A SELF-CALIBRATING ALGORITHM FOR SPEAKER TRACKING BASED ON AUDIO-VISUAL STATISTICAL MODELS

Matthew J. Beal, Nebojsa Jojic, Hagai Attias*

Microsoft Research, One Microsoft Way, Redmond WA

m.beal@gatsby.ucl.ac.uk, {jojic,hagaia}@microsoft.com

Videos available at http://www.gatsby.ucl.ac.uk/~beal/icassp02.html

ABSTRACT

We present a self-calibrating algorithm for audio-visual tracking using two microphones and a camera. The algorithm uses a parametrized statistical model which combines simple models of video and audio. Using unobserved variables, the model describes the process that generates the observed data. Hence, it is able to capture and exploit the statistical structure of the audio and video data, as well as their mutual dependencies. The model parameters are estimated by the EM algorithm; object templates are learned and automatic calibration is performed as part of this procedure. Tracking is done by Bayesian inference of the object location using the model. Successful performance is demonstrated on real multimedia clips.

1. INTRODUCTION

Overwhelmingly, audio and video signals are treated separately in most systems dealing with digital media, be it stored home or professional videos, live audio in speech recognition systems or live video in tracking applications. Since both signals usually come from the same sources (a talking head, a moving car, etc.), it is obvious that an optimal system would exploit correlations among the two modalities.

For example, in Fig. 1, for the audio-visual capture system on the left we show an audio waveform captured by one of the microphones and a few frames captured by the camera. 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 (e.g. [1]). Also, the time delay between the signals arriving at the microphones is correlated with the position of the person in the image as in [2, 3]. In principle, tasks such as tracking may be performed better by taking advantage of these correlations.

However, relevant features 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 impossible 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.

*Current address: Gatsby Computational Neuroscience Unit, University College, London WC1N 3AR, UK.

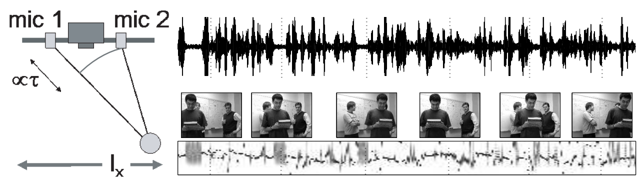

Fig. 1. **(left)** Experimental setup for audio-visual capture: ○ denotes the object of interest, and l_x its horizontal position. **(right top)** Audio waveform. **(right middle)** Selected frames from associated video sequence (120x160 pixels^2) **(right bottom)** Posterior probability over the time delay τ (vertical axis, $\tau \in \{-15, \dots, 15\}$) for each frame of the sequence: darker areas represent higher probability, with each frame separately normalised. The direction of time is left to right.

This paper presents a new framework for jointly modeling audio-visual data and exploiting correlations between the two modalities in a systematic manner. This framework uses statistical models to describe the observed data in terms of the process that generated them. In particular, the 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. Finally, the model also describes the dependence of the time delay on the spatial location.

Statistical models have several important advantages which make them ideal for our purpose. First, since we explicitly model the actual sources of variability in the problem, e.g. object appearance and background noise, the resulting algorithm turns out to be very robust. Second, using statistical models leads to a solution by a Bayes optimal estimation algorithm. Third, parameter estimation and object tracking are both performed efficiently using the EM algorithm, as is customary for models with unobserved variables.

To illustrate the power of the graphical modeling paradigm, we make the problem more difficult by assuming no prior calibration of the system, such as the parameters needed to define the mapping between the object position in video and the waveform delay in audio, or the attenuation parameters of the microphones. Also, we do not allow manual initialization in the first frame, i.e., defining the template or the contours of the object to be tracked. This is in contrast with previous research on the subject, which usually required specific and calibrated configurations, as in [3, 4]. Even the recent paper [2] which, in terms of the probabilistic approach, is similar in spirit to our research, reports results that require 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.

In our approach, the only information the model is allowed to use before or during the tracking is the raw data itself. The tracking algorithm we present in the next few sections will have to find the source of the sound in the images without any help from the user, learn the object's appearance, microphone attenuations, the mapping between the audio waveform delay and the object position in images, and sensor noise parameters for all sensors. Using all these parameters, the system will find the object automatically in all frames, including the first, making the best use of all sensors through various situations, such as increased visual distraction or absence of the audio signal.

2. THE AUDIO-VISUAL GENERATIVE MODEL

Video components: Video frames are modeled using a statistical model introduced in [5] and termed transformed mixture of Gaussians (TMG). This is a simple general-purpose generative model that describes the observed image y in terms of an original image z that has been shifted by $l = (l_x, l_y)$ pixels, and further contaminated by additive noise with precision matrix ψ. To account for the variability in the original image, z is modeled by a mixture model with components s. Component s consists of a template with mean μ_s and precision matrix ϕ_s, and has a prior probability π_s. Hence, we have

$$p(z \mid s) = \mathcal{N}(z \mid \mu_s, \phi_s)\ , \qquad p(s) = \pi_s\ ,$$
$$p(y \mid z, l) = \mathcal{N}(y \mid G_l z, \psi)\ . \tag{1}$$

G_l denotes the shift operator, i.e. $(G_l z)_n = z_{n-l}$, where n runs over the pixel coordinates. The prior probability for a shift l is assumed flat, $p(l) = const$. In our notation, $\mathcal{N}(z \mid \mu, \phi)$ denotes a Gaussian distribution over the random vector z with mean μ and precision matrix (defined as the inverse covariance matrix) ϕ.

The model parameters, including the image templates μ_s, their precisions ϕ_s, and the noise precision ψ, are learned from sequence data using EM. This model has proven successful for tasks such as object tracking and image stabilization [5], when the object of interest is large and salient compared to the noisy background. Tasks are generally performed by Bayesian inference: Tracking, for example, is done by computing the posterior distribution over the shift, $p(l \mid y)$, and identifying the most probable l.

However, on the data in Fig. 1 this model fails to perform tracking (see the last row in Fig. 3) as it sometimes locks on the background. Instead of using more complex and thus more expensive models as in [6], we focus on improving performance by combining it with an equally simple model for the audio data.

Audio components: In analogy with the video model, the audio model describes the observed signals x_1, x_2 in terms if an original signal x, which has been attenuated by a factor λ_i on its way to microphone $i = 1, 2$. It is received at microphone $i = 2$ with a delay of τ time points relative to $i = 1$. To account for the variability in the original signal, x is modeled by a mixture model with components r. Each component has mean zero, a precision matrix η_r, and a prior probability π_r. Viewing it in the frequency domain, the precision matrix corresponds to the inverse of the spectrum template for each component. A similar model was used in [7] to perform noise removal from speech signals. Hence, we have

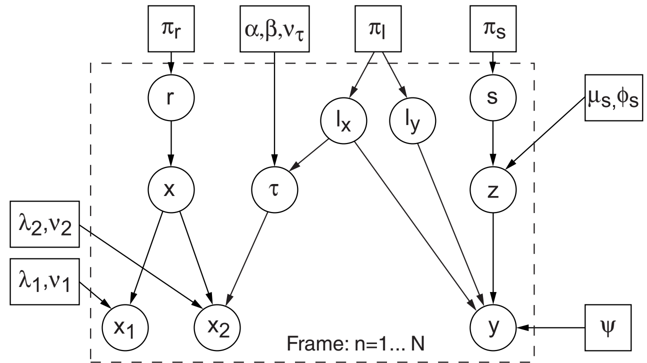

Fig. 2. Graphical model representation of the full Bayes net to model both audio and video signals jointly. The dotted rectangle encompasses the hidden variables *and* the data, and denotes *iid* repetitions over the data (frames of the multimedia sequence). That is to say the same parameters are used to model all the data; just the hidden variables change across time frames.

$$p(x \mid r) = \mathcal{N}(x \mid 0, \eta_r)\ , \qquad p(r) = \pi_r\ ,$$
$$p(x_1 \mid x) = \mathcal{N}(x_1 \mid \lambda_1 x, \nu_1)\ ,$$
$$p(x_2 \mid x, \tau) = \mathcal{N}(x_2 \mid \lambda_2 L_\tau x, \nu_2)\ . \tag{2}$$

L_τ denotes the temporal shift operator, in analogy with the spatial shift operator G_l above.

Audio-visual link: The delay τ on the image location l are related by a noisy linear mapping:

$$p(\tau \mid l) = \mathcal{N}(\tau \mid \alpha l_x + \beta, \nu_\tau)\ . \tag{3}$$

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 (the extension to a mapping involving both dimension is straightforward). It can be shown that the linear approximation is fairly accurate for the pinhole camera and the microphone baseline small in comparison to the object distance. To account for deviations from linearity and other inaccuracies in the simplified model, such as reverberation, we allow the mapping to be noisy, with a noise precision ν_τ.

A graphical representation of the audio-visual generative model is displayed in Fig. 2.

3. PARAMETER ESTIMATION AND OBJECT TRACKING

In the model of Fig. 2, the joint distribution of the observed sensor signals and the unobserved original signals, shifts, and component labels, is given by

$$p(x_1, x_2, y, \tau, l, r, s, x, z) = p(x_1 \mid x) p(x_2 \mid x, \tau) p(x \mid r)$$
$$\cdot\, p(r) p(y \mid z, l) p(z \mid s) p(s) p(\tau \mid l) p(l), \tag{4}$$

which is the product of the joint distributions defined by the audio and the video models.

The model parameters $\theta = \{\lambda_1, \nu_1, \lambda_2, \nu_2, \eta_r, \pi_r, \psi, \mu_s, \phi_s, \pi_s, \alpha, \beta, \nu_\tau\}$ are estimated from the data sequence using the EM algorithm, which is straightforward to derive for this model (some details are given in the appendix). In the E-step we compute sufficient statistics (SS) for the unobserved variables. As is well known, the sufficient statistics are moments of the posterior distribution over the unobserved variables $p(x, \tau, r, z, l, s \mid x_1, x_2, y)$, which is obtained by Bayes' rule. The required SS turn out to be (1) mean and covariance of x given r, τ, (2) mean and covariance of z given s, l, (3) joint distribution of l, s, r, (4) distribution of τ given l_x. As usual, these quantities are conditioned on the observed data, and are computed separately at each frame. Using these SS, the model parameters θ are updated in the M-step. We then recompute the SS and repeat until an appropriate convergence criterion is satisfied.

After estimating the parameters, tracking is performed by computing the posterior distribution $p(l \mid x_1, x_2, y)$ of the location variable l from the joint distribution in (3) above. Our estimate for the object's location at each frame is given by its most likely location given the data, $\hat{l} = \arg\max_l p(l \mid x_1, x_2, y)$.

In Fig. 5 we show how the posterior $p(l, \tau \mid x_1, x_2, y)$ evolves over the iterations of EM.

4. RESULTS

We tested the tracking algorithm on several audio-visual sequences captured by the setup in Fig. 1 consisting of low-cost off the shelf equipment. The video capture rate was 15 frames a second and the audio was digitized at the sampling rate of 16kHz. This means that each audio-visual frame contained one 160x120 image frame and two 1066 samples long audio waveforms. No model parameters were set by hand, and no initialization was required; the only input to the algorithm was the raw data itself. 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 parameters. The processing speed of our Matlab implementation was about 50 frames per second per iteration of EM. Convergence was generally achieved within only 10 iterations.

We present the results on two sequences that had substantial background audio noise and visual distractions. In Fig. 3, we compare the results of tracking using an audio only model, full audio-visual model and the video only model 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 (Fig. 1).

We assumed a single template class s and a single audio source class r. The left two columns in Fig. 3 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 (bottom 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. The model observing only the audio data (top 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 (middle 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 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. 4, we show another example of tracking using the full audio-visual 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.

In Fig. 5 we illustrate the parameter estimation process by showing the progressive improvement in the audio-visual tracking through several EM iterations. Upon random initalization, 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.

5. CONCLUSIONS

We developed a self-calibrating audio-visual tracking algorithm. The algorithm uses a combination of simple structured probability models of video and audio. Parameter estimation, including automatic calibration, is performed by the EM algorithm. We used cheap, off the shelf cameras and microphones.

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. For instance, [8] has recently proposed more sophisticated layered video models that can capture multiple possibly occluding objects in the scenes, while powerful audio source separation and speech denoising algorithms were developed in [9, 7]; both sets of work use the statistical modeling framework. Such models may be incorporated into the present framework and facilitate handling richer multimedia scenarios.

6. REFERENCES

[1] C. Bregler and Y. Konig, "Eigenlips for robust speech recognition," in *Proc. ICASSP*, 1994.

[2] J. Vermaak, M. Gagnet, A. Blake, and P. Pérez, "Sequential Monte-Carlo fusion of sound and vision for speaker tracking," in *Proc. IEEE Intl. Conf. on Computer Vision*, 2001, vol. 1, pp. 741–746.

[3] H. Wang and P. Chu, "Voice source localization for automatic camera pointing system in videoconferencing," in *Proc. ICASSP*, 1997, pp. 187–190.

[4] M. S. Brandstein, "Time-delay estimation of reverberant speech exploiting harmonic structure," *Journal of the Acoustics Society of America*, vol. 105, no. 5, pp. 2914–2919, 1999.

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

[6] B. Frey and N. Jojic, "Fast, large-scale transformation-invariant clustering," in *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002, MIT Press.

[7] H. Attias, "A new method for speech denoising and robust speech recognition using probabilistic models for clean speech and for noise," in *Proc. Eurospeech*, 2001.

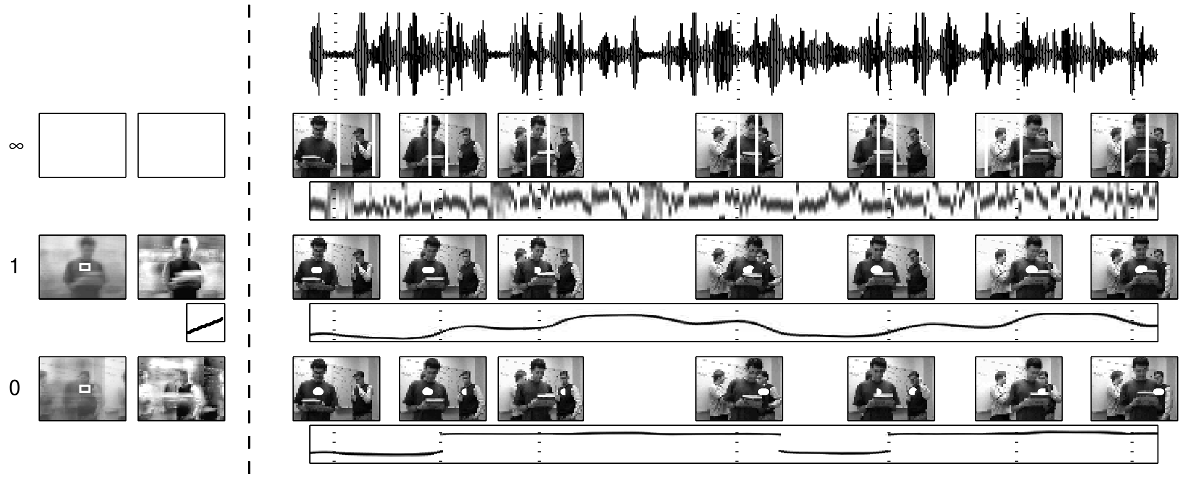

Fig. 3. Tracking results for the **(first row)** audio only, **(second row)** combined, and **(third row)** video only models. Each row consists of **(bottom)** the inference for l_x, and **(top)** selected frames from the video sequence, positioned in time according to the vertical dotted lines. Note that whilst the subject moves horizontally the bottom plot of each row depicts l_x inference on its *vertical* axis for clarity.

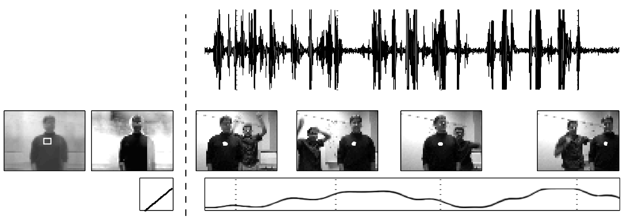

Fig. 4. Tracking results on a data set with significant visual noise.

[8] N. Jojic and B. Frey, "Learning flexible sprites in video layers," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.

[9] H. Attias and C. E. Schreiner, "Blind source separation and deconvolution: the dynamic component analysis algorithm," *Neural Computation*, vol. 10, pp. 1373–1424, 1998.

Appendix: EM algorithm details

E-step. It can be shown that the posterior over the unobserved variables can be written in a product form,

$$p(x,\tau,r,z,l,s \mid x_1,x_2,y) = q(x \mid \tau,r)q(z \mid l,s)q(\tau \mid l)q(l,r,s)$$

where each factor q is understood to be conditioned on the data. $q(z \mid l,s)$ is Gaussian, with mean $\mu^z_{l,s}$ and precision $\nu^z_{l,s}$. $q(x \mid \tau,r)$ is also Gaussian, with mean $\mu^x_{\tau,r}$ and precision $\nu^x_{\tau,r}$. These means and precisions constitute SS and are straightforward to compute. For example,

$$\mu^z_{l,s} = (\nu^z_{l,s})^{-1}(\phi_s\mu_s + G_l^\top \psi y)\ , \qquad \nu^z_{l,s} = \phi_s + \psi \tag{5}$$

with analogous equations for $\mu^x_{\tau,r}$ and $\nu^x_{\tau,r}$. Another SS is the conditional probability table

$$q(\tau \mid l) \propto p(\tau \mid l)\exp(\lambda_1\lambda_2\nu_1\nu^2(\nu^x_{\tau,r})^{-1}r_\tau) \tag{6}$$

where r_τ is the cross-correlation between x_1 and x_2. The last SS is the table $q(l,r,s)$ whose form is omitted.

M step. Where used, $\langle\cdot\rangle$ notation denotes expectation under the hidden state posterior $Q(\tau,l_x,l_y,s,\mathbf{x},\mathbf{z})$ *and* expectation over all the data. $\circ$ denotes element-wise multiplication. Updates for the video model are:

$$\boldsymbol{\mu}_s \leftarrow \nu_s^{-1}\left(\phi_s\boldsymbol{\mu}_s + \psi\left\langle G_l^\top \mathbf{y}|s\right\rangle\right)$$

$$\phi_s^{-1} \leftarrow \nu_s^{-1} + (I-\nu_s^{-1}\phi_s)^2 \circ \boldsymbol{\mu}_s^2 + \psi^2\nu_s^{-2} \circ \left\langle [G_l^\top\mathbf{y}]^2|s\right\rangle$$
$$- 2\psi\nu_s^{-1}\boldsymbol{\mu}_s(I-\nu_s^{-1}\phi_s)\circ\left\langle G_l^\top\mathbf{y}|s\right\rangle$$

Updates for the audio-visual link parameters:

$$\alpha \leftarrow \frac{\langle l_x\tau\rangle - \langle\tau\rangle\langle l_x\rangle}{\langle l_x^2\rangle - \langle l_x\rangle^2} \qquad \beta \leftarrow \langle\tau\rangle - \alpha\langle l_x\rangle$$

$$\frac{1}{\nu_\tau} \leftarrow \langle\tau^2\rangle + \alpha^2\langle l_x^2\rangle + \beta^2 + 2\alpha\beta\langle l_x\rangle - 2\alpha\langle\tau l_x\rangle - 2\beta\langle\tau\rangle$$

For the sake of brevity, we omit the other update equations. They can be derived following the instructions in the paper.

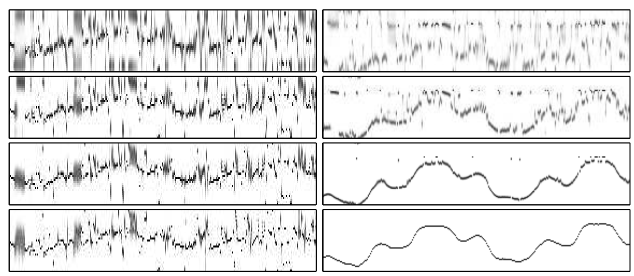

Fig. 5. Learning the combined model with EM iterations. **(left)** Uncertainty in τ represented by the posterior distribution $Q(\tau)$, with darker areas representing more certainty ($\tau \in \{-15,\ldots,15\}$) and **(right)** Uncertainty in horizontal position represented by the posterior distribution $Q(l_x)$, similar shading. The **(rows)** correspond to the inference after 2, 3, 4 and 10 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 naive model in Figure 1.