Robust Real-Time Human Activity Recognition from Tracked Face Displacements*

Paul E. Rybski and Manuela M. Veloso

The Robotics Institute
and Computer Science Department
Carnegie Mellon University, USA
{prybski,mmv}@cs.cmu.edu

Abstract. We are interested in the challenging scientific pursuit of how to characterize human activities in any formal meeting situation by tracking people's positions with a computer vision system. We present a human activity recognition algorithm that works within the framework of CAMEO (the Camera Assisted Meeting Event Observer), a panoramic vision system designed to operate in real-time and in uncalibrated environments. Human activity is difficult to characterize within the constraints that the CAMEO must operate, including uncalibrated deployment and unmodeled occlusions. This paper describes these challenges and how we address them by identifying invariant features and robust activity models. We present experimental results of our recognizer correctly classifying person data.

1 Introduction

Recognizing human activity is a very challenging task, ranging from low-level sensing and feature extraction from sensory data to high-level inference algorithms used to infer the state of the subject from the dataset. We are interested in the scientific challenges of modeling simple activities of people who are participating in formal meeting situations. We are also interesting in recognizing activities of people as classified by a computer vision system.

In order to address these challenges, our group is developing a physical awareness system for an agent-based electronic assistant called CAMEO (Camera Assisted Meeting Event Observer) [1]. CAMEO is an omni-directional camera system consisting of four FireWire cameras mounted in a 360° configuration,

* This research was supported by the National Business Center (NBC) of the Department of the Interior (DOI) under a subcontract from SRI International. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, by the NBC, DOI, SRI, or the US Government.

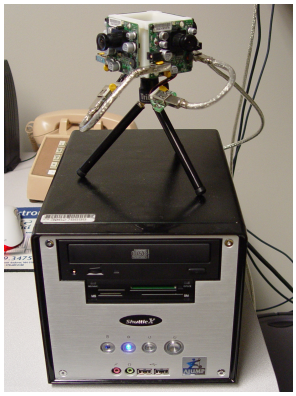

Fig. 1. The CAMEO system consists of a set of FireWire cameras arranged in a panoramic fashion and a small-form-factor PC.

as shown in Figure 1. The individual data streams extracted from each of the cameras are merged into a single panoramic image of the world. The cameras are connected to a Small Form-Factor 3.0GHz Pentium 4 PC that captures the video data and performs image processing.

The panoramic video stream is scanned for human activity by identifying the positions of human faces found in the image. We make the assumption that people can be recognized in the image based on the location of their face. To be able to recognize people's activities within any meeting scenario, the challenge is to find appropriate features in terms of face positions without a global coordinate system. Low-level features are extracted from the raw dataset and are modeled as the observations for the recognition methods. The range of all possible human activities is reduced to a small discrete set.

We successfully solve this problem by two main contributions: (i) the identification of robust meeting features in terms of relative displacements; and (ii) the application of Dynamic Bayesian Networks (DBNs) to this problem, extending their use from other signal-understanding tasks.

2 Related Work

We use DBNs [2] to model the activities of people in the meetings. DBNs are directed acyclic graphs that model stochastic time series processes. They are a generalization of both Hidden Markov Models (HMM) [3] and linear dynamical systems such as Kalman Filters. DBNs are used by [4] to recognize gestures such as writing in different languages on a whiteboard, as well as activities such as using a Glucose monitor. Our system infers body stance and motion by tracking the user's face in a cluttered and general background rather than attempting to track the hands, which is difficult to do in general.

In [5], finite state machine models of gestures are constructed by learning spatial and temporal information of the gestures separately from each other.

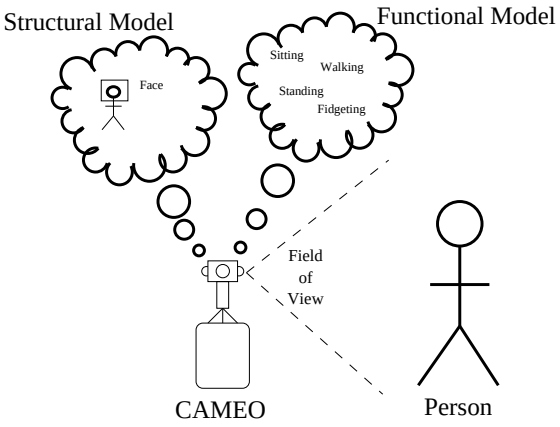

Fig. 2. CAMEO maintains both an image and activity model of people in the environment. The image model is a representation off the person from CAMEO's sensors (e.g. the detected face). The activity model represents how the person is expected to move about and what those movements mean in terms of actions.

However, it is assumed that the gestures are performed directly in front of the camera and that the individual features of the face and hands can be recognized and observed without error.

An extension to the HMM formalism called the Abstract Hidden Markov mEmory Model (AHMEM) [6] is used to represent a hierarchy of both state-dependent and context-free behaviors. However, this work uses a network of cameras set up throughout the entire office space to view hand-labeled locations.

A system for using stereo cameras to infer deictic information through torso and arm position tracking is described in [7]. Our system is essentially monocular and is not intended to be addressed directly where it could observe the full torso and arm positions of everyone attending the meeting.

Recognizing the behaviors of individual robotic (non-human) agents has been studied in [8]. Robots playing soccer against other robots [9] would greatly benefit by being able to classify the different behavioral patterns observed by their opponents. In this work, robots are tracked by an overhead camera and their actions are classified by a series of hand-crafted modified hidden Markov models (called Behavior HMMs).

Much of the related work in activity modeling relies upon fixed cameras with known poses with respect to the objects and people that they are tracking. Our efforts focus very heaviliy on activity models that can be tracked and observed by uncalibrated vision systems which do not have the luxury of knowing their absolute position in the environment. This approach is attractive because it minimizes the cost for setting up the system and increases its general utility.

Fig. 3. A frame of video from a typical meeting as annotated by CAMEO.

3 The CAMEO System

CAMEO is part of a larger effort called CALO (Cognitive Agent that Learns and Organizes) to develop an enduring personalized cognitive assistant that is capable of helping humans handle the many daily business/personal activities in which they engage. In order to be useful in a general set of environments, CAMEO must operate in many different meeting room configurations and should not require any lengthy calibration for distance or lighting conditions.

Raw visual data from the multi-camera system is captured and merged into a single consistent image mosaic [1]. People in the image are located by identifying their faces using the detector in Intel's OpenCV[1] computer vision library. Faces are matched between subsequent video frames by computing a distance metric between sets of tracked faces and the new faces. Matches are those that have the smallest distance. The metric is computed by taking the SVD of the image sets and computing the weighted sum of the most significant eigenvectors. CAMEO is capable of recording mpeg movies of meetings for archival purposes and off-line analysis. All of CAMEO's detection/inference algorithms can be run on either live or recorded video. Finally, streams of tracked facial information are fed into a DBN classifier that identifies the state of each person. The following sections describe the details of CAMEO's state inference mechanisms.

4 Meeting State Inference

As mentioned previously, CAMEO must be able to operate in uncalibrated environments and infer the activities of people in real time. Additionally, activity models that are used in one meeting must be transferable to other meetings with different attendees. This enforces a number of constraints on CAMEO's data processing, the most significant of these are shown in Figure 4. Because the environment contains objects that are unknown and unmodeled, detecting people's bodies is very challenging and difficult to do properly. As a result, the most robust feature for detecting a person becomes their face. Faces are very unique and distinguishing features which greatly simplify the task of determining whether an object is a person or not. The face detection algorithms that we use are a compromise between the need for accurate and robust person detection

[1] http://www.intel.com/research/mrl/research/opencv/

and the needs for CAMEO to operate in real-time. The decision to process facial data only directly affects the selection of specific features for activity recognition. For instance, the absolute Y positions of people's faces are not very meaningful because CAMEO is not aware of the positions of people in the environment, nor is it aware of its own relative distance to those people.

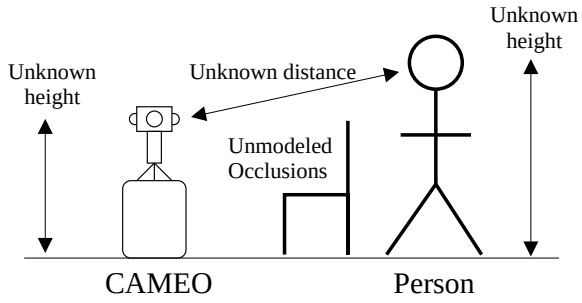

Fig. 4. Unknown spatial quantities that the CAMEO activity recognition system must contend with when attempting to infer activities from observed people.

In order to operate within these constraints, a number of simplifying assumptions are made about the structure of the environments and the behaviors of the participants in the meetings. These assumptions are:

1. CAMEO will be in the center of a meeting and can see all the participants
2. When performing an action, a person is likely to remain doing that action and not rapidly switch from one action to another
3. People's activities are first-order Markovian (the current state can be inferred by only considering the previous state and the current data)

4.1 Observation Model - Features

At every frame of video, all of the faces that have been found return an (x, y) position in image coordinates. The CAMEO software tracks each face from frame to frame and stores a history of the face positions. Because relying on absolute (x, y) positions will be brittle due to the above constraints, we instead look at the difference of the face positions between subsequent frames of video, e.g. $(\Delta x, \Delta y)$, where $\Delta x = x_t - x_{t-1}$ and $\Delta y = y_t - y_{t-1}$.

As an illustrative example, a single person's face was tracked by CAMEO and the resulting raw displacements in the horizontal and vertical direction are shown in Figure 5. In practice, this particular choice of features has proven to be quite robust.

4.2 Observation Model and States

In order to define a DBN, the model states must be identified and observation probabilities distributions must be assigned to those states. Initially, we

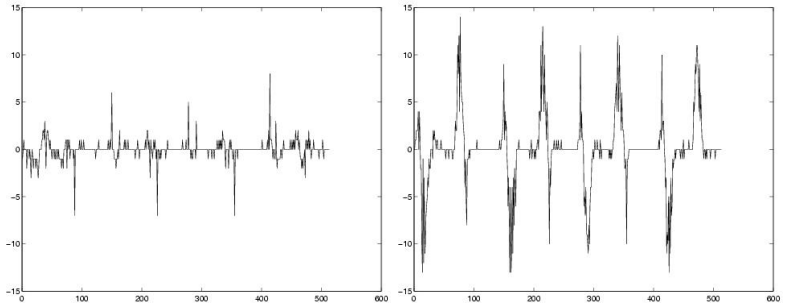

Fig. 5. Raw data from a tracked person. The left graph shows ΔX (horizontal displacement) of the face in pixels over time while the right shows ΔY (vertical displacement) over time. The horizontal axis is time in seconds.

were interested in the following activities: "Standing," "Walking," "Fidgeting," and "Sitting". The observation features as previously introduced consist of real-valued 2D vectors of face *displacements* (to address the challenges of general meeting processing). To correctly define this model, we needed to perform some empirical studies to find out how our observation features were related for these activities that we wanted to track.

Several meetings were recorded and the actions of the people in them were hand-annotated to generate class labels to train the DBN's observation probability distributions. Interestingly, some of our general states could not be distinguished from one another. The stationary standing and sitting states are indistinguishable because the face displacements $(\Delta x, \Delta y)$ are identical given that the person is generally immobile. To address this problem, we refined our model to include states "Sitting down," "Standing up," "Stand still," and "Sit still". Fidgeting and walking left or right could also not be distinguished. In our model, the transitions through intermediate states also resolved this problem. The final observation empirically-defined distributions are shown in Figure 6.

Note the symmetry between the observations in the states associated with "Standing up" and "Sitting down" in Figure 6. The "Walking" and "Fidgeting" states have very similar means, as do the "Stand still" and "Sit still" states. This directly illustrates the uncertainties associated with the CAMEO's observation models, particularly because absolute values in the Y dimension are not directly meaningful.

4.3 Person States

We have used these constraints and assumptions to define a finite state machine which encapsulates a coarse level of possible activities that can be detected in the image data stream. This state machine is illustrated in Figure 7.

The "Sit still" state represents instances when a person is sitting in a chair and is stationary. The "Fidget left" and "Fidget right" states represent motion

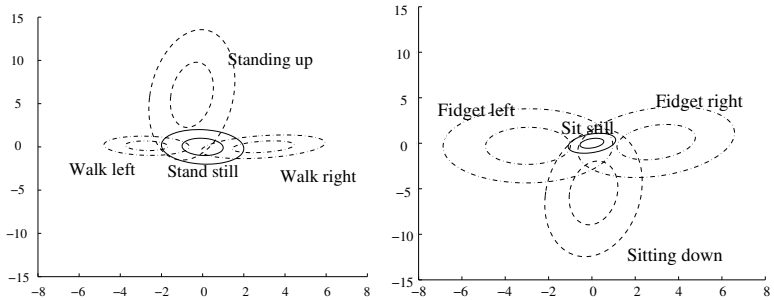

Fig. 6. Learned Gaussian distributions for the real-valued observation vectors corresponding to the four hidden states associated with standing and walking on the left and sitting and fidgeting on the right. The first and second standard deviations are shown for each distribution.

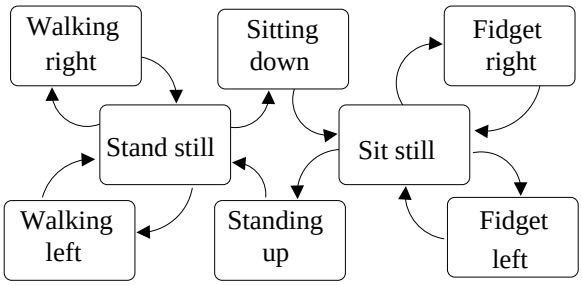

Fig. 7. Example finite state machine for a single person in a meeting.

of those people in their chairs, such as if they look around or lean to one side to talk to someone. The "Standing up" and "Sitting down" states represent the transitions from a "Sit still" and "Stand still" state. These are the actual activities involved with getting up from one's chair and taking one's seat, respectively. Finally, once a person is in the "Stand still" state, they can "Walk left" and "Walk right".

Conceptually, the "Standing up" and "Sitting down" states could be modeled as transitions rather than as actual states, but from observing meeting records, we have seen that people spend anywhere from 3-10 frames in each of these states. This, we feel, warrants that they be treated as states of the system rather than as some sort of transition condition.

Several of our assumptions about how people move in their environment directly dictate the structure for the DBN model. For instance, we have to assume that once people start a particular motion, they will continue doing that motion for a few frames until the observations dictate otherwise. This is done by manually setting the state probability transitions $P(X_t = i|X_{t-1} = j)$, or the

probability that state X at time t is i given that the state at time $t-1$ is j, to:

$$P(X_t = i|X_{t-1} = i) = 0.999999 \tag{1}$$

$$P(X_t = i|X_{t-1} = j) \approx 0.000001(\text{where } i! = j) \tag{2}$$

This probability distribution represents the shortcoming of the standard HMM formalism for modeling timeseries data where the states may have explicit state durations. Without these probabilities, the state transitions are much more likely to change states and to do so very quickly. While the state transition probabilities are set manually in this fashion, the probability of making an observation given a particular state, $P(Y|X)$, is learned from real data.

4.4 The Viterbi Algorithm for State Estimation

The activity recognition system models the human activity framework using a DBN. A layer of hidden nodes represents the discrete set of activities shown previously in Figure 7. The observed output nodes are continuous-value Gaussian distributions over the Δx and Δy signal values returned from the CAMEO system (note that lower-case x and y refer to image coordinates, while upper-case X and Y refer to the state and observation probabilities for a DBN as described next).

Traditionally, the Forward algorithm [3] is used to return the most likely state given an observation sequence. The Forward algorithm computes the most likely state at each timestep by summing the probabilities of the previous states. However, if the state transition $P(X_t = i|X_{t-1} = j) = 0$ at a given timestep, the Forward algorithm can still return that the most likely state at time $t-1$ is j and the most likely state at time t is state i. This is a problem for our model since there are numerous states that do not link to each other.

An algorithm which addresses this shortcoming and provides a more accurate estimate of the model's state sequence is the Viterbi algorithm [3]. Viterbi is a dynamic programming algorithm which takes into account the state transition probabilities to generate the most likely state sequence that could have generated the observation sequence. This is more formally defined as:

$$q_{1:t} = arg \max_{q_1:t} P(q_{1:x}|y_{1:t}) \tag{3}$$

where q_t is the most likely state at time t. Thus, for each timestep, Viterbi computes a term $\delta_t(j)$, defined as:

$$\delta_t(j) = \max_i \left[\delta_{t-1}(i)P(X_t = j|X_{t-1} = i)\right] P(y_t|X_t = j) \tag{4}$$

which is initialized as:

$$\delta_0(i) = P(X_0 = i)P(y_0|X_0 = i) \tag{5}$$

Additionally, the index of the most likely state at time $t-1$ to transition into each state at time t is computed and stored in a table $\psi_t(j)$ as follows:

$$\psi_t(j) = arg \max_i \left[\delta_{t-1}(i)P(X_t = j|X_{t-1} = i)\right] \tag{6}$$

Finally, after the entire observation sequence has been analyzed in this fashion, the state sequence is obtained by backtracking over the table of ψ values ranging from $t = 0$ to $t = T$:

$$q_T = arg\max_i [\delta_T(i)]$$
$$q_t = \psi_{t+1}(q_{t+1}) \tag{7}$$

4.5 Fixed-Length Backtracking Extension to Viterbi

Viterbi is a very powerful algorithm because it is able to employ the entire observation sequence at once to correct for noise that might corrupt the observation sequence. However, because of this, Viterbi requires the full observation sequence to be obtained before it is able to backtrack to resolve the complete state sequence. This will not work with CAMEO when it performs inference in real-time.

In order to achieve the benefits of Viterbi while not requiring that CAMEO wait for the entire state sequence, we have defined a hybrid approach by which backtracking is only done on the latest k states. Thus, when the observation at time t is received, the state at $t - k$ is inferred. In a real-time system, this fixed-window approach will cause a delay in the state estimate, but as long as the delay is not too long, the estimate may still be useful to act upon. Thus, the value of k represents a tradeoff in accuracy and estimation lag in a real-time system. This relationship will be explored further in the experimental results section.

5 Experimental Results

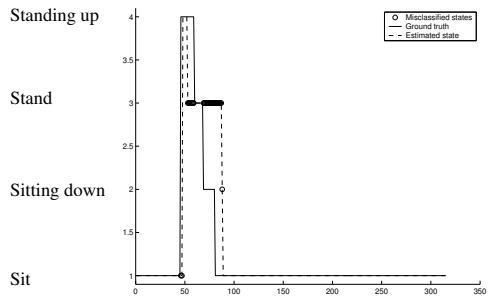

Fig. 8. Classified actions of a person standing up and sitting down. Of the 315 timesteps in this dataset, only 29 of the states were misclassified. The Y-axis represents the state where, 1="Sit", 2="Sitting down", 3="Stand", and 4="Standing up". The X-axis is the frame count. The data was captured at 15 fps.

Figure 8 shows some results from the dynamic Bayesian network action recognition system as compared to hand-labeled ground truth. Data was collected from a short sequence of video that showed a single person standing up and then sitting down again.

In this figure, the solid line shows the hand-labeled ground truth of the person's activities, the dashed line shows the estimated activities, and the circles indicate states that were misclassified. Of the 315 images encoded with person tracked data, only 29 of the states were misclassified. Most of these occurred during the transitions from the "stand" state through the "sitting down" state to the "sit" state. This is primarily due to variances in the way that people move around. However, while the alignments of the activities were slightly off from ground truth (their starting and stopping times), the fact that the person stood up and sat down was not missed.

The performance of the fixed-window Viterbi algorithm with different values of k was evaluated. Figure 9 shows the effect of the window size on the accuracy of the inferred state sequence (the larger the window size, the more accurate the results are). Compare this with the Forward algorithm alone which scored only 67.95% correct. This high error rate of the Forward algorithm is caused primarily by the inference algorithm making illegal state transitions.

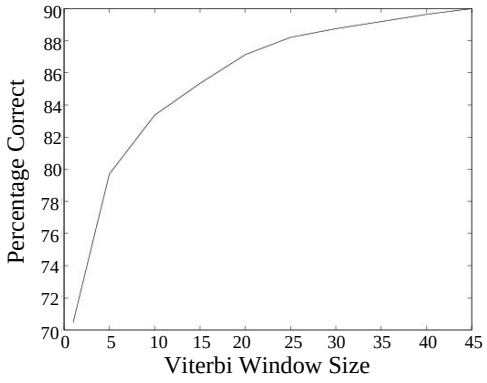

Fig. 9. Plot of the effect that the amount of backtracking in the fixed-window Viterbi algorithm has on the accuracy of the state inference. The larger the window size, the more accurate the state sequence from the ground truth. As a comparison, the Forward algorithm alone was only 67.95% correct on this dataset.

For each frame of video the Viterbi algorithm uses to improve its inference, the longer the time lag between the observation and the state inference. The full CAMEO system, which includes face detection, tracking, and face recognition, typically can track 5-6 people while running in real-time at approximately 3-4 frames per second. This means that each frame of video adds between 0.25-0.3

seconds of latency to the state inference. Any decision process that makes use of CAMEO's state estimation must always take this lag into account.

Finally, the images shown in Figure 10 graphically illustrate the state estimation algorithm running on data captured from CAMEO. The panoramic images have been truncated to only show the areas of interest. In this image sequence, two people enter a room and sit down. Their states are successfully tracked as transitioning through walking, stand still, sitting down, and sit still.

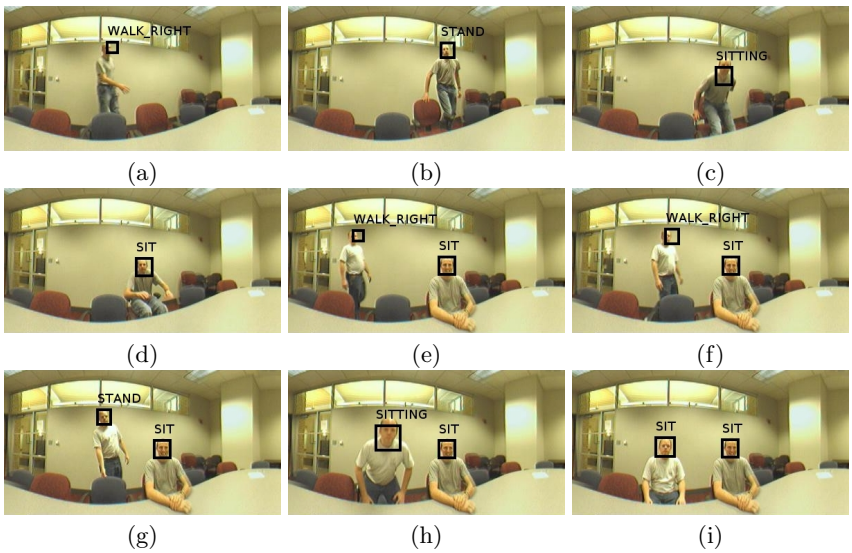

Fig. 10. Sample video captured from CAMEO in which the people's motions are captured by the face detector and classified by the Viterbi algorithm.

The activity recognizer is dependent upon the ability for CAMEO to successfully track people. If the tracker loses a person due to occlusions, or because the person left the room, then the data stream for the activities will be truncated or possibly erroneous. Additionally, if the person moves very slowly, their relative displacements will be lost in the noise model for the states which represent stationary positions. Because of this, the activity recognizer might not classify a transition from standing to sitting (or vice versa). In this case, the activity recognizer will be out of alignment with the person's true position and will not catch up until another transition occurs that the recognizer successfully detects. Such "missed" transitions could be addressed by adding additional features to the classifier, such as using the motions of others in the environment to disambiguate the current state of a person. We are currently looking at ways to enhance our recognition algorithms to use this additional information.

6 Summary

CAMEO is designed to observe and infer state information about people in meetings. To do so, it requires minimal room instrumentation and calibration to operate. Because very little *a priori* information is known about the position (and possible occlusions) of people in the meeting, CAMEO makes use of a robust identification scheme to find and track people's faces in the environment. CAMEO tracks motions of faces and feeds their displacements into a Dynamic Bayesian Network-based classification system used to infer the tracked person's state. This classifier uses a model of human behavior which is encoded into the Bayesian Network's hidden state's conditional probability distribution. The parameters for the observed states are learned from labeled data. A fixed-sized backtracking implementation of Viterbi was implemented to recover the most likely state information from the data. We have shown experimental results from a state model illustrating how CAMEO is able to infer the state of individual people in the meeting. We have also discussed the tradeoffs of this approach between accuracy and real-time operation as well as describing some limitations of the algorithm and future directions of the research.

References

1. Rybski, P.E., de la Torre, F., Patil, R., Vallespi, C., Veloso, M.M., Browning, B.: Cameo: The camera assisted meeting event observer. In: International Conference on Robotics and Automation, New Orleans (2004)
2. Murphy, K.: Dynamic Bayesian Networks: representation, Inference and Learning. PhD thesis, UC Berkeley, Computer Science Division (2002)
3. Rabiner, L.R.: A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of the IEEE **77** (1989) 257–286
4. Hamid, R., Huang, Y., Essa, I.: ARGMode – activity recognition using graphical models. In: Conference on Computer Vision and Pattern Recognition Workshop. Volume 4., Madison, WI (2003) 38–44
5. Hong, P., Turk, M., Huang, T.S.: Gesture modeling and recognition using finite state machines. In: Proceedings of the Fourth IEEE International Conference and Gesture Recognition, Grenoble, France (2000)
6. Nguyen, N., Bui, H., Venkatesh, S., West, G.: Recognizing and monitoring high level behaviours in complex spatial environments. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition. (2003)
7. Darrell, T., Gordon, G., Harville, M., Woodfill, J.: Integrated person tracking using stereo, color, and pattern detection. International Journal of Computer Vision **37** (2000) 175–185
8. Han, K., Veloso, M.: Automated robot behavior recognition applied to robotic soccer. In Hollerbach, J., Koditschek, D., eds.: Robotics Research: the Ninth International Symposium. Springer-Verlag, London (2000) 199–204 Also in the Proceedings of IJCAI-99 Workshop on Team Behaviors and Plan Recognition.
9. Bruce, J., Bowling, M., Browning, B., Veloso, M.: Multi-robot team response to a multi-robot opponent team. In: Proceedings of ICRA'03, the 2003 IEEE International Conference on Robotics and Automation, Taiwan (2003)