INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Turbo Decoders for Audio-visual Continuous Speech Recognition

Ahmed Hussen Abdelaziz

International Computer Science Institute, Berkeley, USA
ahmedha@icsi.berkeley.edu

Abstract

Visual speech, i.e., video recordings of speakers' mouths, plays an important role in improving the robustness properties of automatic speech recognition (ASR) against noise. Optimal fusion of audio and video modalities is still one of the major challenges that attracts significant interest in the realm of audio-visual ASR. Recently, turbo decoders (TDs) have been successful in addressing the audio-visual fusion problem. The idea of the TD framework is to iteratively exchange some kind of soft information between the audio and video decoders until convergence. The forward-backward algorithm (FBA) is mostly applied to the decoding graphs to estimate this soft information. Applying the FBA to the complex graphs that are usually used in large vocabulary tasks may be computationally expensive. In this paper, I propose to apply the forward-backward algorithm to a lattice of most likely state sequences instead of using the entire decoding graph. Using lattices allows for TD to be easily applied to large vocabulary tasks. The proposed approach is evaluated using the newly released TCD-TIMIT corpus, where a standard recipe for large vocabulary ASR is employed. The modified TD performs significantly better than the feature and decision fusion models in all clean and noisy test conditions.

Index Terms: Turbo decoding, audio-visual speech recognition, audio-visual fusion, noise-robustness, ASR

1. Introduction

The performance of modern automatic speech recognition (ASR) systems is very accurate in quite surroundings. However, in uncontrolled varying listening environments, the distortion occurs to the acoustic signals severely affect the accuracy of ASR. With the increasing use of multimedia data in communication technologies, a non-traditional approach to enhance the performance of ASR systems in noisy environments comes from the fact that speech is bi-modal (audio-visual) [1]. In noisy environments, humans start to read the movement of the speaker's lips and combine this information with the speech signals to enhance their intelligibility of speech. Inspired by this phenomenon, the idea of employing visual information in ASR has been adopted in many studies.

One of the most challenging tasks in the realm of audio-visual ASR (AV-ASR) is the fusion of the audio and video modalities [2–5]. Recently, turbo decoders (TD) have been very successful in solving the audio-visual fusion task [6–8]. Turbo codes trace their roots back to digital communication, where they were originally proposed for forward-error correction [9]. The idea of using TD as an audio-visual fusion model is to iteratively exchange soft –also called extrinsic– information between the single modality decoders until convergence. The definition of this soft information varies in literatures. In [6], the *a posteriori* probabilities were used as the exchanged information, where an altered forward-backward algorithm (FBA) was employed to find these probabilities. In [7], the authors reported better results by using a modified version of the FBA-based state posteriors. Viterbi-based soft output scores were also employed in [7] as the exchanged extrinsic information.

Most of the TD-based AV-ASR results reported in [6–8] have been obtained using the GRID audio-visual corpus, which is a small vocabulary task with a fixed grammar. It is, however, known that approaches tested on small vocabulary tasks may be computationally not feasible and do not always work similarly when explored for large vocabulary continuous speech recognition (LVCSR) tasks [10]. For example, the estimation of the extrinsic information in the TD framework may be computationally feasible when applied to the Grid corpus. However, for large vocabulary tasks, where a more complex graph that includes context-dependent acoustic models and a language model are used, the computation of the extrinsic information may be computationally too expensive.

In this paper, an approximation to the FBA-based extrinsic information in [7] is proposed. The idea is to apply the FBA to a lattice of the most-likely utterances instead of using the entire decoding graph to compute the modified posterior probabilities, i.e., the extrinsic information. The lattices are generated in each iteration using modified likelihoods. These modified likelihoods are computed as weighted exponents of the likelihoods of a single-modality recognizer, e.g., audio ASR, and the extrinsic information from the other parallel recognizer.

Due to the lack of publicly available audio-visual LVCSR corpora, which is a problem commonly reported in literatures, the proposed approach is evaluated using the newly published TCD-TIMIT audio-visual continuous speech corpus. Despite the small vocabulary nature of the TCD-TIMIT phone recognition task, applying standard LVCSR approaches to such a corpus can give an insight into the performance of the proposed approach when tested on audio-visual LVCSR tasks [10].

The remaining paper is organized as follows: After a brief summary of the TD-based audio-visual fusion model in Section 2, the proposed approximation of the extrinsic information is introduced in Section 3. In Section 4, the proposed approach is evaluated using clean and distorted versions of the newly published TCD-TIMIT audio-visual corpus. Finally, the paper is concluded in Section 5.

2. Turbo Decoding

To set the stage, the state-conditional likelihoods $\mathrm{p}(\mathbf{o}_s(t)|q_s)$ of a feature vector $\mathbf{o}_s$ at time frame t given a state q_s in a single-modality recognizer are denoted by $b_{q_s}(t)$, where s can be an audio (a) or a video (v) stream. The posterior probability $\mathrm{p}(q_s(t)|\mathbf{O}_s)$ of a state q_s at time frame t given a sequence of T feature vectors $\mathbf{O}_s = \{\mathbf{o}_s(t)\}_{t=1}^{T}$ is denoted by $\gamma_t(q_s)$. The soft (extrinsic) information exchanged by the audio and video recognizers is denoted by $\dot{\gamma}_t(q_s)$. For Q_s states of a single-modality recognizer, the vector notation of the extrinsic information $\dot{\boldsymbol{\gamma}}_{s_t} = \{\dot{\gamma}_t(q_s)\}_{q_s=1}^{Q_s}$ is used. Similarly, the vector notation of other quantities is denoted by a bold font. For the sake of simplicity, the time-frame index t will be dropped.

http://dx.doi.org/10.21437/Interspeech.2017-799

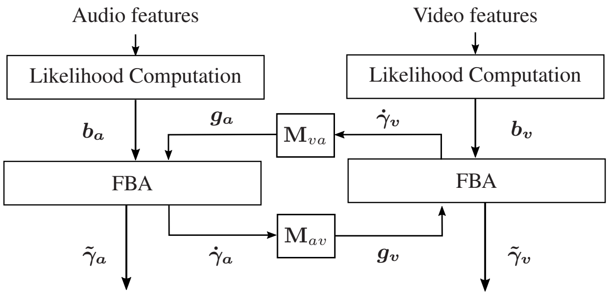

Figure 1: *Turbo decoder framework for audio-visual fusion.*

If the number of states of the audio and video recognizers are not equal, the extrinsic information from one recognizer needs to be linearly mapped via

$$\mathbf{g}_v = \mathbf{M}_{av}\,\dot{\boldsymbol{\gamma}}_a \text{ and } \mathbf{g}_a = \mathbf{M}_{va}\,\dot{\boldsymbol{\gamma}}_v \tag{1}$$

before it can be deployed in the parallel recognizer. The matrices $\mathbf{M}_{av}$ and $\mathbf{M}_{va}$ linearly transform the extrinsic information from the audio to the video state space and vice versa. In this study, same state spaces for the audio and video modalities are used. Thus, $\mathbf{M}_{av}$ and $\mathbf{M}_{va}$ are the identity matrix.

Figure 1 summarizes the basic concept of TD for audio-visual fusion. In each iteration, the state-conditional likelihoods b_{q_s} of a single-modality recognizer are modified using the extrinsic information g_{q_s} from the parallel recognizer. The modified pseudo likelihoods $\tilde{b}_{q_s}$ are estimated via

$$\tilde{b}_{q_a} = b_{q_a}\ (g_{q_a})^{\lambda_v} \text{ and } \tilde{b}_{q_v} = b_{q_v}\ (g_{q_v})^{\lambda_a}. \tag{2}$$

As can be seen in Equation (2) , the relative contribution of the extrinsic information is controlled by the stream exponent λ_s. After modifying the likelihoods using the extrinsic information, the FBA is employed using the pseudo likelihoods to estimate modified state posteriors $\tilde{\gamma}(q_s)$. Finally, decoding is achieved by finding the most likely state $\hat{q}_s$ at each time frame via

$$\hat{q}_s = \underset{q_s \in Q_s}{\arg\max}\{\tilde{\gamma}(q_s)\}. \tag{3}$$

The extrinsic information $\dot{\gamma}(q_s)$ can be estimated by removing the pseudo likelihoods from the modified posteriors via

$$\dot{\gamma}(q_s) \propto \frac{\tilde{\gamma}(q_s)}{\tilde{b}_{q_s}}. \tag{4}$$

As discussed in [7], removing the pseudo likelihoods as in (4) can avoid re-using the information and prevent overemphasizing the likelihoods in each iteration. Equation (4) defines the extrinsic information $\dot{\gamma}(q_s)$ as a prior probability that is estimated by one decoder and provided to the other decoder to help finding the most likely state sequence. Note that the extrinsic information in (4) still needs to be normalized in order to ensure the stochastic constraint.

3. Turbo Decoding for LVCSR

In order to estimate the extrinsic information as shown in (4), the posterior probability $\tilde{\gamma}(q_s)$ should be first estimated using the FBA. Since applying the FBA to the complex graphs usually used in LVCSR systems is computationally too expensive, a simple modification to the TD framework is proposed in Figure 2. As can be seen, the modified likelihood $\tilde{b}_{q_s}$ are first employed to find a simpler graph, i.e., a lattice that contains the most likely state sequences. The FBA can then be applied to the generated lattice to find approximated state posteriors $\tilde{\gamma}(q_s)$. Finally, the extrinsic information can be estimated in a usual manner as in (4) using the approximated posteriors.

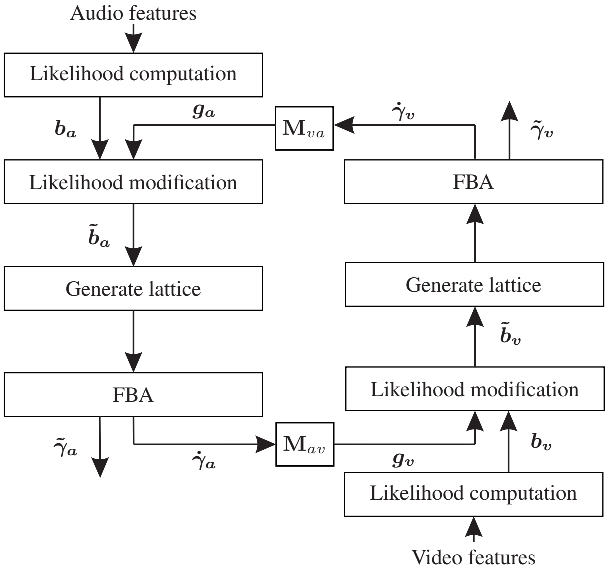
Audio features
Likelihood computation
$\boldsymbol{b_a}$
$\boldsymbol{g_a}$
M_{va}
$\dot{\boldsymbol{\gamma}}_{\boldsymbol{v}}$
$\tilde{\boldsymbol{\gamma}}_{\boldsymbol{v}}$
Likelihood modification
FBA
$\tilde{\boldsymbol{b}}_{\boldsymbol{a}}$
Generate lattice
Generate lattice
$\tilde{\boldsymbol{b}}_{\boldsymbol{v}}$
FBA
Likelihood modification
$\tilde{\boldsymbol{\gamma}}_{\boldsymbol{a}}$
M_{av}
$\boldsymbol{b_v}$
$\dot{\boldsymbol{\gamma}}_{\boldsymbol{a}}$
$\boldsymbol{g_v}$
Likelihood computation
Video features

Figure 2: *Modified turbo decoder for AV-LVCSR.*

3.1. Implementation Issues

It has been noticed that the modified posterior probabilities $\tilde{\gamma}(q_s)$ are sparse and this leads to sparse pseudo likelihoods $\tilde{b}_{q_s}$. To avoid numerical errors caused by estimating the extrinsic information using (4), both $\tilde{\gamma}(q_s)$ and $\tilde{b}_{q_s}$ need to be floored and $\tilde{\gamma}(q_s)$ needs to be re-normalized.

The linearly-transformed extrinsic information g_{q_s} has a very different dynamic range compared to the likelihoods $b(q_s)$. In order for the pseudo likelihoods $\tilde{b}(q_s)$ in (2) not to be dominated by g_{q_s} or $\tilde{b}(q_s)$, a fixed exponent λ_p needs to be applied to g_{q_s} to balance its dynamic range with respect to $\tilde{b}(q_s)$ [8].

The stream weights λ_s have been chosen using a grid search with a minimum phone error rate (PER) criterion. The grid search is conducted in each iteration using a development set. The PER of the development set is also used to determine the convergence of the framework. The optimal stream weight in each iteration as well as the number of iterations until convergence are deployed during testing. A weighting scheme similar to the one proposed in [7] has also been tried. However, no significant differences in the PERs have been noticed.

Very similar results as the ones introduced in Section 4 have been obtained when the modified likelihoods $\tilde{\boldsymbol{b}}_s$ from one iteration are re-modified in the following iteration instead of the original likelihoods $\boldsymbol{b}_s$. Moreover, using $\tilde{\boldsymbol{b}}_s$ instead of $\boldsymbol{b}_s$ has shown very good convergence properties, namely, the TD framework converges faster to more stable results.

4. Experiments and Results

4.1. Dataset

The TCD-TIMIT audio-visual continuous speech corpus [11] has been used for evaluation. TCD-TIMIT is a free audio-visual version of the well known TIMIT database [12]. The corpus contains audio and video recordings of 56 speakers with an Irish accent. It also contains utterances spoken by 3 speakers with non-Irish accents and 3 professional lip speakers. However, utterances from those 6 speakers are not used here. Each speaker utters 98 TIMIT sentences. Like the TIMIT corpus, the task of the TCD-TIMIT is phone recognition in continuous speech.

In addition to a 16 kHz down-sampled version of the clean TCD-TIMIT utterances, 36 noisy versions have been created. Six noise types have been used: White, babble, and car noise from the RSG-10 database [13], living room noise from the second CHiME challenge [14], street and cafe noise from the third CHiME challenge [15]. The noise signals have been added to the clean utterances at 6 signal-to-noise ratios (SNRs) from 20 dB down to -5 dB. The noise and speech signals have been mixed at the appropriate SNRs using the filtering and noise adding tool (FaNT) [16] that has been used to create the Aurora-2 [17] and Aurora-4 [18] databases.

The TCD-TIMIT corpus is originally divided into a 70% - 30% training-testing split, i.e., 39 speakers for the training set (almost 5 hours) and 17 speakers for the test set. Here, the test set is further split into an 8-speaker development set (almost 1 hour) and a 9-speaker test set (1.12 hours).

4.2. Experimental Setup

4.2.1. Features

The 13-dimensional mel-frequency cepstral coefficients (MFCC) with their first and second derivatives have been initially used to train a Gaussian mixture model (GMM)/HMM ASR system. This system has then been used to find a linear discriminant analysis (LDA) [19] and a feature-space maximum likelihood linear regression (fMLLR) [20] transformation matrices. The dimension of the acoustic feature vectors becomes 40 after multiplying the MFCC feature vectors by the LDA matrix and then by the fMMLR matrix. Finally, 11 consecutive (5 previous, current, and 5 future) fMLLR frames have been concatenated to form the 440-dimensional feature vectors used for training the audio-only DNN/HMM hybrid system. The Kaldi speech recognition toolkit [21] has been used for acoustic feature extraction.

Appearance-based visual features have been extracted as follows: Firstly, Viola-Jones algorithm [22] has been used to detect the speakers' faces in the first frame. Next, interest points have been determined using the minimum eigenvalue algorithm [23]. The interest points have then been tracked from a video frame to another using the Kanade-Lucas-Tomasi (KLT) algorithm [24]. The interest points have been updated using the Viola-Jones and minimum eigenvalue algorithm with a refresh rate of 40 frames. Matlab's Computer Vision System Toolbox™ has been mainly used for the face detection algorithm.

From the detected face, the region of interest (ROI), which is a 67×67 bounding box of pixels circumferencing the mouth region, have been extracted using Viola-Jones algorithm. The ROI is then normalized and rotated. The difference between the acoustic frame rate (100 frame/s) and the visual frame rate (30 frame/s) is compensated by repeating visual frames according to the digital differential analyzer (DDA) algorithm [25].

Finally, the actual visual features have been extracted by applying the discrete cosine transform (DCT), and the principal component analysis (PCA). The dimension of the PCA feature vectors, which has been chosen based on empirical results, is 32. Similarly to the acoustic feature extraction, the first and second derivatives have been concatenated and LDA and fMLLR have been applied. The dimension of the visual feature vectors after applying the linear transformations is 40. Finally, 11 consecutive visual frames have been cascaded to form the final 440-dimensional visual feature vectors.

4.2.2. Models

A standard recipe for training LVCSR systems has been used to train speaker-independent acoustic DNN/HMM hybrid models. Firstly, GMM/HMM models have been trained, where 3-states HMMs have been used for modeling acoustic and visual representations of context-dependent tri-phones. The forced alignment algorithm has then been applied to estimate the frame-state alignments that are used as the DNN's training targets.

The DNNs have 6 hidden layers, each of which consists of 1024 neurons with sigmoid activations. The number of units in the input layer equals the feature vector dimension, i.e., 440. The number of units in the output softmax layer is 1953, which is the number of the tied tri-phone states (senones) [26]. The DNN weights have been tuned using stochastic gradient descent and back propagation algorithms so that the sequential minimum bias risk (sMBR) objective function [27] is minimized.

Visual DNN/HMM hybrid models have been trained similarly to the acoustic ones. However, the frame-state alignments used for training the initial visual mono-phone GMM/HMMs and the final visual DNN/HMMs have been obtained from the corresponding acoustic models.

A simple bi-phone language model has been used for decoding. The phone-level transcription of the training set has been used for training the bi-phone language model. The LVCSR training procedure described above is, however, very flexible and any complex language model can be used instead of the bi-phone model.

4.2.3. Feature and Decision Fusion Schemes

The performance of the modified TD framework has been compared to two well known audio-visual fusion schemes. The first one is the feature fusion (FF) scheme, also called direct or early integration. The second fusion model is the decision fusion (DF), also called separate or late integration.

In the FF scheme, fusion is applied at the feature level. The audio and video feature vectors are concatenated to form new audio-visual feature vectors. The new feature vectors are then used to train an AV-ASR system. In this type of fusion, both the audio and video features are assumed to have the same utility and reliability at each time frame.

In the DF models, the audio and video streams are first recognized separately. The recognition results are then combined using a voting scheme. Unlike FF, DF can control the contribution of the audio and video streams to the overall results according to their reliabilities using stream confidence measures. DF, on the other hand, does not account for the natural temporal dependencies between the audio and video streams like the FF.

In this study, ROVER (Recognition Output Voting Error Reduction) [28] has been used to combine the recognition results of the audio-only and video-only recognizers. All hyper parameters required for ROVER have been tuned for each noise condition using the corresponding development sets.

4.3. Results

The performance in all experiments has been evaluated in terms of PER. The test set PER of the speaker-independent video-only ASR is 65.4. Table 1 shows the results of the audio-only ASR system in all noise conditions. The results are obtained from a clean-train-noisy-test setup. This explains the massive increase

Table 1: *PERs of the test set in clean and noisy test conditions obtained using an audio-only ASR system.*

	SNR [dB]	Clean	20	15	10	5	0	-5	Average
Noise Type	Car	21.6	28.7	35.2	44.6	56.2	69.4	80.4	48.0
	White	21.6	40.9	53.1	66.7	77.7	86.7	91.4	62.6
	Babble	21.6	42.2	54.6	68.3	79.2	88.8	92.7	63.9
	L. Room	21.6	47.0	60.3	72.6	81.5	87.7	91.1	66.0
	Street	21.6	50.7	63.2	74.8	84.5	90.8	93.6	68.5
	Cafe	21.6	62.6	69.9	75.2	81.4	89.0	94.5	70.6
	Average	21.6	45.4	56.0	67.0	76.8	85.4	90.6	63.3

Table 2: *PER of the test set obtained using an audio-visual ASR system with a feature fusion scheme.*

	SNR [dB]	Clean	20	15	10	5	0	-5	Average
Noise Type	Car	20.4	25.1	29.6	37.5	48.8	63.3	76.2	43.0
	White	20.4	34.4	45.9	58.8	72.2	83.1	89.9	57.8
	Babble	20.4	35.3	46.4	60.1	73.0	84.1	89.3	58.4
	L. Room	20.4	39.6	51.3	64.1	75.4	84.4	89.2	60.6
	Street	20.4	42.0	54.6	67.3	79.2	87.5	92.3	63.3
	Cafe	20.4	51.3	59.6	67.3	76.0	85.4	92.8	64.7
	Average	20.4	37.9	47.9	59.2	70.8	81.3	88.3	58.0

Table 3: *PER of the test set obtained using an audio-visual ASR system with a decision fusion scheme.*

	SNR [dB]	Clean	20	15	10	5	0	-5	Average
Noise Type	Car	22.5	29.4	35.8	45.1	57.7	65.5	65.6	45.9
	White	22.5	41.4	53.3	65.5	65.6	65.7	65.6	54.2
	Babble	22.5	42.8	54.9	65.5	65.7	65.8	65.6	54.7
	L. Room	22.5	47.4	60.8	65.5	65.6	65.6	65.6	56.1
	Street	22.5	51.0	63.1	65.6	65.7	65.8	65.7	57.1
	Cafe	22.5	62.5	65.4	65.5	65.6	65.7	65.6	59.0
	Average	22.5	45.8	55.6	62.1	64.3	65.7	65.6	54.5

Table 4: *PER of the test set obtained using an audio-visual ASR system with a TD fusion scheme.*

	SNR [dB]	Clean	20	15	10	5	0	-5	Average
Noise Type	Car	19.2	24.6	29.6	42.9	47.4	57.2	62.2	40.4
	L. Room	19.2	37.0	46.0	54.1	60.4	63.5	63.3	49.1
	Babble	19.2	34.2	44.2	55.0	64.0	62.7	64.3	49.1
	Cafe	19.2	38.7	49.8	58.8	63.4	63.5	64.3	51.1
	Street	19.2	41.7	51.8	58.9	62.5	63.9	64.5	51.8
	White	19.2	49.9	55.5	61.2	63.6	63.4	64.3	53.9
	Average	19.2	37.7	46.2	55.2	60.2	62.4	63.8	49.2

in the PERs in all noisy environments. Tables 2-4 show the test set PERs of the FF-, DF-, and TD-based AV-ASR system in all noisy conditions. Although the video-only ASR results is very high, the performance of all AV-ASR systems shown in Tables 2-4 is consistently better than the audio-only ASR system in every test condition. Comparing the AV-ASR results, the approximated TD framework outperforms the FF and DF schemes with average relative PER reduction of 15% and 10%, respectively.

Figure 3 compares the evolution of the audio-only, video-only, and all audio-visual ASR systems with respect to the SNR. The PER results in Figure 3 are the average over the test conditions, i.e., last row in Tables 1-4. As can be seen, the performance of the FF-based AV-ASR system is better than the best performance of the audio-only and video-only ASR systems in high SNRs. In low SNRs, however, FF-based AV-ASR is outperformed by the video-only ASR system. Figure 3 shows also that the performance of the DF-based AV-ASR system follows the best performance of the audio-only and video-only ASR systems if the difference between their performances is large. It only becomes slightly better than both the audio-only and video-only ASR systems at 10 dB when their performances become comparable. Even with the approximation introduced to the TD framework in this study, the TD-based AV-ASR performance has the advantage of the FF at high SNRs and similar behavior of the DF models at low SNRs, where it is always better than the best performance in every test condition.

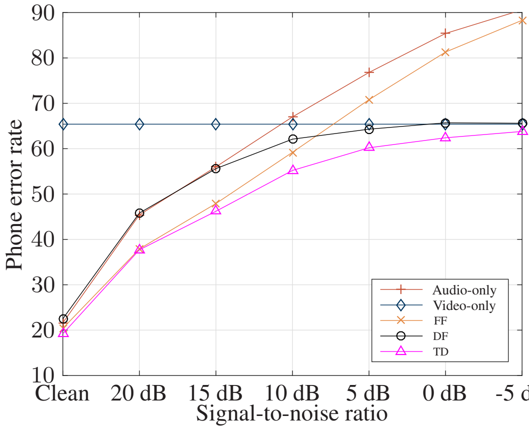

Figure 3: *Test set phone error rate of the audio-only ASR, video-only ASR, FF-based AV-ASR, DF-based AV-ASR, and TD-based AV-ASR systems averaged over the test conditions.*

5. Conclusions

Audio-visual fusion can be conducted using the framework of turbo decoders, where soft information is exchanged between the audio and video decoders until both decoders agree on the decoding results, i.e., until convergence. The estimation of the soft (extrinsic) information typically includes the application of the forward-backward algorithms (FBA) or the soft output Viterbi algorithm to the decoding graphs. Despite being applicable in the context of small vocabulary tasks, it may be computationally too expensive to apply these algorithms to complex graphs usually used in large vocabulary tasks, where context-dependency and language models are included. In this paper, a modified turbo decoding framework for audio-visual fusion has been introduced. The modified framework allows for turbo decoders to be applied to audio-visual large vocabulary automatic speech recognition tasks in a straightforward way. Instead of applying the FBA to the entire decoding graph to estimate the extrinsic information, it can be simply applied to a lattice of the most-likely state sequences. The reduced state space of the lattices makes the estimation of the extrinsic information using the FBA computationally feasible. Because of the lack of publicly available audio-visual LVCSR corpora, the modified TD has been evaluated using the newly released TCD-TIMIT audio-visual continuous speech corpus. However, a typical LVCSR recipe has been applied to the TCD-TIMIT to get an insight into the performance of the proposed approach when applied to audio-visual LVCSR corpora. The approximated TD has outperformed the feature fusion and decision fusion approaches in all clean and noisy conditions.

6. Acknowledgments

This work was supported by the Federal Ministry of Education and Research (BMBF) through the Fit World Wide “FITweltweit” program, administered by the German Academic Exchange Service (DAAD).

The author would like to express his appreciation to Steffen Zeiler for his many useful inputs and valuable comments.

7. References

[1] S. Ouni, M. M. Cohen, H. Ishak, and D. W. Massaro, “Visual contribution to speech perception: Measuring the intelligibility of animated talking heads,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2007, no. 1, 2007.

[2] A. Nefian, L. Liang, X. Pi, X. Liu, and K. Murphy, “Dynamic Bayesian networks for audio-visual speech recognition,” *EURASIP Journal on Applied Signal Processing*, vol. 11, pp. 1274–1288, 2002.

[3] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. W. Senior, “Recent advances in the automatic recognition of audiovisual speech,” *Proceedings of the IEEE*, vol. 91, no. 9, pp. 1306–1326, 2003.

[4] A. H. Abdelaziz, S. Zeiler, and D. Kolossa, “Learning dynamic stream weights for coupled-HMM-based audio-visual speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 5, pp. 863–876, 2015.

[5] A. K. Katsaggelos, S. Bahaadini, and R. Molina, “Audiovisual fusion: Challenges and new approaches,” *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1635–1653, 2015.

[6] B. R. S. Shivappa and M. Trivedi, “Multimodal information fusion using the iterative decoding algorithm and its application to audio-visual speech recognition,” in *Proc. ICASSP*, 2008, pp. 2241–2244.

[7] R. W. S. Receveur and T. Fingscheidt, “Turbo automatic speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 24, pp. 846–862, 2016.

[8] S. Gergen, S. Zeiler, A. H. Abdelaziz, R. Nickel, and D. Kolossa, “Dynamic stream weighting for turbo-decoding-based audiovisual asr,” in *Proc. Interspeech*, 2016, pp. 2241–2244.

[9] A. G. C. Berrou and P. Thitimajshima, “Near shannon limit error-correcting coding and decoding: Turbo-codes,” in *Proc. ICC*, 1993, pp. 1064–1070.

[10] T. N. Sainath, B. Ramabhadran, and M. Picheny, “An exploration of large vocabulary tools for small vocabulary phonetic recognition,” in *Proc. ASRU*, 2009, pp. 359–364.

[11] N. Harte and E. Gillen, “TCD-TIMIT: An audio-visual corpus of continuous speech,” *IEEE Transactions on Multimedia*, vol. 17, no. 5, pp. 603–615, 2015.

[12] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, “TIMIT acoustic-phonetic continuous speech corpus,” *Linguistic data consortium, Philadelphia*, vol. 33, 1993.

[13] H. Steeneken and F. Geurtsen, “Description of the RSG-10 noise database.” TNO Institute for Perception, Tech. Rep., 1988.

[14] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, “The second CHiME speech separation and recognition challenge: Datasets, tasks and baselines,” in *Proc. ICASSP*, 2013, pp. 126–130.

[15] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, “The third ‘CHiME’ speech separation and recognition challenge: Dataset, task and baselines,” in *Proc. ASRU*, 2015, pp. 504–511.

[16] H.-G. Hirsch, “F a N T - filtering and noise adding tool,” International Computer Science Institute, Niederrhein University of Applied Science, Tech. Rep., 2005.

[17] H.-G. Hirsch and D. Pearce, “The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions,” in *Proc. ISCA ITRW ASR2000*, 2000.

[18] N. Parihar, J. Picone, D. Pearce, and H.-G. Hirsch, “Performance analysis of the Aurora large vocabulary baseline system,” in *Proc. the 12th European Signal Processing Conference*, 2004, pp. 553–556.

[19] D. Kolossa, S. Zeiler, R. Saeidi, and R. Astudillo, “Noise-adaptive LDA: A new approach for speech recognition under observation uncertainty,” *IEEE Signal Processing Letters*, vol. 20, no. 11, pp. 1018–1021, 2013.

[20] M. J. Gales, “Maximum likelihood linear transformations for HMM-based speech recognition,” *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[21] D. Povey *et al.*, “The Kaldi speech recognition toolkit,” in *Proc. ASRU*, 2011.

[22] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proc. CVPR*, 2001, pp. I–511.

[23] J. Shi and C. Tomasi, “Good features to track,” in *Proc. CVPR*, 1994, pp. 593–600.

[24] B. D. Lucas, T. Kanade *et al.*, “An iterative image registration technique with an application to stereo vision.” in *Proc. IJCAI*, vol. 81, no. 1, 1981, pp. 674–679.

[25] J. E. Bresenham, “Algorithm for computer control of a digital plotter,” *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.

[26] S. J. Young, J. J. Odell, and P. C. Woodland, “Tree-based state tying for high accuracy acoustic modelling,” in *Proc. workshop on Human Language Technology*. Association for Computational Linguistics, 1994, pp. 307–312.

[27] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, “Sequence-discriminative training of deep neural networks.” in *Proc. Interspeech*, 2013, pp. 2345–2349.

[28] J. G. Fiscus, “A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER),” in *In Proc. ASRU*, 1997, pp. 347–354.