WATCH, LISTEN ONCE, AND SYNC: AUDIO-VISUAL SYNCHRONIZATION WITH MULTI-MODAL REGRESSION CNN

Toshiki Kikuchi and Yuko Ozasa

Keio University
Yokohama, Japan

ABSTRACT

Recovering audio-visual synchronization is an important task in the field of visual speech processing. In this paper, we present a multi-modal regression model that uses a convolutional neural network (CNN) for recovering audio-visual synchronization of single-person speech videos. The proposed model takes audio and visual features of multiple frames as the input and predicts a drifted frame number of the audio-visual pair which we input. We treat this synchronization task as a regression problem. Thus, the model does not need to search with a sliding window which would increase the computational cost. Experimental results show that the proposed method outperforms other baseline methods for recovered accuracy and computational cost.

Index Terms— Audio-visual synchronization, visual speech processing, neural networks

1. INTRODUCTION

Recently, video hosting websites and social networking services spread widely. However, in some uploaded videos the audio-visual synchronization drifts because of many reasons. Our work focuses on recovering audio-visual synchronization of single-person speech videos.

To recover audio-visual synchronization, Liu and Sato [1] proposed a method that uses quadratic mutual information (QMI). Their method computes QMI between audio and visual features, and uses it as the correlation value to determine whether the audio and video are synchronized correctly or not. They use the vertical optical flows of speaking lip image sequences as the visual feature. We also use the optical flow extracted from speaking lip image sequences.

The optical flow is calculated with the changes in the intensity of each pixel. Consequently, the outside of the lip area, which has little correlation with the speaker's voice, can adversely affect the result. To prevent this, we use the convolutional neural network (CNN) to properly weight the optical flow region and extract the feature. In related work that uses the CNN to extract the lip image feature, Assael *et al.* [2] and Chung *et al.* [3] showed state-of-the-art lip-reading results. In addition, Ephrat and Peleg [4] showed the CNN-based model generating acoustic speech audio from video frames. In contrast, we propose a model that takes visual and audio information as the input and predicts the drifted frame number.

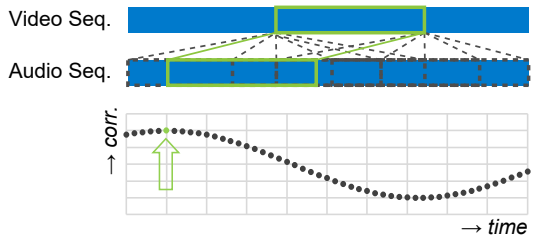

Fig. 1. The sliding window approach.

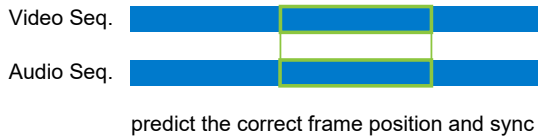

Fig. 2. The regression approach.

For a multi-modal CNN that takes audio-visual information as the input, Arandjelovic and Zisserman [5] attempted to compute the correlation between a video frame and the audio. However, here we consider not only spatial but also temporal features by taking multi-frame visual information as the input.

Here, we define two audio-visual synchronization approaches, the sliding window approach and the regression approach. The sliding window approach is a naïve algorithm to find the correct position of the audio sequence outlined in Fig. 1. First, we calculate correlation values for all possible audio frames with a sliding window. Then, we let the frame number with the highest correlation value be the drifted frame number. In this approach, we have to shift the audio sequence and compute the correlation values for each frame; therefore, we call this brute-force method the sliding window approach. By this definition, Liu and Sato's system [1] is classified as the sliding window approach. The regression approach is another approach for achieving audio-visual synchronization

This work was partially supported by JST CREST Intelligent Information Processing Systems Creating Co-Experience Knowledge and Wisdom with Human-Machine Harmonious Collaboration.

as shown in Fig. 2. This is the approach that pridicts the drifted frame number only by looking at the corresponding audio frame. We call this method which performs without a sliding window as the regression approach. As long as the computational cost is equivalent to the cost of calculating every sliding window in the sliding window approach, the regression approach can be faster. Our system outputs the drifted frame number instead of a sort of correlation value so this approach is classified as the regression approach. As far as we know, this is first work based on this approach.

We propose a multi-modal regression CNN for audio-visual synchronization for single-person speech videos. The proposed model takes audio and visual features of multiple frames as the input and outputs the drifted frame number. We call the propsoed model WLOS (Watch, Listen Once, and Sync).

Our contributions can be summarized as follows: (1) We propose a novel architecture for recovering audio-visual synchronization using a CNN. (2) We show the benefit of treating audio-visual synchronization as a regression problem.

We also show experimental results that demonstrate the proposed method outperforms other baseline methods.

2. APPROACH

In this section, we present a multi-modal regression CNN for audio-visual synchronization for single-person speech videos. This approach treats synchronization as a regression problem. We input the audio and visual features of n-frame into the multi-modal CNN to obtain the drifted frame number d as the output. If the audio and video entered in this model are synchronized correctly, then the expected output d is 0. Looking at only the n-frame audio-visual information, the system can distinguish the drift as long as d satisfies $-(n-1) \leq d \leq +(n-1)$. We assume that $n = 10$ so that $-9 \leq d \leq +9$.

2.1. Input feature representation

2.1.1. Visual feature

Similar to Chung *et al.* [3], we use a grayscale image as the input of the proposed whole system.

We use facial alignment similar to LipNet [2] with facial landmarks detected by Kazemi and Sullivan's method [6]. This obviates the need to consider the position-invariance network; thus, that the system can focus on just extracting the lip motions. We assume that this process performs well without any failures. Fig. 3 (a) shows an example of detected facial landmarks.

We assume that the mouth area has a stronger correlation with the audio than the cheek area. Therefore, we use only an area of a lip of spatial resolution 32×32 as shown in Fig. 3 (b).

(a) Facial landmarks.

(b) Cropped and aligned lip area.

Fig. 3. Images of a speaker's face.

Similar to Liu and Sato [1], we consider that in speaking a lip moves almost up and down. Therefore, we use vertical elements of the optical flows. For each frame t, we compute the optical flows between F^t and a previous frame F^{t-1} with the Gunnar Farneback method [7], where F^t is the cropped image of frame t. We also scale the optical flows to be in the range $[0, 1]$ per each mini-batch and use it as the visual feature V^t.

2.1.2. Audio feature

We use Mel-Frequency Cepstrum Coefficients (MFCCs) [8] as the audio feature. MFCCs are a representation of the spectral information in a short-term sound and have been used extensively in speech or speaker recognition because of a characteristic; it takes human auditory sensitivity into consideration. First, we apply the hamming window whose size is 256 to the audio. Then, we compute the 13 MFCCs and use 12 MFCCs except the very first MFCC which is not informative about the actual spectral content. We normalize the MFCCs to the range $[0, 1]$ and use it as the audio feature A.

2.2. Network architectures

The proposed network consists of three networks which are visual network, audio network and fusion network as shown in Fig. 4. We treat the 10-frame optical flows $\{V^t, V^{t+1}, ..., V^{t+9}\}$ extracted from $\{F^t, F^{t+1}, ..., F^{t+9}\}$ as the input of the visual network. Simultaneously, we input the corresponding audio features A into the audio network. The two networks that use the CNN extract the feature vector VF and AF from $\{F^t, F^{t+1}, ..., F^{t+9}\}$ and A, respectively. We input a concatenated audio-visual feature $[VF, AF]$ into the fusion network to obtain the drifted frame number d as the output of the network using regression.

In this section, we explain the details about the proposed network, WLOS.

2.2.1. Visual network

This network takes 10-frame optical flows $\{V^t, V^{t+1}, ..., V^{t+9}\}$ of spatial resolution 32×32 as the input and extracts the 1024 dimensional feature vector VF.

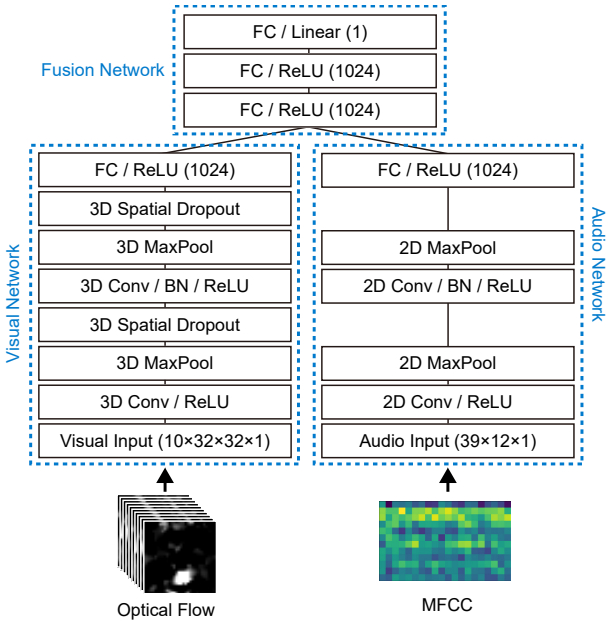

Fig. 4. WLOS network architecture.

The design of the visual network is mainly based on LipNet [2]; therefore, we also use the three-dimensional (3D) convolutional neural network (3D CNN) [9] after them. Recently, a 3D CNN achieved remarkable results in an action recognition task [9]. It enables us to take not only spatial but also temporal representation into account to extract features.

We use two layers of $3 \times 3 \times 3$ convolutions with a stride of $1 \times 1 \times 1$. The number of filters is 64 for the upper layer and 32 for the lower layer. We also use two maximum pooling layers with a stride of $1 \times 1 \times 1$ to reduce the network parameters. The pooling sizes are $2 \times 2 \times 2$ for the upper layer and $1 \times 2 \times 2$ for the other. The input tensor is convolved by two 3D convolutional layers and flattened by the 1024 dimensional fully-connected layer. We also apply batch normalization [10] at the second convolutional layer to accelerate the training. This network is trained with three-dimensional (3D) spatial dropout [11] after every pooling layer at dropout rate $p = 0.5$. All layers use the rectified linear unit (ReLU) as the nonlinear activation function.

2.2.2. *Audio network*

The audio network takes the 39×12 dimensional audio feature A as the input and extracts the 1024 dimensional feature vector AF.

This network consists of traditional 2D convolutional layers. We use two layers of 2×2 convolutions with a stride of 1×1. The number of filters is 128 for the upper layer and 64 for the lower layer. We also use two maximum pooling layers with a stride of 1×1 to reduce the network parameters. The pooling sizes are 2×2 for the upper layer and 1×2 for the lower layer. As with the visual network, we apply batch normalization [10] at the second convolutional layer. The input tensor is convolved by two 2D convolutional layers and flattened by the 1024 dimensional fully-connected layer. All layers use ReLU as the nonlinear activation function.

2.2.3. *Fusion network*

Using regression, the fusion network predicts the drifted frame number d. It takes the 2048 dimensional audio-visual feature vector AVF as the input where AVF is a concatenated audio-visual feature $[VF, AF]$. This network consists of three fully-connected layers. Because this is designed for solving the regression problem, the final fully-connected layer has one dimension and linear activation. All layers except the very final one use ReLU as the nonlinear activation function.

2.3. Fine-tuning

We use pre-trained weights as the initial weights of the visual network and the audio network. The network for pre-training takes the audio and visual information the same as WLOS and predicts whether the audio and visual information are correlated or not. Therefore, this solves the binary classification problem. We call this network classification correlation CNN (C^3). The architecture of C^3 is completely the same as WLOS except the fusion network. The fusion network of C^3 is composed of two fully-connected layers. The dimension of the penultimate layer (input of the fusion network) is 1024 and that of the other layer (output) is 2. They are activated with ReLU and Softmax functions, respectively. Before training WLOS, we train the C^3 network with audio-visual pairs, including correlated and uncorrelated pairs, to minimize the cross-binary entropy. Then, we remove the all of the fusion network of C^3, connect the fusion network of WLOS, and train it.

3. EXPERIMENTS

3.1. Dataset

We made intentionally drifted audio-visual data similar to Liu and Sato. We constructed a dataset with the GRID Corpus [12]. The GRID Corpus [12] contains audio and video clips of 1000 sentences spoken by 34 subjects, labeled *S1* to *S34*. This dataset is widely used for lip-reading tasks. All videos have a 3 second duration with a frame rate of 25 FPS. We performed experiments on the first two men (*S1*, *S2*) and women (*S4*, *S7*) in numerical order.

We extracted the face regions of all videos by using the HoG (histogram of oriented gradients)-based [13] face detector build with Dlib [14] for the first frame and using the KCF (kernelized correlation filters) tracker [15] for the ramaining frames. We divided the video and audio sequences into blocks by shifting them sequentially, frame by frame. Each block

has 10 frames. The frame duration is the same as the video sequence, 25 FPS, as the audio sampling rate is higher than that of video. At this time, we ignored blocks whose maximum MFCC value is lower than the threshold as they contain no speech.

We generated the audio-visual pairs including -9- to $+9$-frame drifted pairs for each video block. Thus, a video block creates 19 pairs. As a result, we have 63739 pairs for *S1*, 64000 pairs for *S2*, 63992 pairs for *S4*, and 63997 pairs for *S7*. We skipped 261 pairs for *S1*, 8 pairs for *S4*, and 3 pairs for *S7* because of face detection or facial alignment errors.

3.2. Baseline Methods

To evaluate the proposed method, we developed three baseline methods.

The first method is based on Liu and Sato's method [1] using kernel density estimation (KDE) and QMI. In this baseline method, we compute QMI between the 10-frame optical flows and the power of the corresponding audio as shifting the audio from -9 frames to $+9$ frames to find the position with the highest correlation.

The second method uses the C^3 network which is used to pre-train the WLOS network. C^3 outputs the probability that the video and the audio are correlated. Similar to the first method, we compute the probabilities as shifting the audio from -9 frames to $+9$ frames and determine the audio position most correlated with a video block. Both methods use sliding windows so these methods are classified as the sliding window.

The third method is the same as the proposed method except the method of initializing network weights. In this method, we train the WLOS network from scratch to evaluate the influence of the pre-training. This method is classified as the regression approach.

3.3. Training details

For the training data, we randomly sampled 80% of the dataset and used the remaining data for validation. We optimized the network using the stochastic gradient descent [16] with a learning rate of 10^{-4}, a momentum of 0.9, and a decay of 10^{-5}. We also used mini-batches containing 4096 samples per iteration. We trained the network until the validation loss no longer decreased, around 2000 iterations.

3.4. Results

Table 1. shows the mean absolute errors of computing drifted frame number d for the validation dataset with each method, from QMI to WLOS with fine-tuning. This table also shows dataset-wise results from *S1* to *S7*.

WLOS with fine-tuning, the proposed method, outperformed the other baseline methods. In comparison with the QMI method and other CNN-based methods, the CNN can perform well to extract good features for audio-visual synchronization. Comparing the WLOS trained from scratch and the WLOS with fine-tuning, we can also see a positive effect of pre-training.

Table 1. Mean absolute error (frame).

Method	S1	S2	S4	S7
QMI (based on [1])	6.305	6.494	6.196	6.479
C^3	1.352	2.134	2.983	1.019
WLOS (scratch)	0.937	1.003	1.116	0.848
WLOS (fine-tune)	**0.907**	**0.916**	**1.038**	**0.799**

In addition, WLOS can determine the drifted frame number approximately 19 times faster than C^3 because WLOS does not use a sliding window. Synchronizing a frame with WLOS, C^3, and QMI took 1.80 ms, 34.00 ms, and 46.07 ms, respectively. The time measurement was done for the whole *S1* dataset with Nvidia TITAN X (Pascal) and Intel Core i7-6900K. This result is theoretically correct because we move the sliding window for 19 frames in the sliding window approach under the condition of $n = 10$.

4. CONCLUSION

We proposed a multi-modal regression CNN for recovering audio-visual synchronization. We also introduced baseline methods, and the results show that the proposed method performs better than the baseline methods. The proposed approach enables us to recover errors without searching with a sliding window which would increase computational cost. The experimental result also shows that CNN-based methods perform well in comparison with other methods based on previous research.

In future work, we will make it possible to correct audio-visual synchronization errors of general videos instead of speech videos.

5. REFERENCES

[1] Yuyu Liu and Yoichi Sato, "Recovery of audio-to-video synchronization through analysis of cross-modality correlation," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 696–701, June 2010.

[2] Yannis M. Assael, Brendan Shillingford, Shimon Whiteson, and Nando de Freitas, "Lipnet: Sentence-level lipreading," *CoRR*, vol. abs/1611.01599, 2016.

[3] Joon Son Chung, Andrew W. Senior, Oriol Vinyals, and Andrew Zisserman, "Lip reading sentences in the wild," *CoRR*, vol. abs/1611.05358, 2016.

[4] Ariel Ephrat and Shmuel Peleg, "Vid2speech: Speech reconstruction from silent video," in *Proceedings of*

the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'17). IEEE, 2017, pp. 5095–5099.

[5] Relja Arandjelovic and Andrew Zisserman, "Look, listen and learn," *CoRR*, vol. abs/1705.08168, 2017.

[6] Vahid Kazemi and Josephine Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14)*. IEEE, 2014, pp. 1867–1874.

[7] Gunnar Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Proceedings of the 13th Scandinavian Conference on Image Analysis (SCIA'03)*, 2003, pp. 363–370.

[8] Steven B. Davis and Paul Mermelstein, "Readings in speech recognition," chapter Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences, pp. 65–74. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[9] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, January 2013.

[10] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[11] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler, "Efficient object localization using convolutional networks," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*. IEEE, 2015, pp. 648–656.

[12] Martin Cooke, Jon Barker, Stuart Cunningham, and Xu Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, November 2006.

[13] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. IEEE, 2005, vol. 1, pp. 886–893.

[14] Davis E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, December 2009.

[15] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings of the 12th European Conference on Computer Vision - Volume Part IV (ECCV'12)*, Berlin, Heidelberg, 2012, pp. 702–715, Springer-Verlag.

[16] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, "Neurocomputing: Foundations of research," chapter Learning Representations by Back-propagating Errors, pp. 696–699. MIT Press, Cambridge, MA, USA, 1988.