SCHOOL OF ENGINEERING - STI
ELECTRICAL ENGINEERING INSTITUTE
SIGNAL PROCESSING LABORATORY

Mihai Gurban

EPFL - FSTI - IEL - LTS
Station 11
Switzerland-1015 LAUSANNE

Phone: +4121 6934682
Fax: +4121 6937600
e-mail: mihai.gurban@epfl.ch

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

AUDIO-VISUAL RELIABILITY ESTIMATES USING STREAM ENTROPY FOR SPEECH RECOGNITION

Mihai Gurban and Jean-Philippe Thiran

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Laboratory

Technical Report LTS-2009-009

September 10, 2009

Audio-visual reliability estimates using stream entropy for speech recognition

LTS Technical Report

LTS-REPORT-2009-009

Mihai Gurban, and Jean-Philippe Thiran

Abstract

We present a method for multimodal fusion based on the estimated reliability of each individual modality. Our method uses an information theoretic measure, the entropy derived from the state probability distribution for each stream, as an estimate of reliability. Our application is audio-visual speech recognition. The two modalities, audio and video, are weighted at each time instant according to their reliability. In this way, the weights vary dynamically and are able to adapt to any type of noise in each modality, and more importantly, to unexpected variations in the level of noise.

Index Terms

Audio-visual speech recognition, multimodal fusion.

I. Introduction

MULTIMODAL fusion, that is, merging information from different modalities, is not trivial. First, the data streams may have different representations, different temporal rates and different ranges of variation. Second, they may be corrupted by noise in varying ways and at different moments. This is why multimodal fusion or integration is a very active area of research.

The case of audio-visual speech recognition (AVSR) is more complex from the multimodal integration point of view as compared to, for example, multimodal person identification. In AVSR both modalities are time-varying, which is not the case in identification, where typically at least some of the modalities are static, like face images or fingerprints.

The simplest multimodal integration method is feature concatenation, putting together the features from all modalities into one high-dimensional vector. However this has the big disadvantage of weighting all modalities equally and not allowing for variations in their relative importance. This problem is solved in decision fusion by assigning weights to each of the modalities. This also allows the dynamic adjustment of the importance of each stream through the weights, according to its estimated reliability.

We are using a dynamic weighting scheme in which the weights are derived from the posterior entropies of each stream, and at each frame. The reason is that, in this way, the weights are allowed to vary quickly in a large range, thus being able to adapt to sudden changes in the quality of the streams.

In the following, we will present our method in detail. We begin by presenting the context of our work, and in particular the most common models used for speech recognition and AVSR, and also the most common fusion methods used in this application. We continue by presenting our method, the general algorithm used and the justification behind it. We then present our results with dynamic weights constrained to sum to one. We continue by justifying that the constant sum constraint is not necessary and show that consistently better results can be obtained by allowing the sum itself to be variable.

The content of this report is partially based on work that we have published in [1] and [2].

II. Multimodal fusion for AVSR

A. Hidden Markov models

Many different classifiers have been applied to the area of speech recognition, which is a difficult classification task due to the fact that the signals involved are time varying and of different temporal lengths.

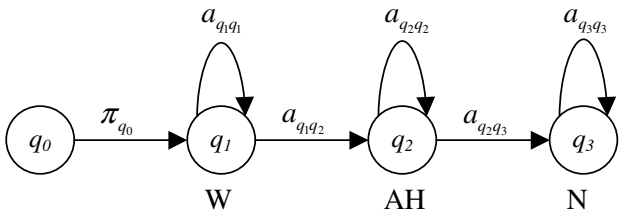

Fig. 1. A left-right HMM for the word "one".

The simplest classifier that can be used for isolated word recognition is template matching, where classification is based on the distance between the test word and stored temporal patterns representing the learned words. This technique was used for AVSR in [3]. However, the speaking rate needs to be more or less the same, otherwise their length would differ and the method would not work.

Dynamic Time Warping (DTW) [4] is a time normalization technique which aims to solve the problem of nonlinear temporal fluctuations caused by speech rate variation. DTW works best on small tasks and for isolated word recognition, and can also be applied to connected speech recognition, as shown in [5]. However, it was replaced by Hidden Markov Models (HMMs) which became the dominant classifier used in virtually all speech recognition systems.

We will give here a very brief overview of HMMs and their use in speech recognition. More detail can be found in [6] [7].

HMMs can model the behavior of systems which can switch between states stochastically. In a discrete, first order *Markov chain*, the probability of being in a particular state at a particular time depends only on the state itself and the previous state. In each of these states, the system emits a symbol which can be observed, the *observation*. In HMMs, the state itself is always hidden and only the observation is visible, thus the name *Hidden* Markov Models.

In speech recognition HMMs can be used to model the temporal evolution of the speech signal. In small vocabulary systems each word can be modeled by an HMM, while in large vocabulary tasks each speech sound has a corresponding HMM. Most recognition systems use left-right models, or Bakis models [8], in which states that have been left are never visited again and the system always progresses from an initial state towards a final one.

The purpose of the recognition process is to choose the most likely word model, given an observation sequence. The word attached to this model is the recognized word. To this end the Viterbi algorithm is employed [7].

Let us take as an example an isolated word speech recognizer, having left-right HMMs as word models. Such a word model is given in Figure 1. Let us define $a_{q_iq_j}$ as the transition probability from state q_i to state q_j. The initial transition probability is considered equal to one ($\pi_{q_0} = 1$). Let us also define the likelihood that observation O_t was emitted by state q_t as $b_{q_t}(O_t)$. The likelihood of the observation sequence $\mathbf{O} = O_1O_2 \cdots O_T$, given a path $\mathbf{Q} = q_1q_2 \cdots q_T$ in the model ω, is:

$$p(\mathbf{O}|\mathbf{Q}, \omega) = b_{q_1}(O_1)b_{q_2}(O_2) \cdots b_{q_T}(O_T), \tag{1}$$

assuming the statistical independence of the observations. The probability of the path itself is given by:

$$P(\mathbf{Q}|\omega) = \pi_{q_0}a_{q_1q_2}a_{q_2q_3} \cdots a_{q_{T-1}q_T}. \tag{2}$$

The joint probability of $\mathbf{O}$ and $\mathbf{Q}$ occurring simultaneously is the product of the two:

$$p(\mathbf{O}, \mathbf{Q}|\omega) = \prod_{q_i \in \mathbf{Q}} b_{q_i}(O_i) \cdot \pi_{q_0} \prod_{(q_i,q_j) \in \mathbf{Q}} a_{q_iq_j}. \tag{3}$$

The likelihood of the observation sequence given the model is the sum of these joint probabilities over all possible state sequences $\mathbf{Q}$:

$$p(\mathbf{O}|\omega) = \sum_{\text{all } \mathbf{Q}} p(\mathbf{O}, \mathbf{Q}|\omega)). \tag{4}$$

This "full" likelihood can be replaced by the "Viterbi" approximation, considering only the most likely path in the model:

$$p(\mathbf{O}|\omega) \simeq \max_{\mathbf{Q}}[p(\mathbf{O}, \mathbf{Q}|\omega)]. \tag{5}$$

This often-used approximation does not lead to a significant performance loss, while facilitating the numerical computation. In the end, the recognized word is given by the most likely word model:

$$\omega_{\text{recognized}} = \arg\max_{\omega}[p(\mathbf{O}|\omega)]. \tag{6}$$

The observation probabilities are modeled using Gaussian Mixture Models (GMMs) [9], in this way:

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm} N(o_t; \mu_{jm}, \Sigma_{jm}) \tag{7}$$

where $N(o; \mu, \Sigma)$ is the value in o of a multivariate gaussian with mean μ and covariance matrix Σ. M gaussians are used in a mixture, each weighed by c_{jm}. Typically, in speech recognition systems, the gaussians have a diagonal covariance matrix, as the coefficients used are assumed to be uncorrelated.

For the training phase of the HMMs, an iterative method called the Baum-Welch algorithm [10] is used. Depending on how well the environment conditions, that is, the acquisition equipment, the room acoustics and ambient noise, match between training and testing data, experiments can be performed in *matched* or *mismatched* conditions. The largest gain from AVSR comes especially in mismatched conditions.

Since the GMMs used to model speech units are not aimed to discriminate between them, a lot of attention has been given to the use of more discriminative models in conjunction with HMMs. In hybrid ANN-HMM systems [11], GMMs are replaced with Artificial Neural Networks (ANNs) [9]. Such models have also been applied to AVSR, as in [12].

Support Vector Machines (SVMs) [13] [14] have also been used as an alternative to GMMs, both in audio-only speech recognition [15] and in AVSR [16]. The main advantage of using discriminative models like ANNs and SVMs is that the scores of the speech units will be better separated and the distinction between these units will be clearer.

Although widely used for speech recognition, Hidden Markov Models have several inherent limitations with respect to their use for modeling speech [7]. The first one is the assumption that successive speech frames are independent, as seen in equation 1. To compensate (partially) for this shortcoming, first and second temporal derivatives can be added to the feature vectors, since they include information about the correlation between frames. A second limitation is the assumption that the probability distribution of observations can be well represented by a GMM, a limitation which is addressed by hybrid systems. Finally, the Markov assumption itself, that is, that only the previous state influences the choice of current state, is flawed, as temporal dependencies for speech can extend for several states.

B. The multimodal integration methods

The integration of audio and visual information [17] can be performed in several ways. The simplest one is *feature concatenation* [18], where the audio and video feature vectors are simply concatenated before being presented to the classifier. Here, a single classifier is trained with combined data from the two modalities.

Although the feature concatenation method of integration does in some cases lead to an improved performance, it is impossible to model the reliability of each modality, depending on the changing conditions in the audio-visual environment.

A second family of integration methods is *decision fusion*. In this method separate audio and video classifiers are trained, and their output log-likelihoods are linearly combined with appropriate weights. There are three possible levels for combining individual modality likelihoods [17]:

- Early integration, in the case when likelihoods are combined at the state level, forcing the synchrony of the two streams.
- Late integration, which requires two separate HMMs. The final recognized word is selected based on the n-best hypothesis of the audio and visual HMMs.
- Intermediate integration, which uses models that force synchrony at the phone or word boundaries.

In the following we present one of the most common integration methods, and the one we chose for our experiments, the Multi-Stream HMM. It belongs to the early integration category, forcing synchrony at the frame level. Our choice is justified by the fact that this type of integration allows very rapid changes in the importance given to each modality, allowing the implementation of systems which can very quickly adapt to changing conditions.

C. Multi-stream hidden Markov models

The Multi-Stream HMM (MSHMM) is a statistical model derived from the HMM and adapted for multimodal processing. Unlike typical HMMs which have one gaussian mixture (GMM) per state, the MSHMM has several GMMs per state, one for each input stream or modality.

The emission likelihood b_j for state j and observation o_t at time t is the product of likelihoods from each modality s weighted by stream exponents λ_s [19]:

$$b_j(o_t) = \prod_{s=1}^{S} \left[\sum_{m=1}^{M_s} c_{jsm} \mathcal{N}(o_{st}; \mu_{jsm}, \Sigma_{jsm}) \right]^{\lambda_s} \tag{8}$$

where $\mathcal{N}(o; \mu, \Sigma)$ is the value in o of a multivariate gaussian with mean μ and covariance matrix Σ. M_s gaussians are used in a mixture, each weighed by c_{jsm}. The product in eq. 8 is in fact equivalent to a weighted sum in logarithmic domain, and typically the weights are chosen in such a way that they sum to one. In practice, the weights λ_s should be tied to stream reliability, such that, when environment conditions (e.g. SNR) change, they can be adjusted to emphasize the most reliable modality.

The product seen here comes from the more general probability combination rules [20], and is one of the most widely used, along with the sum rule, the min rule or the max rule. These rules are compared in [21], with the purpose of combining the outputs of classifiers trained on different types of audio-only features. The product rule was found to be the best performer.

There are two possibilities to train MSHMMs. The first one is to start with normal one-stream HMMs, one per modality, and train each of them on single modality data. These models have to have the same number of states, but the parameters of the GMMs can differ between modalities. No weights are required while training. After the training stage, the models can be combined into a MSHMM. However, there are some drawbacks with this training strategy, one being that there is no guarantee that the models are trained on the same speech segments across modalities. Another drawback is that it is not clear how the transition probabilities should be combined.

The second method of training MSHMMs is to begin with a MSHMM and train it directly on multimodal data. In this way the synchronicity is guaranteed. However, weights are required in training, and a poor choice of training weights can lead to poor performance in testing.

At testing time, the weights given to each stream can be fixed for the whole duration of the test sequence, or can vary dynamically in time. In either case, the weights should be set according to the estimated reliability of each stream. In the following, we present some common stream reliability estimation methods.

D. Stream reliability estimates

The choice between weights which are constant in time and weights which vary dynamically, adapting to the conditions in the environment, is taken in the beginning based on the assumptions made on the context of the problem. If it is assumed that the acoustic and visual environment remains more or less the same, and that the training conditions reasonably match those in testing, fixed weights can be used. However, it is more realistic to assume that conditions will not be matched, and that they will not be constant in time. Indeed, sudden bursts of noise can happen anytime in the audio, like a cough, a pop from a microphone, the door of the conference room that is swung against the wall or any other similar situation can lead to a sudden degradation in the quality of the audio. Similarly, for the video, a lighting change, a speaker that turns his head or gestures with his hands making

his mouth invisible to the camera, all such situations can lead to sudden degradation in the quality of the video stream. These degradations can be temporary as in the case of the cough, or permanent, as the lighting change. In all these conditions, having the stream weights adapt automatically to the conditions which change in time should be beneficial for the performance of the system.

a) Fixed stream weights: can be derived with discriminative training techniques, applied on training or held-out data. They will only be relevant for the particular environment conditions in which that data was acquired. From the methods that are applied directly on the training data, some minimize a smooth function of the word classification error [22], [23]. Another approach is to minimize the frame classification error, as in [24] where the maximum entropy criterion is used. From the methods that use a held-out data set, the simplest is the grid search, when the weights are constant and constrained to sum to 1, as is the case in [24] or [25]. More complex methods need to be employed in other cases, for example when weights are class-dependent, however, this dependency was not proved to have a beneficial effect on recognition results, as shown in [24] or [26].

Yet another approach is to use a small amount of unlabeled data, as in [27], to estimate the stream weights in an unsupervised way. Class specific models and anti-models are first built, and then used to initialize a k-means algorithm on the unlabeled data. The stream weights ratio is then expressed as a non-linear function of intra- and inter-class distances.

In [28], the testing set itself is used in an iterative likelihood-ratio maximization algorithm do determine the stream weights. The algorithm finds the weights that maximize the dispersion of stream emission likelihoods $P(o|c)$, which should lead to better classification results. The measure to be maximized is:

$$L(\lambda_c^A) = \sum_{t=1}^{T} \sum_{c \in C} \Big\{ P(o_t^A|c_t) - P(o_t^A|c) \Big\} \tag{9}$$

where c is the class or HMM state out of the set C of classes, and o^A is the audio observation vector. The measure is computed over a time interval of T frames.

An extension of this algorithm is based on output likelihood normalization [29]. Here, the weights are class-dependent, and the weight for one class is the ratio between the average class-likelihood for a time period T and the average likelihood for that particular class over the same time period, that is:

$$l_{vt}^A = \frac{\frac{1}{NT} \sum_{t=1}^{T} \sum_{c \in C} \log P(o_t^A|c)}{\frac{1}{T} \sum_{t=1}^{T} \log P(o_t^A|v)} \tag{10}$$

Both these methods optimize the audio weights first, and then set the video weights relative to the audio ones.

Stream reliability estimates are however not limited to the AVSR field. For example, in [30], reliability measures are used for the fusion of three streams for multimodal biometric identification. The three streams are audio speech, visual speech and the speaker's face, while the reliability measure used is the difference between the two highest ranked scores, normalized by the mean score.

b) Dynamical stream weights: are however better suited for practical systems, where the noise can vary unexpectedly. Such examples of sudden degradation of one modality can be loud noises in the audio, or loss of face/mouth tracking in the video. Events like these can happen in a practical setup and they prove the need for temporally-varying stream weights. The weights can be adjusted for example based on the estimated signal to noise ratio (SNR), as in [18] [31] [32] [33] [34] [35] [36], or based on the voicing index [37] used in [38]. However these methods are based on reliability measures on the audio only, and the video degradation is not taken into account. Other weighting methods are modality-independent, based only on indicators of classifier confidence, as presented in the following.

In [39] and [40], several classifier confidence measures are used. The first one is the N-best log-likelihood difference, based on the stream emission likelihoods $P(o|c)$. The measure is defined as follows. If o_{st} is the observation for stream s at time t and c_{stn} are the N-best most likely generative classes (HMM states), $n = 1...N$, then the log-likelihood difference at time t for stream s is:

$$l_{st} = \frac{1}{N-1} \sum_{n=2}^{N} \log \frac{P(o_{st}|c_{st1})}{P(o_{st}|c_{stn})} \tag{11}$$

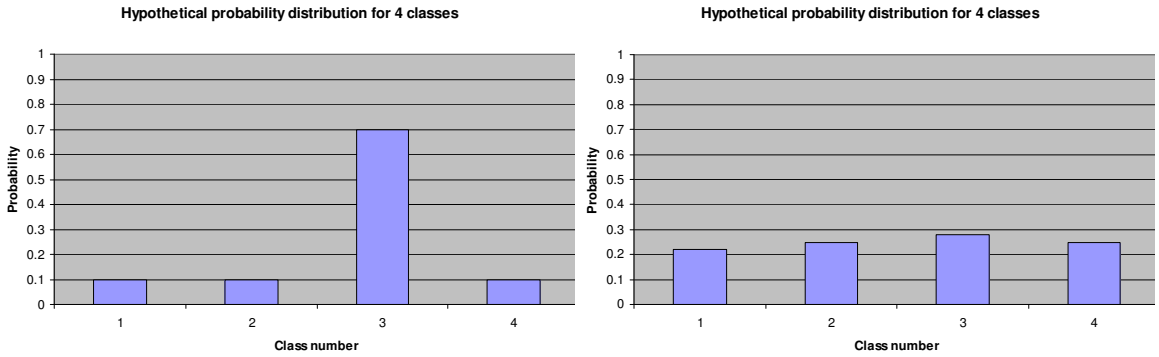

Fig. 2. Two hypothetical four-class posterior distributions. The one from the left has a very distinct peak, so the confidence that we can have in the result of the classification is high, while the one to the right is flat, leading to a low confidence in the classification result.

The justification is that the likelihood ratios give a measure of the reliability of the classification. The same justification can be given for the second measure used, the log-likelihood dispersion, defined as:

$$d_{st} = \frac{2}{N(N-1)} \sum_{m=1}^{N} \sum_{n=m+1}^{N} \log \frac{P(o_{st}|c_{stm})}{P(o_{st}|c_{stn})} \tag{12}$$

Other methods for stream confidence estimation are based on posteriors, not on likelihoods. In [41], the class-posterior probability of the combined stream $P(c|o^{AV})$ is computed as the maximum between three posteriors, derived from three observation vectors, the audio-only one o_t^A, the video-only o_t^V or the concatenated observation o_t^{AV}, that is:

$$P(c_{tn}|o_t) = \max(P(c_{tn}|o_t^A), P(c_{tn}|o_t^V), P(c_{tn}|o_t^{AV})) \tag{13}$$

The stream reliability estimation framework is not only applicable on AVSR, but also in audio-only speech recognition, in the case when multiple feature streams are used in order to exploit their complementarity. For example, in [42] the entropy of the class-posterior distribution is used as a reliability estimator:

$$h_{st} = -\sum_{i=1}^{C} P(c_i|o_{st}) \log P(c_i|o_{st}) \tag{14}$$

where C is the number of classes. The entropy is also a measure of dispersion, but this time used on all the classes, not only the N-best ones.

III. Our stream reliability estimation method

As mentioned before, we are using multi-stream HMMs to recognize words. These recognizers require synchronous audio and visual feature streams, and perform multimodal fusion at the frame level, that is, a decision on the fusion is taken every 10 ms. The emission likelihood b_j for state j and observation o_t at time t is the product of likelihoods from each modality s weighted by stream exponents λ_s [19]:

$$b_j(o_t) = \prod_{s=1}^{S} \left[\sum_{m=1}^{M_s} c_{jsm} N(o_{st}; \mu_{jsm}, \Sigma_{jsm}) \right]^{\lambda_s} \tag{15}$$

The stream weights λ_s are computed based on the estimated stream reliability, which is derived from the entropy of the class-posterior distributions for each stream.

The reasoning is as follows. Consider the case in Figure 2, where a simple hypothetical situation is presented. Assume that, in a multimodal 4-class problem, the posterior distribution has a very clear peak, as in the left figure. This means that there is a very good match between the test sample and the class model for the recognized class, and a very bad match with all the other classes. The confidence that we have in assigning the sample to the class is high, meaning that the confidence in the corresponding stream should also be high. On the other hand, when the

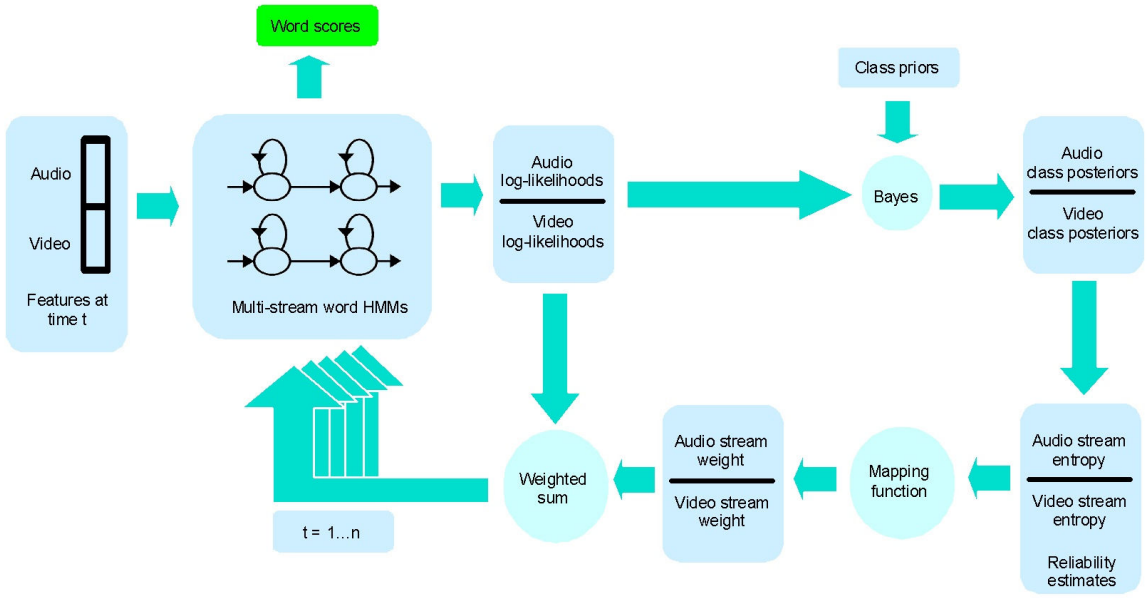

Fig. 3. The steps in the dynamic weights multimodal integration algorithm.

posterior distribution is flat or nearly flat, like in the right figure, there is a high possibility that the sample was assigned to the wrong class, so the confidence is low, both in the classification result and the stream. To measure if the distribution is peaky or flat, we use the entropy:

$$h_{st} = -\sum_{i=1}^{C} P(c_i|o_{st}) \log P(c_i|o_{st}) \tag{16}$$

The steps of the algorithm are presented in Figure 3. First, the audio and visual log-likelihoods are obtained for each class, that is, for each gaussian mixture in each of the states of the HMMs. Then, using Bayes' formula, we obtain the posteriors:

$$P(c_i|o_{st}) = \frac{P(o_{st}|c_i)P(c_i)}{\sum_j P(o_{st}|c_j)P(c_j)} \tag{17}$$

Here, the class priors $P(c_i)$ are the relative durations of the classes, obtained from the training set. The entropies of both distributions are then computed, and finally the weights are obtained through a mapping function.

The mapping function is required since the weights are the inverse of the entropies, and scaling also needs to be applied. Indeed, since a high entropy signifies a low confidence in the corresponding stream, a low weight should be assigned to it. The mapping functions we use will be detailed in the next section.

The big advantage of this algorithm is its flexibility in different environments. If one of the modalities becomes corrupted by noise, the posterior entropy corresponding to it would increase, making its weight, and so its importance to the final recognition, decrease. This is also valid in the case of a complete interruption of one stream. In this case, the entropy should be close to maximum, and the weight assigned to the missing stream close to zero. This practically makes the system revert to one-stream recognition automatically. This process is instantaneous, and also reversible, that is, if the missing stream is restored, the entropy would decrease and the weight would increase to the level before the interruption.

Even in the case of static noise, the relative importance that should be given to each of the modalities may vary. Some speech sounds are easier to distinguish in the visual modality, while others are more distinguishable in the audio. This means that the posterior distribution peaks corresponding to these speech sounds will be more pronounced in one modality than the other, leading to a reduced entropy and a higher weight. Thus, in theory, our algorithm automatically favors the modality in which the sound is easier to distinguish.

All this means that the system can dynamically adapt to all levels of noise, including even the loss of one of the streams.

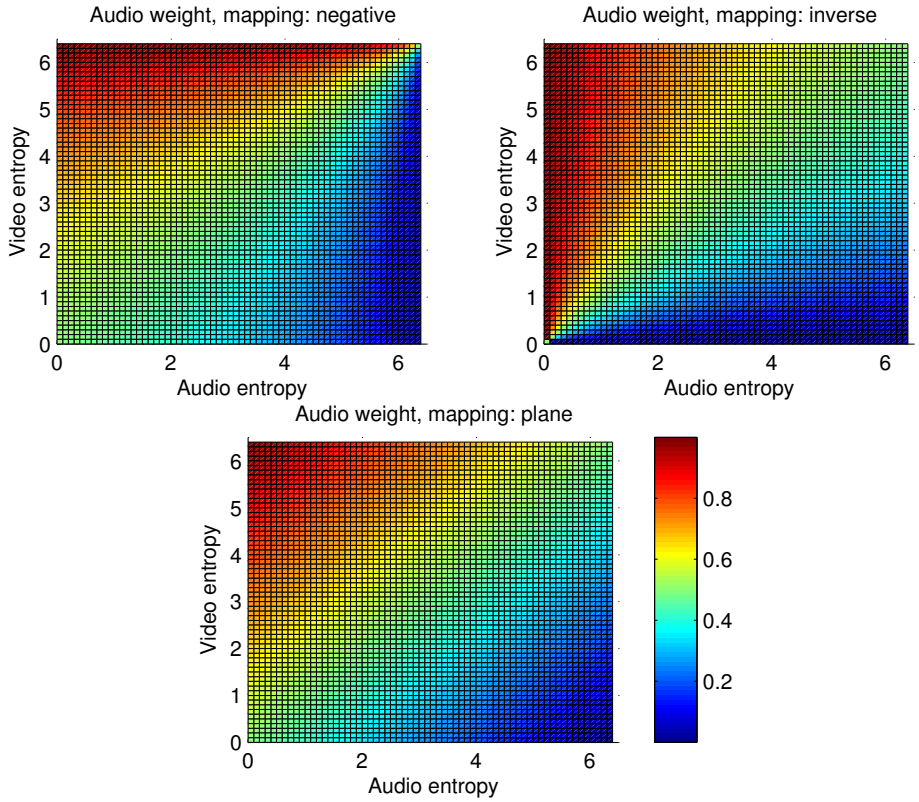

Fig. 4. Three possible mappings from posterior distribution entropy to stream weight.

IV. Choosing a mapping function from entropies to weights

A. Three static mapping functions

We have established that the entropies of the posterior distributions are suitable reliability estimators for the audio and visual streams. What is still needed is a function to transform these estimators into stream weights. This function is required since the weights and the reliability measure are not on the same scale, and their relation is non-linear.

Other approaches use training on a held-out subset of the training dataset, but we try to avoid this, as we consider that the type and intensity of noise in testing conditions is uncertain. This means that having a held-out set for weights training that matches the target noise conditions is very unlikely.

The approach that we take is to find a mapping that satisfies a few basic conditions. In our case, the relationship between the entropies and the weights is inverse, that is, when the entropy is low, the weight should be high, and vice-versa. We also impose for the moment that the sum of the weights be 1, that is $\lambda_a + \lambda_v = 1$.

Let the audio and video streams' entropies be H_a and H_v, and their associated weights λ_a and λ_v. The maximum value that the entropy can reach in our case is $H_{max} = log_2 83 \simeq 6.3$ since we have 83 classes. The mapping should ensure that when $H = 0$, $\lambda = 1$ and when $H = H_{max}$, $\lambda = 0$. Obviously, when $H_a = H_v$, $\lambda_a = \lambda_v = 0.5$.

There are several possible mappings that can be used. Two possibilities are presented below:

$$\lambda_a(t) = \frac{H_{max} - H_a(t)}{\sum_s H_{max} - H_s(t)} \tag{18}$$

$$\lambda_a(t) = \frac{1/H_a(t)}{\sum_s 1/H_s(t)} \tag{19}$$

We will refer to equations 18 and 19 as the "negative entropy" and "inverse entropy" mappings. The first one uses the difference to reverse the entropy, while the second uses the inverse. The difference between them is that the inverse mapping has a bias towards low values of the entropy, while the negative mapping has a bias towards high values.

The two mappings have a common shortcoming: if one of the entropy values is close to an extreme (either zero or H_{max}), a variation in the other entropy's value will have no effect. This can be seen in figure 4. For example, for the negative mapping, when the video entropy is close to H_{max}, the audio weight will be close to 1, irrespective of the value of the audio entropy. The preferable behavior would be for the audio weight to vary with the audio entropy, for example, when the audio entropy is also high, the weight to be close to 0.5. The inverse mapping has the same problem, since, when the audio entropy is close to zero, the audio weight is 1, even if the video entropy is also close to zero.

To avoid these problems, we derived a third mapping, which represents a plane in 3D space. The plane was derived using the following four points:

- $H_a = H_v = 0 \rightarrow \lambda_a = \lambda_v = 0.5$
- $H_a = 0; H_v = H_{max} \rightarrow \lambda_a = 1; \lambda_v = 0$
- $H_a = H_{max}; H_v = 0 \rightarrow \lambda_a = 0; \lambda_v = 1$
- $H_a = H_v = H_{max} \rightarrow \lambda_a = \lambda_v = 0.5$

The resulting equation is:

$$\lambda_a(t) = \frac{H_v - H_a}{2H_{max}} + \frac{1}{2} \tag{20}$$

As can be seen from the figure, using this mapping, the audio weight always varies with both entropies.

In the next section, we present our results with these three mappings.

B. Results with the static mappings

For our experiments, we use sequences from the CUAVE audio-visual database [43]. They consist of 36 speakers repeating the 10 digits. We use only the static part of the database, that is, the first 5 repetitions.

The video sequences are filmed at 30 fps interlaced, so we can effectively double this framerate through deinterlacing. The average length of one video sequence is around 50 seconds (3000 deinterlaced frames).

Out of the 36 sequences, 30 are used for training, and 6 for testing. We use a six-fold crossvalidation procedure, that is, we repeat training and testing 6 times, each time changing the respective sets using a circular permutation. The performance reported is the average on the 6 runs.

We start our visual feature processing by locating the region of the mouth, scaling and rotating it, such that all the mouths have more or less the same size and position. The temporal resolution of the video is then increased through interpolation, to reach 100 fps, since synchrony between the audio and the video streams is required by our integration method.

The visual features that we use are even-frequency discrete cosine transform (DCT) coefficients of the mouth images, since they contain the information related to the symmetrical details of the image, as detailed in [44]. From them, the highest-energy 64 coefficients are selected, with their first and second temporal derivatives, and LDA is applied on them, to obtain a 40-dimensional feature vector.

On the audio side, the features extracted are 13 Mel Frequency Cepstral Coefficients (MFCCs), together with their first and second temporal derivatives. Audio features are extracted 100 times per second, at the same frequency as the visual features. Different levels of white gaussian noise are added in order to show how our dynamic weighting algorithm performs across a large range of SNRs.

We use the HTK library [19] for the HMM implementation. Our word models have 8 states with one diagonal-covariance gaussian per state. The silence model has 3 states with 3 gaussians per state. Two streams are used, audio and video. The grammar consists of any combination of digits with silence in-between. The accuracy that we report is the number of correctly recognized words minus insertions, divided by the total number of test words.

We compare our method with the Maximum Stream Posterior (MSP) method described in [45] [41]. The method is based on the premise that a stream weighting algorithm should outperform when possible either of the two streams when they are reliable, and perform at the same level as one of the stream when the other one is corrupted. The class-posterior probabilities for each stream are used, $P(s|o_a)$ and $P(s|o_v)$, where s is the state and o the observation vector. They are computed from the likelihoods with Bayes' formula. The combined stream class-posterior probability is also used, $P(s|o_{AV})$, derived as follows:

$$P(s|o_{AV}) = \frac{p(o_a|s)p(o_v|s)P(s)}{\sum_{s'} p(o_a|s')p(o_v|s')P(s')} \tag{21}$$

where $p(o_a|s)$ and $p(o_v|s)$ are the likelihoods. This would be equivalent to using the weights $\lambda_a = \lambda_v = 1.0$ in our case. The MSP method consists of using either the combined stream, or one of the monomodal streams, in order to maximize the stream posterior at each frame, as follows:

$$P(s|o) = \max\left[P(c|o_a), P(c|o_v), P(c|o_{AV})\right] \tag{22}$$

Figure 5 shows audio-visual results for dynamic weighting with the three mapping functions. As can be seen from the figure, the negative and plane mapping are practically equivalent, while the inverse mapping is performing slightly worse. For the first two SNR levels, clean and 25 dB, the performance is equivalent to audio-only recognition, but for lower SNRs there is an ever-increasing gain from audio-visual recognition.

The reason for the fact that the negative and plane mappings lead to the same results may be that the entropy values are not covering the whole space between $(0.0, 0.0)$ and (H_{max}, H_{max}). Indeed, in Figure 4, if splitting the images along the second diagonal, between the points $(0.0, H_{max})$ and $(H_{max}, 0.0)$, the lower halves for the negative and the plane mapping images are very similar. If most entropy values are confined in that interval, the results should also be similar indeed.

Figure 6 compares our dynamic weights method with negative mapping with the MSP method and the fixed weights method. The fixed weights methods consists in searching for the optimal pair of weights for a given SNR and using it for the entire sequence, with no time variation. This method is not applicable in practice, as typically the test SNR is not known at training time and is also not necessarily constant. The fixed weights results are presented here as a measure of how well multimodal integration can perform with ideal weights.

The MSP method performs worse than our method at most SNR levels, with the highest difference at 10 dB, 2.2%. The reason for this poor performance may be the fact that the MSP method takes extreme decisions, either considering a 50% - 50% combination of the two streams, or ignoring one of the streams entirely. Our method, by contrast, is more flexible, allowing any weighted combination of scores from the two streams, at any moment.

Both methods perform worse than the fixed weights for clean audio and high SNRs, as the entropy of the clean audio does not become low enough compared to the video to allow the high difference in weight values which leads to high performance for these SNRs.

In fact, the entropies for both audio and video do not cover the entire range from 0.0 to H_{max}. It may be better to use a mapping which is more "sensitive" in the parts of this range where there actually are a lot of audio and video entropy values. The next section investigates such a mapping.

C. A dynamic mapping

The mappings presented earlier cover uniformly the entire space between 0 and H_{max}. If, hypothetically, the entropy values are concentrated in only a small region of this space, the variations in the weights values would be very small. An ideal mapping would, by contrast, concentrate its discriminative power only in that region, making small variations in entropy lead to large variations in the stream weights. If the entropy values of the two streams are consistently close, this type of mapping would strongly favor the stream which has an even slightly higher entropy.

Intuitively, this mapping should be more sensitive for some entropy value intervals compared to others, and those "sensitive" intervals should be the ones that include the entropy values that occur most often. This intuition lead to the following method of constructing a entropy to weights mapping.

First, a histogram of past entropy values is built for both streams. In our case, the histogram has 15 bins and comprises 150 past entropy values from both streams, for a total of 300 samples. Then, a piecewise-linear function is built, mapping low entropy values to high weights and vice-versa. This is done in such a way that the slope of each piece is proportional to the number of points contained in the corresponding histogram bin. Figure 4 shows an example of such a mapping and the histogram from which it was built.

This mapping is dynamic itself. It adapts to the particular configuration of entropy values, with the purpose of having the best discriminating power between the most occurring ones. As can be seen from the figure, the mapping is flat for the intervals where the number of frames with corresponding entropies is low, and steep where the number of frames is high. Its shape changes all the time, according to the particular distribution of entropy values.

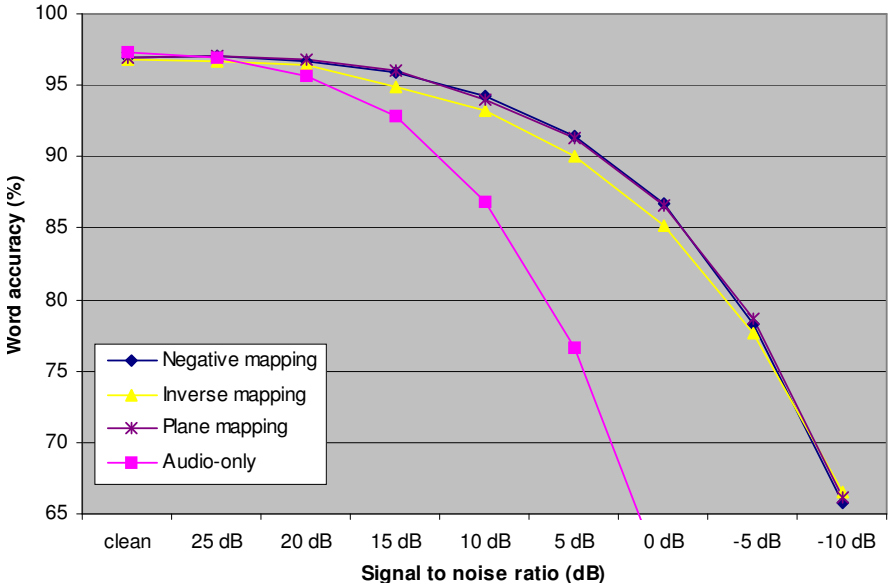

Fig. 5. Audio-visual results with dynamic weights and the three mapping functions, inverse, negative and plane, for all SNRs, with white noise.

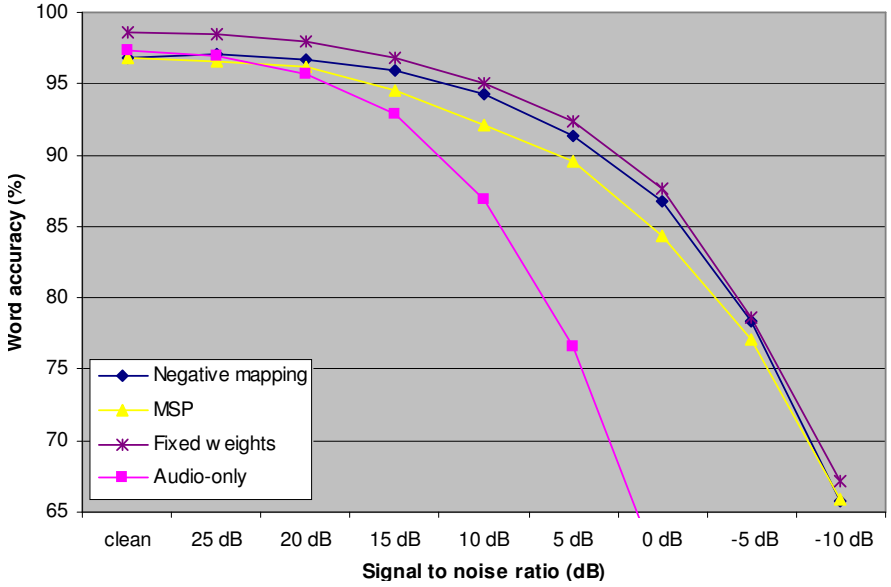

Fig. 6. Results for audio-visual recognition with dynamic weights, fixed weights and the MSP method, for all SNRs, with white noise.

	SNR								
	clean	25 dB	20 dB	15 dB	10 dB	5 dB	0 dB	-5 dB	-10 dB
negative	96.88	97.10	96.65	95.93	94.30	91.39	86.71	78.33	65.82
inverse	96.82	96.65	96.48	94.91	93.18	90.10	85.20	77.70	66.55
plane	96.99	97.10	96.82	95.98	94.02	91.28	86.54	78.61	66.15
MSP	96.76	96.54	96.15	94.47	92.12	89.61	84.41	77.11	65.86
fixed	98.66	98.44	97.93	96.81	95.02	92.34	87.65	78.60	67.17
AO	97.31	96.97	95.61	92.87	86.88	76.64	60.38	40.35	19.55

TABLE I

AUDIO-VISUAL RESULTS FOR DYNAMIC WEIGHTS, FIXED WEIGHTS, AND THE MSP METHOD, FOR ALL SNRS, WITH WHITE NOISE.

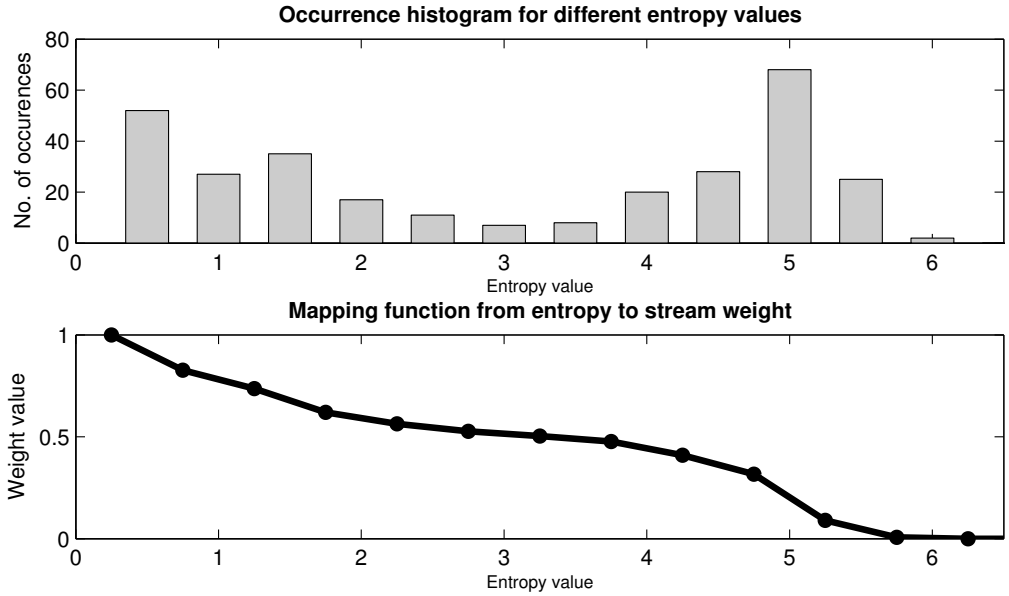

Fig. 7. A flexible mapping function from entropy to weight.

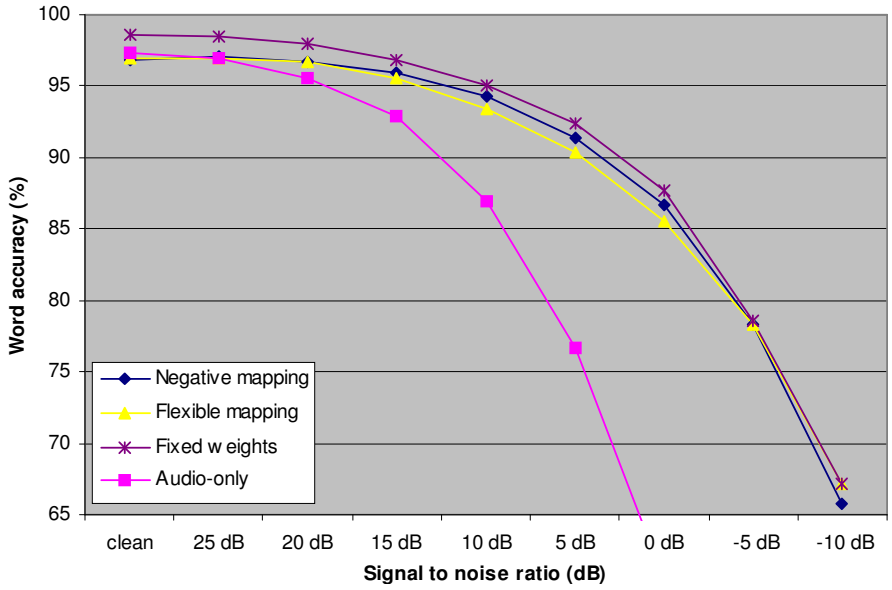

Fig. 8. Audio-visual results with dynamic weights and a flexible mapping, for all SNRs, with white noise.

	SNR								
	clean	25 dB	20 dB	15 dB	10 dB	5 dB	0 dB	-5 dB	-10 dB
negative	96.88	97.10	96.65	95.93	94.30	91.39	86.71	78.33	65.82
flexible	96.93	96.93	96.71	95.53	93.40	90.37	85.53	78.37	67.21
fixed	98.66	98.44	97.93	96.81	95.02	92.34	87.65	78.60	67.17
AO	97.31	96.97	95.61	92.87	86.88	76.64	60.38	40.35	19.55

TABLE II
RESULTS FOR AUDIO-VISUAL RECOGNITION WITH DYNAMIC WEIGHTS AND A FLEXIBLE MAPPING, FOR ALL SNRS, WITH WHITE NOISE.

Figure 8 and Table II show our results obtained with this mapping. As can be seen from the figure, the results are rather disappointing. The flexible mapping performs identical to the negative mapping for high SNRs and slightly worse for lower ones. A notable exception is the -10 dB level, where this mapping performs better than all the

others. However, overall the flexible mapping does not bring any improvement to performance compared to the previously presented methods.

V. The weight sum

A. The role of the weight sum

As mentioned before, most approaches in the literature impose that the sum of the weights should be equal to one. The reason might be to keep the result in the same range as the original likelihoods. However, the score that is computed by combining the likelihoods is not a likelihood (or emission probability) anymore. As shown before, in logarithmic domain the likelihoods are combined as follows:

$$\log b_j(o_t) = \sum_{s=1}^{S} \left[\lambda_s \cdot \log \sum_{m=1}^{M_s} c_{jsm} \mathcal{N}(o_{st}; \mu_{jsm}, \Sigma_{jsm}) \right] \tag{23}$$

For our particular case, with two streams, audio and video, the equation becomes:

$$\log b_j(o_{AV}) = \lambda_a \log b_j(o_a) + \lambda_v \log b_j(o_v) \tag{24}$$

Indeed, there are no guarantees that the combined score would still integrate to 1 over the value range of the features, as the mono-modal likelihoods do, with or without the constraint $\lambda_a + \lambda_v = 1$. So, in fact, the constraint is not required.

To explain the effect that a different weights sum could have on the decoding, we recall the expression of the score of an observation sequence O and a path Q through a model ω, with respect to the emission likelihoods b_{q_i} and the transition probabilities $a_{q_i q_j}$.

$$\log p(O, Q|\omega) = \sum_{q_i \in Q} \log b_{q_i}(o_i) + \sum_{(q_i, q_j) \in Q} \log a_{q_i q_j} \tag{25}$$

which, when replacing the combined score with its expression from Eq. 24, becomes:

$$\log p(O, Q|\omega) = \lambda_a \sum_{q_i \in Q} \log b_{q_i}(o_i^A) + \lambda_v \sum_{q_i \in Q} \log b_{q_i}(o_i^V) + \sum_{(q_i, q_j) \in Q} \log a_{q_i q_j} \tag{26}$$

In a typical speech recognition system, the emission likelihoods b_{q_i} tend to become very small, making log-likelihoods large in absolute terms. Obviously, all log-likelihoods and log-probabilities are negative. In fact, the reason for using the logarithm in the first place is to prevent underflow errors that might be caused when multiplying likelihood values. By contrast, the transition probabilities $a_{q_i q_j}$ are larger than the emission likelihoods, and thus, have a much smaller absolute value in the logarithmic domain. This makes the emission likelihoods have a larger influence on the recognition results compared to transition probabilities. Indeed, if the difference in the range of variation is large, the path through the HMM and in consequence the score of the corresponding word could be influenced only by the emissions, with transitions having only the effect of allowing or not allowing certain paths in the model.

In our AVSR setup, the weights λ_a and λ_v can also be a factor in the balance between emission and transition probabilities, as can be seen from Eq. 26. Indeed, if the sum $\lambda_a + \lambda_v$ is allowed to change, this balance will also change. If, for example, the sum is reduced, the effect of the transition probabilities would be increased.

B. Results with an unconstrained sum

Figures 9, 10 and 11 show the audio-visual performance with audio and visual weights varying from 0.0 to 1.0 with a step of 0.02, without any constraint on their sum. The audio SNRs are as follows: clean, 20 dB, 10 dB, 0 dB, -5 dB and -10 dB, with babble noise. The purple line drawn on all the figures shows the weights for which the sum is equal to 1, $\lambda_a + \lambda_v = 1$, a constraint common to most approaches in the literature. As can easily be seen on all the figures, the optimal performance is always attained for sums which are significantly smaller, that is, the maximum is never on the purple line. In all figures, the lower value on the color bar is also the lower threshold applied on the accuracy values. This thresholding is applied for visualization purposes, as much lower accuracy

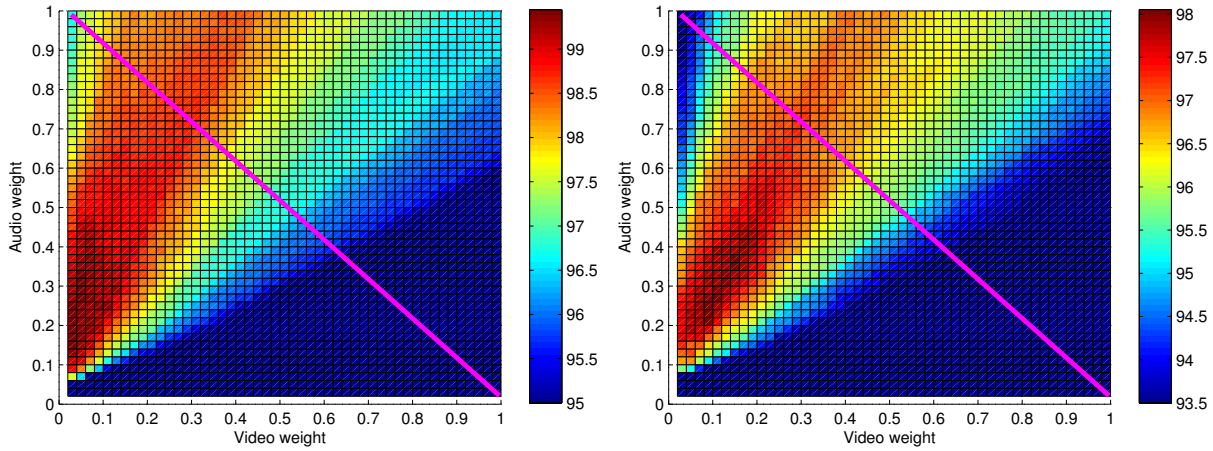

Fig. 9. AV accuracy, with unconstrained weights, for clean and 20 dB.

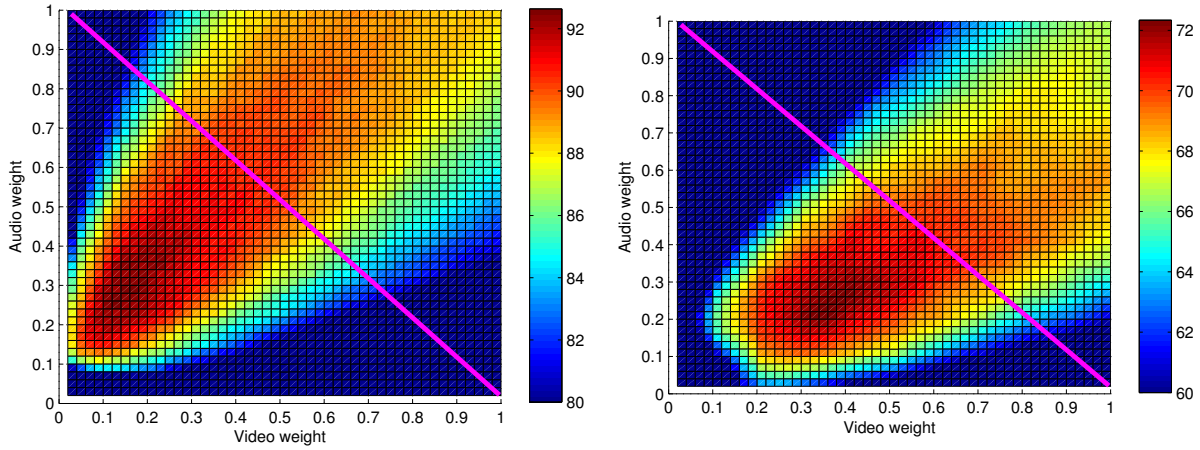

Fig. 10. AV accuracy, with unconstrained weights, for 10 dB and 0 dB.

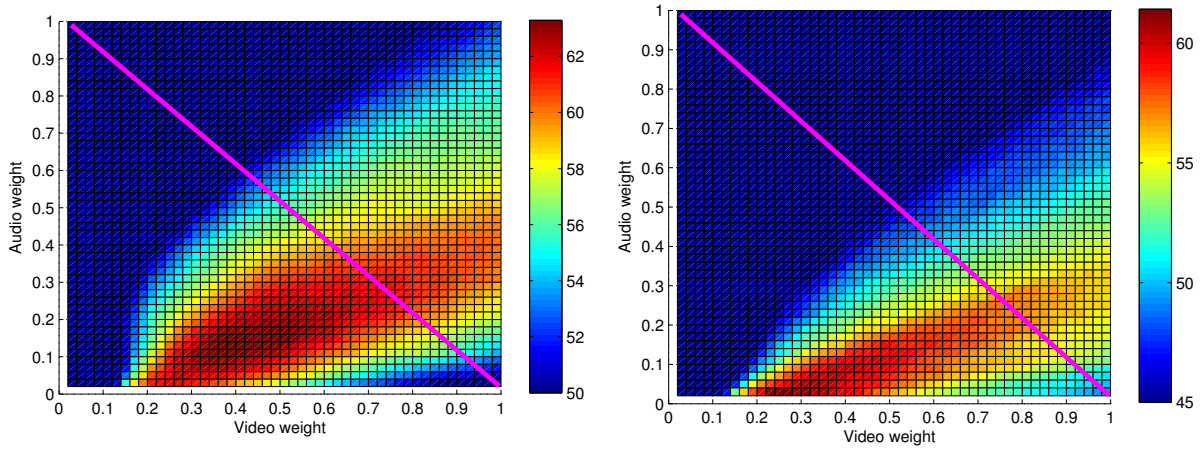

Fig. 11. AV accuracy, with unconstrained weights, for -5 dB and -10 dB.

values are also obtained for some combinations of weights, which lead to a widening of the range which has to be covered by the color map, making the peak indistinguishable in the absence of the threshold.

For clean audio, the value of the video weight at the point where maximum accuracy is very small, $\lambda_v = 0.02$, while the audio weight is $\lambda_a = 0.14$. However, the video still has a significant influence on the final result. Indeed, the AV accuracy is 99.45%, while audio-only performance is just 97.31%.

The same is seen at the other extreme, for an SNR of -10 dB. Here the audio weight is extremely small, $\lambda_a = 0.01$ compared to a video weight of $\lambda_v = 0.24$. As seen with clean audio, such a small weight can still have a big influence on the result. In this case, AV performance is 61.93%, while video-only is just 54.8%.

Figure 12 and Table III show the results obtained with fixed weights when using all combinations of weights, not only the ones summing to 1, at all SNRs with babble noise. The results for $sum \neq 1$ are practically the peak performance values from Figures 9, 10 and 11. It is clear from the figure that sums that are smaller than 1 lead to improved results at all SNRs. Although the gains are small, they are consistent across all noise levels. While for clean audio only 0.8% are gained, the number of errors is reduced by more then half. This is also true at the 25 dB SNR level. The relative reductions in error levels, shown in Table III, are more modest for lower SNRs, but the trend is consistent at all noise levels. Allowing the sum to vary improves recognition results.

The justification for these accuracy improvements should be the factor mentioned above, the balance between transition probabilities and emission likelihoods. Indeed, for sums which are smaller than 1, the influence of the transitions is enlarged, as seen in Eq. 26.

The balance between audio and video is given in this case by the ratio between the audio and the video weights, not their absolute values. Indeed, if we normalize the weights by dividing them to their sum value, we obtain an image which is very similar to that obtained with constrained sum. Figure 13 shows the values of the audio and the video weight, divided by their sum. The audio normalized weight decreases with the SNR, from 0.88 to 0.04, while the video normalized weight increases from 0.12 to 0.96.

C. Adapting the weight sum dynamically

Since a smaller weight sum is beneficial to the recognition results, it might be a good idea to allow the sum itself to vary dynamically. This would lead to a system in which the ratio of the weights is estimated from the individual reliability of each stream, while the sum is estimated globally. But how would a dynamical sum be able to influence recognition results?

In a situation where the noise is variable, there might be instances where both modalities are corrupted simultaneously. In such cases, both emission likelihood distributions may be unreliable, and the only source of reliable information left would be the transition probabilities. In such a case, it may be convenient to reduce the weight sum in such a way that decoding continues based mostly on the transitions, that is, the states in the maximum-likelihood path through the HMM are chosen on the basis of the most likely transitions. This may be better than allowing likelihoods from corrupted modalities to influence the result.

Even when the noise is constant, in instances when the two modalities are contradicting each other, it might be better to ignore the emission likelihoods for a few frames and just continue the decoding based on the transitions. A contradiction between audio and video would be for example if one modality has a peak in the posterior distribution for one phoneme, while the peak in the other modality corresponds to a different phoneme.

To estimate when a reduction in the value of the weight sum may be necessary, we need to detect the instances when both streams are unreliable, or when they are contradicting. For this purpose, we use the entropy of class posteriors again, but on the combined likelihoods. We compute the combined class-posteriors with a formula similar to Eq. 21, that is:

$$P(s|o_{AV}) = \frac{p(o_a|s)^{\lambda_a} p(o_v|s)^{\lambda_v} P(s)}{\sum_{s'} p(o_a|s')^{\lambda_a} p(o_v|s')^{\lambda_v} P(s')} \tag{27}$$

where the weights λ_a and λ_v are computed as before, based on the stream posterior entropies, and with $\lambda_a + \lambda_v = 1$. The entropy of this combined posterior distribution is then computed:

$$h_{av}^t = -\sum_{i=1}^{S} P(s_i|o_{av}^t) \log P(s_i|o_{av}^t) \tag{28}$$

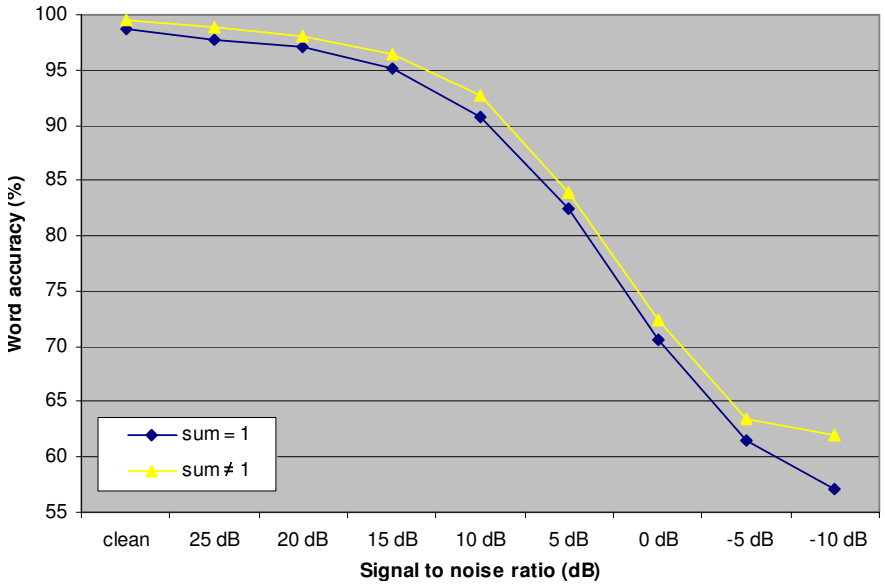

Fig. 12. The impact of removing the constraint on the sum of the weights on audio-visual results, with fixed weights.

	SNR clean	25 dB	20 dB	15 dB	10 dB	5 dB	0 dB	-5 dB	-10 dB
sum=1	98.66	97.77	97.11	95.15	90.74	82.41	70.63	61.53	57.18
sum≠1	99.45	98.89	98.05	96.37	92.64	83.87	72.37	63.42	61.93
gain	+0.79	+1.12	+0.95	+1.22	+1.9	+1.45	+1.74	+1.9	+4.76
% error reduction	58.88	50.26	32.83	25.13	20.52	8.24	5.92	4.94	11.12

TABLE III
RESULTS FOR AUDIO-VISUAL RECOGNITION, WITH FIXED WEIGHTS AND AN UNCONSTRAINED SUM.

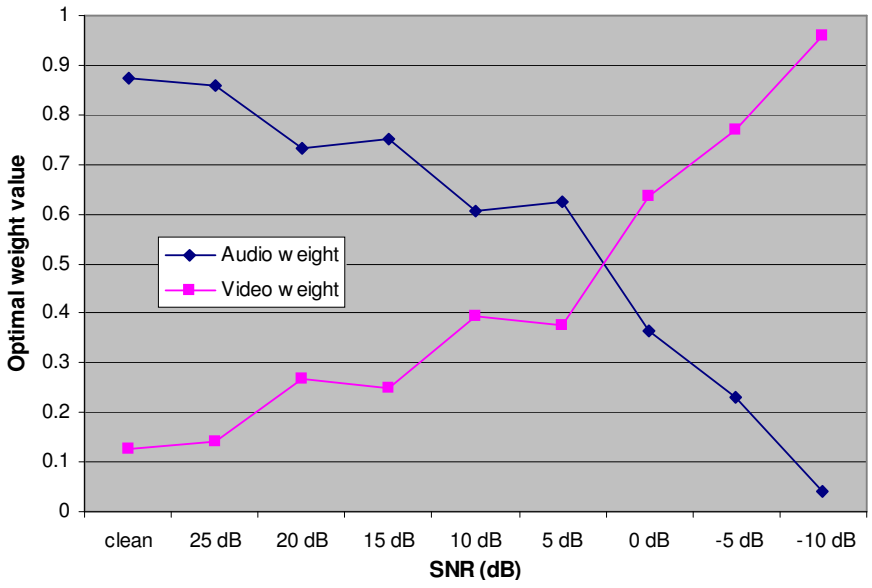

Fig. 13. The values of audio and video weights, normalized by the value of their sum, for all SNRs.

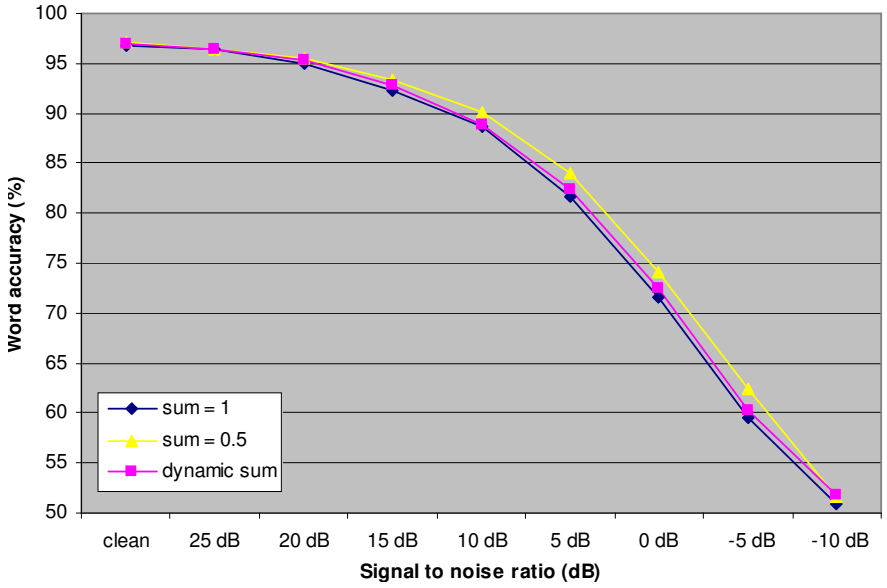

Fig. 14. The impact of removing the constraint on the sum of the weights on audio-visual results, with dynamic weights, compared to the sum fixed to 1 and 0.5.

	SNR								
	clean	25 dB	20 dB	15 dB	10 dB	5 dB	0 dB	-5 dB	-10 dB
sum=1	96.82	96.32	95.04	92.35	88.62	81.64	71.62	59.55	50.81
sum=0.5	97.10	96.49	95.59	93.41	90.07	83.92	74.03	62.36	51.65
dynamic sum	96.93	96.32	95.26	92.74	88.90	82.31	72.46	60.18	51.76

TABLE IV
THE IMPACT OF REMOVING THE CONSTRAINT ON THE SUM OF THE WEIGHTS ON AUDIO-VISUAL RESULTS, WITH DYNAMIC WEIGHTS, COMPARED TO THE SUM FIXED TO 1 AND 0.5.

This entropy will be low in the case when the modalities are in agreement, that is, if there are definite peaks in both the audio and the video posterior distributions, and the peaks coincide. When the peaks do not coincide however, the combined probability distribution $P(s|o_{AV})$ will be flatter than the two mono-modal distributions, leading to a higher entropy. The entropy will also be high when both the audio and the video posterior distributions are flat themselves, because this also leads to a flat combined distribution.

However, in the case when one posterior distribution for one modality is flat, while the other has a definite peak, the weights λ_a and λ_v will be heavily biased toward the reliable modality, making the combined posterior look very similar to its posterior distribution. This means that when one modality is reliable but the other is not, the combined entropy will also be low.

This all shows that the combined entropy should be a good measure for the purpose mentioned above, detecting when both modalities are unreliable or when they are contradicting. The inverse of this entropy can be mapped to a an adjustment factor which will be applied afterwards on both weights, effectively reducing their sum.

Figure 14 and Table IV show results with dynamic weights, for three cases. The first two are with with the sum of the weights constrained to be either 1 or 0.5. The third case is our algorithm of applying an adjustment factor to the sum, based on the entropy of the combined audio-visual posterior probability distribution.

As can be seen, results with the dynamic algorithm are rather disappointing. The performance improves only slightly, but the improvement is present across all SNRs. The fact that there is a consistency in the results shows at least that this improvement is not random. However, a larger improvement and just as consistent across the SNRs can be obtained by simply halving the weights after their estimation from the respective mono-modal entropies. This reduces the weight sum to 0.5 and also has significant influence on the final results.

In our best knowledge, the variation of the sum of stream weights was not attempted before in the literature, neither with fixed weights nor with dynamic ones.

In conclusion, the constraint on the sum of the stream weights is unnecessary, and removing it can lead to significant performance gains across all the SNRs. However, a reliable method of estimating the level at which this sum should be reduced still needs to be found.

VI. Summary

This report presents our multimodal integration method, based on the estimated reliability of each stream. The problem of multimodal integration is solved by merging frame-level probabilities at each time instant, with weights reflecting each stream's importance.

Our stream reliability estimates are based on the entropies of the class-posterior distributions for each stream and for each frame. Since the weights need to be the inverse of these entropies and they also need to be scaled, several mappings from entropies to weights are investigated, both static and dynamic. Experimental results show that dynamic weights perform well in a variety of conditions, leading to improved accuracy compared to audio-only results across a wide range of SNRs, with both white noise and babble noise.

We also investigate the role played by the constraint typically imposed in other work from the literature on the sum of stream weights in multi-stream HMMs. Our findings are that reducing the sum of the weights can lead to significant improvement in the word-level accuracy of the recognizer, across all SNR levels.

The framework presented here is general, since the reliability measure that we use is not particular to either audio or video. The entropy of the posterior distribution can be used in any multimodal context to evaluate the reliability of each modality dynamically, allowing the adjustment of each stream's importance to temporal variations of its quality. The loss of one of the streams is also naturally handled in this framework, as the weights would automatically adjust to ignore the missing stream, reverting to a mono-modal situation.

References

[1] M. Gurban and J. Thiran, "Using entropy as a stream reliability estimate for audio-visual speech recognition," in *Proceedings of the 16th European Signal Processing Conference*, 2008.

[2] M. Gurban, J. Thiran, T. Drugman, and T. Dutoit, "Dynamic modality weighting for multi-stream HMMs in Audio-Visual Speech Recognition," in *Proceedings of the 10th International Conference on Multimodal Interfaces*, 2008.

[3] E. Petajan, "Automatic lipreading to enhance speech recognition," in *Proceedings of the IEEE Communication Society Global Telecommunications Conference*, 1984.

[4] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[5] H. Silverman and D. Morgan, "The application of dynamic programming to connected speech recognition," *IEEE ASSP Magazine*, vol. 7, no. 3, pp. 6–25, 1990.

[6] L. Rabiner and B. Juang, "An introduction to Hidden Markov Models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[7] L. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77(2), 1989.

[8] R. Bakis, "Continuous speech recognition via centisecond acoustic states," *Proceedings of the 91st Meeting of the Acoustical Society of America*, 1976.

[9] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[10] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[11] N. Morgan and H. Bourlard, "Continuous speech recognition, an introduction to the hybrid HMM/connectionist approach," *IEEE Signal Processing Magazine*, vol. 12, no. 3, pp. 25–42, 1995.

[12] M. Heckmann, F. Berthommier, and K. Kroschel, "A hybrid ANN/HMM audio-visual speech recognition system," *Proceedings of the International Conference on Audio-Visual Speech Processing*, 2001.

[13] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[14] B. Schlkopf and A. Smola, *Learning with Kernels*. MIT Press, 2002.

[15] A. Ganapathiraju, J. Hamaker, and J. Picone, "Hybrid SVM/HMM architectures for speech recognition," *Proceedings of the International Conference on Spoken Language Processing*, vol. 4, pp. 504–507, 2000.

[16] M. Gordan, C. Kotropoulos, and I. Pitas, "A support vector machine-based dynamic network for visual speech recognition applications," *EURASIP Journal on Applied Signal Processing*, vol. 2002(11), pp. 1248–1259, 2002.

[17] G. Potamianos, C. Neti, J. Luettin, and I. Matthews, "Audio-visual automatic speech recognition: an overview," in *Issues in audio-visual speech processing*, G. Bailly, E. Vatikiotis-Bateson, and P. Perrier, Eds. MIT Press, 2004.

[18] A. Adjoudani and C. Benoît, "On the integration of auditory and visual parameters in an HMM-based ASR," in *Speechreading by humans and machines*, D. Stork and M. Hennecke, Eds. Springer, 1996, pp. 461–471.

[19] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book*. Cambridge, Entropic Ltd., 1999.

[20] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[21] K. Kirchhoff and J. Bilmes, "Dynamic classifier combination in hybrid speech recognition systems using utterance-level confidence values," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 693–696, 1999.

[22] G. Potamianos and H. Graf, "Discriminative training of HMM stream exponents for audio-visual speech recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 1998, pp. 3733–3736.

[23] S. Nakamura, H. Ito, and K. Shikano, "Stream weight optimization of speech and lip image sequence for audio-visual speech recognition," *Proceedings of the International Conference on Spoken Language Processing*, vol. III, pp. 20–23, 2000.

[24] G. Gravier, S. Axelrod, G. Potamianos, and C. Neti, "Maximum entropy and MCE based HMM stream weight estimation for audio-visual ASR," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2002.

[25] C. Miyajima, K. Tokuda, and T. Kitamura, "Audio-visual speech recognition using MCE-based HMMs and model-dependent stream weights," *Proceedings of the International Conference on Spoken Language Processing*, vol. II, pp. 1023–1026, 2000.

[26] P. Jourlin, "Word dependent acoustic-labial weights in HMM-based speech recognition," *Proceedings of the European Tutorial Workshop on Audio-Visual Speech Processing*, pp. 69–72, 1997.

[27] E. Sanchez-Soto, A. Potamianos, and K. Daoudi, "Unsupervised stream weight computation using anti-models," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 2007.

[28] S. Tamura, K. Iwano, and S. Furui, "A stream-weight optimization method for audio-visual speech recognition using multi-stream HMMs," *Proc. International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 857–860, 2004.

[29] ——, "A stream-weight optimization method for multi-stream HMMs based on likelihood value normalization," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 2005, pp. 468–472.

[30] N. Fox, R. Gross, J. Cohn, and R. Reilly, "Robust biometric person identification using automatic classifier fusion of speech, mouth, and face experts," *IEEE Transactions on Multimedia*, vol. 9, no. 4, pp. 701–714, 2007.

[31] S. Cox, I. Matthews, and A. Bangham, "Combining noise compensation with visual information in speech recognition," *Proceedings of the European Tutorial Workshop on Audio-Visual Speech Processing*, 1997.

[32] S. Gurbuz, Z. Tufekci, E. Patterson, and J. Gowdy, "Application of affine-invariant fourier descriptors to lipreading for audio-visual speech recognition," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 177–180, 2001.

[33] P. Teissier, J. Robert-Ribes, and J. Schwartz, "Comparing models for audiovisual fusion in a noisy-vowel recognition task," *IEEE Transactions on Speech and Audio Processing*, vol. 7, pp. 629–642, 1999.

[34] M. Heckmann, F. Berthommier, and K. Kroschel, "Noise adaptive stream weighting in audio-visual speech recognition," *EURASIP Journal on Applied Signal Processing*, vol. 2002, pp. 1260–1273, 2002.

[35] U. Meier, W. Hurst, and P. Duchnowski, "Adaptive bimodal sensor fusion for automatic speechreading," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 833–836, 1996.

[36] S. Dupont and J. Luettin, "Audio-visual speech modeling for continuous speech recognition," in *IEEE Transactions on Multimedia*, vol. 2, 2000, pp. 141–151.

[37] F. Berthommier and H. Glotin, "A new SNR-feature mapping for robust multistream speech recognition," *Proceedings of the International Congress on Phonetic Sciences*, pp. 711–715, 1999.

[38] H. Glotin, D. Vergyri, C. Neti, G. Potamianos, and J. Luettin, "Weighting schemes for audio-visual fusion in speech recognition," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 173–176, 2001.

[39] G. Potamianos and C. Neti, "Stream confidence estimation for audio-visual speech recognition," *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 2000.

[40] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. Senior, "Recent advances in the automatic recognition of audio-visual speech," *Proceedings of the IEEE*, vol. 91(9), 2003.

[41] R. Seymour, D. Stewart, and J. Ming, "Audio-visual integration for robust speech recognition using maximum weighted stream posteriors," *Proceedings of Interspeech*, 2007.

[42] H. Misra, H. Bourlard, and V. Tyagi, "New entropy based combination rules in HMM/ANN multi-stream ASR," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003.

[43] E. Patterson, S. Gurbuz, Z. Tufekci, and J. Gowdy, "Moving-talker, speaker-independent feature study and baseline results using the CUAVE multimodal speech corpus," *EURASIP Journal on Applied Signal Processing*, vol. 2002(11), pp. 1189–1201, 2002.

[44] G. Potamianos and P. Scanlon, "Exploiting lower face symmetry in appearance-based automatic speechreading," *Proceedings of the International Conference on Audio-Visual Speech Processing*, 2005.

[45] R. Seymour, J. Ming, and D. Stewart, "A new posterior based audio-visual integration method for robust speech recognition," *Proceedings of Interspeech*, p. 1229.