Audio-visual speech recognition (AVSR) is a promising approach to improving the noise robustness of speech recognition in the real world. For AVSR, the auditory and visual units are the phoneme and viseme, respectively. However, these are often misclassified in the real world because of noisy input. To solve this problem, we propose two psychologically-inspired approaches. One is audio-visual integration based on missing feature theory (MFT) to cope with missing or unreliable audio and visual features for recognition. The other is phoneme and viseme grouping based on coarse-to-fine recognition. Preliminary experiments show that these two approaches are effective for audio-visual speech recognition. Integration based on MFT with an appropriate weight improves the recognition performance by $-5$ dB. This is the case even in a noisy environment, in which most speech recognition systems do not work properly. Phoneme and viseme grouping further improved the AVSR performance, particularly at a low signal-to-noise ratio.$^1$
2. Issues and Approaches for Robotic Audio-Visual Speech Recognition

There are essentially two approaches to AVSR – early or late integration. Early integration involves integrating audio and visual information in the feature-extraction stage, i.e., the unit of recognition is a phoneme-viseme (P-V) [10,11]. Late integration involves integrating the recognition results separately: audio-based speech recognizers have phoneme recognition units whereas visual-based speech recognizers have viseme ones [12]. Both approaches have been reported to improve speech recognition [10–12]. However late integration has to consider the different alignments of the two types of recognition result; these are estimated separately in the audio- and visual-based speech recognizers. Thus, it is time consuming to integrate the recognition results, because many hypotheses have to be tested to find the best alignment correspondence between the results. Therefore, early integration is commonly used for AVSR. For example, Tamura et al. used a multi-stream hidden Markov model (HMM) for AVSR based on early integration [11]. The multi-stream HMM is able to weight the audio and visual streams independently. Thus, the AVSR performance improves drastically when the optimal weight of each stream is found. However, Tamura et al. assumed that both streams were always available, and they did not consider dynamic noise changes. Because robots have to work in everyday environments, robotic AVSR has to cope with 1) visual information being unavailable on occasion because of either partial or total obstruction, and 2) dynamically changing noise levels of the input sounds.

To address these issues, we propose to use AVSR with the following two approaches: using MFT to deal with missing and/or unreliable features included in input speech by masking them in recognition, and coarse speech recognition to change the unit of recognition from an individual P-V to a P-V group.

2.1. Missing Feature Theory

Missing feature theory [8] is proposed to cope with noisy speech input. When there are noises, some areas in the spectro-temporal space of speech are unreliable as acoustic features. Ignoring unreliable areas or estimating features in the unreliable parts by using the reliable areas makes it possible to perform noise-robust speech recognition. As a similar approach, multi-band ASR [13,14] has been proposed. This method uses HMMs for each sub-band, and obtains an integrated likelihood by assigning smaller weights to less reliable sub-bands. In this paper, we include the multi-band ASR method in the term MFT. Methods based on MFT show high noise-robustness for both stationary and non-stationary noises if the reliability of the acoustic features is estimated correctly. The main issue in applying them to ASR is how to estimate the reliability of the input acoustic features correctly.

Yamamoto et al. [4] reported a robot audition system that integrates sound source separation with a microphone array and MFT-based ASR with automatic reliability estimation based on inter-channel leak energy. They showed the effectiveness of the robot audition system through the recognition of speech uttered simultaneously by three people. Nishimura et al. [15] focused on speech recognition in the presence of noise due to the robot when either stationary or moving. They captured the noise associated with each motion in advance, and succeeded in improving ASR performance by using the captured noise as a template to estimate the reliability for MFT.

Because MFT is a general framework for dealing with noisy data, it is applicable to AVSR. Each feature in an input AV feature vector is weighted independently by MFT, whereas multi-stream HMM supports only weighting between the audio and visual streams. This means that MFT has a more flexible weighting mechanism, and is able to cope with more complicated noises.

2.2. Coarse Speech Recognition

Coarse speech recognition is inspired by the auditory and visual confusion that arises in human speech recognition in a noisy environment. The auditory confusion suggests that perceptually-close phonemes are perceived as a phoneme group when people attempt to recognize phonemes in a noisy environment [16]. The higher the noise level, the larger the number of phonemes that are unified into a phoneme group. The same is true for visemes in visual speech recognition [17].

This indicates that it is reasonable to output a P-V group that includes multiple P-V candidates rather than to estimate an individual P-V forcefully for the corresponding input, since robotic AVSR must deal with noisy input. Nevertheless, the result of coarse speech recognition is ambiguous because it is a P-V group rather than a sequence of P-Vs. This ambiguity can be resolved by using other knowledge, such as a word dictionary if the vocabulary is limited. Human-like hierarchical recognition known as coarse-to-fine recognition [18] would solve this issue effectively.

As related work, Sugamura et al. reported ASR using a phoneme-like template that was formed by clustering the acoustic features of input speech signals [19]. Other studies have used hierarchical phoneme classes that were structured according to the similarity of articulation [20, 21]. This paper extends that approach to AVSR by generating P-V groups according to the distance between individual P-Vs.
3. System Implementation

Figure 1 depicts the architecture of our AVSR prototype system based on coarse speech recognition and MFT. The system consists of seven modules – 1) Audio Feature Extraction, 2) Visual Feature Extraction, 3) Audio-Visual Feature Generation, 4) Audio Missing Feature Mask Generation, 5) Visual Missing Feature Mask Generation, 6) Audio-Visual Mask Generation, and 7) Coarse Speech Recognition. The following subsections describe each module in detail.

3.1. Audio Feature Extraction

Mel frequency Cepstrum coefficients (MFCCs) are one of the most commonly used features of ASR. Thus, we used a 25-dimensional MFCC-based acoustic feature (12 MFCCs, 12 ΔMFCCs, and 1 Δlog power) as the AVSR acoustic feature. The input speech signals were recorded with a microphone at a sampling rate of 16 kHz in synchronization with image capturing at 125 Hz. We used the Hidden Markov Model Toolkit (HTK) to extract the MFCCs. Although a window length of 25 ms and a window shift length of 10 ms are commonly used for short-time Fourier transform (STFT) extraction of MFCCs, we set the window shift length to 8 ms in order to remain synchronized with the visual features that were extracted at 125 Hz (corresponding to a period of 8 ms).

3.2. Visual Feature Extraction

Visual feature extraction in AVSR is one of the most challenging problems, for which many methods have been proposed. Image-based low-level features obtained by principal component analysis (PCA) [22, 23], fast Fourier transforms (FFTs) [24, 25], or their hybrid [10] do not require a lip model, but they have difficulty in dealing with obstruction and visual noise. Another reported approach is the use of model-based parametric features such as the outline of the lip shape [26], and the length and height of the lips [11]. Using these types of feature leads to high performance in speech recognition, and we propose a model-based parametric feature in this paper.

Our visual feature extraction consists of two stages: lip detection and feature extraction. Because this paper focuses on AVSR based on coarse speech recognition and MFT rather than on visual feature extraction, these stages are specific to our AV data. After mentioning a condition on the input image capturing, these stages are described in detail.

3.2.1. Condition of Image Capturing

An image is captured with a camera (NAC HSV-500C3), and visual features are then extracted. The frame rate of the camera is 125 Hz, whereas that of a conventional camera is 30 Hz. Each image has 720 × 480 pixels with 24-bit color. The camera is set up to take a full-face photograph with a 100-pixel wide mouth, as shown in Fig. 2.

3.2.2. Lip Detection

Figure 3 shows the flow of lip detection. To emphasize the edges in an image, \( G(x, y) \) at a position \((x, y)\) around the mouth area is transformed to binary value \( G_{bin}(x, y) \):

\[
\begin{align*}
G(x, y) &= 3 \cdot G(x, y) - G(x-1, y) - G(x+1, y) \\
&\quad - G(x, y-1) - G(x, y+1), \\
G_{bin}(x, y) &= \begin{cases} 
1, & \text{if } G(x, y) > 0, \\
0, & \text{otherwise}. 
\end{cases} \quad (1)
\end{align*}
\]

This Laplacian-like filter was determined empirically. Horizontal and vertical histograms of \( G_{bin}(x, y) \) are then generated, i.e., \( h_x(x) = \sum_y G_{bin}(x, y) \) and \( h_y(y) = \sum_x G_{bin}(x, y) \).

\[ \sum_{i} G_{\text{bin}}(x, y). \] A continuous part including the center of the mouth (i.e., \((x_c, y_c) = (360, 300)\)) is extracted from each histogram as the width \((x_l)\) or the height \((y_l)\) of the mouth.

\[
x_l = \arg \max x_i - y_{\min} \sum_{k = x_{\min}}^{x_l} h_k(x_c + k),
\]

\[
y_l = \arg \max y_i - y_{\min} \sum_{k = y_{\min}}^{y_l} h_k(y_c + k),
\]

where \(x_{\min}\) and \(y_{\min}\) are distances from \((x_c, y_c)\) to the top-left point of the mouth area; \(x_{\min}\) and \(y_{\min}\) are also estimated while satisfying \(0 \leq x_{\min} \leq x_l\) and \(0 \leq y_{\min} \leq y_l\). We tried using a more sophisticated method, namely LUX-based lip detection [27], but our proposed method performed more robustly with our AV data.

### 3.2.3. Feature Extraction

Let \(x_k\) and \(y_k\) be \(x_l\) and \(y_l\), respectively, extracted from an image at the \(k\)-th frame. \(x_k\) and \(y_k\) are smoothed because lip detection sometimes produces outliers. Since a simple moving-average method cannot remove such outliers, we applied rule-based smoothing as follows:

a) \(x_{k-1}\) is set to \(x_k\) if \(x_k\) is equal to \(x_{k-2}\) or \(x_{k-3}\); \(x_{k-2}\) is also set to \(x_k\) when \(x_k = x_{k-3}\).

b) \(x_{k-1}\) is set to \((x_k + x_{k-2})/2\) if the following conditions are satisfied:

\[
(\Delta x_k) + (x_{k-1} - x_{k-2}) < 0,
\]

\[
|x_k - x_{k-1}| \geq 5, \quad \text{and} \quad |x_{k-1} - x_{k-2}| \geq 5.
\]

c) \(x_{k-2}\) is set to \((x_{k-1} + x_{k-3})/2\) if \((x_k - x_{k-1})(x_{k-1} - x_{k-3}) > 0\).

Differences \(\Delta x_k\) and \(\Delta y_k\) are calculated using \((x_{k-2}, \ldots, x_{k+2})\) and \((y_{k-2}, \ldots, y_{k+2})\) with linear regression defined by

\[
\begin{cases}
\Delta x_k = \frac{-2x_k - x_{k-1} + x_{k+1} + 2x_{k+2}}{10\Delta t} \\
\Delta y_k = \frac{-2y_k - y_{k-1} + y_{k+1} + 2y_{k+2}}{10\Delta t}
\end{cases}
\]

where \(\Delta t\) is the frame shift length (8 ms). Although this filter has a three-frame delay in processing, it works in an online manner.

Finally, a four-dimensional visual feature vector \((x(k), y(k), \Delta x(k), \Delta y(k))\) is extracted and sent to Audio-Visual Feature Generation.

### 3.3. Audio-Visual Feature Generation

This module generates a 29-dimensional AV feature vector by concatenating 25-dimensional acoustic and 4-dimensional visual feature vectors. The resulting AV feature vector is sent to two modules: Audio-Visual Mask Generation and Coarse Speech Recognition.

### 3.4. Audio Missing Feature Mask Generation

An AV missing feature mask (AV-MFM) is used to cover unreliable features in an AV feature vector; it has 29 mask values corresponding to the AV features. Thus, it consists of two types of mask: an audio missing feature mask (A-MFM) and a visual missing feature mask (V-MFM). These masks are estimated independently.

An A-MFM is obtained by comparing acoustic features between inputted speech and corresponding clean speech. This is called an \(a\ pri o r\i\) mask [28] because mask generation heuristics such as clean speech are used. The detailed algorithm is as follows:

1. Let \(A\) and \(C\) be acoustic features of the input speech and the corresponding clean speech, respectively.

2. Let \(M_k(i)\) be a mask value for the \(i\)-th element of the 12-dimensional MFCC in the \(k\)-th frame, where \(M_k(i)\) is obtained by

\[
M_k(i) = \begin{cases}
1 & \text{if } |A_k(i) - C_k(i)| < T, \\
0 & \text{otherwise}.
\end{cases}
\]

Here, \(T\) is an empirically defined threshold \((T = 5.0\) in our system).

3. If both \(M_k(i)\) and \(M_{k-1}(i)\) are equal to 1, \(\Delta M_k(i)\) is set to 1. Otherwise, it is defined by

\[
\Delta M_k(i) = M_{k-2}(i)M_{k-1}(i)M_{k+1}(i)M_{k+2}(i).
\]

4. The mask value for \(\Delta \log\) power is set to 1.

5. Thus, the 25-dimensional A-MFM of the \(k\)-th frame is generated by concatenating the 12-dimensional \(M_k\), the 12-dimensional \(\Delta M_k\), and the one-dimensional \(\Delta \log\) power mask.

### 3.5. Visual Missing Feature Mask Generation

The V-MFM is obtained from visible or invisible information given in advance, i.e., the \(a\ pri o r\i\) mask. When the given information is visible, all values in the mask are set to 1, i.e., reliable. When it is invisible, they are set to 0.

![Fig. 3. Lip detection.](Image)
3.6. Audio-Visual Mask Generation

The A- and V-MFMs are, then, integrated into an AV-MFM by weighting the MFMs. In this case, the weight values of the A- and V-MFMs affect the system performance directly like a multi-stream HMM. Thus, an experiment was performed to estimate the optimal weight.

For the experiment, we recorded AV word datasets uttered by five speakers (four men and one woman). The visual part of the datasets was captured by following the conditions described in Section 3.2.1. The speech was recorded with a microphone that was 30 cm away from the speakers’ mouth. The words included in the datasets were those in the Advanced Telecommunications Research International (ATR) phonetically-balanced 216-word set, except for the addition of an extra word “ASIMO,” i.e., there were 217 words for each speaker.

In the experiment, the speech part of the datasets was contaminated with two types of additive noises: white noise and stationary robotic noise (mainly fans). We contaminated with two types of additive noises: white noise and stationary robotic noise (mainly fans). We changed the SNR of the speech from −5 dB to 20 dB in intervals of 5 dB. The isolated word recognition was measured for the noise contaminated AV datasets by using MFT-based ASR with an a priori mask. In this way, the optimal weight value of the A-MFM was estimated by fixing the weight value of the V-MFM to 1.0.

Figure 4 shows the optimal A-MFM weight for various noise levels. Each filled circle represents an optimal weight value for the stationary robotic noise, and each open circle is the same for the white noise. The two lines approximate the relationship between SNR and optimal weight. The results show that the best weight is 0.8 when the SNR is higher than 5 dB. However, as the SNR is decreased below 5 dB, the weight decreases monotonically. The best weight value w is selected from Fig. 4 according to the input SNR.

Finally, we generate a four-dimensional V-MFM in which all values are either 0 or 1, and a 25-dimensional A-MFM in which each value is either w or 0. These are integrated into a 29-dimensional AV-MFM that is sent to Coarse Speech Recognition.

3.7. Coarse Speech Recognition

Coarse Speech Recognition consists of a speech recognition engine and the knowledge sources of an AV model and a word dictionary. Although no language model is indicated in Fig. 1, we used a simple language model for isolated word recognition.

We used the MFT-based Julian for a speech recognition engine in the Coarse Speech Recognition [15]. This was implemented by modifying Julian3 to cope with a missing feature mask.

The AV model corresponding to the acoustic model in the ASR is a three-state, eight-mixture, triphone-based HMM trained with AV word datasets. Each state of the HMM is represented as a Gaussian mixture model (GMM) with a diagonal covariance matrix because elements in an AV feature vector are independent from each other. Note that MFT cannot be applied if a GMM with a non-diagonal covariance matrix or a neural network model is used. In such cases, the mutual independence of the elements is not guaranteed. Therefore, the corresponding element of the AV-MFM cannot be used as the weight for an element of the AV feature.

Each HMM represents a P-V group which is shown in Table 1. Thus, the design of P-V grouping is crucial in coarse speech recognition. To obtain the best P-V grouping, we used agglomerative clustering based on the distance between P-V HMMs. In other words, the number of clusters corresponding to P-V groups was determined by changing the threshold of the agglomerative clustering so that the best speech recognition performance was achieved. First, the P-V HMMs were trained with the AV datasets. Each trained HMM is a three-state, eight-mixture monophone. It is necessary to use a computationally expensive method such as the Kullback-Leibler distance to calculate the distance between two HMMs accurately. Thus, we used a simple distance-like measure between two P-Vs $P_1$ and $P_2$ defined by

$$L(P_1, P_2) = -\frac{1}{2} \sum_{k=1}^{K} \{ w(k|P_1) f_{P_2}(\mu_{1,k}) + w(k|P_2) f_{P_1}(\mu_{2,k}) \} \ldots (3)$$

where $K$ is the number of Gaussian mixtures, $\mu_{i,k}$ is an average of each Gaussian in P-V $P_i$, $w(k|P_i)$ is the weight of the $k$-th Gaussian in $P_i$, and $f_P(x)$ is the probability density function for $P$ that is used to estimate the output probability.

Table 1. Sample of phoneme-viseme grouping.

<table>
<thead>
<tr>
<th>P-V Group</th>
<th>P-V Group</th>
<th>P-V Group</th>
<th>P-V Group</th>
<th>P-V Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>e</td>
<td>e</td>
<td>by</td>
<td>by</td>
</tr>
<tr>
<td>n</td>
<td>o</td>
<td>o</td>
<td>s</td>
<td>m</td>
</tr>
<tr>
<td>a</td>
<td>i</td>
<td>d</td>
<td>p</td>
<td>z</td>
</tr>
<tr>
<td>u</td>
<td>t</td>
<td>r</td>
<td>r</td>
<td>q</td>
</tr>
<tr>
<td>w</td>
<td>g</td>
<td>g</td>
<td>ry</td>
<td>j</td>
</tr>
</tbody>
</table>

Table 2. Experiment conditions of Ex. 1 and Ex. 2.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Values of A-MFM</th>
<th>Values of V-MFM</th>
<th>P-V grouping</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 AV integration with A-MFM and P-V grouping</td>
<td>$w/0$ (optimal)</td>
<td>1</td>
<td>on</td>
</tr>
<tr>
<td>2 AV integration with A-MFM</td>
<td>$w/0$ (optimal)</td>
<td>1</td>
<td>off</td>
</tr>
<tr>
<td>3 AV integration with multi-stream MFM</td>
<td>$w$</td>
<td>1</td>
<td>off</td>
</tr>
<tr>
<td>4 AV integration without MFM</td>
<td>1</td>
<td>1</td>
<td>off</td>
</tr>
<tr>
<td>5 audio only with A-MFM</td>
<td>$w/0$</td>
<td>0</td>
<td>off</td>
</tr>
<tr>
<td>6 audio only</td>
<td>1</td>
<td>0</td>
<td>off</td>
</tr>
<tr>
<td>7 vision only</td>
<td>0</td>
<td>1</td>
<td>off</td>
</tr>
<tr>
<td>8 AV integration with AV-MFM and P-V grouping</td>
<td>$w/0$ (optimal)</td>
<td>1/0 (optimal)</td>
<td>on</td>
</tr>
<tr>
<td>9 AV integration with AV-MFM</td>
<td>$w/0$ (optimal)</td>
<td>1/0 (optimal)</td>
<td>off</td>
</tr>
</tbody>
</table>

($w$ denotes the optimal audio-stream weight when the visual-stream weight is set to 1.)

The word dictionary contains the 217 words included in the AV word datasets. In the case of ASR, the word dictionary includes a sequence of phonemes corresponding to each word. In our system, a sequence of P-V groups is registered with the corresponding word.

Finally, Coarse Speech Recognition outputs a recognized word for the input AV data.

4. Evaluation

We evaluated the system by means of two experiments.

Exp. 1: Isolated word recognition with noise added to the speech part of the AV data.

Exp. 2: As Exp. 1 but with time frames missing from the visual part of the AV data.

The AV model was trained with clean AV data that were included in the AV datasets. The test data were also the same as the AV data that were used for training the AV model. However, because the test data were contaminated by noise, we reason that the tests other than the clean test were semi-closed tests.

In Exp. 1, the word correct rate (WCR) was measured under the seven conditions given by changing the MFM and P-V grouping parameters described as 1–7 in Table 2. We changed the SNR from $-5$ dB to 20 dB in intervals of 5 dB. Two types of noise were used: white and robotic noise. The robotic noise was recorded with a microphone embedded in our target robot, the Honda ASIMO, when it was switched on. It has several local peaks in its spectrum, whereas the white noise is widely-spread in the frequency domain. The WCR for clean AV data (i.e., an exactly closed test) was also measured to determine the upper-limit performance of this test.

Exp. 2 simulates obstructions and out-of-view situations that occur often in an application of AVSR with a robot. The visual part is missing for 0.8 s in every 1.6 s, as shown in Fig. 5. The WCR is measured under the four conditions that correspond to 8, 9, 2, and 4 in Table 2. The noise conditions are the same as those in Exp. 1. If the visual part is missing, the visual feature vector is set to 0 in Visual Feature Extraction.

4.1. Results

Figures 6a) and b) shows the results of Exp. 1 for white noise and robotic noise, respectively. Figs. 7a) and b) shows the results of Exp. 2 for white noise and robotic noise, respectively. The vertical axes represent the WCR, and the horizontal axes represent the SNR. The labels in the legend correspond to those in Table 2.

First, we note that the type of additive noise does not have much effect on the system performance; a small difference exists between the results with white and robotic noise in Exp. 1 and Exp. 2 from Figs. 6 and 7.

Figure 6 shows that the performance with “vision only” is better than that of “audio only” if the SNR is lower than 10 dB, otherwise “audio only” is better. AV integration without the use of MFT (i.e., “AV integration without MFM”) improved the performance in comparison with “audio only.” However, the performance of “AV integration without MFM” was worse than that of “vision only” when the SNR was equal to or lower than 5 dB. This indicates that AV integration was not effective in this case. In contrast, “AV integration with multi-stream MFM” which uses only a weight value between acoustic and visual feature vectors, was more effective because the performance was better than that of “audio only” and “vision only” regardless of the SNR. A more sophisticated AV integration method, “AV integration with A-MFM,” outperformed these conditions. When it is compared with “audio only with A-MFM,” we note that MFT-based AV integration makes the best use of visual information especially at low SNR, and shows high noise-robustness. P-V
grouping brings a small improvement (1 point) when “AV integration with A-MFM” is compared with “AV integration with A-MFM and P-V grouping.”

Figure 7 shows that the performance of AV integration without using MFM or with the use of A-MFM is not noise-robust any more when a visual feature is missing. When V-MFM is used effectively, the performance improves drastically. For example, at an SNR of 5 dB, a 40-point improvement was attained. In addition, P-V grouping improves the performance, although the improvement is relatively small.

4.2. Discussion

In our experiments, an AV-MFM is optimal because of the a priori mask. Because an a priori mask is unavailable in real situations, the AV-MFM has to be estimated from other cues. As for A-MFM, we already reported automatic MFM generation based on inter-channel leak energy after sound source separation using a microphone array [4] and a single channel speech enhancement technique [29, 30]. In particular, Sound source separation is a promising approach to improving ASR [4, 5]. Therefore, automatic MFM generation and sound source separation should be applied to our system in order to cope with more practical situations.

In Exp. 2, we set 0 for a missing feature. Instead of this, we could use the same value as the previous frame, which would be better when a V-MFM is not used. In this case, the performance would be between the results for condition 4 in Figs. 6 and 7. This remains to be confirmed in future work.

Coarse speech recognition with P-V grouping performs well at isolated word recognition. However, it may do less well at large vocabulary continuous speech recognition (LVCSR) using a statistical language model, because of the increased recognition complexity (e.g., more homonyms). However, it is possible that the use of coarse speech recognition together with context information obtained from a dialog module and situation information from a scene understanding module could improve the total system accuracy. This would be because the recognition results of coarse speech recognition retain ambiguities to be resolved later. This will be considered in future work as an interesting issue to be investigated.

The frame rate of our visual dataset was 125 Hz, but it is impractical to use visual features with such a high frame rate in real-time applications. We tried to clarify the necessary frame rate for visual feature extraction to maintain the performance. For this, we used a dataset consisting of 16 men and 216 words. The visual data was 8-bit monochrome and 640 × 480 pixels in size, and was recorded at 100 Hz using a BASLER A602fc camera. The sampling rate of the speech data was 16 kHz, and every speech was recorded synchronously with visual data. The visual data was downsampled to 10 Hz, 20 Hz, 33 Hz, 50 Hz, and 100 Hz. In addition, cubic spline interpolation was performed to upsample the downsampled visual data.
to 100 Hz. For each condition, isolated word recognition was performed for each visual feature. Note that this interpolation was performed not for raw image data but for visual features.

Figures 8a) and b) shows the results without/with spline interpolation. The horizontal and vertical axes represent the frame rate and the isolation word error rate, respectively. Fig. 8a) shows that a high frame rate results in better performance. However, when spline interpolation is performed, the performance can be maintained, as shown in Fig. 8b). Another interesting phenomenon is that the features interpolated from low frame rates outperformed those from high frame rates. We reason that this is because downsampling to generate features with a lower frame rate resulted in a form of smoothing. This indicates a promising approach of performing feature extraction at 10 Hz, say, and then upsampling to 100 Hz to achieve real-time processing.

We performed isolated word recognition with an AVSR system using MFCC as an acoustic feature and lip height and length as visual features. Recently, deep-learning-based AVSR [31] and VSR [32] have been reported, and both showed high performance. We believe that integrating the key concepts in this paper with deep learning techniques would be promising for further improvements.

5. Conclusion

We proposed a new audio-visual speech recognition (AVSR) system for a robot based on missing feature theory and coarse speech recognition to improve noise robustness. We constructed a prototype AVSR system based on these two approaches, and showed its effectiveness through isolated word recognition for AV data including noisy and missing features. We also discussed possibilities for automatic missing feature mask generation, frame rate reduction of visual features, and integration with deep learning for further improvements of AVSR systems.

Acknowledgements

We thank Prof. Jun-ichi Imura for valuable discussions.

References:


Tomoaki Koiwa
Affiliation: Examiner, Assistant Director, Japan Patent Office
Address: 3-4-3 Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan
Brief Biographical History: 2008 Received Master of Engineering from Tokyo Institute of Technology 2008- Japan Patent Office