A Graphical Model for Audiovisual Object Tracking

Matthew J. Beal, Nebojsa Jojic, *Member*, *IEEE Computer Society*, and Hagai Attias

Abstract—We present a new approach to modeling and processing multimedia data. This approach is based on graphical models that combine audio and video variables. We demonstrate it by developing a new algorithm for tracking a moving object in a cluttered, noisy scene using two microphones and a camera. Our model uses unobserved variables to describe the data in terms of the process that generates them. It is therefore able to capture and exploit the statistical structure of the audio and video data separately, as well as their mutual dependencies. Model parameters are learned from data via an EM algorithm, and automatic calibration is performed as part of this procedure. Tracking is done by Bayesian inference of the object location from data. We demonstrate successful performance on multimedia clips captured in real world scenarios using off-the-shelf equipment.

Index Terms—Audio, video, audiovisual, graphical models, generative models, probabilistic inference, Bayesian inference, variational methods, expectation-maximization (EM) algorithm, multimodal, multimedia, tracking, speaker modeling, speech, vision, microphone arrays, cameras, automatic calibrations.

1 Introduction

In most systems that handle digital media, audio, and video, data are treated separately. Such systems usually have subsystems that are specialized for the different modalities and are optimized for each modality separately. Combining the two modalities is performed at a higher level. This process generally requires scenario-dependent treatment, including precise and often manual calibration.

For example, consider a system that tracks moving objects. Such a system may use video data, captured by a camera, to track the spatial location of the object based on its continually shifting image. If the object emits sound, such a system may use audio data, captured by a microphone pair (or array), to track the object location using the time delay of arrival of the audio signals at the different microphones. In principle, however, a tracker that exploits both modalities may achieve better performance than one that exploits either one or the other. The reason is that each modality may compensate for weaknesses of the other one. Thus, whereas a tracker using only video data may mistake the background for the object or lose the object altogether due to occlusion, a tracker also using audio data could continue focusing on the object by following its sound pattern. Conversely, video data could help where an audio tracker alone may lose the object as it stops emitting sound or is masked by background noise. More generally, audio and video signals originating from the same source tend to be correlated—thus, to achieve optimal performance, a system must exploit not just the statistics of each modality alone, but also the correlations among the two modalities.

The setup and example data in Fig. 1 illustrate this point. The figure shows an audiovisual capture system (left), an audio waveform captured by one of the microphones (top right), and a few frames captured by the camera (middle right). The frames contain a person moving in front of a cluttered background that includes other people. The audio waveform contains the subject's speech but also some background noise, including other people's speech. The audio and video signals are correlated on various levels. The lip movement of the speaker is correlated with the amplitude of part of the audio signal (see, e.g., [7]). Also, the time delay[1] between the signals arriving at the microphones is correlated with the position of the person in the image (see, e.g., [6], [31], [32]). It is the latter type of correlations that we aim for in this paper.

However, in order to use these correlations, a careful calibration procedure much be performed to establish a correspondence between the spatial shift in the image and the relative time delay between the microphone signals. Such a procedure needs to be repeated for each new setup configuration. This is a serious shortcoming of current audiovisual trackers.

The origin of this difficulty is that relevant features in the problem are not directly observable. The audio signal propagating from the speaker is usually corrupted by reverberation and multipath effects and by background noise, making it difficult to identify the time delay. The video stream is cluttered by objects other than the speaker, often causing a tracker to lose the speaker. Furthermore, audiovisual correlations usually exist only intermittently. This paper presents a new framework for fusing audio and video data. In this framework, which is based on probabilisitic generative modeling, we construct a model describing the

- *M.J. Beal is with the Department of Computer Science, University of Toronto, 10 King's College Road, Toronto, M5S 3G4, Canada. E-mail: beal@cs.toronto.edu.*
- *N. Jojic and H. Attias are with Microsoft Research, One Microsoft Way, Redmond, WA 98052. E-mail: {jojic, hagaia}@microsoft.com.*

Manuscript received 30 Aug. 2002; revised 10 Mar. 2003; accepted 21 Apr. 2003.
Recommended for acceptance by W.T. Freeman.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 118203.

1. The posterior probability of the time delay in Fig. 1 is based on the audio correlation and can be computed using (21) with the prior $p(\tau \mid \ell)$ set to uniform.

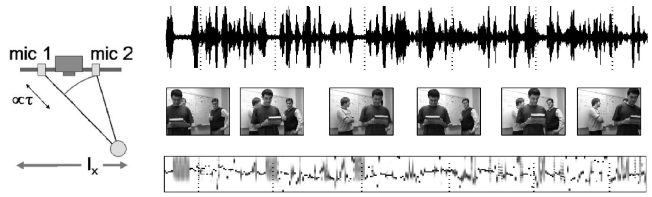

Fig. 1. (Top) audio waveform. (Middle) selected frames from associated video sequence (120×160 pixels^2). (Bottom) posterior probability over time delay τ (vertical axis, $\tau \in \{-15, \ldots, 15\}$) for each frame of the sequence; darker areas represent higher probability, and each frame has been separately normalized. The horizontal direction represents time along the sequence.

joint statistical characteristics of the audio-video data. Correlations between the two modalities can then be exploited in a systematic manner. We demonstrate the general concept by deriving a new algorithm for audiovisual object tracking. An important feature of this algorithm, which illustrates the power of our framework, is that calibration is performed automatically as a by-product of learning with the algorithm; no special calibration procedure is needed. We demonstrate successful performance on multimedia clips captured in real-world scenarios.

2 Related Work

There has been a significant amount of work on detecting and tracking people and moving objects in video (see, e.g., [4], [20], [21], [22]). There has also been much work on tracking speakers using arrays of two or more microphones (see, e.g., [5], [30]). In comparison, the area of audiovisual fusion is relatively new but growing fast. For example, [9], [15] attack the problem of speaker detection. A time delayed neural network is used in [9] to learn audiovisual correlation between a single microphone signal and a camera image, then use it for speaker detection by searching an audio-video sequence for correlated motion and audio that is indicative of a person talking. In [15], the authors construct a speaker detector by fusing multiple audio and video sensors, such as skin color, mouth motion, and audio silence detector, using dynamic Bayesian network which detects people in front of kiosks. The problem of speaker localization using a camera and one or more microphones is treated in [11], [18], [32], [33] in different contexts, among them video conferencing and user interfaces.

Audiovisual tracking (see, e.g., [10], [27], [28], [31], [34]) is a popular topic. Vermaak et al. [31] extend the particle filter-based approach of [4] which focused on video tracking, to include audio by modeling cross correlations between microphone array signals as noisy functions of the speaker's location. Particle filtering is one important approach to dealing with the computational intractability of some probabilistic models which is based on sequential sampling techniques. The tracking algorithm of [28] improves on that approach using the unscented particle filter. Audiovisual detection and tracking algorithms form major components of the system for capturing and broadcasting meetings reported in [10].

Speech enhancement in noisy reverberant environments is another problem where expoiting audio-video correlations may result in a significant gain. In particular, "lip reading" could help disambiguate parts of speech when noise, echoes, or additional speakers blur the difference between them. The authors in [16], [17], [19], [29] describe algorithms that address several aspects of such scenarios. We also mention applications to robotics and human-robot interaction [24], [26], person verification [3], and vehicle collision avoidance [8].

3 Probabilistic Generative Modeling

Our framework uses probabilistic generative models (also termed graphical models) to describe the observed data. The models are termed generative since they describe the observed data in terms of the process that generated them, using additional variables that are not observable. The models are termed probabilistic because, rather than describing signals, they describe probability distributions over signals. These two properties combine to create flexible and powerful models. The models are also termed graphical since they have a useful graphical representation, as we shall see below.

The observed audio signals are generated by the speaker's original signal, which arrives at microphone 2 with a time delay relative to microphone 1. The speaker's signal and the time delay are unobserved variables in our model. Similarly, the video signal is generated by the speaker's original image, which is shifted as the speaker's spatial location changes. Thus, the speaker's image and location are also unobserved variables in our model. The presence of unobserved (hidden) variables is typical of probabilistic generative models and constitutes one source of their power and flexibility.

The delay between the signals captured by the microphones is reflective of the object's position, as can be seen in Fig. 1 where we show the delay estimated by signal decorrelation (bottom right). Whereas an estimate of the delay can, in principle, be used to estimate the object position, in practice, the computation of the delay is typically not very accurate in situations with low signal strength, and is quite sensitive to background noise and reverberation. The object position can also be estimated by analyzing the video data, in which case problems can be caused by the background clutter and change in object's appearance. In this paper, we combine both estimators in a principled manner using a single probabilistic model.

Probabilistic generative models have several important advantages that make them ideal for our purpose. First, since they explicitly model the actual sources of variability in the problem, such as object appearance and background noise, the resulting algorithm turns out to be quite robust. Second, using a probabilistic framework leads to a solution by an estimation algorithm which is Bayes-optimal. Third, parameter estimation and object tracking are both performed efficiently using the expectation-maximization (EM) algorithm.

Within the probabilistic modeling framework, the problem of calibration becomes the problem of estimating the parametric dependence of the time delay on the object position. It turns out that these parameters are estimated automatically as part of our EM algorithm, and no special treatment is required. Hence, we assume no prior calibration of the system and no manual initialization in the first frame (e.g., defining the template or the contours of the object to be tracked). This is in contrast with previous research in this area, which typically requires specific and calibrated configurations, as in [32], [6]. We note, in particular, the method of [31] which, while using an elegant probabilistic approach, still requires contour initialization in video and the knowledge of

the microphone baseline, camera focal length, as well as the various thresholds used in visual feature extraction.

Throughout the paper, the only information our model is allowed to use before or during the tracking is the raw data itself. The EM algorithm described below learns from the data the object's appearance parameters, the microphone attenuations, the mapping from the object position in the video frames to the time delay between the audio waveforms, and the sensor noise parameters for all sensors.

4 A Probabilistic Generative Model for Audio-Video Data

We now turn to the technical description of our model. We begin with a model for the audio data, represented by the sound pressure waveform at each microphone for each frame. Next, we describe a model for the video data, represented by a vector of pixel intensities for each frame. We then fuse the two models by linking the time delay between the audio signals to the spatial location of the object's image.

4.1 Audio Model

We model the audio signals $\mathbf{x}_1$, $\mathbf{x}_2$ received at microphones 1, 2 as follows: First, each signal is chopped into equal length segments termed *frames*. The frame length is determined by the frame rate of the video. Hence, 30 video frames per second translates into $1/30$ second long audio frames. Each audio frame is a vector with entries x_{1n}, x_{2n} corresponding to the signal values at time point n. The number of time points in a frame depends on the sampling rate. For instance, at a sampling rate of 32kHz, a $1/30$s long audio frame would contain roughly 1,000 samples.

The audio frames $\mathbf{x}_1$, $\mathbf{x}_2$ are described in terms of an original audio signal $\mathbf{a}$. We assume that $\mathbf{a}$ is attenuated by a factor λ_i on its way to microphone $i = 1, 2$ and that it is received at microphone 2 with a delay of τ time points relative to microphone 1,

$$\begin{aligned} x_{1n} &= \lambda_1 a_n, \\ x_{2n} &= \lambda_2 a_{n-\tau}. \end{aligned} \tag{1}$$

We further assume that the observed audio frames $\mathbf{x}_1$, $\mathbf{x}_2$ are contaminated by additive sensor noise with precision matrices $\boldsymbol{\nu}_1$, $\boldsymbol{\nu}_2$. In this paper, we will assume that the sensor noise is white and the precision matrices are diagonal with a uniform diagonal, i.e.,

$$\boldsymbol{\nu}_m = \nu_m \mathbf{I}, \quad m = 1, 2. \tag{2}$$

To account for the variability of the hidden signal $\mathbf{a}$, it is described by a mixture model. Denoting the component label by r, each component has mean zero, a Toeplitz precision matrix (inverse covariance matrix) $\boldsymbol{\eta}_r$, and a prior probability π_r. Viewing it in the frequency domain, the precision matrix corresponds to the inverse of the *spectral template* for each component. This matrix is Toeplitz, i.e., we assume that the signal is stationary so that the second order statistics depends only on the distance between samples,

$$\eta_{r_{i,j}} = f(|i - j|). \tag{3}$$

Thus, we will index the elements $\boldsymbol{\eta}_r$ with a single index corresponding to the column index in the first row, i.e.,

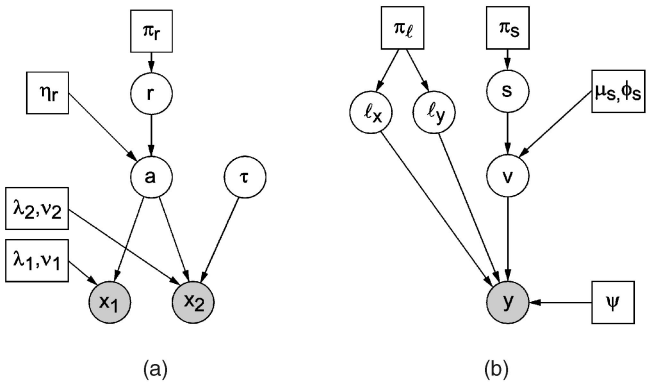

Fig. 2. (a) A simple graphical model of audio recorded by two microphones and (b) a simple graphical model of a video sequence.

$$\eta_{r_k} = \eta_{r_{i,j}}, \quad i - j = k - 1. \tag{4}$$

The first element, η_{r_1}, is the inverse power of the signal $\mathbf{a}$ and it is repeated along the diagonal of $\boldsymbol{\eta}_r$. The inverse of the second element η_{r_2} which is repeated along the first two side diagonals of $\boldsymbol{\eta}_r$, captures the first-order smoothness properties of the signal (or very low-frequency content in the spectral domain), etc.

Hence, we have

$$\begin{aligned} p(r) &= \pi_r, \\ p(\mathbf{a} \mid r) &= \mathcal{N}(\mathbf{a} \mid 0, \boldsymbol{\eta}_r), \\ p(\mathbf{x}_1 \mid \mathbf{a}) &= \mathcal{N}(\mathbf{x}_1 \mid \lambda_1 \mathbf{a}, \boldsymbol{\nu}_1), \\ p(\mathbf{x}_2 \mid \mathbf{a}, \tau) &= \mathcal{N}(\mathbf{x}_2 \mid \lambda_2 \mathbf{L}_\tau \mathbf{a}, \boldsymbol{\nu}_2), \end{aligned} \tag{5}$$

where $\mathbf{L}_\tau$ denotes the temporal shift operator, i.e., $(\mathbf{L}_\tau \mathbf{a})_n = a_{n-\tau}$. If the model is to be used alone, without the video components we describe in the next section, the prior probability for a delay τ can be set to a constant, $p(\tau) = const$. In this case, the estimation of time delay based on the posterior for a single-component model would be very similar to correlation-based speaker tracking [5]. On the other hand, a multiclass model similar to the one described here was used in [1] to perform noise removal from speech signals. In that paper, the joint $p(\mathbf{a}, r)$ served as a speech model with a relatively large number of components, which was pretrained on a large clean speech data set. Here, $p(\mathbf{a}, r)$ has only a few components and its parameters can be learned from audio-video data as part of the full model.

A note about notation. $\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\nu})$ denotes a Gaussian distribution over the random vector x with mean $\boldsymbol{\mu}$ and precision matrix (defined as the inverse covariance matrix) $\boldsymbol{\nu}$,

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\nu}) = |\boldsymbol{\nu}/2\pi|^{\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\nu} (\mathbf{x} - \boldsymbol{\mu})\right]. \tag{6}$$

Fig. 2a displays a graphical representation of the audio model. As usual with graphical models, a graph consists of nodes and edges. A shaded circle node corresponds to an observed variable, an open circle node corresponds to an unobserved variable, and a square node corresponds to a model parameter. An edge (directed arrow) corresponds to a probabilistic conditional dependence of the node at the arrow's head on the node at its tail.

A probabilistic graphical model has a generative interpretation: according to the model in Fig. 2a, the process of generating the observed microphone signals starts with

picking a spectral component r with probability $p(r)$, followed by drawing a signal $\mathbf{a}$ from the Gaussian $p(\mathbf{a} \mid r)$. Separately, a time delay τ is also picked. The signals $\mathbf{x}_1$, $\mathbf{x}_2$ are then drawn from the undelayed Gaussian $p(\mathbf{x}_1 \mid \mathbf{a})$ and the delayed Gaussian $p(\mathbf{x}_2 \mid \mathbf{a}, \tau)$, respectively.

4.2 Video Model

In analogy with the audio frames, we model the video frames as follows: Denote the observed frame by $\mathbf{y}$, which is a vector with entries y_n corresponding to the intensity of pixel n. This vector is described in terms of an original image $\mathbf{v}$ that has been shifted by $\ell = (\ell_x, \ell_y)$ pixels in the x and y directions, respectively,

$$y_n = v_{n-\ell}, \tag{7}$$

and has been further contaminated by additive noise with precision matrix ψ. To account for the variability in the original image, $\mathbf{v}$ is modeled by a mixture model. Denoting its component label by s, each component is a Gaussian with mean $\boldsymbol{\mu}_s$ and precision matrix $\boldsymbol{\phi}_s$, and has a prior probability π_s. The means serve as image templates. Hence, we have

$$\begin{aligned} p(s) &= \pi_s, \\ p(\mathbf{v} \mid s) &= \mathcal{N}(\mathbf{v} \mid \boldsymbol{\mu}_s, \boldsymbol{\phi}_s), \\ p(\mathbf{y} \mid \mathbf{v}, \ell) &= \mathcal{N}(\mathbf{y} \mid \mathbf{G}_\ell \mathbf{v}, \psi), \end{aligned} \tag{8}$$

where $\mathbf{G}_\ell$ denotes the shift operator, i.e., $(\mathbf{G}_\ell v)_n = v_{n-\ell}$. The prior probability for a shift ℓ is assumed flat, $p(\ell) = const$. This model was used in [14], [22] for video-based object tracking and stabilization.

Fig. 2b displays a graphical representation of the video model. Like the audio model, our video model has a generative interpretation. According to the model in Fig. 2b, the process of generating the observed image starts with picking an appearance component s from the distribution $p(s) = \pi_s$, followed by drawing a image $\mathbf{v}$ from the Gaussian $p(\mathbf{v} \mid s)$. The image is represented as a vector of pixel intensities, where the elements of the diagonal precision matrix define the level of confidence in those intensities. Separately, a discrete shift ℓ is picked. The image $\mathbf{y}$ is then drawn from the shifted Gaussian $p(\mathbf{y} \mid \mathbf{v}, \ell)$.

Notice the symmetry between the audio and video models. In each model, the original signal is hidden and described by a mixture model. In the video model, the templates describe the image and, in the audio model, the templates describe the spectrum. In each model, the data are obtained by shifting the original signal, where in the video model the shift is spatial and in the audio model the shift is temporal. Finally, in each model, the shifted signal is corrupted by additive noise.

4.3 Fusing Audio and Video

Our task now is to fuse the audio and video models into a single probabilistic graphical model. One road to fusion exploits the fact that the relative time delay τ between the microphone signals is directly related to the object position ℓ. This is the road we take in this paper. In particular, as the distance of the object from the sensor setup becomes much larger than the distance between the microphones, which is the case in our experiments, τ becomes linear in ℓ. We therefore use a linear mapping to approximate this dependence and model the approximation error by a zero mean Gaussian with precision ω,

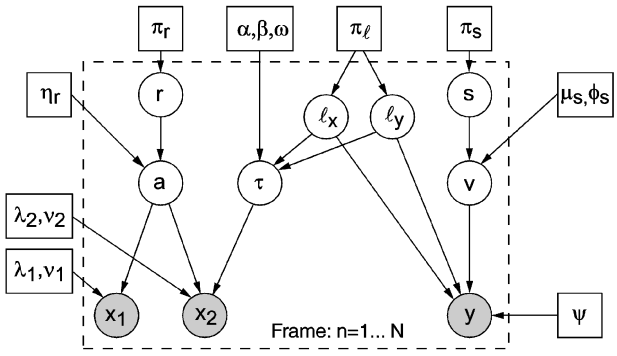

Fig. 3. Graphical model for the joint audio-video data. The dotted rectangle denotes i.i.d. frames and has the following meaning: everything it encompasses, i.e., all model variables, has value that is frame dependent; everything it leaves out, i.e., the model parameters, is frame independent.

$$p(\tau \mid \ell) = \mathcal{N}(\tau \mid \alpha \ell_x + \alpha' \ell_y + \beta, \omega). \tag{9}$$

Note that, in our setup (see Fig. 1), the mapping involves only the horizontal position, as the vertical movement has a significantly smaller affect on the signal delay due to the horizontal alignment of the microphones (i.e., $\alpha' \approx 0$). The link formed by (9) fuses the two models into a single one, whose graphical representation is displayed in Fig. 3.

5 Parameter Estimation and Object Tracking

Here, we outline the derivation of an EM algorithm for the graphical model in Fig. 3. As usual with hidden variable models, this is an iterative algorithm. The E-step of each iteration updates the posterior distribution over the hidden variables conditioned on the data. The M-step updates parameter estimates.

We start with the joint distribution over all model variables, the observed ones $\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}$ and the hidden ones $\mathbf{a}, \tau, r, \mathbf{v}, \ell, s$. As Fig. 3 shows, this distribution factorizes as

$$\begin{aligned} P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, \mathbf{a}, \tau, r, \mathbf{v}, \ell, s \mid \theta) &= p(\mathbf{x}_1 \mid \mathbf{a})\, p(\mathbf{x}_2 \mid \mathbf{a}, \tau) \\ \cdot\ p(\mathbf{a} \mid r) p(r)\, p(\mathbf{y} \mid \mathbf{v}, \ell)\, p(\mathbf{v} \mid s)\, p(s)\, p(\tau \mid \ell)\, p(\ell), \end{aligned} \tag{10}$$

where each term is conditioned on the model parameters θ, but in a more compact notation we omit this condition. In the rest of the paper, all probability distributions depend on the model parameters, even when it is not explicitly stated. The model parameters are

$$\theta = \{\lambda_1, \nu_1, \lambda_2, \nu_2, \boldsymbol{\eta}_r, \pi_r, \psi, \boldsymbol{\mu}_s, \boldsymbol{\phi}_s, \pi_s, \alpha, \alpha', \beta, \omega\}, \tag{11}$$

i.e., the microphone attenuation and noise parameters $\lambda_i, \boldsymbol{\nu}_i$; the spectral characteristics $\boldsymbol{\nu}_r$ and prior distribution over the audio components π_r; video observation noise parameters ψ, video templates $\boldsymbol{\mu}_s$, their precision parameters $\boldsymbol{\phi}_s$, and the prior distribution over the templates π_s; and, finally, the audio-video calibration parameters α, α' that define the mapping between the object position and audio delay, as well as the uncertainty of this mapping ω.

These model parameters control the joint probability distribution over the observed and hidden variables in the model: audio class index r, generated latent audio signal $\mathbf{a}$, the time delay τ, the audio signals $\mathbf{x}_1, \mathbf{x}_2$ received by the two microphones, the video class s, the latent image $\mathbf{v}$, the transformation index $\ell = (\ell_x, \ell_y)$ defining a horizontal ℓ_x, and vertical ℓ_y shift applied on the latent image to produce

the video frame $\mathbf{y}$. In the rest of the paper, we will use a single variable ℓ to denote image transformation whenever possible to shorten the notation, except when the components ℓ_x, ℓ_y are treated differently in equations, for example, due to the dependence between ℓ_x and τ.

Ultimately, we are interested in tracking the object based on the data, i.e., obtaining a position estimate $\hat{\ell}$ at each frame. In the framework of probabilistic modeling, one computes more than just a single value of ℓ. Rather, the full posterior distribution over ℓ given the data, $p(\ell \mid \mathbf{x}_1, \mathbf{x}_2, \mathbf{y})$, for each frame, is computed. This distribution provides the most likely position value via

$$\hat{\ell} = \arg\max_{\ell} p(\ell \mid \mathbf{x}_1, \mathbf{x}_2, \mathbf{y}), \tag{12}$$

as well as a measure of how confident the model is of that value. It can also handle situations where the position is ambiguous by exhibiting more than one mode. An example is when the speaker is occluded by either of two objects. However, in our experiments, the position posterior is always unimodal.

The parameters θ of the system can be estimated based on the entire audio-video sequence by maximizing the average log-likelihood of the data

$$\mathcal{F}(\theta) = \langle \log p(x_1, x_2, y \mid \theta) \rangle$$

with respect to the model parameters. The brackets denote the averaging operator, $\langle f \rangle = \frac{1}{T}\sum_t f_t$, where $t = 1, \ldots, T$ enumerates the observed data samples.

Note that optimizing the average log-likelihood is equivalent to optimizing the sum of log-likelihoods for all audiovisual samples. Using this cost implies that the data points are considered as independently generated from the generative model. Obviously, in practice, the position of the object changes slowly through time and this observation can be used to further constrain the generative process by modeling the evolution of the higher-level hidden variables through time (see, for example, [22]). However, this turns out to be useful only when the data is very noisy. Usually, however, the audio and, especially, visual observations provide such strong clues about the hidden variables that the time series model does not improve the tracking over the i.i.d model. Modeling dependencies through time will probably prove important in case of tracking jointly multiple audiovisual objects in presence of significant intermittent mutual occlusions in both audio and video signals.

Instead of the average log-likelihood, it is possible to focus instead on the negative of a free energy of the model, defined in terms of an auxiliary probability distribution over hidden variables $Q(\mathbf{a}, \tau, r, \mathbf{v}, \ell, s)$ as

$$\left\langle - \int Q(\mathbf{a}, \tau, r, \mathbf{v}, \ell, s) \log \frac{Q(\mathbf{a}, \tau, r, \mathbf{v}, \ell, s)}{P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, \mathbf{a}, \tau, r, \mathbf{v}, \ell, \mathbf{s} \mid \theta)} \right\rangle,$$

where the integration is performed over the hidden variables $(\mathbf{a}, \tau, r, \mathbf{v}, \ell, s)$.

It turns out [25] that this quantity is a lower bound on the average log-likelihood of the data, with the bound becoming tight when Q for each data sample is equal to the exact posterior distribution over the hidden variables. This justifies a number of iterative optimization algorithms that improve this bound in each step and which are often more tractable than direct optimization of the log likelihood. For example, variational methods use parameterized functions Q of constrained forms [23]. The parameters of Q functions for all data samples are estimated as to increase the above bound (generalized E step) and then keeping the Q functions fixed, the parameters θ are found that further increase the bound (generalized M step). These steps are iterated until convergence and, in case of a Q function form which captures the full posterior, this procedure is equivalent to the exact EM algorithm. As the Q function plays the role of the posterior, the tracking comes as a byproduct of parameter optimization. (For more details on various algorithms for minimizing free energy, the reader can also see [13].)

We note again that an iterative EM algorithm makes it possible to avoid prior calibration of the system and to automatically adapt to changes in physical configuration of microphones and cameras. This is done by iterating the inference (E step) and model parameters estimation (M step), where the latter includes updating the audiovisual link parameters. Thus, the tracker is provided only with the *raw* video and audio measurements and the model parameters are learned automatically, starting with random initialization. Of the model parameters listed above, the only set of parameters that would be useful to learn separately from this procedure are the spectral templates captured in precision matrices $\boldsymbol{\nu}_r$. These could be trained on a large corpus of speech data in order to specialize the model to speaker tracking. However, in our experiments, we assume that even the templates are white noise, i.e., we set $\boldsymbol{\nu}_r = \mathbf{I}$, as even such a simple audio model tends to bootstrap the proper tracking in the video component, pushing it out of the local maxima that is not consistent with the audio evidence. Thus, although we demonstrate only speaker tracking, it is likely that the algorithm would work on any object in video that produces sound.

5.1 E-Step

Generally, the posterior over the hiddens is computed from the model distribution by Bayes' rule,

$$p(\mathbf{a}, \tau, r, \mathbf{v}, \ell, s \mid \mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, \theta) = \frac{p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, \mathbf{a}, \tau, r, \mathbf{v}, \ell, s \mid \theta)}{p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y} \mid \theta)},$$

where $p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y} \mid \theta)$ is obtained from the model distribution by marginalizing over the hiddens. To describe the posterior distribution over the hidden variables, we switch to a notation that uses q to denote a posterior distribution conditioned on the data. Hence,

$$\begin{aligned} p(\mathbf{a}, \tau, r, \mathbf{v}, \ell, s \mid \mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, \theta) = \\ q(\mathbf{a} \mid \tau, r) q(\mathbf{v} \mid \ell, s) q(\tau \mid \ell, r) q(\ell, r, s) = Q. \end{aligned} \tag{14}$$

The q notation omits the data, as well as the parameters. Hence, $q(\mathbf{a} \mid \tau, r) = p(\mathbf{a} \mid \tau, r, \mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, \theta)$, etc. For practical reasons, we use the same factorization that we assumed in the model distribution, (10), as the free energy defined above breaks into a sum of terms that involve subsets of variables. However, note that this form of a posterior is still a full joint distribution over all hidden variables and, thus, we can express any probability distribution over hidden variables in this form.

The functional forms of the posterior components q also follow from the model distribution. As our model is constructed from Gaussian components tied together by discrete variables, it can be shown that the audio posterior $q(\mathbf{a} \mid \tau, r)$ and the video posterior $q(\mathbf{v} \mid \ell, s)$ are both Gaussian,

$$q(\mathbf{a} \mid \tau, r) = \mathcal{N}(\mathbf{a} \mid \boldsymbol{\mu}^a_{\tau,r}, \boldsymbol{\nu}^a_{\tau,r}),$$
$$q(\mathbf{v} \mid \ell, s) = \mathcal{N}(\mathbf{v} \mid \boldsymbol{\mu}^v_{\ell,s}, \boldsymbol{\nu}^v_{\ell,s}), \tag{15}$$

while the rest of the posterior is assumed to be a discrete distribution over ℓ, τ, s, r, which is true for all but τ. However, since the time in the audio signal is already discrete and the possible discrete time delays fall typically into a set of 30 or so values, a discrete approximation of τ tends to be nearly exact. This makes the inference described in this section variational, although the resulting distribution would almost perfectly match the exact posterior. (See the discussion at the end of the section).

To compute the posterior, we can optimize the KL divergence between so parameterized q distribution and the posterior for the model. Since the logarithm of the posterior satisfies

$$\log p(\mathbf{a}, \tau, r, \mathbf{v}, \ell, \mathbf{s} \mid \mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, \boldsymbol{\theta}) = \log p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, \mathbf{a}, \tau, r, \mathbf{v}, \ell, s \mid \theta) + const, \tag{16}$$

optimizing the KL divergence is equivalent to optimizing the free energy

$$F = \int_{\mathbf{h}} Q \log \frac{Q}{P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, \mathbf{a}, \tau, r, \mathbf{v}, \ell, \mathbf{s} \mid \theta)}, \tag{17}$$

where the integration is done with respect to all hidden variables $\mathbf{h} = \mathbf{a}, \tau, r, \mathbf{v}, \ell, s$ (see [23], [25], or the tutorial paper [13] in later issue). (The free energy F is equal to the negative log likelihood of the data when Q is the exact posterior and, thus, both E and M steps will be based on minimizing F.) Since all component distributions in Q are either Gaussian or discrete, the integration can be done in a closed form which turns out to be quadratic in the parameters of the Guassians and linear in the parameters of the discrete distributions.

By minimizing F, we obtain the following expressions for the mean and precision of the video posterior:

$$\boldsymbol{\mu}^v_{\ell,s} = (\boldsymbol{\nu}^v_s)^{-1}(\boldsymbol{\phi}_s \boldsymbol{\mu}_s + G_\ell^\top \boldsymbol{\psi} \mathbf{y}),$$
$$\boldsymbol{\nu}^v_s = \boldsymbol{\phi}_s + \boldsymbol{\psi}. \tag{18}$$

Note the precision matrices are diagonal, and $\boldsymbol{\psi}$ further has a uniform diagonal which resulted in the precision matrix independent of the transformation index ℓ.

The mean and precision of the posterior over the hidden audio signal a are

$$\boldsymbol{\nu}^a_r = \boldsymbol{\eta}_r + \lambda_1^2 \boldsymbol{\nu}_1 + \lambda_2^2 \boldsymbol{\nu}_2 \tag{19}$$
$$\boldsymbol{\mu}^a_{r,\tau} = \boldsymbol{\nu}_x^{-1}(\lambda_1 \boldsymbol{\nu}_1 \mathbf{x}_1 + \lambda_2 \boldsymbol{\nu}_2 L_\tau^\top \mathbf{x}_2). \tag{20}$$

Note that the posterior precision $\boldsymbol{\nu}^a_r$ inherits its Toeplitz structure from $\boldsymbol{\eta}_r$, since $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_2$ are uniform diagonal.

Another component of the posterior is the conditional probability table $q(\tau \mid \ell, r) = p(\tau \mid \ell, r, x_1, x_2, y, \theta)$ which turns out to be

$$q(\tau \mid \ell, r) = \frac{1}{Z} p(\tau \mid \ell) \exp(\lambda_1 \lambda_2 \nu_1 \nu_2 c_{\tau,r}), \tag{21}$$

where

$$c_{\tau,r} = \sum_i \sum_j x_{1,i-\tau} x_{2,j} / \nu^a_{r_{|i-\tau-j|+1}} \tag{22}$$

is a generalized cross correlation coefficient that takes into account the expected spectral characteristics. Note that when $\boldsymbol{\eta}_r = \mathbf{I}$, then $\boldsymbol{\nu}^a_r$ is a uniform diagonal matrix and the generalized cross correlation coefficient becomes proportional to the standard cross correlation coefficient

$$c_\tau = (\nu^a_{r_1})^{-1} \sum_i x_{1,i-\tau} x_{2,i}. \tag{23}$$

Thus, the posterior over the delay τ will directly depend on the cross correlation between the observed audio frames, but through the constant Z it will also depend on the posterior over the position of the object. This constant is a part of the rest of the posterior over the discrete variables,

$$q(s, r, \ell_x, \ell_y) = \frac{1}{\Omega} g(s, r) h(s, \ell_x, \ell_y) Z, \tag{24}$$

where the constant Ω normalizes the posterior and

$$\ln g(s, r) = -\frac{1}{2} \ln |\boldsymbol{\nu}^v_s| + \frac{1}{2} \ln |\phi_s| + \ln \pi_s + \ln \pi_r - \frac{1}{2} \boldsymbol{\mu}_s^\top \left(\phi_s - \phi_s (\nu^v_s)^{-1} \phi_s \right) \boldsymbol{\mu}_s \tag{25}$$

$$\ln h(s, \ell_x, \ell_y) = \frac{1}{2} \psi^2 e_{\ell,s} + \psi d_{\ell,s}, \tag{26}$$

with

$$e_{\ell,s} = \mathbf{y}^\top G_\ell (\nu^v_s)^{-1} G_\ell^\top \mathbf{y}, \tag{27}$$
$$d_{\ell,s} = \boldsymbol{\mu}_s^\top \phi_s (\nu^v_s)^{-1} G_\ell^\top \mathbf{y}. \tag{28}$$

Using the above equations, the posterior distribution over all discrete variables $q(\ell, \tau, s, r) = q(\tau | \ell, r) q(s, r, \ell)$ can be computed up to a multiplicative constant and then normalized to add up to one. In this process, the posterior is evaluated for all possible combinations of the discrete random variables. As there is only a small number of physically possible delays τ that are a multiple of the sampling period, the major computational hurdle here is the computation of the terms in (27) and (28). However, these can be efficiently computed in the FFT domain [12], making the inference process very fast.

Before we move on to the parameter update rules in the M step, we note again that the calculation of $q(\tau \mid \ell, r)$ involves a minor but somewhat subtle point. Since throughout the paper we work in discrete time, the delay τ in our model is generally regarded as a discrete variable. In particular, $q(\tau \mid \ell, r)$ is a discrete probability table. However, for reasons of mathematical convenience, the model distribution $p(\tau \mid \ell, r)$ (9) treats τ as continuous. Hence, the posterior $q(\tau \mid \ell, r)$ computed by our algorithm is, strictly speaking, an approximation, as the true posterior in this model must also treat τ as continuous. It turns out that this approximation is of the variational type (for a review of variational approximations see, e.g., [23], [13]). To derive it rigorously, one proceeds as follows: First, write down the form of the approximate posterior as a sum of delta functions,

$$q(\tau \mid \ell, r) = \sum_n q_n(\ell) \delta(\tau - \tau_n), \tag{29}$$

where the τ_n are spaced one time point apart. The coefficients q_n are nonnegative and sum up to one, and their dependence on ℓ is initially unspecified. Next, compute the $q_n(\ell, r)$ by minimizing the Kullback-Leibler (KL) distance between the approximate posterior and the true posterior. This produces the optimal approximate

posterior out of all possible posteriors which satisfy the restriction (29). In this paper, we write $q(\tau \mid \ell, r)$ rather than $q_n(\ell, r)$ to keep notation simple.

5.2 M-Step

The M-step updates the model parameters θ (11). The update rules are derived, as usual, by considering the objective function

$$\mathcal{F}(\theta) = \langle \log p(x_1, x_2, y \mid \theta) \rangle, \tag{30}$$

known as the averaged data likelihood. We use the notation $\langle \cdot \rangle$ to denote averaging with regard to the posterior (14) over all hidden variables that do not appear on the left-hand side and, in addition, averaging over all frames. Hence, $\mathcal{F}$ is essentially the log-probability of our model for each frame, where values for the hidden variables are filled in by the posterior distribution for that frame, followed by summing over frames. Each parameter update rule is obtained by setting the derivative of $\mathcal{F}$ with regard to that parameter to zero.

As noted above, a useful way to express the data likelihood is as the negative of the free energy and, so, the objective function can be written as

$$\mathcal{F}(\theta) = \left\langle - \int_{\mathbf{h}} Q \log \frac{Q}{P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, \mathbf{a}, \tau, r, \mathbf{v}, \ell, \mathbf{s} \mid \theta)} \right\rangle, \tag{31}$$

where the Q distribution is computed for each audio-video frame as described in the previous section. Equating the derivatives of this objective function to zero provides the update rules for model parameters.

For example, for the video model parameters μ_s, ϕ_s, π_s, we have

$$\begin{aligned} \boldsymbol{\mu}_s &= \frac{\langle \sum_\ell q(\ell, s) \boldsymbol{\mu}^v_{\ell,s} \rangle}{\langle q(s) \rangle}, \\ \boldsymbol{\phi}_s^{-1} &= \frac{\langle \sum_\ell q(\ell, s) (\boldsymbol{\mu}^v_{\ell,s} - \boldsymbol{\mu}_s)^2 + q(s) (\boldsymbol{\nu}^v_{\ell,s})^{-1} \rangle}{\langle q(s) \rangle}, \\ \pi_s &= \langle q(s) \rangle, \end{aligned} \tag{32}$$

where the qs are computed by appropriate marginalizations over $q(\ell, r, s)$ from the E-step. Notice that here, the notation $\langle \cdot \rangle$ implies only average over frames. Update rules for the audio model parameters $\boldsymbol{\eta}_r$, π_r are obtained in a similar fashion.

For the audio-video link parameters α, β, we have, assuming for simplicity $\alpha' = 0$,

$$\begin{aligned} \alpha &= \frac{\langle \ell_x \tau \rangle - \langle \tau \rangle \langle \ell_x \rangle}{\langle \ell_x^2 \rangle - \langle \ell_x \rangle^2} \\ \beta &= \langle \tau \rangle - \alpha \langle \ell_x \rangle \\ \omega^{-1} &= \langle \tau^2 \rangle + \alpha^2 \langle \ell_x^2 \rangle + \beta^2 + 2\alpha\beta \langle \ell_x \rangle - 2\alpha \langle \tau \ell_x \rangle - 2\beta \langle \tau \rangle, \end{aligned}$$

where, in addition to averaging over frames, $\langle \cdot \rangle$ here implies averaging for each frame with regard to $q(\tau, \ell)$ for that frame, which is obtained by marginalizing $q(\tau \mid \ell) q(\ell, r, s)$ over r, s.

A note about complexity. According to (32), computing the mean $(\mu_s)_n$ for each pixel n requires summing over all possible spatial shifts ℓ. Since the number of possible shifts equals the number of pixels, this seems to imply that the complexity of our algorithm is quadratic in the number of pixels N. If that were the case, a standard $N = 120 \times 160$ pixel array would render the computation practically intractable. However, as pointed out in [12], a more careful examination of (32), in combination with (18), shows that it can be written in the form of an inverse FFT. Consequently, the actual complexity is not $\mathcal{O}(N^2)$ but rather $\mathcal{O}(N \log N)$. This result, which extends to the corresponding quantities in the audio model, significantly increases the efficiency of the EM algorithm.

For brevity, we omit the update rules for other parameters in the model, such as microphone gain and noise parameters. These are straightforward to derive, and we should note that, except for $\boldsymbol{\eta}_r$, which we set to I, in our experiments we jointly optimize for *all* parameters in the model.

5.3 Tracking

Tracking is performed as part of the E-step using (12), where $p(\ell \mid x_1, x_2, y)$ is computed from $q(\tau, \ell)$ above by marginalization. For each frame, the mode of this posterior distribution represents the most likely object position and the width of the mode a degree of uncertainty in this inference.

6 Results

We tested the tracking algorithm on several audio-video sequences captured by the setup in Fig. 1 consisting of low-cost, off-the-shelf equipment. The video capture rate was 15 frames per second and the audio was digitized at a sampling rate of 16 kHz. This means that each frame contained one 160×120 image frame and two 1,066 samples long audio frames.[2] No model parameters were set by hand and no initialization was required; the only input to the algorithm was the raw data. The algorithm was consistently able to estimate the time delay of arrival and the object position while learning all the model parameters, including the calibration (audio-video link) parameters. The processing speed of our Matlab implementation was about 50 audio-video frames per second per iteration of EM. Convergence was generally achieved within just 10 iterations. This means that our Matlab script processes a minute of video in about three minutes.

We present the results on two sequences that had substantial background audio noise and visual distractions. In Fig. 4, we compare the results of tracking using the audio only model (Fig. 2a), full audio-video model (Fig. 3), and the video only model (Fig. 2b) on the multimodal data containing a moving and talking person with a strong distraction consisting of another two people chatting and moving in the background (see Fig. 1). For tracking using the audio only model, a link between τ and ℓ was added, and the link's parameters are set by hand, to allow computing the posterior $q(\ell)$, without connecting the rest of the video model. This allows us to plot the inferred position based only on the audio model that tracks time delay. The left two columns in Fig. 4 show the learned image template and the variance map. (For the audio model, these images are left blank.) Note that the model observing only the video (third main row) failed to focus on the foreground object and learned a blurred template instead. The inferred position stayed largely flat and occasionally switched as the model was never able to decide what to focus on. This is indicated in the figure both by the white dot in the appropriate position in the frames and in

2. As the audio sampling rate is not exactly a multiple of the video sampling rate, there is a variable number of audio frames that could be captured for each video frame. We simply take the first 1,066 audio samples that are captured during the duration of the video frame.

Fig. 4. Tracking results for the audio only (first row), audio-video (second row), and video only (third row) models. Each row consists of the inference for ℓ_x (bottom), and selected frames from the video sequence (top), positioned in time according to the vertical dotted lines. Note that while the subject moves horizontally, the bottom row of each plot depicts ℓ_x inference on its *vertical* axis for clarity. The area enclosed by the white dots, or *between* the white lines in the case of the audio only model (first row), represents the region(s) occupying the overwhelming majority of the probability mass for the inferred object location.

the position plot (see figure caption). The model observing only the audio data (first main row) provided a very noisy estimate of ℓ_x. As indicated by the white vertical lines, no estimate of ℓ_y could be obtained, due to the horizontal alignment of the microphones.

The full audiovisual model (second main row) learned the template for the foreground model and the variance map that captures the variability in the person's appearance due to the nontranslational head motion and movements of the book. The learned linear mapping between the position and delay variables is shown just below the template variance map. The tracker stays on the object even during the silent periods, regardless of the high background audio noise, and as can be seen form the position plot, the tracker had inferred a smooth trajectory with high certainty, without need for temporal filtering.

In Fig. 5, we illustrate the parameter estimation process by showing the progressive improvement in the audiovisual tracking through several EM iterations. Upon random initialization, both the time delay and location estimates are very noisy. These estimates consistently improve as the iterations proceed and even though the audio part never becomes fully confident in its delay estimate, mostly due to reverberation effects, it still helps the video part achieve near certainty by the tenth iteration. In Fig. 6, we show another example of tracking using the full audio-video model on the data with strong visual distractions. One might note the step-like trends in the position plots in both cases, which really does follow the stepping patterns in the walk of the subjects.

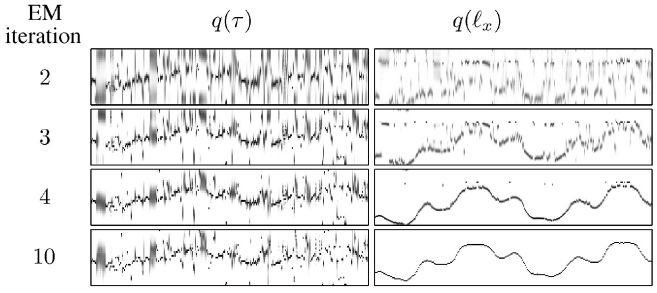

Fig. 5. Learning the combined model with EM iterations. (Left) uncertainty in τ represented by the posterior distribution $q(\tau)$, with darker areas representing more certainty ($\tau \in \{-15, \ldots, 15\}$). Right uncertainty in horizontal position represented by the posterior distribution $q(\ell_x)$, similar shading. The four rows correspond to the inference after 2 (top), 3, 4, and 10 (bottom) iterations, by which point the algorithm has converged. In particular, note how the final uncertainty in τ is a considerable improvement over that obtained by the correlation based result shown in Fig. 1.

7 Conclusions and Future Work

In this paper, we have presented a new approach to building models for joint audio and video data. This approach has produced a new algorithm for object tracking, which is based on a graphical model that combines audio and video variables in a systematic fashion. The model parameters are learned from a multimedia sequence using an EM algorithm. The object trajectory is then inferred from the data via Bayes' rule. Unlike other methods that require precise calibration to coordinate the audio and video, our algorithm performs calibration automatically as part of EM.

Beyond self calibration, our tracker differs from the state of the art in two other important aspects. First, the tracking paradigm does not assume incremental change in object location, which makes the algorithm robust to sudden movements. At the same time, the estimated trajectories are smooth as the model has ample opportunity to explain noise and distractions using data features other than the position itself. This illustrates the power of modeling the mechanism that generates the data.

Second, the paradigm can be extended in several ways. Multiobject situations may be handled by replicating our single object model. Such cases typically involve occlusion, which may be approached using models such as the one proposed in [21]. Multiobject situations also pose the problem of interfering sound from multiple sources. This

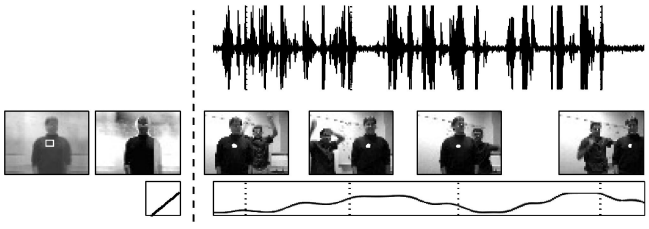

Fig. 6. Tracking results on a data set with significant visual noise.

aspect of the problem may be handled by source separation algorithms of the type developed in [2]. Such models may be incorporated into the present framework and facilitate handling richer multimedia scenarios.

REFERENCES

[1] H. Attias, L. Deng, A. Acero, and J.C. Platt, "A New Method for Speech Denoising and Robust Speech Recognition Using Probabilistic Models for Clean Speech and for Noise," *Proc. Eurospeech,* 2001.

[2] H. Attias and C.E. Schreiner, "Blind Source Separation and Deconvolution: The Dynamic Component Analysis Algorithm," *Neural Computation,* vol. 10, 1998.

[3] S. Ben-Yacoub, J. Luttin, K. Jonsson, J. Matas, and J. Kittler, "Audio-Visual Person Verification," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2000.

[4] A. Blake and M. Isard, *Active Contours.* Springer, 1998.

[5] *Microphone Arrays,* M. Brandstein and D. Ward, eds. Springer, 2001.

[6] M.S. Brandstein, "Time-Delay Estimation of Reverberant Speech Exploiting Harmonic Structure," *J. Accoustic Soc. Am.,* vol. 105, no. 5, pp. 2914-2919, 1999.

[7] C. Bregler and Y. Konig, "Eigenlips for Robust Speech Recognition," *Proc. IEEE Conf. Acoustics, Speech, and Signal Processing,* 1994.

[8] K. Cheok, G. Smid, and D. McCune, "A Multisensor-Based Collision Avoidance System with Application to Military HMMWV," *Proc. IEEE Conf. Intelligent Transportation Systems,* 2000.

[9] R. Cutler and L. Davis, "Look Who's Talking: Speaker Detection Using Video and Audio Correlation," *Proc. IEEE Conf. Multimedia and Expo,* 2000.

[10] R. Cutler, Y. Rui, A. Gupta, J.J. Cadiz, I. Tashev, L.-W. He, A. Colburn, Z. Zhang, Z. Liu, and S. Silverberg, "Distributed Meetings: A Meeting Capture and Broadcasting System," *Proc. ACM Multimedia,* 2002.

[11] R. Duraiswami, D. Zotkin, and L. David, "Active Speech Source Localization by a Dual Coarse-to-Fine Search," *Proc. IEEE Conf. Acoustics, Speech, and Signal Processing,* 2001.

[12] B. Frey and N. Jojic, "Fast, Large-Scale Transformation-Invariant Clustering," *Proc. Advances in Neural Information Processing Systems 2001,* vol. 14, 2002.

[13] B.J. Frey and N. Jojic, "Advances in Algorithms for Inference and Learning in Complex Probability Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* pending publication.

[14] B.J. Frey and N. Jojic, "Transformation-Invariant Clustering Using the EM Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 1, Jan. 2003.

[15] A. Garg, V. Pavlovic, and J.M. Rehg, "Audio-Visual Speaker Detection Using Dynamic Bayesian Networks," *Proc. IEEE Conf. Automatic Face and Gesture Recognition,* 2000.

[16] R. Goecke, J.B. Millar, A. Zelinsky, and J. Robert-Ribes, "Stereo Vision Lip-Tracking for Audio-Video Speech Processing," *Proc. IEEE Conf. Acoustics, Speech, and Signal Processing,* 2001.

[17] J. Hershey and M. Case, "Audio-Visual Speech Separation Using Hidden Markov Models," *Proc. Advances in Neural Information Processing Systems 2001,* vol. 14, 2002.

[18] J. Hershey and J.R. Movellan, "Using Audio-Visual Synchrony to Locate Sounds," *Proc. Advances in Neural Information Processing Systems 1999,* S.A. Solla, T.K. Leen, and K.-R. Muller, eds., vol. 12, 2000.

[19] J.W. Fisher III, T. Darrell, W.T. Freeman, and P.A. Viola, "Learning Joint Statistical Models for Audio-Visual Fusion and Segregation," *Proc. Advances in Neural Information Processing Systems 2000,* vol. 14, 2001.

[20] A.D. Jepson, D.J. Fleet, and T. El-Maraghi, "Robust, On-Line Appearance Models for Vision Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* Dec. 2001.

[21] N. Jojic and B.J. Frey, "Learning Flexible Sprites in Video Layers," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2001.

[22] N. Jojic, N. Petrovic, B.J. Frey, and T.S. Huang, "Transformed Hidden Markov Models: Estimating Mixture Models of Images and Inferring Spatial Transformations in Video Sequences," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2000.

[23] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul, "An Introduction to Variational Methods for Graphical Models," *Learning in Graphical Models,* M.I. Jordan, ed. Norwell Mass.: Kluwer Academic Publishers, 1998.

[24] K. Nakadai, K. Hidai, H. Mizoguchi, H.G. Okuno, and H. Kitano, "Real-Time Auditory and Visual Multiple-Object Tracking for Robots," *Proc. Int'l Joint Conf. Artificial Intelligence,* 2001.

[25] R.M. Neal and G.E. Hinton, "A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants," *Learning in Graphical Models,* M.I. Jordan, ed. pp. 355-368, Norwell Mass.: Kluwer Academic Publishers, 1998.

[26] H.G. Okuno, K. Nakadai, and H. Kitano, "Social Interaction of Humanoid Robot Based on Audio-Visual Tracking," *Proc. Int'l Conf. Industrial and Eng. Applications of Artificial Intelligence and Expert Systems,* 2002.

[27] G. Pingali, G. Tunali, and I. Carlborn, "Audio-Visual Tracking for Natural Interfaces," *Proc. ACM Multimedia,* 1999.

[28] Y. Rui and Y. Chen, "Better Proposal Distributions: Object Tracking Using Unscented Particle Filter," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2000.

[29] M. Slaney and M. Covell, "Facesync: A Linear Operator for Measuring Synchronization of Video Facial Images and Audio Tracks," *Proc. Advances in Neural Information Processing Systems 2000,* vol. 14, 2001.

[30] D.E. Sturim, M.S. Brandstein, and H.F. Solverman, "Tracking Multiple Talkers Using Microphone-Array Measurements," *Proc. IEEE Conf. Acoustics, Speech, and Signal Processing,* 1997.

[31] J. Vermaak, M. Gangnet, A. Blake, and P. Perez, "Sequential Monte Carlo Fusion of Sound and Vision for Speaker Tracking," *Proc. IEEE Int'l Conf. Computer Vision,* 2001.

[32] H. Wang and P. Chu, "Voice Source Localization for Automatic Camera Pointing System in Cideoconferencing," *Proc. IEEE Conf. Acoustics, Speech, and Signal Processing,* 1997.

[33] K. Wilson, N. Checka, D. Demirdjian, and T. Darrell, "Audio-Video Array Source Localization for Perceptual User Interfaces," *Proc. Workshop Perceptive User Interfaces,* 2001.

[34] D.N. Zotkin, R. Duraiswami, and L.S. Davis, "Joint Audio-Visual Tracking Using Particle Filters," *EURASIP J. Applied Signal Processing,* vol. 11, pp. 1154-1164, 2002.

Matthew J. Beal received the bachelor and master degrees in physics from the University of Cambridge and completed the doctoral degree in 2003 at University College London, United Kingdom, on the subject of variational algorithms for approximate Bayesian inference. He is now a postdoctoral fellow at the University of Toronto, Canada, working in the Machine Learning and Statistics groups.

Nebojsa Jojic received the doctoral degree in 2002 from the University of Illinois at Urbana-Champaign (UIUC), where he received the Robert T. Chien Memorial Award for his research on generative models for computer vision. In addition to conducting research at the University of Illinois and Microsoft, he has consulted at the Hong Kong University of Science and Technology and spent a semester at the University of Illinois at Chicago. Currently, he is a researcher in the Interactive Visual Media Group at Microsoft Research. His research interests include signal processing, machine learning, computer vision, and computer graphics. He has published more than 30 papers in these areas. He is a member of the IEEE Computer Society.

Hagai Attias received the PhD degree in theoretical physics from Yale University. He was a Sloan postdoctoral fellow at the University of California, San Francisco, and a senior research fellow at the University of London. He is a researcher in the machine-learning group at Microsoft Research. He develops intelligent robots that use probabilistic models for perception, planning, and learning.