several humans in real time and give experimental results obtained in a real-world environment with several people.

1 Introduction

Autonomous robotic systems such as service robots need maps in order to complete their tasks. This is partially due to the fact that navigation in completely unknown environments is still an unsolved problem. Even if this was possible maps would be needed to specify the robots task or constraints in its path. As robust collision avoidance in moderately difficult environments has been achieved in the last years these maps do not have to provide the complete geometrical information but often an approximate geometrical or topological representation is sufficient (e.g. the PRIAMOS system [3]).

The maps the robot uses are usually created by a human operator and contain a static representation of possible paths the robot can take. This is cumbersome as it requires constant updating to a changing environment. Some advanced mobile systems can explore their environment and build such maps by themselves. This method has the problem that most sensors will not detect all possible types of obstacles. Some obstacles as yellow lines on the floor or signs saying "don't enter" can not be detected at all without large amounts of contextual knowledge.

The approach we suggest to solve this problem is looking at people. In indoor environments people and robots consider similar things as obstacles. The only common exception here are steps and stairs. As our environment is build be safe for people the robot can usually rely upon them to make few mistakes.

Vision systems for looking at people have made significant advances in the last years and can now be built with industry standard hardware [4]. This development has equally been driven by new technology, new approaches to finding humans and a large number of applications ranging from intelligent man-machine interfaces over tele-conferencing to security.

Recently a new paradigm of seeing such systems has evolved. They are no longer seen as an isolated vision system but as a sensor component of the intelligent infrastructure of the space they look at. Such intelligent spaces are able to watch what is happening in them, build a model of themselves, communicate with their inhabitants and act based on decisions they make. Examples for such spaces are the Intelligent Room [2] or the Smart Space [1]. With networking capability being built into many common appliances and prices for sensor hardware dropping rapidly this vision seems indeed not too distant.

Our research is particularly interested in how these intelligent spaces can cooperate with robots that move in them. One interesting application here is how the room can serve as a high level, context sensitive interface to robots. The second research area is on how the room and the robot can share information on the geometry and semantics of the space. Classically a robot has a representation of its environment in a world model he uses alone. In an intelligent space the world model becomes a service that is offered by the environment, in which the robot participates and eventually contributes to.

In this paper we first give an introduction to the Intelligent Space project at the University of Tokyo. The main focus however is on how the system can derive 3D geometrical information by using the vision system of the intelligent space. We first describe the architecture of the vision system in section 3, illustrate how

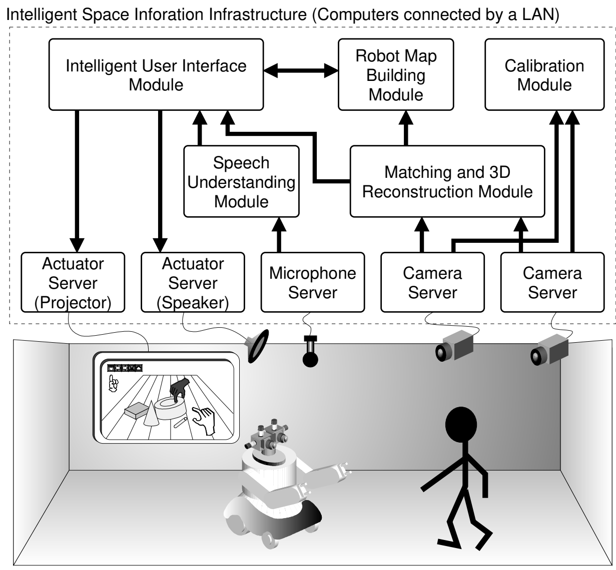

Figure 1: The Architecture of the Intelligent Space

a topological path can be derived in section 4, show experimental results of the system in section 5 and conclude with a perspective on ongoing research.

2 The Intelligent Space

An Intelligent Space is an area such as a room, a corridor or a street that is equipped with sensors and actuators. The sensors have the main purpose to perceive what is happening in the space, especially what humans inhabiting it are doing. They include active and passive cameras, microphones, microphone arrays and tactile sensors. Actuators are currently mainly used to provide information to the inhabitants. This is done with speakers, screens, pointing devices, switches or robots inside the room.

The soft- and hardware architecture of the intelligent space must posses a number of properties:

Modular. As often components will be added or removed from the intelligent space it has to be modular and should be reconfigurable during runtime.

Scalable. The architecture should be valid not only for a single room but should allow integration of local spaces into larger systems.

Integration It should be simple to integrate existing intelligent components or services into the room.

Low Cost. As many intelligent components are needed the cost of a single component should be low. This requires the use of industry standard hardware.

Easy Configuration and Maintenance. Setting up and maintaining an intelligent space should be possible with minimal effort. The space must be able to learn about itself (e.g. model building, auto-calibration) and site specific adaptation has to be done easily.

The modularity dictates that the components of the intelligent space have to be connected by a network. As the volume of the data generated by some of its sensors (e.g. cameras) is too high for todays network architectures it is clear that data preprocessing has to be done locally.

On the software side we have three different types of tasks with different characteristics. They are distributed as individual processes over the computers of the network.

Sensor and Actuator Servers. For the data preprocessing highly specialized modules are needed that derive relevant information from the sensors and offer this information on the network.

Intermediate Processing. On an intermediate level processes collect data from one or several sensor servers to which they connect as clients. Typical tasks are sensor fusion, temporal integration and model building. As sometimes this requires some real-time capability they should be located close to the sensor computers. The intermediate results are again offered on the network.

Application Processes. These processes perform the actual applications of the space. As they usually require low volumes of data and slower reaction times optimization is less critical. They should however be easily portable across architectures and easily maintainable by the user.

Similar architectures have already been used for the control of other types of intelligent systems such as mobile robots (e.g. [5]).

Our intelligent space is integrated into the laboratories computer network. Its processes run on PCs, SGI and Sun workstations running different types of UNIX/Linux. The Robot participates in the network by wireless LAN.

While for the sensor servers and intermediate processes, system specific compiler languages are used the application processes use interpreted Tcl/Tk [6]. This allows extremely fast application development and easy creation of user interfaces. It additionally is used for rapid prototyping of lower level processes.

Unlike previous projects our intelligent space it is not a special room but a real laboratory that is constantly used by students and researchers working in it. This requires a high degree of robustness for the components of the intelligent space.

Figure 2: Different stages of the detection process. (a) sample camera image (b) potential skin color pixels (c) foreground pixels (d) foreground skin color pixels (e) cluster bounding boxes (f) centroids and first moments of clusters.

3 The Vision System

To build a map the intelligent space tracks the movements of humans. For this purpose the vision system is the only sensor that is used. Recognizing the human is done in two steps. First the area or shape of a human is separated from the background. Second features of the human as head, hands, feet, eyes etc. are located. Taking the images of several cameras we can then calculate the 3D position of the human.

3.1 Separating the Human Shape

To separate the shape of a human from the background the two most frequently used approaches are background separation or motion detection by optical flow. It has been suggested to find humans directly using color information alone [7] however this only works robustly in front of simple backgrounds. Finding humans by detecting motion of edges [8] has been suggested however it seems less robust for obtaining the complete shape.

Background separation. Object moving in front of a static background viewed with a static camera can be identified using background separation. If we define background B and image I at the time t as RGB trippels:

$$B^t = (B^t_r, B^t_g, B^t_b) \quad (1)$$
$$I^t = (I^t_r, I^t_g, I^t_b) \quad (2)$$

the all pixels of the Image I for which

$$I_t(x,y) \neq B_0(x,y) \quad (3)$$

are foreground. This technique however faces several difficulties. First the image values $I^t(x,y)$ tend to be noisy. This noise can differ significantly for different regions of the image e.g. zero for saturated areas and high for the region that contains a television screen. This can be solved by using an experimentally obtained local noise model σ. The criterion for foreground is:

$$I^t(x,y) - B^0(x,y) > \sigma(x,y) \quad (4)$$

A second common problem are shadows. There detection can be avoided if additional color information is used. For an intensity normalized representation such as:

$$r = \frac{R}{R+G+B} \quad g = \frac{G}{R+G+B} \quad b = \frac{B}{R+G+B}$$

the color of a lambertian surface is invariant under varying intensity of a single light source. In this model shadows will not appear. In images taken of real objects using a ccd camera and frame grabber this is usually only an approximation. Deviations are due to factors such as different light sources, saturation, specular reflections, nonlinearities introduced by the camera and frame grabber etc. For our application however this model has proven to be sufficient.

No background will remain static over a long period of time. Changes in lightning, moved objects and the camera signals will alter the image slightly. Here a continuously updated background model is needed:

$$B_{t+1} = \lambda \cdot B_t + (1 - \lambda) I_t \quad (5)$$

We obtain as criterion for background separation:

$$|I_r^t - B_r^t| + |I_g^t - B_g^t| + |I_b^t - B_b^t| > \sigma \quad (6)$$

Motion and Optical Flow. Optical flow is another commonly used technique to separate moving objects from the background however most of its implementations are too slow for real-time implementation on standard hardware. Often simple image differencing is used instead. It selects all pixels for which:

$$I^t - I^{t-1} > \sigma \quad (7)$$

This method has the problem that it will include all pixels of the background which are covered in the last frame. This is acceptable if the objects movement between two frames is small compared to its size. If the objects movement is faster secondary differencing has successfully be used in the Intelligent Room [9]:

$$I^t - I^{t-1} > \sigma \quad \texttt{AND} \quad I^{t-1} - I^{t-2} > \sigma \quad (8)$$

This too fails however if several objects with overlapping trajectories move (e.g. hands and arms). Image differencing equally has a problem with shadows that can equally be solved by using color. Background separation and simple image differencing can both be characterized by equation 6 with $1 > \lambda > 0$ for adaptive background separation and $\lambda = 0$ for image differencing.

3.2 Finding Head and Hands

To calculate 3D from several camera views point correspondences are needed. To establish these correspondences directly from the shape of the human is difficult. Instead we first find the head and hands of the human and use their centers for matching. A second motivation to further analyze the shape is that adaptive background separation in complex scenes detects recently displaced objects.

Too identify parts of the human body feature detectors that find these parts directly can be constructed. For finding faces neural nets [10, 11] have been used. A different approach uses edges in gradient images [12]. For high resolution face images recognition can be further enhanced by using active contour models [13]. Templates or sets of templates can equally be used for finding faces however robust person independent location under changing conditions seems difficult. The contour shape of a human has frequently been used to identify body parts [14, 15, 16]. It only works well however for parts which are stretched out.

Another common technique is to use color information. By segmenting the shape into colored regions [17] or looking for skin colored regions directly [7] hands and head can be identified. As we only try to find head and hands the second approach is sufficient for our system.

In an luminance normalized color space skin color is remarkably constant over a wide range of ethnic origins and illuminations. Again as an approximation the transformation from equation 6 can be used. As in this equation $(r + g + b) = 1$ only two components have to be considered. The criterion for skin pixels is thus:

$$Hist(r, g) = \begin{cases} 0 & \texttt{Skin Pixel} \\ 1 & \texttt{Other} \end{cases} \quad (9)$$

Problems arise however when skin color is too dark to contain substantial color information.

An interesting and more complex skin color detection scheme using probability histogram instead of a binary histogram has be develloped by Schiele et al. [7].

3.3 Calibration and Reconstruction

To obtain 3D coordinates from the color regions we have identified three steps are necessary. We have to obtain the geometric parameters of the cameras (calibration), we have to establish point correspondences between the two images (matching) and finally have to find an approximate 3D position to the corresponding points (reconstruction).

Calibration. The geometric model used for the camera system as well as the algorithm for calibrating it are those of Tsai [18]. It is attractive as it offers a fast, high precision solution that is able to cope with lens distortion as it is encountered when using wide angle lenses. Additionally the highly portable implementation of Willson [19] is publicly available in source code. Calibration is done using a grid of circular targets at known positions on the floor. The center of the targets is calculated with sub-pixel accuracy. For each camera of the system the calibration matrix is stored and can be accessed by the corresponding camera server. A very interesting development in the field is the self calibrating vision system recently presented by Azarbayejani et al [14].

Clustering and Matching. Before being able to calculate the 3D position of a point we need a correspondence of to point in the two images. This faces us with two problems.

First we have to reduce the skin color regions to a single point. The first step to do this is to cluster potential skin pixels to clusters. For this a simple but fast self-developed one pass clustering algorithm is used. For each line cluster hypotheses are generated. Possible clusters that do not have a certain amount of continuety over several lines are discarded. In a final step overlapping clusters are merged. For the point correspondences we could now use the centers of the bounding boxes of these clusters however this methods is highly susceptible to noise. We instead calculate the center of gravity $(\bar{x}, \bar{y})$ of all the pixels p_i of the cluster as:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{10}$$

Second we have to find out which clusters in one image corresponds to which cluster in the other image. One criterion is the size of the clusters. Again the size of the bounding box is a highly susceptible to noise. A more robust measure is the matrix of the first moments:

$$\begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{pmatrix} = \begin{pmatrix} \frac{1}{N}\sum_1^N x_i^2 - \bar{x}^2 & \frac{1}{N}\sum_1^N x_i y_i - \bar{x}\bar{y} \\ \frac{1}{N}\sum_1^N x_i y_i - \bar{x}\bar{y} & \frac{1}{N}\sum_1^N y_i^2 - \bar{y}^2 \end{pmatrix}$$

For the matching four indicators are used.

Cluster Size. We compare the standard deviation of the clusters σ_{xx} and σ_{yy}.

Cluster Orientation. From the matrix of the first moments we equally know the inclination of the clusters. Only inclinations with a meaningful 3D position are considered.

Epipolar Constraint. For two cameras with known geometry the two dimensional search for correspondences can be reduced to a single dimension along a line in th other image, the epipolar line (see e.g. [20]). The distance from this line corresponds to an error probability.

Model Constraints. After 3D reconstruction of the position we confirm that the position is meaningful. Positions above the ceiling, under the floor or outside the room are discarded.

Reconstruction. Once two matching 2D positions are obtained the 3D reconstruction done by finding the least square solution for the projective geometric equations (e.g. see [20]). As a fast implementation is required the optimized LAPACK libraries are used.

A more general approach to using first moments to describe 3D color distributions from which some of the above ideas were borrowed is used for the system described in [14].

A modeling stage that uses a model of humans to further enhance the robustness of the tracking has recently been developed however it is not used for the experiments in this article.

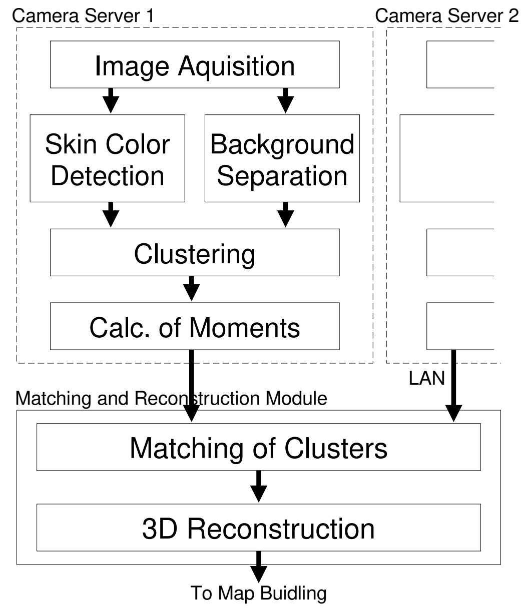

Figure 3: The architecture of the vision system.

3.4 Architecture

The above algorithms are implemented in three different modules of the intelligent space as shown in figure 3.

Camera Server. This module connects directly the the frame grabber. It performs the shape separation, skin color detection, clustering, and moment calculation. The result is a 6 parameter description ($\bar{x}$, $\bar{y}$, σ_{xx}, σ_{yy}, σ_{xy}, color) of each cluster. An unlimited number of clients can connect to each server.

3D Reconstruction Module. For each pair of cameras of the space a 3D reconstruction module can be started. It connects to the two camera servers, loads its calibration data and offers a stream of reconstructed 3D color clusters on the network. As we want to be able to add and remove cameras by simply adding them to the network we can not use synchronized cameras. This leads to a deviation of the reconstructed 3D position for fast moving objects. We counter this problem by rejecting cluster data for which the time difference between the two sources is larger than 50ms. The resulting error in this case is small [4].

Calibration Client. The calibration module is an application module that is interactively launched by an operator. It uses a graphical interface to specify the points of the grid and their coordinates.

3.5 Optimization Issues

In order to make the above system run on standard personal computer hardware a number of optimization issues are important.

First it is important to understand that the main constraint of the system is not calculation time but the amount of data that can be transfered to the processor. As skin color detection can be done with less memory transfer than background separation it is done first.

Before the histogram lookup for skin color detection can be performed the colors have to be normalized as equation 6. The calculation steps to do this however are time consuming. A second possibility is to transfer the histogram from the normalized (r, g) to the unnormalized (R, G, B) representation. Now the color components can directly be used for lookup. The problem now is that the histogram becomes big (256^3 = 16 Million entries). The solution is to use a 15 bit representation with 5 bit of color information for each channel:

$$I(x, y) = 0RRRRRGGGGGBBBBB_2$$

This reduces the histogram size to 32 kByte which fits into the second level cache of the processor. As the frame grabber can directly generate this format the whole skin color decision process is reduced to a single binary memory lookup operation.

The skin color detection, background separation and clustering of the images are all calculated in a single pass over the image. An additional pass for the moment calculation is only needed for skin color areas.

The hardware of the system are PCI-Bus PCs using Matrox Meteor PCI frame grabbers and Chinon CCD cameras. As the most critical part of the system is the PCI-DMA transfer selection of a high performance chip-set is crucial. Overall cost of the system is less then U.S. $ 2000 per camera.

4 Generating Topological Maps

The topological map generation is done by a seperate process that connects to the human tracking module. First we have to determine if the human that is observed is walking or doing something else. Second we have to transfor the human positions into a topological map that is usefull for the robot.

4.1 Identifying walking areas

To determine which areas people walk in we generally need to determine what people are doing when they where seen. People sitting on chairs, leaning over tables or having their hands on tables while working all generate 3D skin color blobs which have to be filtered out. A number of clues can be used to identify a persons walking path e.g. height of the head, speed of movement, relative position of head and hands. For our application it has proven sufficient to assume that if a person is standing upright its head is above its walking area. People standing upright can be filtered out by simple height thresholding. Hands can be filtered out by removing all blobs that are too small for faces.

4.2 Extracting a Topological Path

Next the positions are filtered for outliers which are usually due to incorrectly matched blobs. Then the positions are dilated with a morphological operator to obtain a connected walking area.

The next step is finding a path inside this walking area. As the positions of humans will be noisy due to the reconstruction error and mismatches, we need an algorithm that is robust against this type of noise. As speed is a less important issue we are looking for a safe rather than the shortest path.

In [21] it has been pointed out that the safest path in terms of having the biggest distance to obstacles can be found by using the skeleton of the passable area. As a method of skeletonizing, we used Zhang-Suen's skeleton algorithm [22], which is frequently used in image processing. It is a iterative procedure of decomposing cells by considering relations to other cells. This algortihm has several important properties that are necessary for our task:

- It never cuts connected areas. This is necessary to ensure that the topological path reflects all possible ways the robot can move to a point.
- The result of skeleton process is always placed on the center of the object. This guarantees us maximum safety and makes the algorithm robust against evenly distributed noise.
- One iterative process decomposes only one layer of object cells around a contour. By this way it is possible to generate skeletons of variable thickness. This is interesting for robots that have a size that is significantly bigger than the size of humans.

While the algorithm can generate skeletons of arbitrary thickness we currently use it to generate a skeleton of minimal thickness which is then transformed into a graph with nodes for each fork of the skeleton. Approximate geometrical path information for each connection is stored. The robot can then navigate its environment by driving to a close node using collision avoidance, use the topological map to get to a node close to is destination and navigate there with collision avoidance as detailed in [5].

5 Experimental Results

5.1 The Vision System

Experimental evaluation of the system at the output level of the 3D reconstruction shows a speed of 20

Frames/s with an image resolution of 640x480. An example standard deviation for heads at a distance of 10 meters from the camera plane is about 3.5 centimeters in the camera plane and 7.5 cm orthogonal to it. Fast movements of a human (e.g. running, waving quickly) slightly increases the error. See [4] for more experimental results.

5.2 Building Maps

The experiments where performed in the intelligent laboratory room whose ground plan is shown in 4. Positions A-C denote places where people sat and worked during the experiment. Position D and E contain a large and small obstacle that are not in the world model. F is a large box on the floor that is part of the world model.

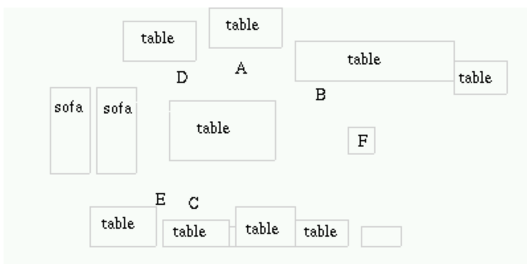

Figure 4: Ground plan of the Intelligent Space

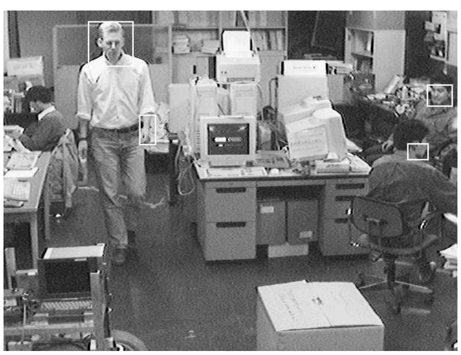

Figure 5: Example scene during the tracking. The head to the very left was not found at it had not moved over a longer period of time, the left arm because it is fragmented and too dark and thus contains too little color information.

The room watched the movements of the people in it for about one hour. During all of the time 2-4 people where in the room working, talking to each other or walking around. Figure 5 shows an example scene.

Positions of moving persons were obtained with about 20 Hz. Only positions with a vertical height between 1.65 and 2.00 meters and only blobs with at least 0.6 times the size of a head were taken into account. Due to these parameter settings about two thirds of the correct head positions were discarded but wrong matches, hand positions etc. were filtered out with a high accuracy.

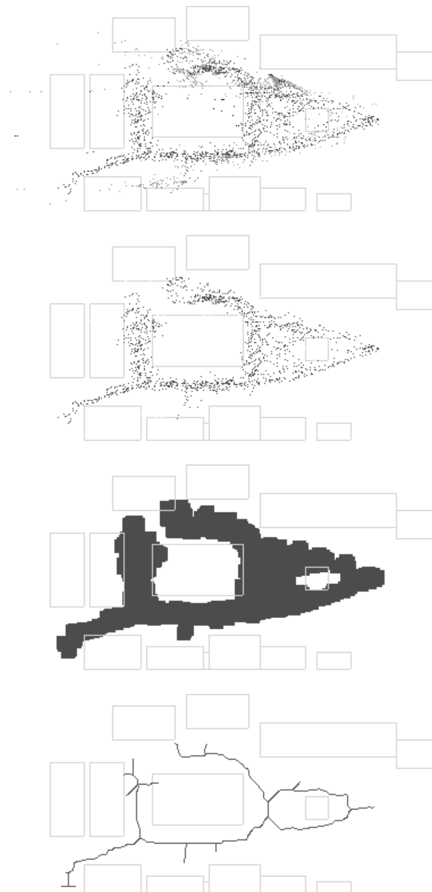

Figure 6: Different stages of the map building algorithm. (a) initial position of found skin blobs (b) filtered positions (c) dilated walkable area map (d) safest path through map.

In figure 6a-d the different stages of the algorithm can be seen. Filtering out all suspected non-head positions reduces some hand blobs everywhere but the strongest effect is visible in areas where people worked. Here positions were filtered out as head and hands are much lower than that of walking people. The last stop shows the topological map of the room. The topological map correctly avoids all static obstacles as well as the dynamic ones at position D and E that where not in the world model. The small error in the left of the image is probably due to tall people leaning over the table.

6 Conclusion and Outlook

We have presented a system that is capable of learning topological paths for robots from the movements of people in the room that works robustly in a complex, cluttered real world environment with multiple people. In a usual laboratory environment the system can generate a new topological map within one hour. We consider the system as a special case of cooperation between intelligent robots and intelligent rooms.

Currently research is under way to additionally use the room as enhanced user interface to the robots. Using the camera system the room will be capable to resolve spatial references in human speech. Commands as "drive there" become possible.

A more general question is what a room can learn from looking at its contents. Fusing the data gained by tracking humans and that of object recognition the room should not only be able to gain a better geometric model but annotate it with semantic information such as classifying areas according to their use, understanding tasks humans perform etc. Here the learning of tasks from looking at the actions of humans could be particularly interesting.

References

[1] C. R. Wren, F. Sparacino, A. J. Azarbayejani, T. J. Darell, T. E. Starner, A. Kotani, C. M. Chao, M. Hlavac, K. B. Russell, and A. P. Pentland, "Perceptive spaces for performance and entertainment: Unethered interaction using cumputer vision and audition," technical report, M.I.T. Media Laboratory, 1995.

[2] M. C. Torrance, "Avances in human-computer interaction: The intelligent room," in *Proceedings of the CHI'95 Research Symposium, Denver, Colorado U.S.A*, May 1995.

[3] J. K. R. Dillmann and F. Wallner, "Priamos: An experimental platform for reflexive navigation," tech. rep., Institute for Real Time Computer Systems and Robotics, University of Karlsruhe, Germany, 1992.

[4] G. Appenzeller, Y. Kunii, and H. Hashimoto, "A low-cost real-time stereo vision system for looking at people (submitted)," in *International Symposium on Industrial Electronics*, July 1997.

[5] R. Dillmann, J. Kreuziger, and F. Wallner, "The control architecture of the mobile system priamos," in *Proc. of the 1st IFAC International Workshop on Intelligent Autonomous Vehicles, Southampton*, 1993.

[6] J. K. Ousterhout, *Tcl and the Tk Toolkit.* Addison Wesley, 1994.

[7] B. Schiele and A. Waibel, "Gaze tracking based on face-color," in *Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition*, June 1995.

[8] H. Gu, Y. Shirai, and M. Asada, "Mdl-based segmentation and motion modelling in a long image sequence of scene with multiple moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 59–64, Jan. 1996.

[9] H. J. Kim, "Motion based multi-person tracking," tech. rep., Artificial Intelligence Laboratory, M.I.T., Boston, MA, 1995.

[10] S. McKenna and S. Gong, "Tracking faces," in *International Conference on Automatic Face and Gesture Recognition*, pp. 271–276, Oct. 1996.

[11] M. Collobert, R. Feraud, G. L. Tourneur, and O. Berenier, "Listen: A system for locating and tracking individual speakers," in *International Conference on Automatic Face and Gesture Recognition*, pp. 283–288, Oct. 1996.

[12] K. C. Yow and R. Cipolla, "A probabilistic framework for perceptual grouping of features for human face detection," in *International Conference on Automatic Face and Gesture Recognition*, pp. 16–21, Oct. 1996.

[13] H. Wu, T. Yokoyama, D. Pramadihanto, and M. Yachida, "Face and facial feature extraction from color images," in *International Conference on Automatic Face and Gesture Recognition*, pp. 345–350, Oct. 1996.

[14] A. Azarbayejani and A. Pentland, "Real-time self-calibrating stereo person tracking using 3d shape estimation from blob features," in *International Conference on Pattern Recognition*, pp. 627–632, Aug. 1996.

[15] D. Geiger and T.-L. Liu, "Recognizing articulated objects with information theoretic methods," in *International Conference on Automatic Face and Gesture Recognition*, pp. 45–50, Oct. 1996.

[16] M. K. Leung and Y.-H. Yang, "First sight: A human body outline labeling system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 359–377, Apr. 1995.

[17] C. Wren, A. Azarbayejani, T. Darrel, and A. Pentland, "Pfinder: Real-time tracking of the human body," in *International Conference on Automatic Face and Gesture Recognition*, pp. 51–56, Oct. 1996.

[18] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Transactions on Robotics and Automation*, vol. RA-3, pp. 323–344, Aug. 1987.

[19] R. Willson, "Camera calibration using tsai's method - revision 3.0b3," tech. rep., Carnegie Mellon University, http://www.ius.cs.cmu.edu/afs/cs.cmu.edu/user/rgw/www/TsaiCode.html, 1994.

[20] O. Faugeras, *Three-Dimensional Computer Vision, a Geometric Viewpoint.* MIT Artificial Intelligence, The MIT Press, 1993.

[21] J. H. Lee, "A study on advanced path planning and navigation algorithm for autonomous mobile robot," Master's thesis, Korea University, 1995.

[22] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Comm. ACM*, pp. 236–239, 1984.