ARGMode — Activity Recognition using Graphical Models*

Raffay Hamid, Yan Huang, Irfan Essa

Georgia Institute of Technology
GVU Center / College of Computing
Atlanta, GA 30332-0280 USA
{raffay, huangy, irfan}@cc.gatech.edu
http://www.cc.gatech.edu/cpl/projects/ARGMode/

Abstract

This paper presents a new framework for tracking and recognizing complex multi-agent activities using probabilistic tracking coupled with graphical models for recognition. We employ statistical feature based particle filter to robustly track multiple objects in cluttered environments. Both color and shape characteristics are used to differentiate and track different objects so that low level visual information can be reliably extracted for recognition of complex activities. Such extracted spatio-temporal features are then used to build temporal graphical models for characterization of these activities. We demonstrate through examples in different scenarios, the generalizability and robustness of our framework.

1. Introduction and Related Work

Video-based recognition activity is a challenging problem as most activities are dynamic and occur in environments with significant visual clutter. The former poses challenges in representing these spatio-temporally changing activities adequately. While, the later presents difficulties in terms of extracting low level visual information to track the movements, and then to build the representative stochastic models.

Over the past few years, stochastic finite state machines such as Hidden Markov Models (HMMs) have been quite popular for recognizing gestures [1, 23], sign language [26, 24] and actions [17, 27, 3]. Some work also attempts to incorporate the underlying structure of the target activity to improve upon the performance of stochastic models (using coupled HMMs [3] and hierarchical HMMs [8]). Additionally, work on recognizing human interaction with surrounding objects proposes the use of context to recognize activities [18]. However, most systems for activity recognition rely on only one handed interactions with one object at a time (including [21]). This constraint of using only one hand at a time restricts the system from recognizing more complex activities as most of them involve both the hands acting simultaneously such actions as 'opening', 'closing', *etc.*

In most activity recognition approaches, the focus is on the modeling and recognition. These methods use simple background subtraction and color based segmentation of different objects. For example, Moore and Essa [17, 18] use color table to locate person's hands when person's location is assumed to have stabilized which is inferred from background subtraction. Such simple low-level measures fail when the background changes or when occlusion exists. Hongeng and Nevatia [10] explore knowledge about ground plane and events to track objects robustly. However, it's not suitable for a general purpose activity recognition system due to its dependence on specific scenarios and contexts. KidsRoom [2, 11] is also a context-based tracking system based on "closed-world regions" in which the specific context of what is in the region is assumed to be known. In contrast, we use a particle filter to track movements, without context knowledge of the image sequence, which enable our system to be more robust and more general.

Ever since the incorporation of particle filter into computer vision community [13], there has been a lot of extension to it for tracking purpose. Contours has been used by Isard and Blake [13] to track foreground objects which are modelled as curves. However, contours tends to fail in the general purpose tracking tasks where the objects of interest can be of any kind so that there is no prior known nature such as outlines about the objects to be tracked. Color histogram is an appealing feature for such tracking tasks. Color-based particle filter trackers have been recently proposed in [5, 6, 20]. Such trackers are robust and versatile at a modest computational cost. However, color does not incorporate spatial adjacency of pixels in the image and may

*This work was supported in part by a National Science Foundation ITR Grant #0121661 and DARPA HumanID Grant #F49620-00- 1-0376.

lead to inaccuracies in the presence of large background blobs that exhibit similar color patterns as the moving objects. In this case, orientation histogram that describe low-level the shape information on the basis of edges, is a robust and fast feature. Therefore, we combine color histogram and orientation histogram as the tracking features.

In our framework, we incorporate color and shape based particle filter to track scene objects. Using this tracking data we extract various spatio-temporal features such as the relative distances between objects and the agents as well as their velocities. These features are used to train Dynamic Bayesian Networks (DBNs) and then used for recognition. To show the generalizability for our framework, we experiment on different scenarios with varying levels of difficulty, which include activities in classroom, living room, and Glucose control monitoring system in Section 3. These experiments are intentionally chosen as the complexity of number of objects being tracked and the order of actions observed to recognize and activity is different in each scenario.

2. ARGMode

Our system incorporates a bottom up architecture that involves cascading layers of multiple object tracking, feature extraction and recognition methods. In the following section, these steps are presented in detail.

2.1 Tracking

We build our tracking system on the well-known *condensation* algorithm [13, 12] to provide more robust tracking support. In a typical deterministic tracking framework, starting from the final location in the previous frame, the current frame is searched for a region whose content best matches a reference model. However, it works poorly in dense visual clutter whose elements may mimic parts of foreground features and give rise to multi-modal density functions. *Condensation* uses random sampling techniques to handle such situation.

In *condensation* algorithm, multi-modal state distribution is discretely sampled by a finite set of particles (or samples). A *state* is a set of parameters aligning an object model with the input data. Since we associate a rectangular box with each object of interest in the image plane, as shown in Figure 1, a vector of four values $x_t = (t_x, t_y, r_\theta, s)$ is employed to represent the state. t_x and t_y are the translation along x and y directions respectively, θ is the angle of rotation in the image plane and s is the scaling factor applied to both x and y direction. Currently, we only deal with the situations where objects of interest do not disappear from the camera view.

To propagate the past particle set to the future particle set, three steps are performed at each time-step.

Selection: A sample set $(s_t^{'n}, n = 1, \cdots, N)$ is generated from the prior density $p(x_{t-1}|z_{t-1})$ using factored sampling techniques.

Prediction: $(s_t^n, n = 1, \cdots, N)$ is generated from $(s_t^{'n}, n = 1, \cdots, N)$ according to a dynamical model. In our case, dynamics is applied to state parameters as:

$$s_t^n = s_t^{'n} + A \cdot v_{t-1} + B \cdot w_t, w_t \sim N(0, \Sigma) \qquad (1)$$

where v_{t-1} is the velocity vector obtained from the previous steps, A and B are matrices representing the deterministic and stochastic components of the dynamical model respectively. In all of our experiments, A is set to an identity matrix while B is set to a diagonal matrix where the diagonal elements associated with t_x and t_y are set as 1 pixel/frame, the diagonal element associated with r_θ is set as 5 degrees/frame, and the diagonal element associated with s is set as 0.1/frame. w is a noise term defined by a vector of standard normal random variates.

Measurement: Compute the probability of the state $p(z_t|x_t = s_t^n)$ and normalize the probabilities of all particles so that they sum to one, producing weights $\pi_t^n = p(z_t|x_t = s_t^n)/\Sigma_{i=1}^N p(z_t|x_t = s_t^i)$.

Based on the discrete approximation of $p(z_t|x_t = s_t^n)$, different estimates of the "best" state at time t can be devised. We use the state with the maximum likelihood $\hat{x_t} = \arg\max_{s_t^n} p(z_t|x_t = s_t^n)$ as the tracker output at time t.

Within this framework, the input image sequences are searched against the fixed image-based models of the objects that are extracted from the first frame and kept frozen. We define the probability of a state in terms of how well it matches the data and employ two statistical features, namely color histogram [25] and orientation histogram [9], to measure such similarity. Both of these two features are computationally simple and efficient and are insensitive to variation in image scaling, rotation and translation.

Color Model: Color model is obtained by histogramming techniques in YUV color space. Since intensity is more sensitive to lighting variance, Y channel is more coarsely sampled than the other two. The color similarity between the image template and the state is given by histogram intersection:

$$S_c^{H_{I_t^n}, H_M} = \frac{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} \min(H_{I_t^n}(i,j,k), H_M(i,j,k))}{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} H_M(i,j,k)} \qquad (2)$$

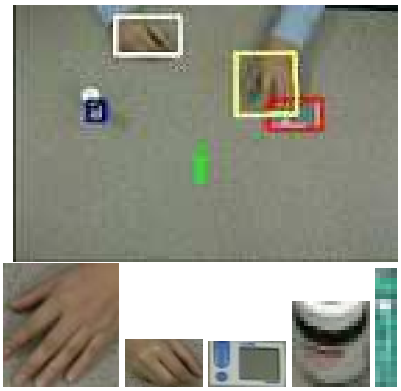


Figure 1: Regions of Interest: The image in the first row is an input frame with five objects of interest. The second row shows templates for each object of interest, including left hand, right hand, glucometer, solution bottle and testing strip. For visualization purpose, bounding boxes have been thickened. The same modification applies to all of the following figures.

where $H_{I_t^n}(i,j,k)$ and $H_M(i,j,k)$ are the normalized color histograms of the state and the image template respectively. The value of $S_c^{H_{I_t^n},H_M}$ lies in the interval [0,1].

Since color histogram is invariant under rotation and takes very few changes when there's little scaling, it's an efficient and effective feature in our application.

Shape Model: We describe shape information on the basis of significant edges. Sobel derivative filters are used to retrieve edge points, and an histogram of edge orientation is employed to represent shape attribute. Orientation histogram is rotation invariant by histogram shifting, while scale invariance can be achieved by normalizing the histogram with respect to the number of edge points in the image. Moreover, the fact that local orientation measurements are less sensitive to lighting changes makes orientation histogram more appealing in tracking problems. Histogram intersection (similiar to Eq. 2) is used to calculate the shape similarity $S_s^{H_{I_t^n},H_M}$ between the image template and the state.

Integration of Color and Shape Features: The similarities given by each individual feature are integrated by combining the associated similarity values. Let S_c be the similarity index on the basis of color ($S_c^{H_{I_t^n},H_M}$), and S_s be the similarity index on the basis of shape ($S_s^{H_{I_t^n},H_M}$), we define the integrated similarity index $S_f^{H_{I_t^n},H_M}$ as:

$$S_f^{H_{I_t^n},H_M} = \frac{w_c * S_c + w_s * S_s}{w_c + w_s} \tag{3}$$

where w_c and w_s are the weights assigned to the color-based similarity and shape-based similarity respectively. In most of our experiments illustrated in Section 3, both w_c and w_s are set to 1 except for the BOOKSHELF experiment where w_c is set to 2 and w_s is set to 1.

The larger $S_f^{H_{I_t^n},H_M}$ is, the more similar the distributions are. Therefore, we define the distance between two distributions as $d = \sqrt{1 - S_f^{H_{I_t^n},H_M}}$ and $p(z_t|x_t = s_t^n)$ is estimated via:

$$p(z_t|x_t = s_t^n) = exp(-\lambda d^2) \tag{4}$$

Lacking a way to estimate satisfactorily the parameter λ, we fix it to the same value $\lambda = 5$ in all the experiments in Section 3.

To tackle multi-target tracking problem, most of the previous efforts [16, 14, 4] explore data association to handle overlapping or occlusion. This produces more precise objects states in a very high computational cost. The state $x_t = (x_1^t \dots x_k^t)$ is defined as a concatenation of k single-object states with each element x_i^t associated with an object template to be matched. It results in high-dimensional state space which relies on very large number of particles to model a distribution.

We avoid such computational burden by placing one tracker for each object and tracking each object in parallel without concatenating their state vectors or taking potential occlusions into account. Since the absence of multiple identical objects is assumed (even in case of two hands, the left hand and right hand have different orientations), parallel trackers will not collapse into one object. Such tracking can be considered both a strength and a weakness. The strength is that the computational complexity is reduced dramatically. The weakness is that the ability to deal with occlusion is not as robust as the previous approaches. However, since occluded object would still be the best match with its unoccluded reference model in most of the cases, we assume this will not seriously affect the tracker in case of occlusion or overlapping. Moreover, since outputs of the trackers will be passed to provide low-level visual information for recognizing activities whose patterns do not require rigorous trajectories, we believe such tracking performance would not seriously affect the recognition results either.

2.2 Inference and Learning

For technical completeness we first give a brief overview of the Dynamic Bayesian Networks. This is followed by a description of how we select the feature vectors and topology of our network. Dynamic Bayesian Networks (DBNs) are stochastic Dynamic Acyclic Graphical Models (DAGs),

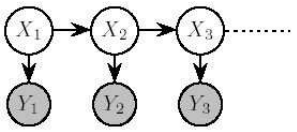


Figure 2: DBN equivalent of HMM. X represents the hidden state, while Y represents the observations.

which can model time-series data in a stochastic framework [19]. A DBN is defined to be a pair $(B_1, B_{\rightarrow})$ where B_1 is a Bayesian Network which defines the prior $P(Z_1)$ where Z is random variable, and $B_{\rightarrow}$ is a two slice temporal Bayesian Net (2TBN) that defines $P(Z_t|Z_{t-1})$ by means of a DAG as follows:

$$P(Z_t|Z_{t-1}) = \prod_{i=1}^{N} P(Z_t^i|P_a(Z_t^i)), \tag{5}$$

where Z_t^i is the node at time t and $P_a(Z_t^i)$ are the parents of Z_t^i in the graph. N represents the number of hidden states. The parents of a node are assumed to follow the first order Markov assumption.

$$X_{t+1} \perp X_{t-1}|X_t \text{ and } Y_t \perp Y_t'|X_t \text{ for } t' \neq t \tag{6}$$

where X_t = hidden state and Y_t = observation vector at time t. Figure 2 shows a DBN equivalent of a Hidden Markov Model.

Inference in DBNs is used to compute the marginals, *i.e.*, $P(X_t^i|y_{1:t})$. We briefly describe the forwards-backwards (FB) algorithm for calculating the inference for DBNs. A detailed explanation of this algorithm for HMMs can be found in [22] which is equally valid for DBNs. The idea of FB algorithm is to calculate the forward co-efficient $\alpha_t(i) \equiv P(X_t = i|y_{1:t})$ in the forward pass and $B_t(i) \equiv P(y_{t+1:T}|X_t = i)$, and then to combine them to produce the final answer $\gamma_t(i) \equiv P(X_t = i|y_{1:T})$,*i.e.*,

$$P(X_t = i|y_{1:T}) = \frac{P(y_{t+1:T}|X_t = i)}{P(y_{1:T})} \cdot P(X_t = i|y_{1:t})$$
$$\gamma_t(i) \propto \alpha_t(i)\beta_t(i) \tag{7}$$

The EM algorithm for learning that we describe here is a generalization of Baum-Welch algorithm [22]. This idea can be applied to Bayesian Nets and can be easily extended to DBNs. For known structure and partially observable model, the log-likehood is given by $L = \Sigma_m log P(D_m)$, where D is the data and m is the number of independent cases. The basic idea behind EM is to get a lower bound on L (the E-step), and then to iteratively maximize this lower bound (the M-step) [7, 15, 28].The reason for using the DBN equivalent of HMMs is that besides their performance advantages, DBNs provide a more general framework in which more intricate inherent model structure of the problem can be more easily incorporated. Exploiting the inherent structure of the recognition problem in different domains can be more efficiently achieved by leveraging the flexibility that a more general framework such as DBNs provide. Now we describe how we select our observation vectors and the topology of our graphical model.

Observation Vectors: Visual information available to determine which object is being acted upon in a given instant of time, requires tracking the relative positions of the actor and the object. Unlike some of the previous work done in this domain [17] where the observation vectors are only the positions of the hands, we use the relative distances between the hands and the objects of interest and their velocities. There are two reasons for doing so. Firstly, measuring only the x,y position of the hands as feature vectors makes the system too dependent on the setup of the scenario. Secondly, feature vectors based on position only, do not give enough information for modelling of relatively complex actions.

Hidden States: The hidden states in our system are atomic actions such as 'holding', 'moving', 'inserting' *etc.* The observation vectors in different states can have different observation vectors, *i.e.*, the observation vectors for the case of 'holding' is the relative distance between the hand and the object of interest, while in case of 'moving' it is the velocities of the object of interest and the relative distances between them. These atomic actions are purposely designed to be simple enough so that they can be extended to different scenarios. The topology of the model is left-to-right. The reason for not using a fully ergodic structure in place of a left-to-right structure is that they tend to require much more training data for correct parameter estimation of state transition matrix. Moreover, in different scenarios, due to the prior information about the sequence that the activity is supposed to follow, we assign requisite probabilities as initial conditions to the state transition matrix to achieve parameter learning quickly. We present in Section 3 scenario 1 that by using the prior knowledge about the context of the scenario, we can achieve high level knowledge based on the simple activities.

3. Experiments and Results

To demonstrate the generalizability and robustness of our system, we provide examples of our approach in three different scenarios, classroom, living room and Glucose Control monitoring system. The first two examples act as a

preamble to the third one which incorporates salient features of the first two. These activities involve partial occlusions which are effectively dealt by our framework. Each activity example mentioned below is a 360x240 color video. As for the histogram parameters used in the tracking stage, the Y and U channels of the color histogram for all the videos has a bin size of 8 while for intensity channel V it is 16. The orientation histogram bin size for all the videos is 20 degrees. All videos are available on project web page listed earlier.

Scenario 1: Classroom Using the prior knowledge about the context of the activity, we can obtain high level knowledge of what is going on in the video. Leveraging this fact, we find out what activity is being performed by the actor on the white board in a classroom based on the motion of his hands. In our example we consider 3 main activities in all, Activity1: writing, Activity2: drawing and Activity3: erasing. Activity1 is constituted further of three subactivities, *i.e.*, Action 1a: writing English, 1b: Urdu and 1c: Chinese.

We are able to distinguish in which language the actor is writing, based on the direction of motion of the user as we have a priori knowledge that English is written from left to right, Urdu/Arabic from right to left, while Chinese can be written from top to bottom. Thus the features in this case are the velocities of the users' hand. The number of states in this scenario for each model are 2, i.e. writing a particular language and waiting.

The distinction between the Activity 2 and Activity 3 is made purely on the bases of the context since for both these activities, the pattern of movement of hand is the same. However, the fact whether the actor is holding an eraser or not distinguishes whether he is drawing or erasing. We estimate if the user is holding an eraser or not by the maximum probability of the samples used to track the eraser. If this probability is significantly less, then we conclude that the user is not holding an eraser. We used 20 video sequences, four for each activity to train our models. These activities were performed by different users with different styles of writing. We further used 15 test videos, out of which 10 were synthetically generated by adding scaled uniform random noise to the features of the 5 real test videos. We combined these videos with the 20 training videos to test a total of 35 sequences. This was done primarily to have enough data to allow for testing. For all the videos in this example, the number of particles in our tracking algorithm was only 100. Figure 3 shows key-frames representing each of the five activities being tracked and recognized in this example. The following table shows the results of the recognition of these sequences:

	English	**Urdu**	**Chinese**	**Drawing**	**Erasing**
English	85.7%	0	0	14.3%	0
Urdu	0	85.7%	0	14.3%	0
Chinese	0	14.3%	85.7%	0%	0
Drawing	14.3%	0	28.6%	57.1%	0
Erasing	0	0	0	0	100%

Table 1: The percentage distribution of recognition of the different sequences recognized when given as the input data to different models.

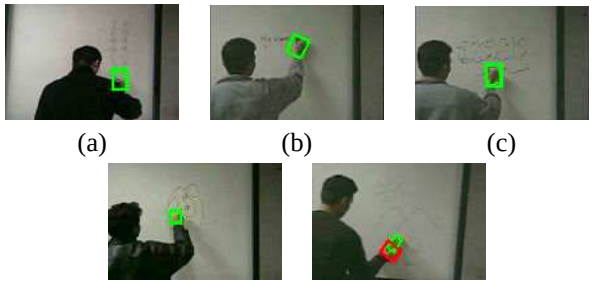


Figure 3: Key-Frames showing different activities including writing in Chinese (a), English (b), Urdu (c), Drawing (d), and Erasing (e) being performed in a classroom. The trajectories of the motion of hand and the context whether the user is holding a pen or an eraser help recognize the different activities.

Scenario 2: Living Room This is a two class classification example in which the actor is moving different objects on a bookshelf in different order, *i.e.*, moves the cup first in one and moves the doll first in the other. The actor is only using one hand to move three objects in the cup-board, so we have a total of four objects of interest. These activities are defined by the sequence in which the actor moves the objects therefore the relative distances of the objects and the hands are the extracted feature vectors. There are four hidden states, three representing whether a person is moving any one of the three objects while the fourth representing the 'roaming' state which models the duration when the hand is not moving any object.

This example can be easily extended to other everyday activities such as dusting, cleaning *etc*. As shown in the above scenario, adding context of the environment can even allow us to further extend this scenario to even more intricate scenarios. We used 5 video sequences for each of the two activities. We further produced 6 test videos and combined them with the 10 train videos to test a total of 16 sequences. For all the videos in this example, the number of particles in our tracking algorithm was only 250. The reason for an increase in the number of particles is the increase in the visual clutter in the scene. Figure 4 shows key-frames representing the two activities being tracked and recognized in this example. The following table (Table 2 shows the re-

Figure 4: Key-Frames Showing different activities of moving different objects in different sequences being performed in a living room.

sults of the recognition of these sequences:

	Activity 1	**Activity 2**
Activity 1	87.5%	12.5%
Activity 2	33.3%	66.7%

Table 2: The percentage distribution of recognition of the different sequences recognized when given as the input data to the two different models. Activity 1 is where comprised of moving the doll, moving the red box, and moving the cup in the order. Activity 2, has the same actions, but in a different order.

Scenario 3: Glucose Control Monitoring System As our final example, we present the preliminary results of our system to recognize the different activities performed by the user while calibrating blood glucose measuring meter (Glucometer) used to measure ones blood glucose level. Since most of the subroutines in this test follow some temporal structure, we believe that systems based on stochastic temporal models can play a significant role to improve the lives of many people by recognizing if the patient is performing the tasks in the requisite manner or not.

Here we present an important Glucometer calibration test called Litmus solution test. The activity is defined as: (1) Turn on the Glucometer, (2) Insert the strip in the Glucometer, (3) Open the bottle of litmus solution, and (4) Bring the bottle close to the meter and drop a few drops of this solution on the strip.

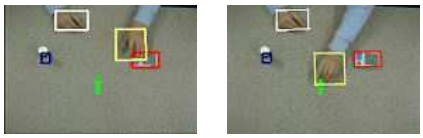
This activity is quite complex since it requires multiple objects as well as the actions involving both hands at the same time such as opening the bottle. There are total number of five objects including the glucometer, the testing strip, the solution bottle and both the hands of the user. The relative distances between these objects as well as their velocities form the feature vectors in this experiment. Each sub-activity consists of different states. The first sub-activity 'turn on glucometer' consists of two states, *i.e.*, whether the hand is close enough to the glucometer and the second is the 'roaming' state. The second sub-activity, 'insert strip in glucometer' consists of three states, *i.e.*, 'hold the strip', 'bring it close to the glucometer' and the third is 'roaming'. The third sub-activity 'open the bottle of litmus solution' consists of three states, 'left and right hands holding bottle simultaneously', 'either of the two hands move away from the bottle while the other keeps holding the bottle' and 'roaming'. Similarly the fourth sub-activity has three states, 'either of the hands holding the bottle', 'bottle brought close to the glucometer' and 'roaming'. We test our trained models on activities in which crucial parts of the entire exercise were not performed. In one of the test videos, the user does not turn on the meter and performs the rest of the activities in order while in the second test, the actor forgets to drop the litmus solution on the strip. We used 10 training videos in all of which the activity was performed correctly. We further took 6 videos, two performing the activity correctly while the rest of the four performing the test wrongly. All videos use 250 particles for tracking each of the five objects. Figure 5 shows key-frames representing critical portions of the activity being tracked and recognized in this example. The following table shows the results of the recognition of these sequences:

	Correct Activity	**Wrong Activity**
Correct Activity	92.3%	7.7%
Wrong Activity 1	0%	100%
Wrong Activity 2	33.3%	66.7%

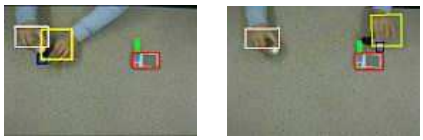
Table 3: The results when the different test sequences are given to the trained model. Wrong activity 1 is where the user missed the step of turning on the glucometer, and wrong activity 2 is where the user did not pour the solution.

4. Conclusions and Future Work

This paper presents a framework for recognizing human activities in different scenarios by using particle filter tracking and DBN based recognition. The example of Glucometer is an ongoing project and in the future more complex DBNs will be looked into to model the activity. Moreover the possibility that there might be an inherent hierarchy in the structure of this problem will be looked into by experimenting with Hierarchical HMMs. For recognizing complex activities over long periods of time, we believe that we should make use of the context to improve the robustness of our system. Currently when tracking multiple targets, we totally ignore the data association to avoid high computational cost. However, although such approach works for activity recognition tasks for the most of the cases, it tends to



a: frame 37 b: frame 93



c: frame 226 d: frame 284

Figure 5: Key-Frames Showing different activities being performed in litmus solution test: turning on (a); inserting the test strip in the glucometer (b); opening the solution bottle (c); and dropping solution on the test strip (d).

fail under full occlusion. New techniques will be explored and incorporated into our system to tell when full occlusion is going to happen and to dynamically take data association into account if full occlusion is about to happen.

References

[1] A. F. Bobick and A. D. Wilson. A state based approach to the representation and recognition of gesture. *PAMI*, 19(12):1325–1337, December 1997.

[2] A.F. Bobick, S.S. Intille, J.W. Davis, F. Baird, C.S. Pinhanez, L.W. Campbell, Y.A. Ivanov, A. Schuette, and A. Wilson. The kidsroom: A perceptually-based interactive and immersive story environment. In *Vismod*, 1996.

[3] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *CVPR*, 1997.

[4] Patrick Prez Carine Hue, Jean-Pierre Le Cadre. A particle filter to track multiple objects. In *WOMOT'01*, page 61, 2001.

[5] H.T. Chen and T.L. Liu. Trust-region methods for real-time tracking. In *ICCV01*, pages II: 717–722, 2001.

[6] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *CVPR00*, pages II: 142–149, 2000.

[7] D. Koller E. Bauer and Y. Singer. Batch and on-line parameter estimation in bayesian networks.

[8] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, 32(1):41–62, 1998.

[9] W. Freeman and M. Roth. Orientation histogram for hand gesture recognition. In *Int'l Workshop on Automatic Face-and Gesture-Recognition*, 1995.

[10] S. Hongeng and R. Nevatia. Multi-agent event recognition. In *ICCV01*, pages II: 84–91, 2001.

[11] S.S. Intille, J.W. Davis, and A.F. Bobick. Real time closed world tracking. In *Vismod*, 1997.

[12] M. Isard. Visual motion analysis by probabilistic propagation of conditional density, phd thesis, oxford university, 1998.

[13] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.

[14] E. Koller-Meier and F. Ade. Tracking multiple objects using the condensation algorithm. *Journal of Robotics and Autonomous Systems*, 34:93–105, 2-3 2001.

[15] S. L. Lauritzen. Propagation of probabilities, means and variances in mixed graphical association models. *Journal of American Statistical Association*, 87(420):1098–1108, 1992.

[16] J.P. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. *IJCV*, 39(1):57–71, August 2000.

[17] D. Moore, I. Essa, and M. Hayes. Context management for human activity recognition. In *Proceedings of Audio and Vision-based Person Authentication*, 1999.

[18] D.J. Moore, I. Essa, and Monson H. Hayes. Exploiting human actions and object context for recognition tasks. In *ICCV*, pages 80–86, 1999.

[19] K. Murphy. Dynamic bayesian networks: Representation, inference and learning, phd thesis, uc berkeley, july 2002.

[20] P. Prez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *ECCV02*, page I: 661 ff., 2002.

[21] M. Shah R. Russo. A computer vision system for monitoring production of fast food. In *ACCV*, 2002.

[22] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Alex Weibel and Kay-Fu Lee (eds.), Readings in Speech Recognition*, pages 267–296, 1990.

[23] J. Schlenzig, E. Hunter, and R. Jain. Recursive identification of gesture inputs using hidden markov models. In *WACV94*, pages 187–194, 1994.

[24] T. Starner, J. Weaver, and A.P. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *PAMI*, 20(12):1371–1375, December 1998.

[25] Michael J. Swain and Dana H. Ballard. Color indexing. *IJCV*, 7(1):11–32, 1991.

[26] C. Vogler and D. Metaxas. A framework for recognizing the simultaneous aspects of american sign language. *CVIU*, 81(3):358–384, March 2001.

[27] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using a hidden Markov model. In *CVPR1992*, pages 379–385, 1994.

[28] N. L. Zhang. Irrelevance and parameter learning in bayesian networks. *Artificial Intelligence, An International Journal*, pages 359–373, 1988.