Appears in *Computer Vision and Pattern Recognition (CVPR 99)*, Ft. Collins, CO, June, 1999.

Vision-Based Speaker Detection Using Bayesian Networks

James M. Rehg
Cambridge Research Lab
Compaq Computer Corp.
Cambridge, MA 02139
rehg@crl.dec.com

Kevin P. Murphy
Dept. of Computer Science
University of California
Berkeley, CA 94720
murphyk@cs.berkeley.edu

Paul W. Fieguth
Dept. of Systems Design Eng.
University of Waterloo
Waterloo, Ontario N2L 3G1
pfieguth@ocho.uwaterloo.ca

Abstract

The development of user interfaces based on vision and speech requires the solution of a challenging statistical inference problem: The intentions and actions of multiple individuals must be inferred from noisy and ambiguous data. We argue that Bayesian network models are an attractive statistical framework for cue fusion in these applications. Bayes nets combine a natural mechanism for expressing contextual information with efficient algorithms for learning and inference. We illustrate these points through the development of a Bayes net model for detecting when a user is speaking. The model combines four simple vision sensors: face detection, skin color, skin texture, and mouth motion. We present some promising experimental results.

1 Introduction

Human-centered user-interfaces based on vision and speech present challenging sensing problems in which multiple sources of information must be combined to infer the user's actions and intentions. Statistical inference techniques therefore play a critical role in system design. This paper addresses the application of Bayesian network models to the task of detecting whether a user is speaking to the computer. This is a challenging task which can make use of a variety of sensors. It is therefore a good testbed for exploring statistical sensor fusion techniques. Speaker detection is also a key building block in the design of a conversational interface.

Bayesian networks [16, 9] are a class of probabilistic models which graphically encode the conditional independence relationships among a set of random variables. Bayesian networks are attractive for vision applications because they combine a natural mechanism for expressing domain knowledge with efficient algorithms for learning and inference. They have been successfully employed in a wide range of expert system and decision support applications. One example is the Lumière project [6] at Microsoft, which used Bayesian networks to model user goals in Windows applications.

In this paper we demonstrate the use of Bayesian networks for visual cue fusion. We present a network, shown in Figure 4(c), which combines the outputs of four simple "off-the-shelf" vision algorithms to detect the presence of a speaker. The structure of the network encodes the context of the sensing task and knowledge about the operation of the sensors. The conditional probabilities along the arcs of the network relate the sensor outputs to the task variables. These probabilities are learned automatically from training data.

While Bayesian network models are not yet in widespread use within the computer vision community, there is a growing body of work on their application to object recognition [11], scene surveillance [2], video analysis [22, 7], and selective perception [19]. Much of this earlier work relies upon expert knowledge to instantiate network parameters. In contrast, we have explored the ability to learn network parameters from training data. Learning is a key step in fusing sensor outputs at the data level.

This paper makes two contributions. First, we use a series of examples to illustrate the power of Bayesian networks in combining noisy measurements and exploiting context. We present a network architecture (network F in Figure 4(b)) that can infer the frontal orientation of a face even though we have no explicit pose sensor.

Second, we present a solution to the speaker-detection problem which is based on commonly available vision algorithms and achieves a classification rate of 91% on a simple test set. This result suggests that Bayesian network classifiers can provide an interesting alternative to the standard decision tree or neural network classifiers commonly used in vision applications.

2 The Speaker Detection Task

Speaker detection is an important component of a conversational interface for a *Smart Kiosk* [17, 23, 3], a freestanding computer system capable of social interaction with multiple users. The kiosk uses an animated synthetic face to communicate information, and can sense its users with touch-screens, cameras, and microphones (see Figure 1). In this setting we would like to model and estimate a wide range of user states, from concrete attributes such as the presence of a user or whether they are speaking, to

more abstract properties such as the user's level of interest or frustration.

Figure 1: The Smart Kiosk

In a kiosk interface, speaker detection consists of identifying users who are facing the kiosk display and talking. In particular, we want to distinguish these users from others who may be conversing with their neighbors. The public, multi-user nature of the kiosk application domain makes this detection step a critical precursor to any speech-based interaction.

To solve the speaker detection task, we use a combination of four "off-the-shelf" vision sensors: the CMU face detector [20], a Gaussian skin color detector [24], a face texture detector, and a mouth motion detector. They are explained in more detail below. These components have the advantage of either being easy to implement or easy to obtain, but they have not been explicitly tuned to the problem of speaker detection.

In combining the outputs of these sensors we would like to exploit *contextual knowledge* about their performance characteristics and about the physical design of the kiosk. For example, our kiosk design aligns the camera axis with the primary viewing direction of the kiosk display. Users who want to speak to the kiosk must be facing the display and in close proximity if they expect to be heard. As a result of this camera placement, speaking users will generate frontal face images in which lip and jaw motion is visible. Thus the detection of frontal faces provides an important cue for the presence of speakers. We will show in Section 3 that Bayesian networks provide a powerful tool for integrating vision sensors and exploiting context.

A complete solution to the speaker detection problem must include an architecture for searching an input video sequence over all possible positions, scales, and orientations. This could be done through a combination of heuristics and brute force search as in [20]. In this paper we address a simpler task: Given an image region of a specified size and position within a video frame, compute the probability that it contains a speaker. The resulting region-based speaker detector could be the basis for a global search architecture.

Each sensor can be viewed as an operator that takes an input region and outputs a scalar feature. We illustrate

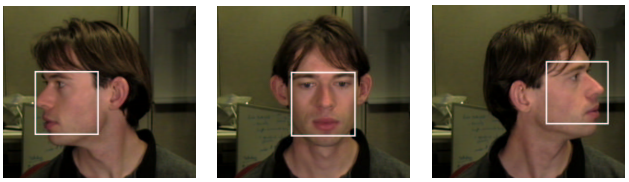


Figure 2: Frames 10, 25, and 40 from a sequence in which a talking head rotates from left to right.

the variation in these features using the sample image sequence shown in Figure 2. We applied each sensor to two sequences of input regions of length seven. The first set of regions tracks the face as the pose varies from left to right across the sequence, as illustrated in the figure. The resulting feature trajectories are plotted with solid lines in Figure 3. They illustrate the pose dependence of the sensor outputs.

A second set of regions was obtained by scanning a window from left to right in image coordinates within a single frame. Region number four in this sequence corresponds to the middle frame in Figure 2. It is identical to region four in the pose sequence. The resulting feature trajectories are plotted with dashed lines in Figure 3. They illustrate the selectivity of the sensors with respect to the face.

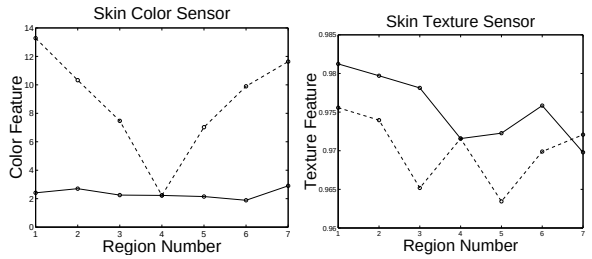
We see that all four sensors respond selectively to frontal faces, in the sense that their responses peak when the input window is centered on the face. All of them except for the face detector are fairly insensitive to the pose of the face. The skin color sensor was the most stable under pose variation. We now describe each sensor in more detail.

Skin Sensor

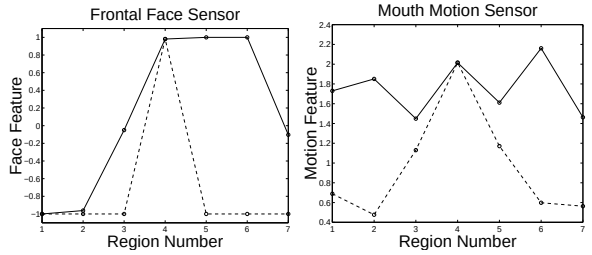
We employ skin color as a basic cue for detecting a visible face in the input window, as it is largely unaffected by the facial pose. Given skin color measurements obtained during a training phase, we fit a single gaussian color model as described in [24]. The feature is the average of the log-likelihood over the input region. The solid line in Figure 3(a) shows the stability of the skin color feature as a function of the pose of the face. The dashed line shows a gradual degradation as the input region is contaminated with background pixels.

Texture Sensor

It is well-known that many objects, such as walls, are similar in color to skin. We designed a simple texture feature to help discriminate regions containing faces from regions containing either very smooth patterns such as walls or highly textured patterns such as foliage. A correlation



(a) Skin (b) Texture



(c) NN (d) Mouth

Figure 3: Plots of the four sensor outputs for two sequences of image regions. The solid lines show the response as the pose of the face varies. The dashed lines show the result of sweeping the window across a single image.

ratio

$$T = \frac{E\left[g(x,y) \cdot g(x+\tau,y)\right]}{E\left[g^2(x,y)\right]}$$

defines the feature, where τ is set to one twelfth the width of the region of interest — on the order of facial feature sizes, and where g denotes the gray component of the input color image. (In our experiments we simply used the green channel.) We found correlation in X to be more stable than correlation in Y. Variation in this feature is illustrated in Figure 3(b).

NN face Sensor

The CMU face detector [20] uses a neural network (NN) architecture to search for frontal, upright faces in images. Since we are given a specific image position and scale to evaluate, we employ the verification network from the CMU system. Since this network is sensitive to small position errors, it is evaluated over a fixed range of displacements around the desired location and the highest score is returned.

The output of this detector is plotted in Figure 3(c). The solid curve shows the continuous output of the NN as the pose of the face varies. The output is highly saturated and orientation-sensitive. The feature is equally sensitive to position within an image (the dashed curve) and falls off rapidly around the face (region 4).

Mouth Sensor

This sensor uses the motion energy in the mouth region of a stabilized image sequence to measure chin and lip movement. A weighting mask is used to identify mouth and nonmouth pixels inside the target region. Affine tracking of the nonmouth pixels is used to cancel small face motions. The residual error in the mouth region averaged over five frames is then used as the feature. It is normalized by dividing by the residual error over the remainder of the face. This is an approximation to the optical flow approach to lip motion analysis proposed in [12].

In the absence of an accurate segmentation of the face pixels, the sensor is sensitive to significant head rotation. As the face pose approaches a profile view, residuals around the occluding contour increase, biasing the sensor. This effect is apparent in the "jaggedness" of the solid curve in Figure 3(d).

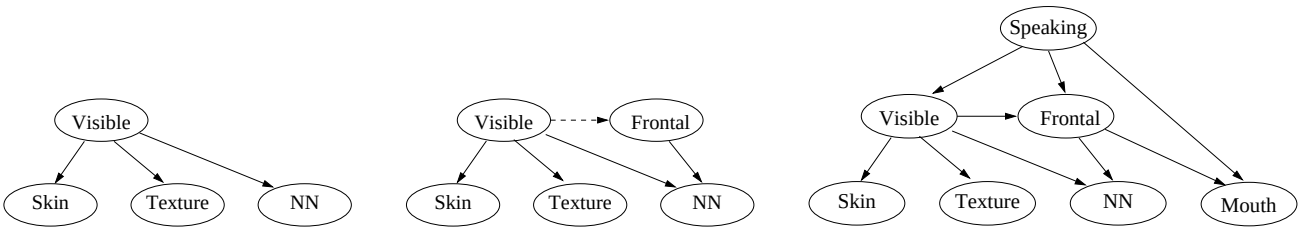
We selected the skin, texture, neural net, and mouth sensors described above on the basis of their availability, simplicity, and relevance to the task. Other sensors could undoubtedly be used. In the next section we demonstrate how Bayesian networks can be used to combine these simple sensors into a more complex speaker detector.

3 Bayesian Networks for Speaker Detection

A Bayesian network [16, 9] is a directed acyclic graph in which nodes represent random variables, and the absence of arcs represents conditional independence in the following formal sense: A node is independent of its nondescendants given its parents. Informally, we can think of a node as being "caused" by its parents. Figure 4(a) gives an example of a simple network which models the presence of a face in the input region.

Given a Bayesian network graph, we can factor the joint distribution over all of the variables into the product of local terms: $\Pr(X_1, \ldots, X_n) = \prod_i \Pr(X_i | \mathrm{Pa}\,(X_i))$, where $\mathrm{Pa}\,(X_i)$ are the parents of node X_i, and $\Pr(X_i|\mathrm{Pa}\,(X_i))$ is the conditional distribution of X_i given its parents. If all of the nodes are discrete (as we assume throughout this paper), the conditional distributions can be represented as conditional probability tables, called CPTs. (See Table 2 for an example.) However, we can also allow the nodes to be continuous and employ conditional Gaussians. Both CPTs and Gaussian parameters can be learned from training data using EM. See [13] for more details.

There are two computational tasks that must be performed in order to use these networks as classifiers. After the network topology has been specified, the first task is to obtain the local CPT for each variable conditioned on its parent(s). Once the CPTs have been specified (either through learning or from expert knowledge), the remaining task is inference, i.e., computing the probability of one



(a) Network N (b) Networks P and F (c) Network S

Figure 4: (a) Naive Bayes classifier. (b) Polytree (network P) without dashed arc, final face detector (network F) with dashed arc. (c) Final speaker detector. Note that the leaves represent the output of sensors, the other nodes represent hidden states.

set of nodes (the query nodes) given another set of nodes (the evidence nodes). In speaker detection the evidence nodes are the discretized outputs of the four vision sensors and the query node is the probability of a detected speaker. See [9] for more details on the standard Bayesian network algorithms.

We now explore the representational power of Bayesian networks through a series of four examples, culminating in the speaker detection network. The first example is the *naive Bayesian classifier* (network N) shown in Figure 4(a). The leaves represent observable features (the outputs of our sensors, suitably discretized), and the root node represents an unobserved variable, *visible*, which has value 1 if a face is visible in the input region, and 0 otherwise. This network acts as a face detector.

We are interested in computing $\Pr(V|S,T,N)$, where V represents *visible*, S represents the color-based *skin* sensor, T represents the face *texture* sensor, and N represents the *NN face* sensor. This quantity can be used in a decision rule, such as inferring that a face is present whenever $\Pr(V=1) > \Pr(V=0)$.

Network N is a poor model for a visible face because it fails to take into account the fact that the *NN face* sensor can only detect frontal faces. This missing contextual knowledge can easily be incorporated into our network model by means of an additional hidden variable F, for *frontal*. F takes on the values 1 for frontal faces, 0 for non-frontal faces, and 2 for not-applicable (in the case where $V=0$.)

We can build a separate naive Bayes classifier for F, with just one child, N. When we combine the two classifiers into a single network, we end up with a *polytree* structure (network P). This is shown in Figure 4(b) as the graph in which the dotted edge is absent. A polytree is a directed graph whose underlying undirected graph is a tree, i.e., an acyclic graph. Intuitively, we can think of a polytree as multiple directed trees grafted together in such a way as to not introduce any undirected cycles.

Polytrees are more powerful than naive Bayes models, since variables such as *NN face* can have multiple parents. However, the fact that *frontal* depends upon *visible* (since $\Pr(F=2|V=0)=1.0$) is not encoded in network P. We can model this additional fact by adding an extra arc, shown as a dotted line in Figure 4(b). This results in a graph with an undirected cycle, which we will call network F (the complete face detection network).

Network F has some interesting properties. For example, consider the case where $N=0$, meaning that the neural network has not detected a face, but $S=1$ and $T=1$, meaning that the skin and texture sensors have detected a face. In the case of network N, these contradictory sensor readings would have the effect of reducing $\Pr(V=1)$. In network F, however, the fact that $N=0$ can be *explained away* by the fact that $F=0$ despite the fact that $V=1$, since we know that the neural network cannot detect non-frontal faces. Hence we not only increase the classification accuracy on V, but we also infer the value of F without directly measuring it. The phenomena of explaining away is a key property of Bayesian network models for cue fusion.

The complete vision-based speaker detection network (network S) is shown in Figure 4(c), where we have introduced an additional measurement variable *mouth* motion (M) and hidden variable *speaking* (S). S is the desired output, the probability of a speaker being present in the input region. Note that the arcs connecting *speaking* to *visible* and *frontal* encode the contextual knowledge about camera placement described in Section 2.

Notice also that network F can be viewed as being "plugged in" as a module into network S. This is because the *visible* and *frontal* nodes *separate* (in a certain technical sense) all of the nodes in network F from the additional nodes *speaking* and *mouth*. The idea of reusing network components by plugging them into larger networks is formalized in [10] under the name *object-oriented Bayesian networks*.

4 Experimental Results

We conducted two experiments using a common dataset. The first experiment compared the face detection

performance of networks P and F in order to quantify the benefit of the more complex network topology. The second experiment tested the speaker detection performance of network S. Our implementations were based on the *Bayes Net Toolbox* for Matlab 5 which is freely available from the second author.[1]

The dataset for both experiments was generated from 80 five-frame video clips of faces. For each clip we manually labeled the position (bounding box) and pose (frontal, non-frontal, or not applicable) of the face in the first frame. We also randomly sampled 80 non-face regions from the backgrounds of these clips. We applied each of the four sensors to these 160 regions. The color, texture, and neural network sensors were applied to the first frame in each clip, while the mouth motion sensor used all five frames. We discretized the results using two bins for the skin detector, two for the neural network detector, and three for the texture detector. We used half of our data for training and half for testing. When training, we presented the values of all the nodes to the network. When testing, we presented the values of the sensors, and computed the marginal probabilities of the hidden nodes.

4.1 Face Detection Experiment

The first experiment compared the ability of networks P and F in Figure 4(b) to estimate V and F. We declared $V = 1$ if $\Pr(V = 1) > \Pr(V = 0)$. Equivalently, we declared $F = \arg\max \Pr(F)$. An error was counted if either V or F were incorrect. The results are shown in Table 1.

Network	Train	Test
P	72	75
F	95	94

Table 1: Face detection results. Percentage of cases in which both V and F are estimated correctly by the networks of Figure 4(b).

It is clear that the full network model performs better than the polytree model. To understand why, we examined the CPT for the *NN face* node, shown in Table 2. We can see that it has learned that the neural network is good at detecting frontal faces, but not good at detecting non-frontal faces; the general model (but not the polytree model) can exploit this to infer pose, as we discussed earlier. The increased expressive power of network F comes at the cost of more complicated algorithms (e.g. the join tree algorithm described in [9]). Fortunately, a number of freely available software packages contain good implementations of these routines.

[1] See http://www.cs.berkeley.edu/∼murphyk/Bayes/bnt.html for more information.

V	F	$\Pr(N = 0)$	$\Pr(N = 1)$
0	0	0.5	0.5
1	0	0.8377	0.1623
0	1	0.5	0.5
1	1	0.0055	0.9945
0	2	0.9980	0.0020
1	2	0.5	0.5

Table 2: The learned CPT for the neural network detector node in network G. When the face is visible and frontal (fourth row), the probability that the neural network will detect it is 0.9945; but when the face is visible and non-frontal (second row), the probability it will detect it is only 0.1623. Rows with 0.5 in them correspond to values of the parent nodes that were never seen in the training data (because they are impossible).

In this experiment, all of the errors were due to incorrectly estimating F for images where $V = 1$. This reflects the inherent ambiguity in the concept of "frontal pose". The threshold on the pose angle used by the human labeler is likely to be inconsistent with that implicitly defined by the neural network, resulting in errors in F. This explains why the performance on the test set can exceed the performance on the training set (as in the polytree case).

4.2 Speaker Detection Experiment

In the second experiment we evaluated the speaker detector (network S) using three sets of test data. The first set contained regions with *frontal* faces equally divided between speaking and nonspeaking. The second, *nonfrontal* set contained faces at a variety of nonfrontal poses. The final *nonface* set consisted of regions that did not contain a face. As before, we computed $S = \arg\max \Pr(S)$ in scoring the network output. The results for the training and testing data are given in Table 3. The average test score on face regions was 91%.

Dataset	Train	Test
Frontal	100	94
Nonfrontal	93	89
Nonface	94	98

Table 3: Speaker detection results. Percentage of correct estimates of S by network S (see Figure 4(c)).

In 90 % of the test cases, errors in estimating S seemed to result from estimating F incorrectly (i.e., F was incorrect and the mouth feature supported speaking). This suggests that the *mouth* sensor was fairly reliable for frontal faces.

The controlled lighting and lack of background motion in our dataset undoubtedly contributed to the success of

these two experiments. We plan to validate our network designs futher under more challenging experimental conditions, including variable lighting and moving background clutter.

5 Conclusions and Future Work

We have demonstrated a general approach to solving vision tasks in which Bayesian networks are used to combine the outputs of simple sensing algorithms. Bayesian networks provide an intuitive graphical framework for expressing contextual knowledge, coupled with efficient algorithms for learning and inference. They can represent complex probability models, but their learning rules are simple closed-form expressions given a fully-labeled data set.

Context is a particularly powerful cue in user-interface applications since it can be exploited and reinforced in the design of the interface. For the speaker detection task we exploited two contextual cues: the fact that a speaker's face image will be frontal, and the fact that the CMU face detector can only detect frontal faces. One result is network F in Figure 4(b), which can infer the frontal orientation of a face even though we have no explicit pose sensor.

The combination of multiple vision algorithms based on contextual information is a feature of many successful vision systems. For example, the vision-based kiosk described in [5] also exploits the alignment of camera and display axes and uses a combination of multiple sensing modules. It includes a clever hardware design for physically integrating the camera and the display. The KidsRoom system [8] at the M.I.T. Media Lab is another relevant example.

An alternative to fusing many simple sensors is to design complex algorithms that jointly measure a large number of hidden states. For example, speaker detection could also be performed using the output of a real-time head and lip tracking system such as LAFTER [14]. In this instance the primary advantage of our sensor fusion approach is its simplicity of implementation. It is quite likely that greater accuracy could be obtained with a more complex and specialized sensor.

However, as we move from sensing well-defined attributes like speech production to more abstract quantities such as the user's interest level, it becomes increasingly difficult to imagine designing a single highly specialized sensor. We believe that the full power of the Bayesian network approach will become apparent in this limit.

Our speaker detection experiments using the network of Figure 4(c) demonstrated classification rates of 91% on a controlled test set. This result suggests that Bayesian networks can provide an interesting alternative to the standard decision tree and neural network classifiers that are often used in vision applications.

In future work we plan to add speech sensing to the speaker detection network and experiment with multimodal inference. We will further validate our network designs on a large subject population under realistic conditions of background clutter. We also plan to explore the use of dynamic Bayesian networks (DBNs) to capture temporal attributes of users. Some interesting previous work in dynamic cue fusion includes the SERVP [4] and IFA [21] architectures, coupled HMM models [1], and mixed-state DBNs [15].

Going beyond low-level cue fusion, we would like to use Bayes nets as a framework for integrating high-level reasoning with low-level sensing. With a suitable utility model it should be possible to close the loop between sensing and action in a sound, decision-theoretic manner [6].

Acknowledgements

We would like to thank Henry Rowley for his help with the CMU face detector. We would also like to thank the reviewers for their detailed comments. An earlier version of this paper appeared as [18].

References

[1] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *Computer Vision and Pattern Recognition*, pages 994–999, 1997.

[2] H. Buxton and S. Gong. Advanced visual surveillance using bayesian networks. In *ICCV '95 Workshop on Context-Based Vision*, pages 111–122, Cambridge MA, 1995.

[3] A. D. Christian and B. L. Avery. Digital smart kiosk project. In *ACM SIGCHI '98*, pages 155–162, Los Angeles, CA, April 18–23 1998.

[4] J. Coutaz, F. Bérard, and J. L. Crowley. Coordination of perceptual processes for computer mediated communication. In *Proc. of 2nd Intl Conf. Automatic Face and Gesture Rec.*, pages 106–111, 1996.

[5] T. Darrell, G. Gordon, J. Woodfill, and M. Harville. A virtual mirror interface using real-time robust face tracking. In *Proc. of 3rd Intl Conf. Automatic Face and Gesture Rec.*, pages 616–621, Nara, Japan, 1998.

[6] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The Lumière project: Bayesian user modeling for inferring the goals and needs of software users. In *Proc. of the 14th Conf. on Uncertainty in AI*, pages 256–265, 1998.

[7] S. Intille and A. Bobick. Representation and visual recognition of complex, multi-agent actions using belief networks. In *CVPR '98 Workshop on Interpretation of Visual Motion*, 1998. Also see MIT Media Lab TR 454.

[8] S. S. Intille, J. W. Davis, and A. F. Bobick. Real-time closed-world tracking. In *Computer Vision and Pattern Recognition*, pages 697–703, 1997.

[9] F. V. Jensen. *An Introduction to Bayesian Networks*. Springer-Verlag, 1996.

[10] D. Koller and A. Pfeffer. Object-oriented bayesian networks. In *Proc. of the 13th Conf. on Uncertainty in AI*, pages 302–313, Providence, RI, Aug 1997.

[11] W. B. Mann and T. O. Binford. An example of 3–D interpretation of images using bayesian networks. In *DARPA IU Workshop*, pages 793–801, 1992.

[12] K. Mase and A. Pentland. Automatic lipreading by optical-flow analysis. *Systems and Computers in Japan*, 22(6):67–76, 1991.

[13] K. P. Murphy. Inference and learning in hybrid Bayesian networks. Technical Report 990, U.C. Berkeley, Dept. Comp. Sci, 1998.

[14] N. Oliver, A. P. Pentland, and F. Bérard. LAFTER: Lips and face real time tracker. In *Computer Vision and Pattern Recognition*, pages 123–129, 1997.

[15] V. Pavlović, B. J. Frey, and T. S. Huang. Time-series classification using mixed-state dynamic bayesian networks. In *Computer Vision and Pattern Recognition*, Ft. Collins, CO, June 1999. In this proceedings.

[16] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

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

[18] J. M. Rehg, K. P. Murphy, and P. W. Fieguth. Vision-based speaker detection using bayesian networks. In *Workshop on Perceptual User-Interfaces*, pages 107–112, 1998.

[19] R. D. Rimey and C. M. Brown. Control of selective perception using bayes nets and decision theory. *Intl. J. of Computer Vision*, 12(2/3):173–207, 1994.

[20] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In *Computer Vision and Pattern Recognition*, pages 203–208, 1996.

[21] K. Toyama and G. D. Hager. Incremental focus of attention for robust visual tracking. In *Computer Vision and Pattern Recognition*, pages 189–195, San Fransisco, CA, June 1996.

[22] N. Vasconcelos and A. Lippman. A bayesian framework for semantic content characterization. In *Computer Vision and Pattern Recognition*, pages 566–571, 1998.

[23] K. Waters, J. M. Rehg, M. Loughlin, S. B. Kang, and D. Terzopoulos. Visual sensing of humans for active public interfaces. In *Computer Vision for Human-Machine Interaction*, pages 83–96. Cambridge University Press, 1998.

[24] J. Yang and A. Waibel. A real-time face tracker. In *Proc. of 3rd Workshop on Appl. of Comp. Vision*, pages 142–147, Sarasota, FL, 1996.