Audio-Video Sensor Fusion with Probabilistic Graphical Models

Matthew J. Beal[1,2], Hagai Attias[1], and Nebojsa Jojic[1]

[1] Microsoft Research, 1 Microsoft Way,
Redmond, WA 98052, USA
{hagaia,jojic}@microsoft.com
[2] Gatsby Computational Neuroscience Unit, University College London,
17 Queen Square, London WC1N 3AR, UK
m.beal@gatsby.ucl.ac.uk

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.

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 which 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 using also audio data could continue focusing on the object by following

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2350, pp. 736–750, 2002.

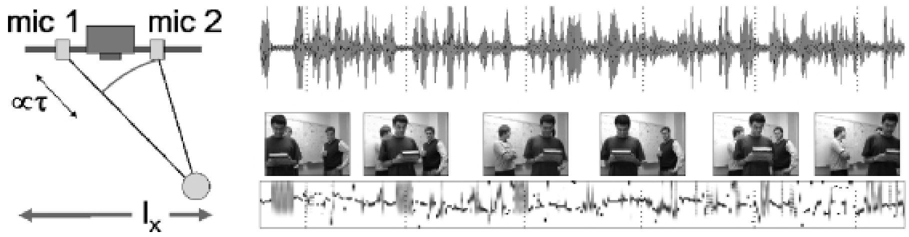

Fig. 1. (Top) audio waveform. (Middle) selected frames from associated video sequence (120×160 pixels^2). (Bottom) posterior probability over time delay τ (vertical axis, $\tau \in \{-15, \ldots, 15\}$) for each frame of the sequence; darker areas represent higher probability, and each frame has been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

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 audio-visual 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., [4]). Also, the time delay between the signals arriving at the microphones is correlated with the position of the person in the image (see, e.g., [9],[10]). 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 audio-visual 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, audio-visual correlations usually exist only intermittently. This paper presents a new framework for fusing audio and video data. In this framework, which is based on probabilistic generative modeling, we construct a model describing the 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 audio-visual 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 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 of 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 which 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 [10],[3]. We note in particular the method of [9] which, while using a 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 this 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.

3 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 model by linking the time delay between the audio signals to the spatial location of the object's image.

3.1 Audio Model

We model the audio signals x_1, 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.

x_1, x_2 are described in terms of an original audio signal a. We assume that 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 a is contaminated by additive sensor noise with precision matrices ν_1, ν_2. To account for the variability of that signal, it is described by a mixture model. Denoting the component label by r, each component has mean zero, a precision matrix η_r, and a prior probability π_r. Viewing it in the frequency

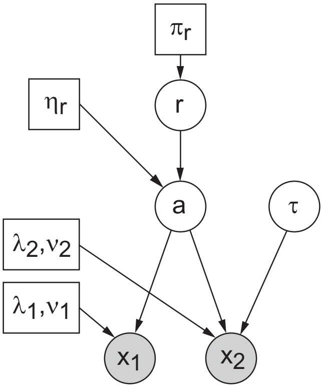

Fig. 2. Graphical model for the audio data.

domain, the precision matrix corresponds to the inverse of the *spectral template* for each component. Hence, we have

$$\begin{aligned} p(r) &= \pi_r \ , \\ p(a \mid r) &= \mathcal{N}(a \mid 0, \eta_r) \ , \\ p(x_1 \mid a) &= \mathcal{N}(x_1 \mid \lambda_1 a, \nu_1) \ , \\ p(x_2 \mid a, \tau) &= \mathcal{N}(x_2 \mid \lambda_2 L_\tau a, \nu_2) \ , \end{aligned} \tag{2}$$

where L_τ denotes the temporal shift operator, i.e., $(L_\tau a)_n = a_{n-\tau}$. The prior probability for a delay τ is assumed flat, $p(\tau) = const$. A similar model was used in [2] to perform noise removal from speech signals. In that paper, the joint $p(a, r)$ served as a speech model with a relatively large number of components, which was pre-trained on a large clean speech dataset. Here, $p(a, r)$ has only a few components and its parameters are learned from audio-video data as part of the full model.

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

$$\mathcal{N}(x \mid \mu, \nu) \propto \exp\left[-\frac{1}{2}(x-\mu)^T \nu (x-\mu)\right] \ . \tag{3}$$

Fig. 2 displays a graphical representation of the audio model. As usual with graphical models (see, e.g., [8]), 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. 2, the process of generating the observed microphone signals

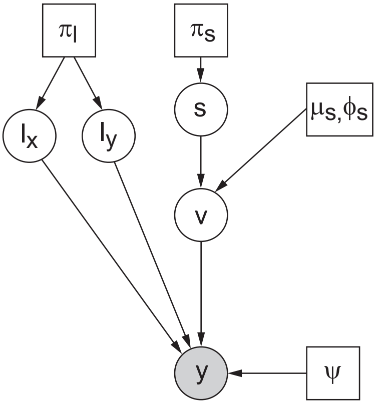

Fig. 3. Graphical model for the video data.

starts with picking a spectral component r with probability $p(r)$, followed by drawing a signal a from the Gaussian $p(a \mid r)$. Separately, a time delay τ is also picked. The signals x_1, x_2 are then drawn from the undelayed Gaussian $p(x_1 \mid a)$ and the delayed Gaussian $p(x_2 \mid a, \tau)$, respectively.

3.2 Video Model

In analogy with the audio frames, we model the video frames as follows. Denote the observed frame by 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 v that has been shifted by $l = (l_x, l_y)$ pixels in the x and y directions, respectively,

$$y_n = v_{n-l} \ , \tag{4}$$

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

$$\begin{aligned} p(s) &= \pi_s \ , \\ p(v \mid s) &= \mathcal{N}(v \mid \mu_s, \phi_s) \ , \\ p(y \mid v, l) &= \mathcal{N}(y \mid G_l v, \psi) \ , \end{aligned} \tag{5}$$

where G_l denotes the shift operator, i.e. $(G_l v)_n = v_{n-l}$. The prior probability for a shift l is assumed flat, $p(l) = const$. This model was used in [5] for video based object tracking and stabilization.

Fig. 3 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. 3, 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 v from the Gaussian $p(v \mid s)$. 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 y is then drawn from the shifted Gaussian $p(y \mid v, l)$.

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.

3.3 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 ν_τ,

$$p(\tau \mid l) = \mathcal{N}(\tau \mid \alpha l_x + \alpha' l_y + \beta, \nu_\tau) \ . \tag{6}$$

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 Eq. (6) fuses the two models into a single one, whose graphical representation is displayed in Fig. 4.

4 Parameter Estimation and Object Tracking

Here we outline the derivation of an EM algorithm for the graphical model in Fig. 4. 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 x_1, x_2, y and the hidden ones a, τ, r, v, l, s. As Fig. 4 shows, this distribution factorizes as

$$\begin{aligned} p(x_1, x_2, y, a, \tau, r, v, l, s \mid \theta) \ &= \ p(x_1 \mid a)\, p(x_2 \mid a, \tau)\, p(a \mid r) \\ &\quad \cdot p(r)\, p(y \mid v, l)\, p(v \mid s)\, p(s)\, p(\tau \mid l)\, p(l) \ . \end{aligned} \tag{7}$$

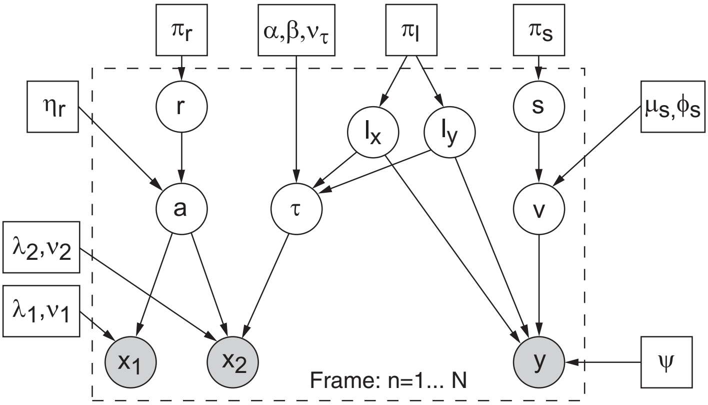

Fig. 4. Graphical model for the joint audio-video data. The dotted rectangle denotes i.i.d. frames and has the following meaning: everything it encompasses, i.e., all model variables, has value that is frame dependent; everything it leaves out, i.e., the model parameters, is frame independent.

This is the product of the joint distributions defined by the audio and video models and their link. The model parameters are

$$\theta = \{\lambda_1, \nu_1, \lambda_2, \nu_2, \eta_r, \pi_r, \psi, \mu_s, \phi_s, \pi_s, \alpha, \alpha', \beta, \nu_\tau\} \ . \tag{8}$$

Ultimately, we are interested in tracking the object based on the data, i.e., obtaining a position estimate $\hat{l}$ 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, $p(l \mid x_1, x_2, y)$, for each frame, is computed. This distribution provides the most likely position value via

$$\hat{l} = \arg\max_l p(l \mid x_1, x_2, y) \ , \tag{9}$$

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.

4.1 E-step

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

$$p(a, \tau, r, v, l, s \mid x_1, x_2, y, \theta) = \frac{p(x_1, x_2, y, a, \tau, r, v, l, s \mid \theta)}{p(x_1, x_2, y \mid \theta)} \ , \tag{10}$$

where $p(x_1, x_2, y \mid \theta)$ is obtained from the model distribution by marginalizing over the hiddens. In our model, it can be shown that the posterior has a factorized form, as does the model distribution (7). To describe it, we switch to a notation that uses q to denote a posterior distribution conditioned on the data. Hence,

$$p(a, \tau, r, v, l, s \mid x_1, x_2, y, \theta) = q(a \mid \tau, r) q(v \mid l, s) q(\tau \mid l) q(l, r, s) \ . \qquad (11)$$

This factorized form follows from our model. The q notation omits the data, as well as the parameters. Hence, $q(a \mid \tau, r) = p(a \mid \tau, r, x_1, x_2, y, \theta)$, and so on.

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 $q(a \mid \tau, r)$ and the video posterior $q(v \mid l, s)$ are both Gaussian,

$$\begin{aligned} q(a \mid \tau, r) &= \mathcal{N}(a \mid \mu^a_{\tau,r}, \nu^a_r) \ , \\ q(v \mid l, s) &= \mathcal{N}(v \mid \mu^v_{l,s}, \nu^v_s) \ . \end{aligned} \qquad (12)$$

The means $\mu^a_{\tau,r}$, $\mu^v_{l,s}$ and precisions ν^a_r, ν^v_s are straightforward to compute; note that the precisions do not depend on the shift variables τ, l. One particularly simple way to obtain them is to consider (11) and observe that its logarithm satisfies

$$\log p(a, \tau, r, v, l, s \mid x_1, x_2, y, \theta) = \log p(x_1, x_2, y, a, \tau, r, v, l, s \mid \theta) + \mathit{const.} \ (13)$$

where the constant is independent of the hiddens. Due to the nature of our model, this logarithm is quadratic in a and v. To find the mean of the posterior over v, set the gradient of the log probability w.r.t. v to zero. The precision is then given by the negative Hessian, and we have

$$\begin{aligned} \mu^v_{l,s} &= (\nu^v_s)^{-1}(\phi_s \mu_s + G_l^\top \psi y) \ , \\ \nu^v_s &= \phi_s + \psi \ . \end{aligned} \qquad (14)$$

Equations for the mean and precision of the posterior over a are obtained in a similar fashion.

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

$$q(\tau \mid l) \propto p(\tau \mid l) \exp(\lambda_1 \lambda_2 \nu_1 \nu_2 (\nu^a_r)^{-1} c_\tau) \ , \qquad (15)$$

where

$$c_\tau = \sum_n x_{1n} x_{2,n+\tau} \qquad (16)$$

is the cross-correlation between the microphone signals x_1 and x_2. Finally, the last component of the posterior is the probability table $q(l, r, s)$, whose form is omitted.

The calculation of $q(\tau \mid l)$ involves a minor but somewhat subtle point. Since throughout the paper we work in discrete time, the the delay τ in our model is generally regarded as a discrete variable. In particular, $q(\tau \mid l)$ is a discrete probability table. However, for reasons of mathematical convenience, the model distribution $p(\tau \mid l)$ (6) treats τ as continuous. Hence, the posterior $q(\tau \mid l)$ 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., [8]). To derive it rigorously one proceeds as follows. First, write down the form of the approximate posterior as a sum of delta functions,

$$q(\tau \mid l) = \sum_n q_n(l)\delta(\tau - \tau_n) \; , \tag{17}$$

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 $q_n(l)$ 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 (17). In this paper we write $q(\tau \mid l)$ rather than $q_n(l)$ to keep notation simple.

4.2 M-step

The M-step performs updates of the model parameters θ (8). The update rules are derived, as usual, by considering the objective function

$$\mathcal{F}(\theta) = \langle \log p(x_1, x_2, y, a, \tau, r, v, l, s \mid \theta) \rangle \; , \tag{18}$$

known as the averaged complete data likelihood. We use the notation $\langle \cdot \rangle$ to denote averaging w.r.t. the posterior (11) 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 $\mathcal{F}$ w.r.t. that parameter to zero.

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

$$\begin{aligned} \mu_s &= \frac{\langle \sum_l q(l,s)\mu^v_{ls} \rangle}{\langle q(s) \rangle} \; , \\ \phi_s^{-1} &= \frac{\langle \sum_l q(l,s)(\mu^v_{ls} - \mu_s)^2 + q(s)(\nu^v_{ls})^{-1} \rangle}{\langle q(s) \rangle} \; , \\ \pi_s &= \langle q(s) \rangle \; , \end{aligned} \tag{19}$$

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

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

$$\begin{aligned}
\alpha &= \frac{\langle l_x \tau \rangle - \langle \tau \rangle \langle l_x \rangle}{\langle l_x^2 \rangle - \langle l_x \rangle^2} \\
\beta &= \langle \tau \rangle - \alpha \langle l_x \rangle \\
{\nu_\tau}^{-1} &= \left\langle \tau^2 \right\rangle + \alpha^2 \left\langle l_x^2 \right\rangle + \beta^2 + 2\alpha\beta \langle l_x \rangle - 2\alpha \langle \tau l_x \rangle - 2\beta \langle \tau \rangle \ , \qquad (20)
\end{aligned}$$

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

A note about complexity. According to Eq. (19), 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 [6], a more careful examination of Eq. (19), in combination with Eq. (14), 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.

4.3 Tracking

Tracking is performed as part of the E-step using (9), where $p(l \mid x_1, x_2, y)$ is computed from $q(\tau, l)$ above by marginalization. For each frame the mode of this posterior distribution of l represents the most likely translation for the object template, and the width of the distribution the degree of uncertainty in this inference.

5 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 16kHz. This means that each frame contained one 160×120 image frame and two 1066 samples long audio frames. 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 frames per second per iteration of EM. Convergence was generally achieved within just 10 iterations.

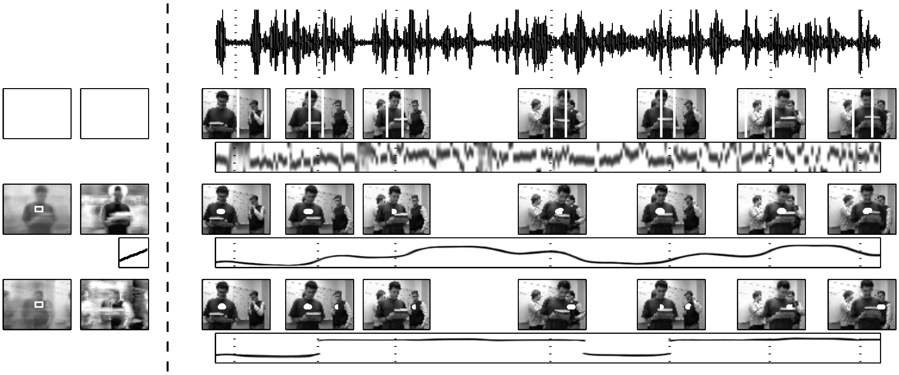

Fig. 5. Tracking results for the audio only (first row), audio-video (second row), and video only (third row) models. Each row consists of the inference for l_x (bottom), and selected frames from the video sequence (top), positioned in time according to the vertical dotted lines. Note that while the 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.

We present the results on two sequences that had substantial background audio noise and visual distractions. In Fig. 5, we compare the results of tracking using the audio only model (Fig. 2), full audio-video model (Fig. 4), and the video only model (Fig. 3) 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 (whose parameters were computed separately) to allow computing the posterior $q(l)$. The left two columns in Fig. 5 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 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 audio-visual 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 non-translational 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

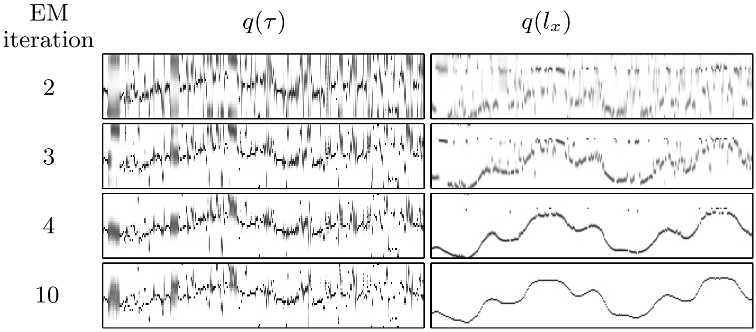

Fig. 6. Learning the combined model with EM iterations. (Left) uncertainty in τ represented by the posterior distribution $q(\tau)$, with darker areas representing more certainty ($\tau \in \{-15, \ldots, 15\}$). Right uncertainty in horizontal position represented by the posterior distribution $q(l_x)$, 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.

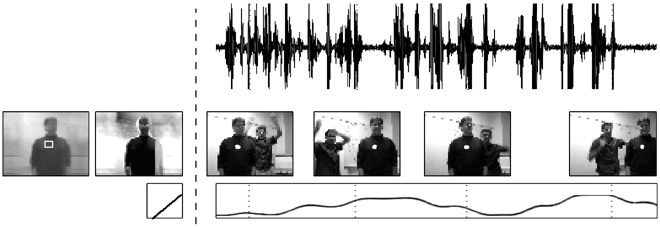

Fig. 7. Tracking results on a data set with significant visual noise.

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. 6 we illustrate the parameter estimation process by showing the progressive improvement in the audio-visual 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. 7, 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.

6 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 which 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. Multi-object 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 [7]. Multi-object situations also pose the problem of interfering sound from multiple sources. This aspect of the problem may be handled by source separation algorithms of the type developed in [1]. Such models may be incorporated into the present framework and facilitate handling richer multimedia scenarios.

References

[1] H. Attias and C.E. Schreiner (1998), Blind source separation and deconvolution: the dynamic component analysis algorithm. Neural Computation 10, 1373-1424.

[2] H. Attias et al (2001), A new method for speech denoising and robust speech recognition using probabilistic models for clean speech and for noise. Proc. Eurospeech 2001.

[3] M. S. Brandstein (1999). Time-delay estimation of reverberant speech exploiting harmonic structure. Journal of the Accoustic Society of America 105(5), 2914-2919.

[4] C. Bregler and Y. Konig (1994). Eigenlips for robust speech recognition. Proc. ICASSP.

[5] B. Frey and N. Jojic (1999). Estimating mixture models of images and inferring spatial transformations using the EM algorithm. Proc. of IEEE Conf. on Computer Vision and Pattern Recognition.

[6] B. Frey and N. Jojic (2001). Fast, large-scale transformation-invariant clustering. Proc. of Neural Information Processing Systems, December 2001, Vancouver, BC, Canada.

[7] N. Jojic and B. Frey (2001). Learning flexible sprites in video layers. Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, Maui, HI.

[8] Jordan, M.I. (Ed.) (1998). *Learning in Graphical Models*. MIT Press, Cambridge, MA.

[9] J. Vermaak, M. Gagnet, A. Blake and P. Pérez (2001). Sequential Monte-Carlo fusion of sound and vision for speaker tracking. Proc. IEEE Intl. Conf. on Computer Vision.

[10] H. Wang and P. Chu (1997). Voice source localization for automatic camera pointing system in videoconferencing. Proc. ICASSP, 187-190.