Audio-visual fuzzy fusion for robust speech recognition

M. Malcangi, K. Ouazzane, and P. Patel

Abstract— Improvements of robustness of speech recognition is one of the hottest topics in speech signal processing, particularly when applied within a noisy environment. Most of the research efforts focused in combining audio and visual data to implement an audio-visual speech recognition (AVSR) system. Bimodal approach demonstrated that a superior performance can be gained compared to the separate audio or visual approach. This paper proposes a fuzzy logic-based data fusion method that combines the recognition capabilities of two independent working systems namely the automatic speech recognition system (ASR) and the automatic visual recognition system (AVR). The main purpose is to boost the whole system's performance keeping the ASR separate from the AVR. This approach provides a powerful method that enables simpler data fusion at decision level rather than the more complex at data and features level. Such complexity is also lowered due to the fuzzy logic-based implementation of the data fusion engine. Preliminary experimental results confirms the proposed approach.

I. INTRODUCTION

Wireless devices such as mobiles, PDA etc. use speech recognition for the name-dialing feature. Each voice tag is trained with a single utterance by the user and assigned to a single phone book entry. When the user wants to place a call, user pushes a button and speaks a person’s name. The phone answers with the recognized voice tag as acoustic feedback, and then automatically sets up the call [1].

In general, speech technology needs to offer a significant advantage over the existing interface to trigger users to change their behavior [2]. On wireless platforms, the technology has not yet reached a level of performance and usefulness to compel users to use the voice interface. Visual modality is very important to human speech perception [3][4]. This fact needs to be exploited to develop a bimodal speech recognition system robust to noisy environment. This research is ultimately aimed to increase ease of use and productivity by producing not a command recognition technology, but a feature rich, continuous, speaker independent bimodal speech recognition technology that enables users to easily activate, navigate and operate their wireless devices.

Another motivation is the huge number of surveillance systems installed in shops, banks, public places, etc. For the most systems, the security officers in the back office have to switch amongst a large number of video data. It is unfeasible to listen to multiple audio streams at once and interpret the information. In such a situation, a system is required that analyzes the visual speech and provides summary information about the speech conversations from every camera on a place to security officers. After receiving the sensitive conversation information, security officers can take appropriate action promptly [5] [6] [7] [8].

To implement speech recognition systems robust to noisy environment and with very low error rate, it is necessary to integrate both the auditory and visual abilities that human beings use to recognize a word uttered by another human beings. Speech recognition systems are very sensitive to noisy environment, so their performance can be improved significantly if more information can be available at decision level.

To realize the ability of some human beings who can read the speech from the speaker’s speech video is a challenging phase in audiovisual speech processing [9]. The visemes and part of the speech for the same phoneme are widely varying from one speaker to another. However, the human beings can surprisingly distinguish between different shapes of the lips, its movement, and associate the lip movement to a group of phonemes (that is, the viseme class).

Speech recognition fusion and decision logic needs to be fuzzy so that audio and video information can be optimally combined to infer about the right text corresponding to the uttered word [10] and about the identity of who is uttering.

In this research the integration of a DTW-based speech recognition system with a viseme recognition system is described [11]. The system can perform also speaker identification using the features extracted from an uttered password at audio and visual level. Two separate fuzzy logic-based infer the first about the word recognition and the second about the speaker identity. The purpose of this integration is to improve the whole performance of the system by converting utterance in text. As the viseme recognition system works at quasi-phoneme level, the text coming from the speech recognition system needs to be converted in its phonetic transcription [12]. Then a smart logic inference engine is tuned to process both the phoneme sequence and the viseme sequence of an uttered word so that the correct text can be generated. The smart logic tries to
emulate the human beings way to correct phoneme mismatch looking to the viseme. If a phoneme do not match with the viseme, the phoneme can be replaced with one belonging to the viseme class. Then, a new text string can be generated from the recovered phoneme sequence.

The purpose of the speaker identification is two-fold, one is to improve security level, the other is to switch automatically to a speaker dependent database that helps to improve speech recognition itself. The weaknesses of speech in identifying people due to the behavioral nature of the utterance can be overcome if speech biometric measurement is combined with visual information leading to an accurate decision. Given that visual information is not behavioral, it is useful to validate weak decisions from speech identification level. Lips embed enough information about speaker’s identity as demonstrated in several researches [13] [14].

How to perform the fusion is the main challenge for AVSR. The low level data and features audio-visual fusion find their main difficulties in higher dimensionality of data that imply nonlinear transformations, often based on neural networks or Kalman filtering, demonstrating no significant improvements in AVSRs performance. Intermediate or high-level fusion [15] uses simpler and linguistic-like information, so that simple combination techniques such as linear or pattern matching methods (HMM or DTW) are usefully applied. Fuzzy logic-based data fusion is a good tradeoff between complexity and the AVSR performance improvement.

II. INTEGRATING A DTW-BASED SPEECH RECOGNITION WITH A VISUAL SPEECH READING SYSTEM

A. System framework

As illustrated in Fig. 1, the system framework consists of four main subsystems, namely the DTW-based automatic speech recognition/speaker identification (ASR/ASI), the text-to-phoneme transcription (TTP), the viseme recognition (VR) and the phoneme-to-text transcription/speaker identification (PTT/SI) smart logic.

The DTW-based ASR subsystem recognizes a word in an utterance, then provides its text transcription. The text sequence is processed by a text-to-phoneme transcription algorithm that generates its phoneme transcription. The viseme recognition subsystem recognizes and generates the viseme sequence from the visual uttered speech. Phoneme and viseme sequences are then processed for text generation related to the uttered speech.

B. DTW-based speech recognition

The DTW-based ASR subsystem is tuned for speaker-independent isolated word recognition. The uttered word is end-pointed for feature extraction, then passed through a DTW pattern alignment algorithm, and matched on a set of reference words (templates) using Euclidean distance measurement method for similarity evaluation.

The end-pointed uttered word is segmented in short frames (20 ms duration) and from each frame a 27 LPC-cepstral coefficients are computed. A frame-by-frame DTW alignment is executed between the uttered word and each of the word templates to match the closer to the uttered word.

The DTW process is defined as:

\[
\begin{align*}
M & = x + y \\
N & = \frac{M}{2} + \frac{y}{2} \\
2 & = \frac{M}{3} - \frac{N}{3}
\end{align*}
\]

Fig. 1. Framework of the bimodal speech recognition and speaker identification system

Fig. 2. Boundaries model to align utterance feature target pattern to a reference template.

Key points are A and B:

\[
\begin{align*}
ax & = \frac{2}{3}M - \frac{N}{3} \\
xb & = \frac{4}{3}N - \frac{2}{3}M
\end{align*}
\]

These points are very useful to understand if the alignment of the two patterns is reasonable. The alignment condition is:

\[
\begin{align*}
2M - N & \geq 3 \\
2N - M & \geq 2
\end{align*}
\]

When the alignment of two patterns \(a\) and \(b\) is possible, the alignment cost is then computed using the Euclidean distance computation method:

\[
d = \sqrt{\sum_{m=0}^{N-1} \left( a_m - b_m \right)^2}
\]
Using this distance computation method, cepstral coefficients for features measurement, and a threshold to filter false positives, an average error of 11.18 is performed.

C. DTW-based speaker identification

The following features were used for speaker identification:

Root Mean Square (RMS):

\[ RMS_j = \sqrt{\frac{1}{N} \sum_{m=0}^{N-1} s_j^2(m)} \]  

Zero-Crossing Rate (ZCR):

\[ ZCR_j = \sum_{m=0}^{N-1} 0.5 \left| \text{sign}(s_j(m)) - \text{sign}(s_j(m-1)) \right| \]  

Auto Correlation (AC):

\[ AC_j = \sum_{i=1}^{N} \sum_{j=1}^{N-1} s_j(j)s_j(i + j - 1) \]  

Cepstral Linear Prediction Coefficients (CLPC):

\[ CLPC_j = a_m + \sum_{k=1}^{n-m} \frac{k}{m} c_k a_{m-k} \]  

This method scores the person’s identity. It is based on distance measurement of the dynamic time warping—k-nearest neighbor (DTW-KNN) algorithm. This combines the dynamic-time-warping measurement with k-nearest neighbor decision algorithm, resulting lower in false-positive and false-negative rates during identification than the original DTW algorithm used for speech recognition. The cost function is computed using Euclidean distance and KNN algorithm is used for k minimal distance matching.

D. Text-to-phoneme transcription

Text-to-phoneme transcription is needed after a text representation of the utterance is available and ASR completes the speech recognition process. The phoneme sequence needs to be compared to the viseme sequence to lower the false positive error rate of the ASR.

Text-to-phoneme transcription subsystem consists of a rule-based algorithm capable to transcribe a text word into its phonetic sequence (its pronunciation). A set of language-specific rules is passed through a regular expression-based algorithm to match and expand the rules that embed the correct phonetic transcription of a text pattern.

To transcribe the alphabetical text of a word into its phoneme sequence, a set of language-specific rules are then applied. Such rules have the following format:

\[ C(A)D = B \]  

Where A is the text transformed into the phonetic utterance B if the text to which it belongs matches A in the sequence C:AD. C is a pre-context string and D is a post-context string.

A sample of the rules encoding the character (P) is as follows (the dash in a phone slot represents its boundaries and that numbers preceding the phone indicate duration):

\[ \begin{align*}
(P) &= /1p/1i/ \\
(PA)STE &= /1p/1eI/1j/ \\
(PH) &= /1f/2I/1z/ \\
(PH) &= /1f/1o/1w/ \\
(PH) &= /1f/1o/1t/ \\
(PC) &\text{STE} = /1p/1eI/1j/ \\
(PE) &\text{PO}E = /1p/1o/1E/ \\
(POE)T &= /1p/1o/1E/ \\
(POU) &= /1p/1o/13/ \\
(POW) &= /1p/1O/1u/ \\
(PP) &= /1p/ \\
(PP) &= /1p/2H_f/ \\
(P) &= /1p/4h/4/ \\
(PA)STE &= /1p/1eI/1j/ \\
(PH) &= /1f/2I/1z/ \\
(PH) &= /1f/1o/1w/ \\
(PH) &= /1f/1o/1t/ \\
(PC) &\text{STE} = /1p/1eI/1j/ \\
(PE) &\text{PO}E = /1p/1o/1E/ \\
(POE)T &= /1p/1o/1E/ \\
(POU) &= /1p/1o/13/ \\
(POW) &= /1p/1O/1u/ \\
(PP) &= /1p/ \\
(PP) &= /1p/2H_f/ \\
(P) &= /1p/ \\
\end{align*} \]

The transcription standard is based on X-SAMPA (right context).

E. Viseme recognition

The viseme recognition subsystem processes the visual representation of the uttered speech and then recognizes the related sequence of visemes.

In visual speech reading, the alphabet detection process involves three main steps, which are given below.

- Manually extracting visual features for system training and as a reference.
- Extracting visual features automatically from speaker’s facial image sequence.
- Comparing these automatically extracted (observed) visual features with manually extracted features to find the similarity of spoken alphabet with other alphabets in the database.

As shown in the Fig. 3, the viseme classifier obtains the visual features from speaker’s facial image sequence using the feature extractor. The reference visual features are obtained from the dataset. Subsequently, the classifier compares the extracted visual features with the visual features in the dataset. The comparison procedure involves finding how similar the characteristics of the alphabet
uttered by speaker are with the characteristics of all English alphabets stored in the dataset. It is achieved by finding the similarity of the visual features such as height, width, time and mouth gestures. The similarities of these visual features have a diverse level of influence on overall similarity; some features are more significant than the others in overall similarity. After finding the similarity scores, the top three alphabets having visual features highly similar to the extracted visual features are generally considered. The methods of finding the similarity of these features and selecting the influence levels (weights) for various features are described in the following sections.

F. Visual feature similarity

The number of frames for any alphabetic utterance is not constant for every speaker. It is due to the fact that for the same alphabet a different speaker normally takes longer or shorter to pronounce it. Instead of constraining the solution to a fixed number of frames, two different versions of the same algorithms are developed. One version is used for features with almost the same number of frames and the other version is used for features with difference in the frame numbers. Since the visual features considered are the lip height-width, the time taken for utterance and the facial gestures, the overall similarity can only be derived by finding the similarity of all four visual features:

i. Height similarity score
ii. Width similarity score
iii. Time (no of frames) similarity score
iv. Gesture similarity score.

G. Height and Width Similarity

The procedure of finding the height or width similarity scores means finding the similarity of height or width variations between the observed visual features and the dataset visual features. For the visual feature sets with similar number of frames, the feature vectors are compared on one-to-one basis. The feature vector consists of the direction and vector itself. Method 1 compares the visual features on one-to-one basis at every 1/10th or 1/15th frame number. Method 2 employs a different approach to find the similarity score. In this approach, for vectors with direction mismatch, both an observed vector and a dataset vector are checked to see if either is unchanged. If any of them is unchanged, the vectors in the previous and next frames are used to see if the overall direction of the vector variation is the same in both feature vectors. In the methods shown above, the direction and the vector itself are equally important. The results of using such an equal weight system for the height similarity score determination are given in the Table I.

<table>
<thead>
<tr>
<th>Alphabet</th>
<th>Height Similarity</th>
<th>Overall Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.47</td>
<td>62.04</td>
</tr>
<tr>
<td>B</td>
<td>0.35</td>
<td>57.42</td>
</tr>
<tr>
<td>W</td>
<td>0.28</td>
<td>52.05</td>
</tr>
<tr>
<td>T</td>
<td>0.43</td>
<td>50.56</td>
</tr>
<tr>
<td>D</td>
<td>0.41</td>
<td>48.18</td>
</tr>
</tbody>
</table>

For alphabet ‘A’

<table>
<thead>
<tr>
<th>Alphabet</th>
<th>Height Similarity</th>
<th>Overall Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>0.46</td>
<td>70.72</td>
</tr>
<tr>
<td>R</td>
<td>0.42</td>
<td>67.36</td>
</tr>
<tr>
<td>H</td>
<td>0.46</td>
<td>63.42</td>
</tr>
<tr>
<td>L</td>
<td>0.46</td>
<td>61.19</td>
</tr>
<tr>
<td>A</td>
<td>0.27</td>
<td>46.73</td>
</tr>
</tbody>
</table>

Although such an equal weight approach gives good results, it has some weaknesses. For example two speakers with a huge physical appearance difference, a similar lip movement would be identified as non-similar. This is explained in details using two variation pattern examples below.

A sample height variation pattern for a speaker is shown in Fig. 4.a. The height variation pattern is compared with height variation patterns of all alphabets stored in the dataset. Assume that a sample dataset height variation pattern shown in Fig. 4.b is being compared with the variation pattern shown in Fig. 4.a.

Though the scale is different, the manner of the variation is the same for both patterns shown in Fig 4.a and Fig. 4.b. Here, due to the difference in the range of vector values, using an equal weight for vector-direction approach described above would identify these patterns as non-similar.

Such a problem was overcome by giving more importance to the actual pattern of variation than the feature value (vector) itself. The visual features for height variation are described by the actual vector representing the feature value and the sign ‘+’, ‘-’ or ‘=’ representing the direction of these variations. Thus, more importance is given to the direction, that is, the sign of the vector than the actual feature vector.

As described above the vector weight is dependent on the direction weight, that is, if the directions are different then the same vectors would have no score. The reason make the vector weight dependent on the direction weight is explained using the example below.
For example, a speaker’s height variation pattern shown in Fig. 4.c is compared with the height variation pattern shown in Fig. 4.d. The line charts in Fig. 4.c and 4.d show that the feature value varies between 10 and 15 pixels. Though the range of variation is the same, the actual lip movements are entirely opposite to each other. Here, it can be noted that the vectors itself would be very similar for both lip movements. However, the direction of vectors would be different. Thus, the independent weights for a vector and direction would interpret them as similar lip movement due to the vector (value) similarity. Thus, the vector weight is made dependent on the direction weight such that the vectors are only matched if the directions are similar.

A weight proportion of 80-20% was selected for the direction-vector weights. The results of using such a weight structure are given for alphabets ‘A’ and ‘B’ in Table II. Table IIa shows that the height vectors of alphabet ‘A’ were found to be more similar to height vectors of alphabet ‘N’, ‘R’ and ‘J’. Consequently, the overall similarity of alphabets ‘N’, ‘R’, ‘J’ and ‘H’ was higher. Other results in Table IIb show that the height vectors of alphabet ‘B’ were 58% similar to alphabet ‘P’ and 60% similar to alphabet ‘B’. Since both of them are from the bilabial phonemes group, the results can be considered as accurate.

<table>
<thead>
<tr>
<th>Alphabet</th>
<th>Height Similarity</th>
<th>Overall Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>0.73</td>
<td>76.2</td>
</tr>
<tr>
<td>R</td>
<td>0.69</td>
<td>74.2</td>
</tr>
<tr>
<td>H</td>
<td>0.60</td>
<td>68.9</td>
</tr>
<tr>
<td>J</td>
<td>0.69</td>
<td>66.6</td>
</tr>
<tr>
<td>A</td>
<td>0.62</td>
<td>65.6</td>
</tr>
</tbody>
</table>

a. For alphabet ‘A’

<table>
<thead>
<tr>
<th>Alphabet</th>
<th>Height Similarity</th>
<th>Overall Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.60</td>
<td>73.4</td>
</tr>
<tr>
<td>B</td>
<td>0.58</td>
<td>71.8</td>
</tr>
<tr>
<td>W</td>
<td>0.49</td>
<td>67.6</td>
</tr>
<tr>
<td>T</td>
<td>0.49</td>
<td>65.4</td>
</tr>
<tr>
<td>G</td>
<td>0.45</td>
<td>52.0</td>
</tr>
</tbody>
</table>

b. For alphabet ‘B’

**H. Time Similarity**

The time taken feature for any alphabet is measured in terms of the number of image frames taken for the utterance. Thus, the ‘Time Similarity’ is determined by comparing the number of frames for the observed features to the number of frames for the dataset features. If the difference in the number of frames of the observed and the dataset features is huge (for example, more than twice), the time similarity score is made zero, that is, non-similar.

The time taken for the utterance of most alphabets was found to be approximately equivalent. However, some alphabets take longer or shorter than average time taken by most other alphabets. Such a phenomenon was noticed for alphabets such as ‘W’, ‘H’ etc which took longer than average time. It was noticed that these alphabets take longer time due to its ‘biphone’ or ‘triphone’ nature. Some other alphabets such as ‘k’ and ‘z’ were observed to take shorter time than most other alphabets. The results found were not reproduced for every speaker. The alphabets which took longer or shorter for few speakers did not take the same time for every other speaker. Furthermore, the time taken for various alphabets was observed to be very similar. For most alphabets with a single phoneme, the number of frames required was very similar to the average number of frames required for most alphabets. Though the time similarity score is not profoundly consistent, it could still be used in a modest mode to classify a group of alphabets. To determine the time similarity score, the number of frames used for an alphabet utterance by a speaker was compared against the number of frames taken by every alphabet stored in the dataset. A time feature similarity score was defined as the ratio of the number of frames in speaker’s features to the number of frames in the dataset features. A sample pseudo code for time similarity derivation is given in Fig. 5.
The overall weight of the time feature is kept low to 5 to 10%. Since the results for the time taken feature were not reproduced and for most of the alphabets the time taken remains the same, thus it is given less significance, that is, less weight. A sample time similarity scores for some alphabets are given in Table III.

TABLE III
TIME SIMILARITY SCORES

<table>
<thead>
<tr>
<th>Alphabet</th>
<th>Time Similarity</th>
<th>Overall Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>1.0</td>
<td>77.5</td>
</tr>
<tr>
<td>U</td>
<td>1.0</td>
<td>76.0</td>
</tr>
<tr>
<td>Q</td>
<td>0.75</td>
<td>76.0</td>
</tr>
<tr>
<td>H</td>
<td>1.0</td>
<td>67.7</td>
</tr>
<tr>
<td>G</td>
<td>1.0</td>
<td>66.3</td>
</tr>
</tbody>
</table>

For alphabet ‘O’

<table>
<thead>
<tr>
<th>Alphabet</th>
<th>Time Similarity</th>
<th>Overall Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>0.69</td>
<td>61.9</td>
</tr>
<tr>
<td>H</td>
<td>0.85</td>
<td>61.7</td>
</tr>
<tr>
<td>X</td>
<td>0.45</td>
<td>60.0</td>
</tr>
<tr>
<td>S</td>
<td>0.94</td>
<td>58.7</td>
</tr>
<tr>
<td>R</td>
<td>0.94</td>
<td>55.6</td>
</tr>
</tbody>
</table>

For alphabet ‘L’

As seen in Table III, the alphabets ‘O’ and ‘U’ uttered by the speaker take the same number of frames as alphabet ‘Q’ in the dataset. Hence, they have the time similarity scores of 1.0. The alphabet ‘Q’ uttered by the speaker is found to have the time feature similarity of 75%. The highly similar time features contributes significantly in the overall similarity to make these alphabets the most similar alphabet. However, it should be noted that the alphabets ‘H’ and ‘G’ uttered by the speaker also have the time features completely similar. Hence it incorrectly makes them the fourth and fifth similar alphabets. The other results for alphabet ‘L’ show that the correct alphabet was not found to be the most similar according to the time feature. Due to the inconsistency of the time feature, it was given less significance.

1. Gesture Similarity

During the utterance of some of the alphabets, a unique mouth gesture is seen on the speaker’s face. To determine the mouth gesture similarity the presence of five mouth gestures is identified from the speaker’s image sequence. These mouth gestures are: 1. Mouth closed 2. Mouth Full Open 3. Mouth Narrow 4. Mouth Wide Open 5. Mouth Width Constant. The mouth states such as mouth open, closed, wide open narrow were identified manually and stored in the dataset. While finding the gesture similarity for the alphabetic utterance, these extracted mouth gesture states are compared with the mouth gesture states of the feature vectors in the dataset. A sample pseudo code to obtain the gesture similarity is given in Fig. 6.

Fig. 6. Pseudo code for mouth gesture similarity score

As illustrated in the pseudo code, not only the features that are present but the absent features are also matched. In this case, if a mouth gesture is absent in speaker’s features then it must be absent in the visual features of alphabet being compared in the dataset to score full points. No points are given for any mismatch in the mouth gesture status.

It is also important to match the gesture time to find the similarity. For instance, assume that the mouth closed gesture is observed at the start or end of the speaker’s utterance sequence. Such a mouth closed gesture is not the same as the mouth closed gesture seen during the utterance of the bilabial sounds such as ‘B’, ‘P’, ‘M’ etc. The ‘mouth closed’ state is seen in such bilabial sounds in the middle of the utterance in contrast to the start or the end of utterance for the ‘silence’ sound. Thus, it is crucial to match the time of mouth gesture presence in observed sequence with the time of gesture presence in the stored feature sequence in the dataset.

Using the above described method, gesture similarity scores for gestures such as mouth full open, closed, narrow, wide open and mouth width constant are obtained. The mouth gesture ‘width constant’ is determined by comparing the maximum change in the lip width with certain threshold and does not involve any event. Thus, for this gesture, the event time matching is not needed. Each of these mouth gestures is assumed to have the same precedence and thus it carries the same 20% weight. The combined gesture similarity is obtained using the following equation.

\[ Gs = \frac{1}{n} \sum_{i=1}^{n} GsFi \]  \hspace{1cm} (10)

Here, \( GsFi \) is the gesture similarity for feature \( Fi \) and \( Gs \) is the final gesture similarity.

J. Overall Similarity

The overall or final similarity is calculated by using four feature similarities described above collectively and in different proportion. These features are given variable weight in the overall similarity depending on its significance in the viseme classification. Since the lips are the most externally visible speech articulators they were identified as the most important articulator that provides crucial visual
feature information regarding the speech. The variation in the position of the articulator ‘lips’ is described by lip movement in the horizontal and vertical direction, which is presented by width and height vectors respectively. Thus, the width and height similarity are more important in viseme classification as they express the lip movement in both directions. Consequently, they share 70% of the total weight among them. For the majority of the characters the time taken to utter a character remains almost the same. Very few characters take unusually long or short time to utter it. Thus, the time factor is given less importance and carries just 10% weight. Similar to the time factor, most features for the majority of the characters remains the same. For example a mouth is neither full open nor wide for most of the characters. Thus, feature similarity is also less significant than the lip movement.

The overall feature similarity is obtained using the equation given below.

\[ Os = 0.35Hs + 0.35Ws + 0.10Ts + 0.20Gs \]  

(11)

Here, \( Os \) is the overall or final similarity and \( Hs, Ws, Ts, Gs \) are the height, width, time and gesture similarity respectively.

<table>
<thead>
<tr>
<th>Alphabet</th>
<th>Similarity Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>63.6</td>
</tr>
<tr>
<td>Q</td>
<td>62.4</td>
</tr>
<tr>
<td>U</td>
<td>52.8</td>
</tr>
<tr>
<td>W</td>
<td>53.5</td>
</tr>
</tbody>
</table>

### TABLE IV

**Overall Similarity Results Using Constant Weights for All Alphabets**

<table>
<thead>
<tr>
<th>Alphabet</th>
<th>Similarity Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>66.5</td>
</tr>
<tr>
<td>V</td>
<td>64.4</td>
</tr>
<tr>
<td>Q</td>
<td>60.5</td>
</tr>
<tr>
<td>W</td>
<td>59.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alphabet</th>
<th>Similarity Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>62.2</td>
</tr>
<tr>
<td>X</td>
<td>56.8</td>
</tr>
<tr>
<td>A</td>
<td>55.8</td>
</tr>
<tr>
<td>G</td>
<td>52.3</td>
</tr>
</tbody>
</table>

Sample of overall similarity results using fixed weights for all alphabets are given in table 4. As shown in the results, alphabet ‘O’ was correctly recognized as either ‘O’ or ‘Q’. An alphabet ‘V’ was detected as ‘Y’ or ‘V’. The last sample results for alphabet ‘H’ shows that it was not detected correctly. Furthermore, the similarity score was around 60-65%, which is not a very high similarity score. For alphabet ‘V’, alphabets ‘Q’ and ‘W’ were found to be 60% similar.

K. Smart logic for text transcription

A smart logic validates the phoneme sequence using the viseme sequence as reference, corrects phoneme mismatches where they occur and finally generates the text transcription of the uttered word.

The phoneme and the viseme sequences are the two inputs of the smart logic engine tuned to emulate the process that human beings execute when they recognize a word jointly from the sound and from the view of the uttered word. If the phoneme and viseme sequences match, then the phoneme sequence is used for phoneme-to-text conversion. Otherwise the smart logic engine tries to recover phoneme errors where they occur.

The proposed smart decision logic for audio-visual speech recognition is represented in Fig. 7. It consists of four main subsystems, namely the video fuzzifier, the audio fuzzifier, the rule-based classifier, and the defuzzifier.

![Fig. 7. Proposed smart decision logic for audio-visual speech recognition.](image)

The basic idea is that the phoneme recognition unit and the viseme recognition unit define respectively an audio class \( A \) and an image class \( V \) so that a set of rules such as this can be applied:

\[
\text{IF} \quad \text{Sound IS } A_i \text{ AND Image IS } V_j \quad \text{THEN} \quad \text{Phoneme IS } P_k
\]

\( P_{i,j,z} \) is the degree matching of \( A_i \) and \( V_j \) in the phonetic class \( P_z \).

These basic rules can be reinforced with predecessors that consider the previous and the next audio and image too:

\[
\text{IF} \quad \text{previous sound IS } A_m \text{ AND current sound IS } A_i \text{ AND next sound IS } A_n \text{ AND previous image IS } V_h \text{ AND current image IS } V_j \text{ AND next image IS } V_k \quad \text{THEN} \quad \text{phoneme is } P_z
\]

Each class \( A \) and \( V \) are fuzzy evaluations executed at audio and image recognition frontend level. So a fuzzy inference engine can be modeled as illustrated in Fig. 8.

Singleton membership functions are used to represent output classes \( P_z \) and the center-of-gravity (COG) method is then applied to defuzzify the decision.

\[
\text{Crisp_out} = \frac{\sum \text{(fuzzy_out) (singleton_position)}}{\sum \text{fuzzy_out}}
\]

(12)
A phoneme-to-text backend processing is then applied to the defuzzified outputs for text transcription. Text generated at ASR level is considered at this final stage because it is helpful for phoneme to text transcription.

Fig. 8. Inference model to score phonemes from audio and image fuzzy scores.

Rules such as those tuned for speech recognition are then applied as illustrated below:

IF

\[
\text{Sound score}_i \text{ IS high AND}
\text{lips IS L}_j
\]

THEN

\[
\text{Speaker IS S}_i
\]

\[
\text{Sound score}_i \text{ is the identification score belonging to}
\text{speaker S}_i; \text{L}_j \text{ is the lips’ features belonging to the same}
\text{speaker.}
\]

III. CONCLUSIONS AND FUTURE DEVELOPMENTS

This research aims to establish a framework for the development of bimodal speech recognition and speaker’s identification system based on a fuzzy logic fusion methodology. Speech recognition and speaker identification are executed as standalone processes on isolated uttered words. Visual recognition is also executed as a standalone process, focusing solely on lips’ movements during utterance. Fuzzy logic executes data fusion recognition at a higher level, keeping the system complexity low while boosting the performance.

The simulation of the AVSR proposed framework shows an average success of 0.86 for the audio (speech and speaker) recognition process in a standalone mode. This performance rises to 0.9 when scored data from visual recognizer are fuzzily fused with the scored data coming from audio recognizer. This demonstrates that the increase in performance for audio recognition can be obtained considering the visual information. This increase is more evident when audio noise and interferences are heavily masking speech. In such situations visual data act like a recovery information that keep high the overall recognition rate while the standalone audio recognition module performs bad.

The most important addressed task in this framework concerns data fusion of audio-visual speech information. Fuzzy logic demonstrates to be an appropriate solution as it enables to emulate the decision logic that human being successfully apply in speech recognition and speaker identification. Hence, future improvements of this framework should focus on speech analysis and feature extraction at phonetic level.

The similarity between audio and visual features at higher level allows for a better integration of the ASR/ASI and the AVR. Segmentation of speech in phonetic units for a better synchronization between audio and visual recognition tasks should also be explored for further improvements of the framework. Prosodic features such as stress and duration will be extracted from phonetic units (phonemes and allophones) for usage to improve data fusion logic. Phonetic segmentation of utterance will be an important improvement leading to a gain of unlimited vocabulary speech-to-text capabilities for the system. More information can be derived from phonetic units that can be correlated with visual information, so that higher speech recognition level will be achieved. Another long term development is to address the audio-visual natural user interface (AV-NUI), including audio-visual speech and speaker synthesis.

REFERENCES


