Chapter 13
Use of Missing and Unreliable Data for Audiovisual Speech Recognition

Alexander Vorwerk, Steffen Zeiler, Dorothea Kolossa, Ramon Fernandez Astudillo, Dennis Lerch

Abstract Under acoustically distorted conditions, any available video information is especially helpful for increasing recognition robustness. However, an optimal strategy for integrating audio and video information is difficult to find, since both streams may independently suffer from time-varying degrees of distortion. In this chapter, we show how missing feature techniques for coupled HMMs can help to fuse information from both uncertain information sources. We also focus on the estimation of reliability for the video feature stream, which is obtained from a linear discriminant analysis (LDA) applied to a set of shape- and appearance based features. The approach has resulted in significant performance improvements under strongly distorted conditions, while in conjunction with stream weight tuning being lower-bounded in performance to the best of the two single stream recognizers under all tested conditions.

Alexander Vorwerk
Electronics and Medical Signal Processing (EMSP), Technische Universität Berlin (TU Berlin), Einsteinufer 17, 10587 Berlin, e-mail: alexander.vorwerk@tu-berlin.de

Steffen Zeiler
EMSP, TU Berlin, e-mail: s.zeiler@ee.tu-berlin.de

Dorothea Kolossa
EMSP, TU Berlin, e-mail: dorothea.kolossa@gmx.de

Ramon Fernandez Astudillo
EMSP, TU Berlin, e-mail: rast79@gmail.com

Dennis Lerch
EMSP, TU Berlin, e-mail: dennis_lerch@gmx.de

13.1 Introduction

The inclusion of the visual modality can significantly improve the robustness of automatic speech recognition systems. This can be achieved using HMMs or more generally graphical models. Examples of successful applications have been described in e.g. [15, 22, 32, 33]. However, both the audio signal and the video images may suffer from noise or artifacts, and the video features are furthermore subject to misdetections of the face and mouth. Thus, for optimal performance, the audiovisual recognition system should be capable of considering each of the modalities as uncertain. This is of special interest here, because unreliable segments of either audio or video data can be compensated at least in part by the other modality, making missing or uncertain feature recognition especially attractive for multimodal data.

As one such strategy, uncertainty compensation has previously been employed in [36], using uncertainty decoding to deal with unreliable features. This approach is used here as well, and it is compared with both binary uncertainties and with *modified imputation*, an uncertain recognition strategy that is proving even more beneficial in the considered context.

This chapter is organized as follows. At first, Section 13.2 will introduce the use of coupled HMMs in audiovisual speech recognition and will describe the changes that are necessary to accommodate missing or uncertain feature components. Next, in Section 13.3 and 13.4, an overview of strategies for the calculation of audio and video features will be presented. In Section 13.5 an audiovisual recognition system is described. This includes the presentation of the audiovisual speech recognizer JASPER, which will be used for all subsequent experiments. This system is based on and allows for asynchronous streams as long as synchrony is again achieved at word boundaries. Subsequently, the feature extraction and uncertainty estimation are presented. Afterwards, the utilized strategy for multistream missing feature recognition is described. Section 13.6 deals with an efficient implementation of the presented recognizer and also shows the results for this system on the GRID database, a connected word small-vocabulary audiovisual database. All results and further implications are discussed in Section 13.7.

13.2 Coupled HMMs in Audiovisual Speech Recognition

Using a number of streams of audio and/or video features may significantly increase robustness and performance of automatic speech recognition [30,33]. In audiovisual speech recognition (AVSR), usually two streams of feature vectors are available, which are denoted $x_{\mathrm{a}}(t)$ for the acoustical and $x_{\mathrm{v}}(t)$ for the visual feature vector at time t. Both streams are not necessarily synchronized. Technical influences, like differing sampling rates or other recording conditions, may be compensated for by a variety of measures, e.g. synchronous sampling or interpolation. However, other causes of asynchronicities are based in the speech production process itself, in which variable delays between articulator movements and voice production lead to time-varying lags between video and audio. Such lags may have a duration of up to 120 ms, which corresponds to the duration of up to an entire phoneme [27].

In order to allow for such delays, various model designs have been used, e.g. multistream HMMs, coupled HMMs (CHMMs), product HMMs or independent HMMs [32]. These differ especially in the degree of required synchrony between modalities from the one extreme of independent HMMs, where both feature streams can evolve with no coupling whatsoever, to the other extreme of multistream HMMs, in which a state-wise alignment is necessary or at least implicitly assumed. Coupled HMMs allow for both streams to have lags or evolve at different speeds, with the obligation that they must again be synchronous at all word boundaries. Since this introduces some constraints but does not force unachievable frame-by-frame alignment, they present a reasonable compromise and have been used for all the following work.

13.2.1 Two-Stream Coupled HHM for Word Models

In coupled hidden Markov models, both feature vector sequences are retained as separate streams. As generative models, CHMMs can describe the probability of both feature streams jointly as a function of a set of two discrete, hidden state variables, which evolve analogously to the single state variable of a conventional HMM.

Thus, CHMMs have a two-dimensional state $\mathbf{q}$ which is composed of an audio and a video state, q_a and q_v, respectively, as can be seen in Fig. 13.1.

Each sequence of states through the model represents one possible alignment with the sequence of observation vectors. To evaluate the likelihood of such an alignment, each state pairing is connected by a transition probability, and each state is associated with an observation probability distribution.

The transition probability and the observation probability can both be composed from the two marginal HMMs. Then, the coupled transition probability becomes

$$\begin{aligned} &\mathrm{p}\Big(q_\mathrm{a}(t+1)=j_\mathrm{a}, q_\mathrm{v}(t+1)=j_\mathrm{v}|q_\mathrm{a}(t)=i_\mathrm{a}, q_\mathrm{v}(t)=i_\mathrm{v}\Big) \\ &\qquad = a_\mathrm{a}(i_\mathrm{a}, j_\mathrm{a}) \cdot a_\mathrm{v}(i_\mathrm{v}, j_\mathrm{v}) \end{aligned} \tag{13.1}$$

where $a_\mathrm{a}(i_\mathrm{a}, j_\mathrm{a})$ and $a_\mathrm{v}(i_\mathrm{v}, j_\mathrm{v})$ correspond to the transition probabilities of the two marginal HMMs, the audio-only and the video-only single-stream HMMs, respectively.

For the observation probability, both marginal HMMs could equally be composed to form a joint output probability by

$$\mathrm{p}(\mathbf{x}|\mathbf{i}) = b_\mathrm{a}(x_\mathrm{a}|i_\mathrm{a}) \cdot b_\mathrm{v}(x_\mathrm{v}|i_\mathrm{v}). \tag{13.2}$$

Here, $b_\mathrm{a}(x_\mathrm{a}|i_\mathrm{a})$ and $b_\mathrm{v}(x_\mathrm{v}|i_\mathrm{v})$ denote the output probability distributions for both single streams. In this case, there is a fairly large degree of independence between both models, and coupling takes place only in so far, as both streams are constrained

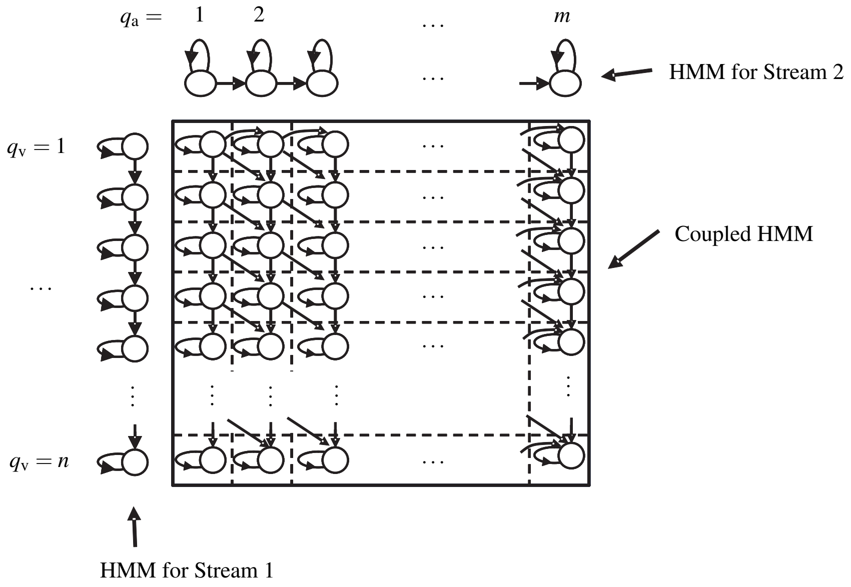

Fig. 13.1: A coupled HMM consists of a matrix of interconnected states, which each correspond to the pairing of one audio- and one video-HMM state, q_a and q_v, respectively.

to be generated by the same word model, i.e., stream synchronicity is enforced at word boundaries by the CHMM structure.

However, such a formulation does not allow to take into account the different reliabilities of audio and video stream. Therefore, Eq. (13.2) is commonly modified by an additional stream weight γ as follows[1]

$$\mathrm{p}(\mathbf{x}|\mathbf{i}) = b_a(x_a|i_a)^{\gamma} \cdot b_v(x_v|i_v)^{1-\gamma}. \tag{13.3}$$

This approach, described in more detail e.g. in [33], is also adopted in the audio-visual speech recognizer (JASPER) that is presented in Section 13.5.4.

13.2.2 Missing Features in Coupled HMMs

Recognition of speech in noisy or otherwise difficult conditions can greatly profit from so-called *missing feature approaches*. In these methods, those features within a signal that are dominated by noise or overly distorted by other detrimental effects, are considered “missing,” and subsequently disregarded in speech recognition [5].

[1] Please note that the resulting function does not correspond a probability density as such, since it does not adhere to the normalization condition unless $\gamma = 1$. However, we will still refer to these non-normalized functions with a $\mathrm{p}(\cdot)$ for simplicity of notation.

While binary uncertainties are beneficial in many situations, allowing for real-valued feature uncertainties leads to a Bayesian framework for speech recognition, which can utilize reliability information in a more precise and flexible way, as described in detail in Chapter 2 of this book. This can also allow for feature uncertainties to be estimated in one suitable domain, e.g. in the short-time Fourier transform (STFT) domain, and then propagated to the recognition domain of interest. Appropriate techniques for such an *uncertainty propagation* have been summarized in Chapter 3 and form the basis of the *uncertain feature recognition* we use here as the second alternative.

In the following section, a short overview of the utilized missing and uncertain feature techniques is given and the integration of both uncertain feature streams in a *missing-feature coupled HMM* is described.

13.2.2.1 Missing Feature Theory

In real world applications, some parts of a speech signal often are occluded by noise or interference, while the visual modality may suffer from varying lighting conditions or misdetected faces or mouths. In these cases, missing feature theory may help the recognizer to focus only on features that are considered reliable. Both continuous-valued and binary methods exist to incorporate such uncertainties in the recognition process. In either case, following the notation from Chapter 2, we assume that the clean observation vector $\mathbf{x}(t)$ is not directly observable, but that rather, only a noisy or otherwise distorted vector $\mathbf{y}(t)$ is accessible.

In the realm on binary uncertainties, two main approaches can be distinguished, marginalization and imputation [39]. In both cases, a binary mask is needed, which marks reliable and unreliable regions in the feature domain. In the following, the uncertainty value for the audio feature stream is denoted as $u_{\mathrm{a}}(k,t)$ and is determined both feature-wise and frame-wise. Where the video stream is concerned, $u_{\mathrm{v}}(t)$ denotes a frame-wise uncertainty value.

In a standard HMM with M Gaussian mixture components, for a given state q the output probability is usually computed by

$$\mathrm{b}\big(\mathbf{x}(t)\big) = \sum_{m=1}^{M} w_m \cdot \mathscr{N}\big(\mathbf{x}(t), \boldsymbol{\mu}_{q,m}, \boldsymbol{\Sigma}_{q,m}\big) \tag{13.4}$$

with $\boldsymbol{\mu}_{q,m}$ and $\boldsymbol{\Sigma}_{q,m}$ as mean and covariance matrix of mixture m and w_m as the mixture weight.

Utilizing marginalization to recognize uncertain audio features as described above, the calculation of the output probability is modified to

$$\mathrm{b}\big(\mathbf{y}(t)\big) = \sum_{m=1}^{M} w_m \cdot \mathscr{N}\big(\mathbf{y}^r(t), \boldsymbol{\mu}^r_{q,m}, \boldsymbol{\Sigma}^r_{q,m}\big). \tag{13.5}$$

In Eq. (13.5), $\mathbf{y}^r(t)$ describes a reduced feature vector, which only contains those components k that are reliable at the given time, i.e. for which the uncertainty value $u_\mathrm{a}(k,t)$ equals 0, so that we can assume $\mathbf{y}^r(t) = \mathbf{x}(t)$. Similarly, $\boldsymbol{\mu}^r_{q,m}$ is a reduced mean vector and $\boldsymbol{\Sigma}^r_{q,m}$ is the reduced covariance matrix, from which all rows and columns k with an uncertainty of $u_\mathrm{a}(k,t) = 1$ have been removed.

13.2.2.2 Uncertainty Decoding

Uncertainty decoding is based on computing observation probabilities by integrating over all possible values for the true feature vector $\mathbf{x}$. As shown in [9], this results in the time varying diagonal uncertainty covariance matrix $\boldsymbol{\Sigma}_y$ becoming an additive component to the state covariance matrix $\boldsymbol{\Sigma}_{q,m}$, which is later substituted by

$$\tilde{\boldsymbol{\Sigma}}_{q,m}(t) = \boldsymbol{\Sigma}_{q,m} + \boldsymbol{\Sigma}_y \tag{13.6}$$

in the likelihood evaluation of the m'th mixture component, so

$$b_m\big(\mathbf{y}(t)\big) = \mathcal{N}\left(\mathbf{y}(t), \boldsymbol{\mu}_{q,m}, \tilde{\boldsymbol{\Sigma}}_{q,m}(t)\right). \tag{13.7}$$

13.2.2.3 Modified Imputation

Modified imputation [21] is based on the idea of computing an estimated observation vector $\hat{\mathbf{x}}$, which is a weighted average of the mean of the currently evaluated mixture m in state q, and of the corrupted feature vector $\mathbf{y}(t)$. The relative weighting is determined from the uncertainty covariance matrix $\boldsymbol{\Sigma}_y$ and the mixture's covariance matrix:

$$\hat{\mathbf{x}}(t) = \left(\boldsymbol{\Sigma}_y^{-1} + \boldsymbol{\Sigma}_{q,m}^{-1}\right)^{-1} \left(\boldsymbol{\Sigma}_y^{-1}\mathbf{y}(t) + \boldsymbol{\Sigma}_{q,m}^{-1}\boldsymbol{\mu}_{q,m}\right). \tag{13.8}$$

Finally, the observation probability of mixture component m is obtained from

$$b_m\big(\mathbf{y}(t)\big) = \mathcal{N}\left(\hat{\mathbf{x}}(t), \boldsymbol{\mu}_{q,m}, \boldsymbol{\Sigma}_{q,m}\right).$$

13.2.2.4 The Missing Feature Coupled HMM

In the case of coupled HMMs, according to Eq. (13.3) the output probability computation is factorized into two streams. For that reason, marginalization, (modified) imputation, or uncertainty decoding can also be carried out independently for each stream.

Since we will be using binary uncertainties and marginalization for the video stream, that leaves the following three options for the implementation of the observation probability evaluation:

$$\mathrm{p}(\mathbf{y}|\mathbf{q}) = \mathrm{b}_{\mathrm{a,UD}}(\mathbf{y}_a|q_a)^{\gamma} \cdot \mathrm{b}_{\mathrm{v,MA}}(\mathbf{y}_v|q_v)^{1-\gamma} \quad \text{or} \tag{13.9}$$

$$\mathrm{p}(\mathbf{y}|\mathbf{q}) = \mathrm{b}_{\mathrm{a,MI}}(\mathbf{y}_a|q_a)^{\gamma} \cdot \mathrm{b}_{\mathrm{v,MA}}(\mathbf{y}_v|q_v)^{1-\gamma} \quad \text{or} \tag{13.10}$$

$$\mathrm{p}(\mathbf{y}|\mathbf{q}) = \mathrm{b}_{\mathrm{a,MA}}(\mathbf{y}_a|q_a)^{\gamma} \cdot \mathrm{b}_{\mathrm{v,MA}}(\mathbf{y}_v|q_v)^{1-\gamma}. \tag{13.11}$$

If the video feature uncertainty is computed only once for each frame and extends to the entire video feature vector at that time (see Section 13.5.2), for each frame and each HMM state $\mathbf{q} = (q_{\mathrm{a}}, q_{\mathrm{v}})$ the evaluation is simplified to the expression

$$\mathrm{p}(\mathbf{y}|\mathbf{q}) = \begin{cases} b_{\mathrm{a}}(y_{\mathrm{a}}|q_{\mathrm{a}})^{\gamma} \cdot b_{\mathrm{v}}(y_{\mathrm{v}}|q_{\mathrm{v}})^{1-\gamma} & \text{for } u_{\mathrm{v}}(t) = 0, \\ b_{\mathrm{a}}(y_{\mathrm{a}}|q_{\mathrm{a}})^{\gamma} & \text{for } u_{\mathrm{v}}(t) = 1. \end{cases} \tag{13.12}$$

13.3 Audio Features

As audio features, we are comparing two feature sets. On the one hand RASTA-PLP-cepstra have been chosen due to their comparatively large robustness with respect to noisy and reverberant conditions. Part of this robustness is due to the RASTA-PLP feature extraction having been designed to concentrate on features with a certain rate of change, namely that rate of change which is typical of speech. For that purpose, features are band-pass filtered. This makes them more robust to variations in room transfer function and in speaker characteristics, and to changes caused by background noise varying more quickly or slowly than speech signals [16].

On the other hand, the ETSI advanced front end introduced in [12] was used. The features specified there comprise Mel-cepstrum coefficients with deltas and accelerations, which have been optimized for achieving maximum robustness, using both a double Wiener filter and a cepstrum normalization step to enhance the tolerance for adverse environmental influences.

13.4 Video Features

Finding the face region and deriving significant video features is a crucial task in any audiovisual recognition process. In this section, an overview on face finding algorithms followed by a description of possible video features is given.

13.4.1 Face Recognition

Any video feature extraction algorithm strongly depends on a robust detection of the speaker's face. A wide range of methods have been presented for that task, an overview is given in [45]. Usually, four methods are distinguished:

- Appearance-based: use of models that are based on the appearance (intensity values) of the face, e.g. Eigenfaces, classification using artificial neural networks (ANN), naive Bayes, Independent Component Analysis (ICA)
- Feature invariant: searching for features that are invariant to changes in pose and lighting condition, e.g. skin color, texture, facial features
- Template matching: correlation with typical templates, e.g. of the face shape
- Knowledge based: methods using rules based on expert knowledge, e.g. a multiresolution approach in order to find typical face parameters in every stage according to heuristic rules.

Despite this categorization, hybrid approaches exist, i.e. a combination of the above mentioned face finding methods is used. For example, in [17], a skin tone detection is combined with a search for eyes and mouth via maps generated by knowledge-based methods that are applied to different scales of the image.

While the selection of a method depends on the given task (e.g. single speaker close-up view vs. multi-speaker video conference scenario) it has been shown that color information is useful for detection in general. In order to detect faces independently from brightness, color images are often converted from the red/green/blue (RGB) representation to a color space that parts luminance and chrominance. A very discriminative color space that has been widely used is YCbCr, which consists of one luma and two color components (e.g. [43], [40], [13]).

An example for a hybrid face finding approach including the localization of face components like eyes, nose and mouth is described in Section 13.5.1.

13.4.2 Feature Extraction

Once the location of the face in a speaker image is known, the region of interest (ROI), which the desired features are to be calculated from, has to be determined. Usually, the mouth region, sometimes extended to parts of the lower face, is extracted. Various methods have been reported for that task, such as correlation with templates [41], binarization methods based on color information thresholds of the lips [44] or eyes [26] and Gaussian estimations of lip color densities [37].

Based on the ROI, visual features applicable to audiovisual speech recognition have to be derived. Three basic classes of visual feature determination are distinguished: appearance based methods, shape or contour based methods and a combination of them [29]. As a preprocessing step, ROI normalization to a defined size as well as mean or variance adjustments are often conducted [44]. Subsequently to feature calculation, postprocessing stages such as linear discriminant analysis (LDA) or maximum likelihood linear transform (MLLT) can significantly improve classification results [38].

13.4.2.1 Appearance Based Methods

These methods encompass algorithms that provide visual features on an image pixel basis. This includes statistical methods like principal component analysis (PCA) as well as linear image transforms, e.g. discrete cosine and wavelet transforms (DCT and DWT, respectively). The DCT of the speaker's gray level image is very popular and a widely used technique because of its good results [1, 26] combined with low calculation costs. A further aspect in using the mentioned transforms is the dimensionality reduction which is reached when choosing only high energy coefficients.

In addition to appearance based features that are determined on single images, it has been reported that motion information may be useful in AVSR tasks (e.g. [7]). For instance, information on ROI movements of adjacent video frames can be extracted pixelwise using optical flow (OF) estimations [6]. Examples of using OF-based visual features are given in [28, 35, 42].

13.4.2.2 Shape Based Methods

An intuitive way of exploiting visual information is using shape parameters of the mouth or a broader face region. Such parameters may be geometrical properties of the lips, such as width, height or the area inside of the lip contours [14], image moments or Fourier descriptors. Furthermore, statistical lip models, like active shape models (ASM), have been used in the community [23]. In any method using shape information, a robust detection of the contours of the lips is necessary. For that purpose, data is either hand-labeled or automatically gathered using e.g. deformable templates (parabolas, [2]) or so-called snakes.

Active Contours (Snakes)

Active contours in an image are somewhat similar in appearance to snakes in nature and therefore often simply called snakes. Snakes are used in image processing for object segmentation and will be applied here to approximate the lip region boundary by a closed curve. This curve in two dimensional space is approximated by a set of sorted discrete points along the curve boundaries. Each point is connected to its neighbors by a straight line. The shape of the snake is influenced by different so-called forces, with a distinction being made between *internal* and *external* forces. The internal forces represent the properties of the snake shape. The two most common internal forces (sometimes formulated as energy terms), described in detail in [18], model the elastic behavior of the active contour as that of an elastic band. First of all there is the attractive force (see Eq. (13.13)), which is calculated with k as the index of the current, $k-1$ the index of the previous and $k+1$ the index of the next point. $\mathbf{p}_k$ is the position vector, and α_k stands for a weighting factor:

$$\begin{aligned}\mathbf{f}_{k,\text{attractive}} &= \alpha_k \cdot (\mathbf{p}_{k-1} - \mathbf{p}_k) + \alpha_{k+1} \cdot (\mathbf{p}_{k+1} - \mathbf{p}_k) \\ &= [\alpha_k, \alpha_{k+1}] \cdot \begin{bmatrix} \mathbf{p}_{k-1} - \mathbf{p}_k \\ \mathbf{p}_{k+1} - \mathbf{p}_k \end{bmatrix} \\ &= \boldsymbol{\alpha} \cdot \tilde{\mathbf{f}}_{k,\text{attractive}}. \end{aligned} \tag{13.13}$$

The so-called bending force, shown in Eq. (13.14), has a smoothing effect on the contour. Sharp corners lead to high bending forces, and with great weighting factors β_k, $(\beta_k \gg 1)$ the snake tends pass over local structural discontinuities and avoid any acute angles.

$$\begin{aligned}\mathbf{f}_{k,\text{bending}} &= \beta_{k-1}\,(2\,\mathbf{p}_{k-1} - \mathbf{p}_{k-2} - \mathbf{p}_k) - \\ &\quad 2\,\beta_k\,(2\,\mathbf{p}_k - \mathbf{p}_{k-1} - \mathbf{p}_{k+1}) + \\ &\quad \beta_{k+1}\,(2\,\mathbf{p}_{k+1} - \mathbf{p}_k - \mathbf{p}_{k+2}) \\ &= [\beta_{k-1}, \beta_k, \beta_{k+1}] \cdot \begin{bmatrix} 2\,\mathbf{p}_{k-1} - \mathbf{p}_{k-2} - \mathbf{p}_k \\ -2\,(2\,\mathbf{p}_k - \mathbf{p}_{k-1} - \mathbf{p}_{k+1}) \\ 2\,\mathbf{p}_{k+1} - \mathbf{p}_k - \mathbf{p}_{k+2} \end{bmatrix} \\ &= \boldsymbol{\beta} \cdot \tilde{\mathbf{f}}_{k,\text{bending}} \end{aligned} \tag{13.14}$$

The total internal force of every discrete point k becomes

$$\mathbf{f}_{k,\text{int.}} = \mathbf{f}_{k,\text{attractive}} + \mathbf{f}_{k,\text{bending}}. \tag{13.15}$$

Image contents and other peripheral effects are considered as external forces. Object boundaries (or edges in the image) are the most important external force and will usually be included in the force calculation via image gradients $\mathbf{g}_k$. Another useful force, which can be used to allow concave active contours, is the center of gravity force

$$\mathbf{f}_{k,\text{gravity}} = \frac{\mathbf{r}_k}{|\mathbf{r}_k|} \cdot |\mathbf{r}_k|^i = \mathbf{r}_k \cdot |\mathbf{r}_k|^{i-1} \ , \tag{13.16}$$

with $\mathbf{r}_k = \mathbf{b} - \mathbf{p}_k$, $\mathbf{b}$ as the position vector of the center of gravity, and i as the order of the vector norm. Thus, the total external force, with extra weighting factors μ_k and ν_k, is

$$\mathbf{f}_{k,\text{ext.}} = \mu_k \cdot \mathbf{g}_k + \nu_k \cdot \mathbf{f}_{k,\text{gravity}} \ . \tag{13.17}$$

Active contours are deformed iteratively so as to minimize the total force

$$\mathbf{f}_{k,\text{total}} = \mathbf{f}_{k,\text{int.}} + \mathbf{f}_{k,\text{ext.}} = \boldsymbol{\alpha} \cdot \tilde{\mathbf{f}}_{k,\text{attractive}} + \boldsymbol{\beta} \cdot \tilde{\mathbf{f}}_{k,\text{bending}} + \mu_k \cdot \mathbf{g}_k + \nu_k \cdot \mathbf{f}_{k,\text{gravity}}, \tag{13.18}$$

until either $\mathbf{f}_{k,\text{total}} \approx \mathbf{0}$ or the maximum number of allowed iterations is exceeded. In Eq. (13.18), $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are weighting factors, which are independent of the index k in the following considerations. The deformation is an iterative minimization process. To find the optimum shape of a snake, it is necessary to calculate the partial derivatives with respect to the discrete point coordinates and set them equal to zero, for which the derivation is given in [18]. The equivalence between the energy and force formulation is shown in [24]. An application can be found in Section 13.5.2.2.

13.4.2.3 Combined Shape and Appearance Based Methods

Instead of simply concatenating appearance and shape based features, an approach of incorporating both sources of information using a single statistical model has been proposed which is called active appearance model (AAM). In this model, intensity values of all points inside the predetermined lip contour are used to build a statistical model based on principal components analysis (for a detailed description, see e.g. [29]).

13.4.2.4 Fisher Linear Discriminant

After features have thus been computed, a dimensionality reduction if often called for. In order to find a projection of the features that is most amenable for subsequent classification, linear discriminant analysis has proved beneficial in many studies.

The Fisher Linear Discriminant (FLD) gives a projection matrix **W** that reshapes the scatter of a data set to maximize class separability, defined as the ratio of the between-class scatter matrix to the within-class scatter matrix. This projection defines features that are optimally discriminative.

Let $\mathbf{x}_i$ be one among N column vectors of dimension D. The mean of the dataset is

$$\boldsymbol{\mu}_x = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i. \tag{13.19}$$

For K classes $\{C_1, C_2, \ldots, C_K\}$, the mean of class k containing N_k members is

$$\boldsymbol{\mu}_{xk} = \frac{1}{N_k}\sum_{\mathbf{x}_i \in C_k}\mathbf{x}_i. \tag{13.20}$$

The between class scatter matrix is

$$\mathbf{S}_B = \sum_{k=1}^{K} N_k(\boldsymbol{\mu}_{xk} - \boldsymbol{\mu}_x)(\boldsymbol{\mu}_{xk} - \boldsymbol{\mu}_x)^T, \tag{13.21}$$

and the within class scatter matrix is

$$\mathbf{S}_W = \sum_{k=1}^{K}\sum_{\mathbf{x}_i \in C_k}(\mathbf{x}_i - \boldsymbol{\mu}_{xk})(\mathbf{x}_i - \boldsymbol{\mu}_{xk})^T. \tag{13.22}$$

In order to find the within-class scatter matrix $\mathbf{S}_W$, either labelled data or the covariance matrices of a trained single-mixture full-covariance HMM may be used. The transformation matrix that repositions the data to be most separable is the matrix **W** that maximizes

$$\frac{\det(\mathbf{W}^T\mathbf{S}_B\mathbf{W})}{\det(\mathbf{W}^T\mathbf{S}_W\mathbf{W})}. \tag{13.23}$$

Let $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_D\}$ be the generalized eigenvectors of $\mathbf{S}_B$ and $\mathbf{S}_W$, which can be determined as the non-trivial solutions of

$$\mathbf{S}_B\mathbf{w} = \lambda\mathbf{S}_W\mathbf{w} \tag{13.24}$$

with λ as a scalar value $\neq 0$.

Then, $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_D]$ gives a projection space of dimension D. A projection space of dimension $d < D$ can be defined by using the generalized eigenvectors with the largest d eigenvalues to give $\mathbf{W}_d = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_d]$. The projection of vector $\mathbf{x}$ into the subspace of dimension d is $\mathbf{y} = \mathbf{W}_d^T\mathbf{x}$. The resultant feature vector $\mathbf{y}$ is subsequently used for recognition.

13.5 Audiovisual Speech Recognizer Implementation and Results

In this section, an implementation of a complete audiovisual recognizer is presented. After an overview of the used face detection method, localizing face features and deriving uncertainty measures from the calculated audio and video features, the utilized speech recognizer and the resulting recognition rates will be described. All experiments have been conducted on the GRID database. It contains utterances from 34 speakers recorded under strictly defined conditions, showing the speaker in a front view reading an English 6-word sentence (for details, see [8]).

13.5.1 Face Detection

The detection of the face utilizes a hybrid approach and is divided into detection of the skin, fitting an ellipse and a rotation correction. Afterwards, the mouth region is determined and, if necessary, a correction of the extracted region over a video sequence takes place.

13.5.1.1 Skin Detection

According to an algorithm proposed by [31], the skin detection is carried out in the YCbCr color space. Based on training images selected randomly from the database, a Gaussian model was trained for the color components in order to eliminate the influence of lighting conditions. Using this model, the color image of the original sequence is transformed into a gray scale image. The gray values represent the likelihood of every pixel for belonging to the skin region. An adaptive threshold is used to convert the gray scale into a binary image, in which the face is then detected as a region of at least a minimum size of connected skin pixels with characteristic holes (mouth, eyes, eyebrows) in it.

13.5.1.2 Ellipse Fitting and Rotation Correction

Because the face may be partly occluded and the image may also contain other skin regions besides the face, the binary skin mask is usually not the exact face region. In order to cope with that, an ellipse with the same balance point and moment of inertia as the binary face region is searched for. From that ellipse and the location of characteristic holes in the skin region that mark the position of eyes and mouth, an angle of rotation of the face is elicited. This angle is used to rotate the face back into a vertical position, so that all further processing steps can assume an upright face.

13.5.1.3 Mouth Region Extraction

The aim of this processing step is the robust determination of a rectangular mouth region. At first, an edge filter is applied to the face region followed by a summation in order to find horizontal lines that belong to the eyebrows, eyes, nostrils, mouth and the chin. Only along those lines that may represent the location of the mouth, correlation with a mouth template is carried out. Starting from the point of highest correlation, the color image is searched for edges in order to find the upper and lower lip, which, after addition of a resolution dependent number of pixels, mark the boundaries for the ROI rectangle.

Sequence Processing

When seeking the mouth in a sequence of images, one can assume that the change of location from one frame to another will be limited to a certain value according to the frame rate and resolution. In order to speed up the process of tracking the ROI, the mouth region found in one image is presented as starting point to the search engine for the next frame until a threshold, basically a certain number of frames, is reached. After that, a complete run, i.e. a reinitialization, of the mouth localization algorithm is started.

Post Processing

It has been found that the dimension of the extracted mouth region changes over the course of a sequence to an undesired degree, especially in reinitialization frames. Since this will decrease the ability to train reliable models based on visual features, lip corners are determined for all frames of a sequence. This is done using repeated morphological filter operations on a color corrected version of the ROI (similar to [25]). All detected mouth corner locations are then checked using geometrical preconditions. Based on valid locations, a mean position of the mouth over the entire sequence is determined that will be used to replace wrongly detected ROIs. Furthermore, all valid mouth corners are used to recalculate the size of the ROI which they were extracted from. As a consequence, all mouth regions of a sequence will only show smooth changes in size over frames. Before the calculation of visual features, the regions of interest are resized to a 64x64 pixel image.

13.5.2 Video Features and their Uncertainties

As video features, DCT coefficients and active contours are used. To compute the uncertainties of video features, frame-wise methods are used that provide an uncertainty measure for both features. These uncertainties are combined afterwards using a simple decision function to be incorporated into the recognition process as binary uncertainties.

13.5.2.1 DCT Uncertainty

Based on the mouth regions extracted as described above, coefficients of a 2-dimensional discrete cosine transform according to Eq. (13.25) are calculated with $0 \leq k \leq M-1, 0 \leq l \leq N-1$ and α_k, α_l as normalization constants according to the size of the ROI. Beforehand, the color image of the mouth region is transformed to gray level.

$$DCT_{kl} = \alpha_k \alpha_l \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} ROI_{mn} \cos \frac{\pi(2m+1)k}{2M} \cos \frac{\pi(2n+1)l}{2N} \tag{13.25}$$

In order to obtain an uncertainty measure for the DCT video features, a speaker-dependent mouth-model is trained on 20 hand labelled sequences, which were selected to contain at least 30 wrongly estimated mouth regions in total. From these sequences, intact mouth regions are learned separately from a model combining both non-mouth and cropped-mouth regions. The model consists of a single Gaussian probability density function (pdf) of the same first 64 DCT coefficients which are also used for HMM training. Therefore, no additional feature extraction needs to take place.

Subsequently, a frame will be counted as reliable, if the log-likelihood p_{m} of the trained mouth model

$$p_{\mathrm{m}} = \log \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}_{\mathrm{m}}|}} \mathrm{e}^{-\frac{1}{2}(\mathbf{y}_{\mathrm{v}}(t)-\boldsymbol{\mu}_{\mathrm{m}})' \boldsymbol{\Sigma}_{\mathrm{m}}^{-1} (\mathbf{y}_{\mathrm{v}}(t)-\boldsymbol{\mu}_{\mathrm{m}})} \tag{13.26}$$

with mean $\boldsymbol{\mu}_{\mathrm{m}}$ and covariance matrix $\boldsymbol{\Sigma}_{\mathrm{m}}$ exceeds that of a non-mouth model

$$p_{\mathrm{n}} = \log \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}_{\mathrm{n}}|}} \mathrm{e}^{-\frac{1}{2}(\mathbf{y}_{\mathrm{v}}(t)-\boldsymbol{\mu}_{\mathrm{n}})' \boldsymbol{\Sigma}_{\mathrm{n}}^{-1} (\mathbf{y}_{\mathrm{v}}(t)-\boldsymbol{\mu}_{\mathrm{n}})} \tag{13.27}$$

with parameters $\boldsymbol{\mu}_{\mathrm{n}}$ and $\boldsymbol{\Sigma}_{\mathrm{n}}$. Thus, the uncertainty $u_{\mathrm{v,DCT}}(t)$ of all DCT features in frame t is given by

$$u_{\mathrm{v,DCT}}(t) = \begin{cases} 0 \text{ for } p_{\mathrm{m}} \geq p_{\mathrm{n}}, \\ 1 \text{ otherwise.} \end{cases} \tag{13.28}$$

A typical example of the resultant labeling of video frames can be seen in Fig. 13.2.

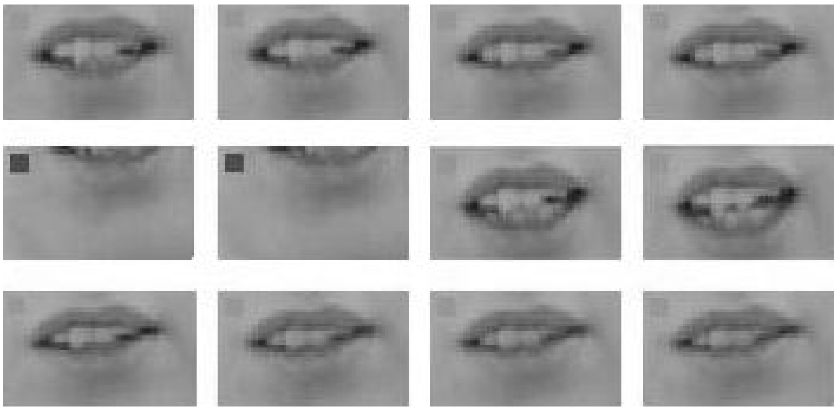

Fig. 13.2: Mouth regions and their associated labels. A light square means that the frame has been counted as reliable, otherwise, a dark square is shown.

13.5.2.2 Active Contour Uncertainty

In AVSR, active contours (a.k.a. snakes) consist of a certain number of coordinate pairs that, in an ideal case, mark exactly the location of the lip contour in the region of interest. From these contour points, it is possible to derive different criteria that provide information on the uncertainty of the calculated snake. Such measures are of a geometrical kind, e.g. the enclosed area, or utilize shape information like subsets of points that describe the upper and lower lip curves. Criteria on both characteristics are described in this section.

Snake Calculation

All of the above mentioned forces (see Sec. 13.4.2.2) are used with optimal weighting parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \mu_k, \nu_k$ according to Eq. (13.18) to obtain the outer lip curve. Due to the iterative curve fitting process, it is necessary to have an initial curve. For good segmentation results, it is recommended that the initial snake should outline the desired lip region in each frame as closely as possible. Since the coordinates of the mouth corners are already known from previous processing steps, we have initialized the snake as an ellipse with a major axis equal to the mouth corner distance and a minor axis equal to half the mouth region height. In every step of the iteration, the coordinates of the discrete points localised initially in the mouth corners are held constant and only the other points are adjusted. The optimal weighting parameters and further investigations can be found in [24].

The calculation of the image gradients is done by a differentiated three dimensional Gaussian filter with standard deviation of $\sqrt{1.5}$ pixels in every dimension. In order to calculate the gradients in boundary image regions, the last pixel values are replicated according to the filter size.

One of the arising problems is a considerable shift of the discrete points over the iteration process. The uniformly distributed points along the shape at the beginning are finally concentrated in regions indicated by the highest gradients. From this, it follows that in these regions the discrete points approximate the lip shape very well but in other sectors too little accuracy is reached. A modified gradient approach, which guarantees a constant relative distance between neighboring points, solves the problem. In principle, it uses a projection of the image gradients onto a line perpendicular to the secant connecting the two neighbor points $k-1$ and $k+1$ (see Fig. 13.3).

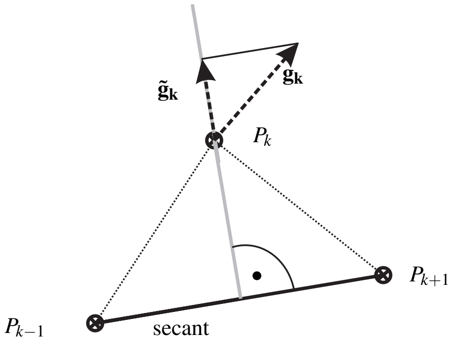

Fig. 13.3: Principle of the gradient projection. $\tilde{\mathbf{g}}_\mathbf{k}$ is the used projected gradient in the point of interest.

The coordinates of the discrete contour points are finally used for calculating the video features.

Surface Area

The enclosed surface area of a snake may help to identify miscalculations. For example, a very low surface value should not appear given a valid region of interest. Taking only the area into account is not robust enough to tolerable changes in the original size of the region of interest. Therefore, this ROI size (*rS*) is incorporated into the area criterion in two ways, as can be seen in Eq. (13.29). The first measure *pC* is derived from the product of the size of the surface area (*sA*) and the number of points of the original ROI, whereas the second measure *sC* is a weighted sum of normalized inputs. For both measures, a threshold is calculated that is subsequently used to denote an active contour as an uncertain candidate.

$$\begin{aligned} pC &= sA \cdot rS \\ sC &= \frac{2}{3} sA_{\text{norm}} \cdot \frac{1}{3} rS_{\text{norm}} \end{aligned} \tag{13.29}$$

Curve Parameters

The active contour shape may deliver further information on its validity. In order to recognize inappropriate curve parameters, a model has to be trained from labeled data. For this work, Artificial Neural Networks (ANN) have been trained on coefficients of 5th order polynomials that have been fitted to the coordinates of the upper and lower lip, respectively. All calculated snakes are classified using these ANNs, so that two additional criteria regarding uncertainty of the active contour are available.

13.5.2.3 Video Feature Uncertainty Combination for Decoding

By rule-based combination of all four criteria described above, a binary uncertainty flag is derived for the snake features, as shown in Fig. 13.4. Black dots in the corners of the images show validity criteria being met, whereas diamonds, squares and pentagrams indicate a miscalculated snake. Gray bars crossing an image stand for an uncertainty $u_{\text{v,AC}} = 1$ for this particular frame, while all other images have an uncertainty $u_{\text{v,AC}} = 0$.

13.5.3 Audio Features and their Uncertainties

13.5.3.1 Rasta-PLP

For the RASTA-PLP cepstrum, 12 coefficients and their first and second derivatives were used, which were obtained from a power spectrum of the speech signal using a window size of 25 ms with 15 ms overlap. RASTA-filtering takes place in the log-

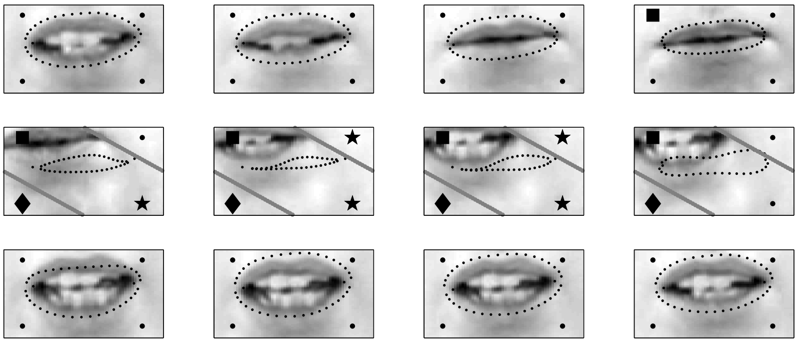

Fig. 13.4: Active contour uncertainty. Black dots near the corners indicate valid snakes. Black squares and diamonds mark invalid snakes according to surface area criteria, whereas pentagrams stand for invalid upper and lower lips, respectively. Gray bars visualize the combined hard decision of an invalid frame.

bark-spectrum, using the transfer function

$$H(z) = \frac{0.2z^4 + 0.1z^3 - 0.1z^{-3} - 0.2z^{-4}}{z^3(z - 0.94)} \tag{13.30}$$

given in [16]. Subsequently, the LPC-cepstrum is obtained from the RASTA-filtered log spectrum, using an LPC model of order 12. For these computations, the Rastamat toolbox was used, which is available from [11]. Finally, first and second order time derivatives were appended to the 12 RASTA-PLP-cepstral coefficients, which further improved robustness.

13.5.3.2 ETSI Advanced Front End

The feature extraction for the ETSI advanced front end follows the standard described in [12]. It uses 13 Mel-Frequency Cepstrum Coefficients, delta and acceleration parameters, which form a 39 dimensional feature vector. In it, the zero'th cepstral coefficient is replaced by the logarithm of the frame energy.

13.5.3.3 Uncertainties

For audio feature uncertainties, numerous approaches exist to estimate either binary or continuous-valued uncertainties. However, in most cases, binary uncertainty values are only used for spectra and log-mel-spectra, which are not very robust to variations in the speaker, the environment or the background noise. In contrast, a number of more elaborate mechanisms for estimating continuous uncertainty values in the feature domain have been developed over the past years, see e.g. [3, 9]. In the following, we will be considering both types of uncertainty.

For binary uncertainties, which are easy to compute but less robust than ideally achievable, a simple strategy is to compare each feature vector component $y_{\mathrm{a}}(k,t)$ at time t to the level of the estimated background noise n_{a}. These two vectors are compared for each of the feature components, and a reliability decision is made by

$$u_{\mathrm{a}}(k,t) = \begin{cases} 0 \text{ for } n_{\mathrm{a}}(k,t) < \alpha \cdot y_{\mathrm{a}}(k,t), \\ 1 \text{ otherwise,} \end{cases} \tag{13.31}$$

i.e. a feature is deemed unreliable if the value of the background noise feature exceeds the value of α times the observed signal feature. This binary feature uncertainty will be used in the following as the uncertainty of the RASTA-PLP cepstrum only. Its application for ETSI-advanced front end has not proven advantageous, possibly to the integrated de-noising stages of the ETSI-AFE, which would require a more advanced decision function for binary uncertainties.

As a second, continuous-valued uncertainty strategy, we will be considering the ETSI advanced front end. Since this feature extraction already contains a Wiener filter, an estimate of the estimators error variance is available also within the system. The continuous-valued uncertainties have been obtained from the Wiener filter estimator error variance in the complex spectrum domain [4], and propagated to the ETSI-MFCC backend via the uncertainty propagation techniques described in Chapter 3, Section 6.

13.5.4 Audiovisual Recognition System

The JASPER (JAVA AUDIOVISUAL SPEech RECOGNIZER) System, developed for robust single- and multistream speech recognition, can combine audio recognition and lipreading with the help of audiovisual CHMMs. It is based on a flexible token passing architecture applicable for a wide range of statistical speech models.

The system allows for a tight integration of the MATLAB and JAVA environments and capabilities, with an interface that lets preprocessing and feature extraction be carried out in MATLAB, whereas model training and recognition take place in JAVA.

JASPER is based on an abstract model in which connected word recognition is viewed as a process of passing tokens around a transition network [46]. Within this network, each vocabulary element is represented by a word model. These word models are statistical descriptions of the evolution of the feature stream over time within the associated words. Word models may be realized as conventional HMMs, coupled or product HMMs, or even templates, or a range of graphical models, which is an advantage of the abstraction inherent in the implemented token passing approach. Fig. 13.5 shows an example of a possible word net structure for the GRID database.

In addition to the word models, further network elements are link nodes and non-terminal nodes. Link nodes provide the connections between non-terminal nodes and word models and associate a linked list of possible word alternatives to all tokens passing through them. Non-terminal nodes allow grouping of nodes into dif-

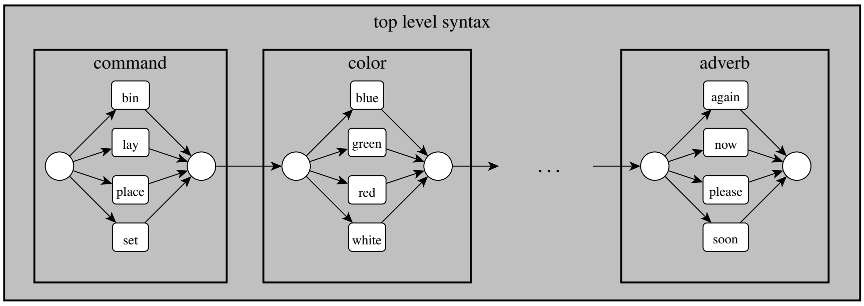

Fig. 13.5: An example of a word network for recognition of the GRID grammar described in [8]. Link nodes are depicted by circles, non-terminal nodes by bold black rectangles and word models are shown as rounded rectangles.

ferent hierarchical levels. The highest level in the network is the non-terminal node that represents the entire language model. The elements on the lowest level are the word models.

The recognition process starts with a single token entering the top level syntax. Every time step is split into two half steps. At first, all link nodes propagate their tokens down the hierarchy, until the lowest level, the word models, are reached. The actual calculations of word model log-likelihoods given the observed features happen in the second phase. At the end of the second phase, all link nodes collect tokens from their incoming connections and build a connected list of the n-best tokens. Tokens are ranked by a score, corresponding to their accumulated log-likelihood, and a global, adaptive threshold is used for efficient pruning. This process is iterated until a complete observation sequence has been processed and the outgoing link node of the top level syntax contains the token with the highest score given the model and the observed data.

13.6 Efficiency Considerations

Coupled HMMs involve, in general, a significantly increased search space when compared to one-dimensional or multistream HMMs. Therefore, computational efficiency is of some significance.

Profiling the search reveals it to be dominated in a large number of cases by the computation of output distributions. In case of missing or uncertain features, this becomes of even greater significance, since the computation of uncertain feature likelihoods causes an increase in computational demands. Therefore, the following section will describe some considerations that have proven valuable for efficient likelihood evaluations in CHMMs, both without and with missing data techniques.

13.6.1 Gaussian Density Computation

The likelihood computation for an observation vector $\mathbf{y}$ using a multivariate normal distribution

$$\mathscr{N}(\mathbf{y} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\mathsf{T} \boldsymbol{\Sigma}^{-1} (\mathbf{y}-\boldsymbol{\mu})\right) \tag{13.32}$$

with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ is prone to overflow errors because of the limited dynamic range of common floating-point number formats. To prevent overflows, the log-likelihood

$$\log(\mathscr{N}) = -\log\left(\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}\right) - \frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\mathsf{T} \boldsymbol{\Sigma}^{-1} (\mathbf{y}-\boldsymbol{\mu}) \tag{13.33}$$

is normally used instead.

For an efficient implementation, a transformation into a quadratic form

$$\log(\mathscr{N}) = s + \mathbf{u}^\mathsf{T}\mathbf{y} - (\mathbf{V}\mathbf{y})^\mathsf{T} (\mathbf{V}\mathbf{y}). \tag{13.34}$$

is helpful. After sorting terms, (13.33) gives

$$\begin{aligned}\log(\mathscr{N}) &= -\frac{1}{2}\Big(\log(2\pi)D + \log(|\boldsymbol{\Sigma}|)\Big) - \frac{1}{2}\Big(\mathbf{y}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{y} - 2\boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{y} + \boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\Big) \\ &= -\frac{1}{2}\Big(\log(2\pi)D + \log(|\boldsymbol{\Sigma}|) + \boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\Big) + \boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{y} - \frac{1}{2}\mathbf{y}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{y}.\end{aligned} \tag{13.35}$$

By means of the Cholesky decomposition, the inverse covariance matrix $\boldsymbol{\Sigma}^{-1}$ is factored into a product of upper triangular matrices $\mathbf{U}^\mathsf{T}\mathbf{U}$, so for the computation of the Mahalanobis distance, we have

$$\mathbf{y}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{y} = \mathbf{y}^\mathsf{T}\mathbf{U}^\mathsf{T}\mathbf{U}\mathbf{y} = \Big(\mathbf{U}\mathbf{y}\Big)^\mathsf{T}\Big(\mathbf{U}\mathbf{y}\Big). \tag{13.36}$$

This matrix $\mathbf{U}$ can also be written as the matrix square root $\mathbf{U} = \boldsymbol{\Sigma}^{-\frac{1}{2}}$. Inserting this into (13.35) yields

$$\begin{aligned}\log(\mathscr{N}) = &-\frac{1}{2}\Big(\log(2\pi)D + \log(|\boldsymbol{\Sigma}|) + \boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\Big) \\ &+ \boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{y} - \frac{1}{2}\Big(\boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{y}\Big)^\mathsf{T}\Big(\boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{y}\Big).\end{aligned} \tag{13.37}$$

After equating with (13.34), this gives the required coefficients:

$$s = -\frac{\log(2\pi)D + \log\left(|\boldsymbol{\Sigma}|\right) + \boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}}{2}, \tag{13.38}$$

$$\mathbf{u} = \boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}, \quad \mathbf{V} = \frac{\boldsymbol{\Sigma}^{-1/2}}{\sqrt{2}}. \tag{13.39}$$

These can be used for efficient evaluation of full covariance Gaussian mixture models.

13.6.1.1 Diagonal Model Implementations

Independently of the structure of the covariance matrix, the likelihood of a multivariate observation $\mathbf{y}$ always depends on the Mahalanobis distance

$$D_M(\mathbf{y}) = \sqrt{(\mathbf{y}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})} \tag{13.40}$$

for given mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. The following paragraph describes the efficient likelihood computation for diagonal covariance matrices and the necessary extensions for methods using observations with uncertainties.

13.6.2 Conventional Likelihood

For diagonal covariance matrices

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_D^2 \end{pmatrix}, \tag{13.41}$$

with the variances σ_i^2 on the main diagonal, the equations for the log-likelihood computations simplify to

$$\log\left(\mathcal{N}\right) = -\frac{1}{2}\left(\log(2\pi)D + \log\left(\prod_{i=1}^{D}\sigma_i^2\right) + \sum_{i=1}^{D}\frac{\boldsymbol{\mu}_i^2}{\sigma_i^2}\right) + \sum_{i=1}^{D}\frac{\boldsymbol{\mu}_i}{\sigma_i^2}y_i - \sum_{i=1}^{D}\frac{1}{2\sigma_i^2}y_i^2 \tag{13.42}$$

and the calculation can be expressed as a dot product

$$\left(s, \frac{\boldsymbol{\mu}_1}{\sigma_1^2}, \cdots, \frac{\boldsymbol{\mu}_D}{\sigma_D^2}, -\frac{1}{2\sigma_1^2}, \cdots, -\frac{1}{2\sigma_D^2}\right) \cdot \left(1, y_1, \cdots, y_D, y_1^2, \cdots, y_D^2\right)^{\mathsf{T}}, \tag{13.43}$$

between the parameter vector and an augmented observation vector [10]. This is especially helpful for parallel implementations and has been used to obtain faster results by extending JASPER to CASPER[2].

13.6.3 Uncertainty Decoding

In uncertainty decoding, the diagonal matrix of feature uncertainties $\boldsymbol{\Sigma}_y$ is used to obtain a modified covariance matrix for computing

$$p(\mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\Sigma}_y) = \mathcal{N}\left(\mathbf{y} \mid \boldsymbol{\mu}, \tilde{\boldsymbol{\Sigma}}\right). \tag{13.44}$$

Here, the model covariance $\boldsymbol{\Sigma}$ is substituted by $\tilde{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma} + \boldsymbol{\Sigma}_y$.

Because the entries of the covariance matrix are thus modified by the uncertainties, the factor s from Eq. (13.38) has to be recalculated in each frame by

$$s = -\frac{1}{2}\left(\log(2\pi)D + \log\left(\prod_{i=1}^{D} \tilde{\sigma}_i^2\right)\right). \tag{13.45}$$

Otherwise, the considerations from Section 13.6.2 apply.

13.6.4 Modified Imputation

In contrast to uncertainty decoding, modified imputation adapts the observation vector $\mathbf{y}$ with uncertainties $\boldsymbol{\Sigma}_y$

$$p(\mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\Sigma}_y) = \mathcal{N}\left(\hat{\mathbf{x}}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right), \tag{13.46}$$

according to

$$\hat{x}_i = \frac{y_i \sigma_{yi}^{-1} + \mu_i \sigma_i^{-1}}{\sigma_{yi}^{-1} + \sigma_i^{-1}}. \tag{13.47}$$

Here, μ_i and σ_i denote the i'th component of the model mean and the square root of the i'th diagonal element of the covariance matrix for the currently evaluated state, respectively. σ_{yi} is the i'th component of the feature uncertainty and $\hat{x}$ is determined elementwise for each dimension $1 \leq i \leq D$.

The following simplifications are useful:

[2] CUDA AUDIOVISUAL SPEECH RECOGNIZER, this is the JASPER System with all log-likelihood computations done in CUDA, see [20].

$$\left(\frac{\hat{x}-\mu}{\sigma}\right)^2 = \frac{1}{\sigma^2}\left(\frac{\frac{y}{\sigma_y}+\frac{\mu}{\sigma}}{\frac{1}{\sigma_y}+\frac{1}{\sigma}}-\mu\right)^2 \tag{13.48}$$

$$= \frac{1}{\sigma^2}\left(\frac{\frac{y}{\sigma_y}+\frac{\mu}{\sigma}-\frac{\mu}{\sigma_y}-\frac{\mu}{\sigma}}{\frac{1}{\sigma_y}+\frac{1}{\sigma}}\right)^2 \tag{13.49}$$

$$= \frac{1}{\sigma^2}\left(\frac{\frac{y}{\sigma_y}-\frac{\mu}{\sigma_y}}{\frac{1}{\sigma_y}+\frac{1}{\sigma}}\right)^2 \tag{13.50}$$

$$= \frac{1}{\sigma^2}\left(\frac{y-\mu}{1+\frac{\sigma_y}{\sigma}}\right)^2 \tag{13.51}$$

$$= \frac{(y-\mu)^2}{(\sigma+\sigma_y)^2}. \tag{13.52}$$

This last expression can be used to compute the Mahalanobis distance more efficiently than with the original definition of modified imputation.

13.6.5 Binary Uncertainties

For binary uncertainties, only the reliable subvector of $\mathbf{y}$, composed of only the reliable components of $\mathbf{y}$ and denoted by $\mathbf{y}_r$, is used. If full covariances are needed, the considerations from Section 13.6.2 apply, otherwise, computation of a reduced inner product according to

$$(\tilde{s}, \boldsymbol{\mu}_r)\cdot(1,\mathbf{y}_r)^{\mathsf{T}}. \tag{13.53}$$

will give an implementation that is potentially suitable for utilizing the vector lanes of SIMD architectures. However, it is first necessary in each frame to recompute the scaling factor s via $\frac{1}{\sqrt{(2\pi)^d \det(\Sigma_q+\Sigma_y)}}$, with d as the number of reliable components in $\mathbf{y}_r$, and to obtain the reliable subvector of $\boldsymbol{\mu}$, a process which is hard to parallelize over time because the number of reliable components tends to change quickly.

13.6.6 Overview of Uncertain Recognition Techniques

An overview of all uncertainty-of-observation techniques is shown in Table 13.1. In the last line of Table 13.1, the binary uncertainty pdf is only computed for reliable features $y_i \in \mathbf{y}_r$ and is set to 1, otherwise.

Since these computations need to take place for each of the GMM mixture components and at each time, they also parallelize well in principle. An overview of possible architectures for exploiting this fact using graphics processors with Nvidia's

Table 13.1: Mahalanobis distance and likelihood computation for diagonal covariance models in the standard implementation, and using uncertainty decoding, modified imputation, and binary uncertainties.

Likelihood Evaluation	Mahalanobis Distance	Scaling Factor s
diagonal covariance	$\sqrt{\frac{(\mathbf{y}-\boldsymbol{\mu})^2}{\sigma^2}}$	s is unchanged an can be precomputed offline
uncertainty decoding	$\sqrt{\frac{(\mathbf{y}-\boldsymbol{\mu})^2}{\sigma^2+\sigma_y^2}}$	s changes and has to be updated
modified imputation	$\sqrt{\frac{(\mathbf{y}-\boldsymbol{\mu})^2}{(\sigma+\sigma_y)^2}}$	s is unchanged an can be precomputed offline
binary uncertainties	$\sqrt{\frac{(\mathbf{y}_r-\boldsymbol{\mu}_r)^2}{\sigma_r^2}}$	s changes and has to be updated

Compute Unified Device Architecture (CUDA) [34] are shown in [10] for audio-only, and in [20] for audiovisual speech recognition. A more detailed discussion of parallel kernels for missing feature probability computations is given in [19].

13.6.7 Recognition Results

The system has been tested in two setups. In the first, continuous-valued uncertainties were used. Here, as the audio feature extraction, the ETSI advanced front end was used, since its integrated Wiener filter yields the posterior error variance of spectrum domain features. This value is useful directly for modified imputation or uncertainty decoding, once it has been propagated to the MFCC domain used in the ETSI standard as described in Chapter 3. Results for this setup are shown in Table 13.2.

Table 13.2: Recognition results for clean video and additive babble noise.

SNR	-30 dB	-20 dB	-10 dB	0 dB	10 dB	20 dB	30 dB
Video only	83.1	83.1	83.1	83.1	83.1	83.1	83.1
Audio only	18.4	27.4	71.0	96.9	98.1	98.1	98.3
Conventional AVSR	84.1	85.6	92.1	97.8	**98.2**	**98.3**	**98.4**
AVSR + MI	84.4	**86.7**	**93.5**	**98.0**	**98.2**	**98.3**	**98.4**
AVSR + UD	**84.5**	86.5	93.1	**98.0**	**98.2**	**98.3**	98.3

As it can be seen, there are consistent, however rather small improvements in accuracy over the entire range of SNRs. In contrast, in the case when video data is also distorted by additive noise, the improvement due to uncertain recognition becomes more noticeable, in form of a much more graceful degradation of performance for low SNRs. Here, modified imputation outperforms uncertainty decoding, as will also be visible in the following set of results, given in Table 13.3.

Table 13.3: Recognition results for noisy video and additive babble noise.

SNR	-30 dB	-20 dB	-10 dB	0 dB	10 dB	20 dB	30 dB
Video only	44.1	44.1	44.1	44.1	44.1	44.1	44.1
Audio only	18.4	27.4	71.0	96.9	98.1	98.1	98.2
Conventional AVSR	45.5	48.0	77.7	97.6	**98.2**	**98.3**	**98.4**
AVSR + MI	45.6	**53.0**	**92.3**	**98.0**	**98.2**	**98.3**	**98.4**
AVSR + UD	**45.7**	52.7	92.1	97.9	98.1	**98.3**	**98.4**

Tables 13.4 and 13.5 show the results when no stream weight adaptation is carried out, but rather, the same stream weight is used over the entire range of SNRs.[3] Here, it is noticeable that is is still possible to retain a performance above that of the video model, as long as an SNR greater than about -20dB is given, indicating the usefulness of the audio stream even under severely distorted conditions. Also, on average the performance is clearly in excess compared to that of conventional AVSR for noisy video data, showing the helpfulness of missing data techniques for the integration of multiple uncertain streams.

However, since the use of stream adaptation still results in clearly best performance, it is likely that the best overall system design will rely on a combination of both optimizations, that of adaptive stream weighting and that of uncertain recognition.

Table 13.4: Recognition results for clean video and additive babble noise without stream weight tuning.

SNR	-30dB	-20dB	-10dB	0dB	10dB	20dB	30dB
Video only	**83.1**	**83.1**	83.1	83.1	83.1	83.1	83.1
Audio only	18.4	27.4	71.0	96.9	**98.1**	**98.1**	**98.3**
Conventional AVSR	30.0	50.5	86.7	95.5	96.6	96.6	96.9
AVSR + MI	26.6	51.8	**89.5**	**97.0**	97.6	98.0	98.1
AVSR + UD	32.5	56.2	89.4	95.7	96.6	96.8	97.0

Results for Videos with Missing Frames

For videos where the main source of error lies in missing frames or frames with misdetected mouths, binary uncertainties are the natural choice. In the following set

[3] For each of the methods, the stream weights were chosen for best average performance over the SNR range from -10dB to 30dB.

Table 13.5: Recognition results for noisy video and additive babble noise without stream weight tuning.

SNR	-30dB	-20dB	-10dB	0dB	10dB	20dB	30dB
Video only	**44.1**	44.1	44.1	44.1	44.1	44.1	44.1
Audio only	18.4	27.4	71.0	**96.9**	98.1	98.1	98.2
Conventional AVSR	19.9	42.7	66.3	92.6	98.0	98.1	98.0
AVSR + MI	23.1	**49.8**	**80.1**	96.1	**98.3**	**98.2**	**98.3**
AVSR + UD	21.1	45.0	70.9	94.4	98.1	98.1	98.1

of experiments, binary uncertainties have been used for the audio features as well, where they were estimated as described in Section 13.5.3.3.

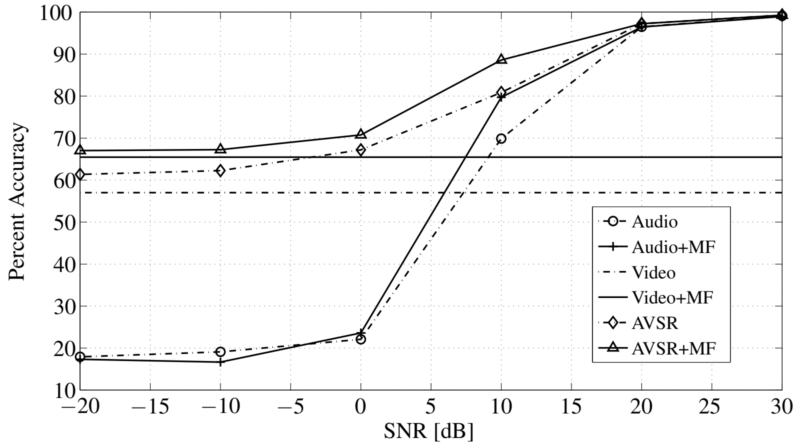

Fig. 13.6: Recognition results with purely binary uncertainties, applied both on the audio and video features.

In case of binary uncertainties, it can be seen that despite the very simple strategy, results of missing-feature based recognition clearly outperform conventional AVSR, especially in the area around 10 dB. This is a rather typical result from many such experiments, and it reflects the fact that missing data techniques are apparently most successful in aiding stream integration when both sources of information have roughly equivalent contributions to the final result.

13.7 Conclusion

The use of missing data techniques can help increase the robustness of audiovisual speech recognition. In all tested cases and scenarios, both using binary and continuous-valued uncertainties, the best performance was achieved with the help of missing data techniques. When the stream weights are adjusted appropriately, the performance is also, in all cases, lower-bounded by the best performance of either audio-only or video-only speech recognition.

Best performance gains are generally observed, when both audio and video signal have some mismatch to the model, otherwise, it can be seen that the second modality is very well able to compensate the shortcomings of the first, when the stream weights are chosen well. However, it can be argued that in most cases where it will be needed, audiovisual speech recognition is confronted with both types of mismatch in general, since video features are not yet as independent to changes in environment and speaker as it would be desirable, and since distorted audio conditions will be the major reason for including the computationally expensive visual modality.

For future work, it will be interesting to use missing data techniques in conjunction with model adaptation, which has not been tried in this context but has been shown a successful strategy for dealing with reverberant data in Chapter 9 of this book.

In order for audiovisual speech recognition to remain computationally feasible, it is also important to consider implementation strategies which make use of the fast-growing potential of parallel processing. This is another target of future work, to make real-time human-machine interaction with large vocabularies feasible also with the loose stream integration that CHMMs can offer in conjunction with the missing data techniques discussed here.

References

1. Ahmad, N., Datta, S., Mulvaney, D., Farooq, O.: A comparison of visual features for audiovisual automatic speech recognition. In: Acoustics 2008, Paris, pp. 6445–6448 (2008). DOI 10.1121/1.2936016
2. Aleksic, P.S., Williams, J.J., Wu, Z., Katsaggelos, A.K.: Audio-visual speech recognition using MPEG-4 compliant visual features. EURASIP Journal on Applied Signal Processing **11**, 1213–1227 (2002)
3. Astudillo, R.F., Kolossa, D., Orglmeister, R.: Uncertainty propagation for speech recognition using RASTA features in highly nonstationary noisy environments. In: Proc. ITG (2008)
4. Astudillo, R.F., Kolossa, D., Orglmeister, R.: Accounting for the uncertainty of speech estimates in the complex domain for minimum mean square error speech enhancement. In: Proc. Interspeech (2009)
5. Barker, J., Green, P., Cooke, M.: Linking auditory scene analysis and robust ASR by missing data techniques. In: Proceedings WISP 2001 (2001)
6. Barron, J., Fleet, D., Beauchemin, S.: Performance of optical flow techniques. International Journal of Computer Vision **92**, 236–242 (1994). URL citeseer.ist.psu.edu/barron92performance.html
7. Cetingl, H., Yemez, Y., Erzin, E., Tekalp, A.: Discriminative analysis of lip motion features for speaker identification and speech-reading. Image Processing, IEEE Transactions on **15**(10), 2879–2891 (2006). DOI 10.1109/TIP.2006.877528
8. Cooke, M., Barker, J., Cunningham, S., Shao, X.: An audio-visual corpus for speech perception and automatic speech recognition. Acoustical Society of America Journal **120**, 2421–2424 (2006). DOI 10.1121/1.2229005
9. Deng, L., Droppo, J., Acero, A.: Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion. IEEE Trans. Speech and Audio Processing **13**(3), 412–421 (2005)
10. Dixon, P.R., Oonishi, T., Furui, S.: Harnessing graphics processors for the fast computation of acoustic likelihoods in speech recognition. Comput. Speech Lang. **23**(4), 510–526 (2009). DOI http://dx.doi.org/10.1016/j.csl.2009.03.005
11. Ellis, D.P.W.: PLP and RASTA (and MFCC, and inversion) in Matlab. http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/ (2005). Online web resource, last checked: 01 July 2010
12. ETSI: Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech recognition; Front-end feature extraction algorithm; Compression algorithms, ETSI ES 202 050 v1.1.5 (2007-01) (January 2007)
13. Gejguš, P., Šperka, M.: Face tracking in color video sequences. In: SCCG '03: Proceedings of the 19th spring conference on Computer graphics, pp. 245–249. ACM, New York, NY, USA (2003). DOI http://doi.acm.org/10.1145/984952.984992
14. Goecke, R.: A stereo vision lip tracking algorithm and subsequent statistical analyses of the audio-video correlation in australian english. Ph.D. thesis, Australian National University, Canberra, Australia (2004). URL citeseer.ist.psu.edu/goecke04stereo.html
15. Gowdy, J., Subramanya, A., Bartels, C., Bilmes, J.: DBN based multi-stream models for audio-visual speech recognition. In: Proc. ICASSP, vol. 1, pp. I–993–6 vol.1 (2004). DOI 10.1109/ICASSP.2004.1326155
16. Hermansky, H., Morgan, N.: Rasta processing of speech. Speech and Audio Processing, IEEE Transactions on **2**(4), 578–589 (1994). DOI 10.1109/89.326616
17. Hsu, R.L., Abdel-Mottaleb, M., Jain, A.K.: Face detection in color images. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**, 696–706 (2002)
18. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision **1**(4), 321–331 (1988). URL http://www.springerlink.com/content/q7g93335q86604x6/fulltext.pdf
19. Kolossa, D., Astudillo, R.F., Zeiler, S., Vorwerk, A., Lerch, D., Chong, J., Orglmeister, R.: Missing feature audiovisual speech recognition under real-time constraints. Accepted for publication in ITG Fachtagung Sprachkommunikation (2010)

20. Kolossa, D., Chong, J., Zeiler, S., Keutzer, K.: Efficient manycore CHMM speech recognition for audiovisual and multistream data. Accepted for publication in Proc. Interspeech (2010)
21. Kolossa, D., Klimas, A., Orglmeister, R.: Separation and robust recognition of noisy, convolutive speech mixtures using time-frequency masking and missing data techniques. In: Proc. WASPAA, pp. 82–85 (2005). DOI 10.1109/ASPAA.2005.1540174
22. Kratt, J., Metze, F., Stiefelhagen, R., Waibel, A.: Large vocabulary audio-visual speech recognition using the Janus speech recognition toolkit. In: DAGM-Symposium, pp. 488–495 (2004)
23. Lan, Y., Theobald, B.J., Ong, E.J., Bowden, R.: Comparing visual features for lipreading. In: Int. Conf. on Auditory-Visual Speech Processing (AVSP2009). Norwich, UK (2009)
24. Lerch, D.: Audiovisuelle Spracherkennung unter Berücksichtigung der Unsicherheit von visuellen Merkmalen. Diploma thesis, TU Berlin, dennis_lerch@gmx.de (2009)
25. Lewis, T.W., Powers, D.M.W.: Lip feature extraction using red exclusion. In: VIP '00: Selected papers from the Pan-Sydney workshop on Visualisation, pp. 61–67. Australian Computer Society, Inc., Darlinghurst, Australia, Australia (2001)
26. Lucey, P.J., Dean, D.B., Sridharan, S.: Problems associated with current area-based visual speech feature extraction techniques. In: AVSP 2005, pp. 73–78 (2005). URL `http://eprints.qut.edu.au/12847/`
27. Luettin, J., Potamianos, G., Neti, C.: Asynchronous stream modelling for large vocabulary audio-visual speech recognition. In: Proc. ICASSP (2001)
28. Mase, K., Pentland, A.: Automatic lip-reading by optical flow analysis. Trans. Systems and Computers in Japan **22**, 67–76 (1991)
29. Matthews, I., Cootes, T., Bangham, J., Cox, S., Harvey, R.: Extraction of visual features for lipreading. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(2), 198–213 (2002). DOI 10.1109/34.982900
30. Metze, F.: Articulatory features for conversational speech recognition. Ph.D. thesis, Universität Fridericiana zu Karlsruhe (2005)
31. Naseem, I., Deriche, M.: Robust human face detection in complex color images. IEEE International Conference on Image Processing, ICIP 2005. **2**, 338–341 (2005). DOI 10.1109/ICIP.2005.1530061
32. Nefian, A., Liang, L., Pi, X., Liu, X., Murphy, K.: Dynamic bayesian networks for audio-visual speech recognition. EURASIP Journal on Applied Signal Processing **11**, 1274–1288 (2002)
33. Neti, C., Potamianos, G., Luettin, J., Matthews, I., Glotin, H., Vergyri, D., Sison, J., Mashari, A., Zhou, J.: Audio-visual speech recognition. Tech. Rep. WS00AVSR, Johns Hopkins University, CLSP (2000). URL `citeseer.ist.psu.edu/neti00audiovisual.html`
34. NVIDIA Corporation: NVIDIA CUDA Compute Unified Device Architecture Programming Guide (2007)
35. Otsuki, T., Ohtomo, T.: Automatic lipreading of station names using optical flow and HMM. Technical report of IEICE. HIP **102**(473), 25–30 (2002). URL `http://ci.nii.ac.jp/naid/110003271904/en/`
36. Papandreou, G., Katsamanis, A., Pitsikalis, V., Maragos, P.: Adaptive multimodal fusion by uncertainty compensation with application to audiovisual speech recognition. Audio, Speech, and Language Processing, IEEE Trans. **17**(3), 423–435 (2009). DOI 10.1109/TASL.2008.2011515
37. Patterson, E.K., Gurbuz, S., Tufekci, Z., Gowdy, J.N.: Moving-talker, speaker-independent feature study, and baseline results using the CUAVE multimodal speech corpus. EURASIP J. Appl. Signal Process. **2002**(1), 1189–1201 (2002). DOI http://dx.doi.org/10.1155/S1110865702206101
38. Potamianos, G., Neti, C., Luettin, J., Matthews, I.: Audio-visual automatic speech recognition: An overview. In: E. Vatikiotis-Bateson, P. Perrier (eds.) Issues in Visual and Audio-Visual Speech Processing. MIT Press (2004)
39. Raj, B., Stern, R.: Missing-feature approaches in speech recognition. Signal Processing Magazine, IEEE **22**(5), 101–116 (2005)

40. Schwerdt, K., Crowley, J.L.: Robust face tracking using color. In: Proc. of 4th International Conference on Automatic Face and Gesture Recognition, pp. 90–95. Grenoble, France (2000). URL `citeseer.ist.psu.edu/schwerdt00robust.html`
41. Shdaifat, I., Grigat, R.R., Lütgert, S.: Recognition of the german visemes using multiple feature matching. In: B. Radig, S. Florczyk (eds.) Lecture Notes in Computer Science, Pattern Recognition, vol. 2191/2001, pp. 437–442. Springer-Verlag Berlin Heidelberg (2001)
42. Tamura, S., Iwano, K., Furui, S.: A robust multi-modal speech recognition method using optical-flow analysis. In: Multi-Modal Dialogue in Mobile Environments, ISCA Tutorial and Research Workshop (ITRW). ISCA, Kloster Irsee, Germany (2002)
43. Vezhnevets, V., Sazonov, V., Andreeva, A.: A survey on pixel-based skin color detection techniques. In: Proc. Graphicon, pp. 85–92. Moscow, Russia (2003). URL `citeseer.ist.psu.edu/vezhnevets03survey.html`
44. Wang, X., Hao, Y., Fu, D., Yuan, C.: ROI processing for visual features extraction in lipreading. In: 2008 International Conference on Neural Networks and Signal Processing, pp. 178–181. IEEE (2008)
45. Yang, M.H., Kriegman, D.J., Ahuja, N.: Detecting faces in images: A survey. IEEE Transactions on pattern analysis and machine intelligence **24**(1), 34–58 (2002)
46. Young, S., Russell, N., Thornton, J.: Token passing: a simple conceptual model for connected speech recognition systems. Tech. Rep. CUED/FINFENG /TR.38, Cambridge University Engineering Department (1989)