Vision-Based Speaker Detection Using Bayesian Networks

James M. Rehg
Cambridge Research Lab
Compaq Computer Corporation
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 perceptual user interfaces requires the solution of a challenging statistical inference problem: The intentions and actions of multiple individuals must be inferred from noisy and ambiguous vision and speech data. We argue that Bayesian network models are an attractive statistical framework for cue fusion in PUI 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

Perceptual 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 PUI 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 [14, 8] are a class of probabilistic models which graphically encode the conditional independence relationships among a set of random variables. Bayesian networks are attractive for PUI 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. Of particular interest to the PUI community is the Lumière project [5] at Microsoft, which used Bayesian networks to model user goals in Windows applications.

In this paper we explore 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.

The goals of this paper are to illustrate the process of applying Bayesian networks to PUI applications and to explore the performance of the Bayesian network approach for the speaker detection task. We define the speaker detection problem in Section 2. In Section 3, we present a series of networks that illustrate the modeling process. We present experimental results in Section 4.

2. The Speaker Detection Task

We are interested in speaker detection as one of the components of a PUI for a *Smart Kiosk* [15, 21], a free-standing 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. 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.

In the context of our 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.

Our approach to speaker detection has two attributes. First, we want to exploit the context of the sensing environment as directly as possible. The use of context to simplify image interpretation is an attribute of many computer vision systems [19]. The main contextual cue which we exploit

is the popular strategy of aligning 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 the camera placement, these speaking users will generate frontal face images in which lip and jaw motion is visible. Figure 1 shows a picture of the kiosk configuration.

Figure 1. The Smart Kiosk

Another kiosk user-interface which exploits the alignment of camera and display axes is described in [4]. It includes a clever hardware design for physically integrating the camera and the display. The KidsRoom system [7] at the M.I.T. Media Lab is another good example of a PUI that exploits context.

The second attribute of our solution is the use of many simple sensors in combination to solve a more complex task. We employ four visual cues to perform speaker detection: the CMU face detector [18], Gaussian skin color detector [22], skin texture detector, and mouth motion detector. 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. We combine the output of these sensors with a Bayesian network. We show that the network can infer the frontal orientation of a face even though we have no explicit pose sensor.

A complete solution to the speaker detection problem must include a procedure for searching an input video sequence over all possible positions, scales, and orientations. This could be done, for example, through a combination of heuristics and brute force search [18]. The focus of this paper is on the design of the detector that would form the basis of such a search process. Given an image region at a certain size and position within a certain frame, the goal of the detector is to compute the probability that a speaker is present.

In this context, each sensor can be viewed as an operator that takes an input region in a video frame and outputs a scalar feature. We illustrate 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

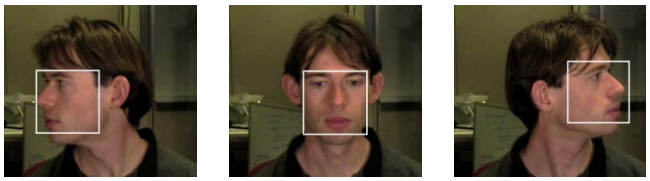

Figure 2. Frames 10, 25, and 40 from a sequence in which a talking head rotates from left to right.

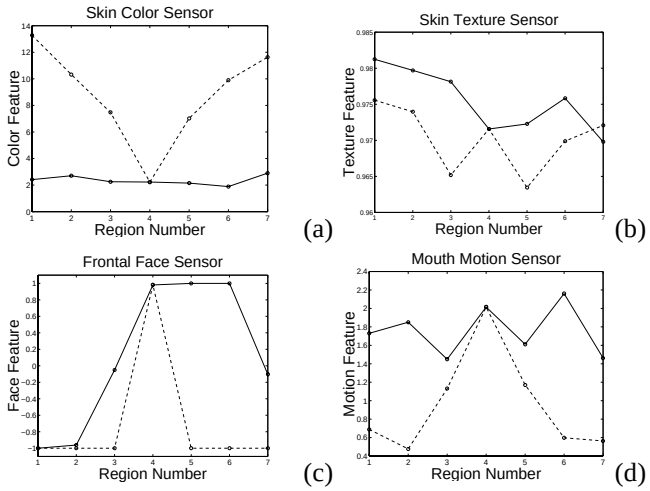

Figure 3. Plots of the four sensor outputs for two sequences of image regions. The solid lines show the response as the pose of the face varies. The dashed lines show the result of sweeping the window across a single image.

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 (number 25 in the input sequence). Region number four in this sequence contains a frontal face and corresponds to Figure 2(b). It is identical to region four in the pose sequence. The resulting feature trajectories illustrate the response of the sensors to the background. They are plotted with dashed lines in Figure 3. We now briefly describe the four vision sensors (more details can be found in [16]).

Color-Based 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 [22]. 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-Based Skin Sensor

It is well-known that many objects, such as walls, are similar in color to skin. We employ a 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 simple correlation 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 green channel in the input color image. Variation in this feature is illustrated in Figure 3(b).

Frontal Face Sensor

The CMU face detector [18] 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. This network is very sensitive to small position errors, so we search over a small region around the desired location and return the highest score.

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 Motion 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 [11].

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).

Of the four sensors described above, only the face detector could be applied to unrestricted video input with some chance of success. Furthermore, none of these sensors can directly measure the presence of a speaking user. It is important to note that we selected these particular sensing algorithms on the basis of their availability, simplicity, and relevance to the task. Our claim is not that they are optimal. Rather, our concern is to explore Bayesian networks as a principled method for combining such simple features in solving PUI tasks.

3. Bayesian Networks for Speaker Detection

A Bayesian network [14, 8] 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 non-descendants 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 Bayes net which models the presence of a face in the input region.

Given a Bayes net 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 Gaussians or conditional Gaussians. Both CPTs and Gaussian parameters can be learned from training data using EM. See [12] for more details.

There are two computational tasks that must be performed in using Bayes nets 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 set of nodes (the query nodes) given another set of nodes (the evidence nodes). In our example the evidence nodes are the discretized outputs of the four vision sensors and the query node is the probability of a detected speaker. See [16] for more details on the standard Bayes net algorithms.

In the remainder of this section we explore the representational power of Bayes nets through a series of examples that culminate in the speaker detector network. The first example is the naive Bayesian classifier of 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. We are interested in computing $\Pr(V|S,T,N)$, where V represents *visible*, S represents the *skin color* sensor, T represents the *skin texture* sensor, and N represents the *NN face* sensor. This quantity

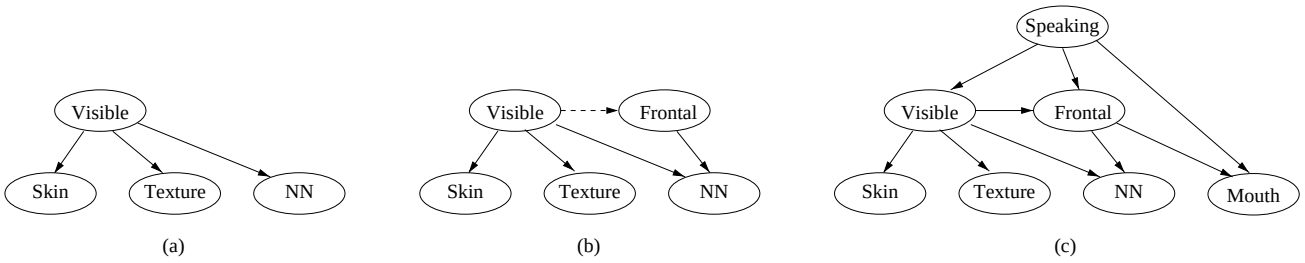

Figure 4. (a) A naive Bayes classifier. (b) Bayes net for visible/frontal face detection. Without the dotted arc, the graph is a polytree. (c) The speaker detection Bayes net. The leaves represent the output of sensors, the other nodes represent hidden states.

can be used in a decision rule, such as inferring that a face is present whenever $\Pr(V = 1) > \Pr(V = 0)$.

The network of Figure 4(a) 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 the *polytree* structure shown in Figure 4(b) (where 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 $\Pr(F = 2|V = 0) = 1.0$) cannot be encoded in a polytree. We can model this fact by adding an extra arc, shown by a dotted line in Figure 4(b). This results in a graph with an undirected cycle, which we will call G.

Network G in Figure 4(b) 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 naive Bayes case of network (a), these contradictory sensor readings would have the effect of reducing $\Pr(V = 1)$. In G, 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 Bayes net models for cue fusion.

In Figure 4(c), we have introduced an additional measurement variable *mouth motion* (M) and hidden variable *speaking* (S) to obtain the complete vision-based speaker detection network. 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 the network G from Figure 4(b) can be viewed as being "plugged in" as a module into the larger speaker detection network of (c). This is because the *visible* and *frontal* nodes separate (in a certain technical sense) all of the nodes in graph G from the additional nodes in the new speaker detection graph. The idea of reusing Bayes net components by plugging them into larger networks is formalized in [9] under the name object-oriented Bayes nets.

4. Experimental Results

We recorded 80 five-frame video clips of faces, and labeled the position (bounding box) and pose (frontal, profile, 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 bounding boxes. The color, texture, and neural network sensors were applied to the first frame in each clip, while the mouth motion sensor employed all five frames. We discretized the results by hand, 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. We conducted two sets experiments corresponding to the networks of Figure 4(b) and (c).

4.1. Face Detection Experiment

The first set of experiments explored the ability of the polytree and general (G) networks 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

Table (a)	Train	Test	Table (b)	Train	Test
Polytree	72	75	Frontal	100	94
General	95	94	Nonfrontal	93	89
			Nonface	94	98

Table 1. (a) Percentage of cases in which both V and F are estimated correctly using the models of Figure 4(b). (b) Percentage of correct estimates of S by the network of Figure 4(c) for three sets of video clips containing frontal faces, nonfrontal faces, and no faces.

V	F	$\Pr(N=0)$	$\Pr(N=1)$
0	0	0.5	0.5
1	0	0.0055	0.9945
0	1	0.5	0.5
1	1	0.8377	0.1623
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 (second row), the probability that the neural network will detect it is 0.9945; but when the face is visible and nonfrontal (fourth 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).

was counted if either V or F were incorrect. The results are shown in Table 1(a).

It is clear that the general model (network G) 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.

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 detection network of Figure 4(c) using three sets of test data. The first set contained video clips with visible, *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 clips 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 1(b).

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 motion* sensor was fairly reliable for frontal faces. The controlled lighting and lack of background motion in our experiments undoubtedly contributed to this success. We plan to validate these network designs futher under more challenging experimental conditions, including natural lighting and moving background clutter.

5. Previous Work

While Bayes net models are not yet in wide-spread use within the human sensing and computer vision communities, there is a growing body of work on their application to object recognition [10], scene surveillance [2], video analysis [20, 6], and selective perception [17]. 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.

While our focus has been on cue fusion in static images, there has been some interesting work on dynamic cue fusion for PUI problems. One example is the SERVP architecture of Coutaz et. al. [3]. Another is the coupled HMM models used by Brand et. al. [1]. See [16] for a more detailed discussion of the literature.

6. Discussion

We have demonstrated a general approach to solving perceptual user-interface tasks by fusing simple sensing algorithms using Bayesian networks. The primary advantage of Bayes nets is the ease with which contextual knowledge can be encoded. Context is a particularly powerful cue in PUI applications since it can be controlled and reinforced in the design of the interface.

In the speaker detection task which was the focus of this paper 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. The resulting network demonstrated the ability to estimate whether a

presented face was frontal, in spite of the fact that there was no sensor for face pose.

A second advantage of Bayesian networks in cue fusion applications is the existence of well-behaved learning algorithms for inferring the relationship between sensors and task variables. Bayes nets can represent complex probability models, but their learning rules are simple closed-form expressions, given a fully-labeled data set.

Our experiments on real data using the face and speaker detection networks of Figure 4(b) and (c) demonstrate the promise of Bayes net models for PUI applications. It would be premature to draw strong conclusions about the success of these particular network architectures from our current experiments. We plan to validate our design choices on a larger subject population under more challenging illumination and background conditions.

The approach of combining simple sensors with contextual cues to solve a PUI task is an alternative to approaches which build complex sensors for large numbers of user states. For example, speaker detection could also be performed using the output of a real-time head and lip tracking system such as LAFTER [13]. For the task of speaker detection, the primary advantage of the 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 Bayes net approach will become apparent in this limit.

In future work we plan to add speech sensing to the speaker detection network and experiment with multimodal inference. We also plan to explore the use of dynamic Bayes nets to capture temporal attributes of users. 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 [5].

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.

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 *Proc. of the Workshop on Context-Based Vision*, pages 111–122, Cambridge MA, 1995.

[3] 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.

[4] 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, April 14–16 1998.

[5] 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.

[6] S. Intille and A. Bobick. Representation and visual recognition of complex, multi-agent actions using belief networks. Technical Report 454, MIT Media Lab, 1998.

[7] 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.

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

[9] D. Koller and A. Pfeffer. Object-oriented bayesian networks. In *Proc. of Uncertainty in AI*, 1997.

[10] 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.

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

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

[13] 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.

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

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

[16] J. M. Rehg, K. P. Murphy, and P. W. Fiegut. Vision-based speaker detection using bayesian networks. Technical Report CRL 98/7, Compaq Cambridge Research Lab., 1998.

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

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

[19] T. M. Strat and M. A. Fischler. Context-based vision: Recognizing objects using information from both 2-d and 3-d imagery. *PAMI*, 13(10):1050–1065, 1991.

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

[21] 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.

[22] 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.