Discovering Patterns in Visual Speech

Stephen Cox

School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, UK.
s.j.cox@uea.ac.uk

Abstract

We know that an audio speech signal can be unambiguously decoded by any native speaker of the language it is uttered in, provided that it meets some quality conditions. But we do not know if this is the case with visual speech, because the process of lip-reading is rather mysterious and seems to rely heavily on the use of context and non-speech cues. How much information about the speech content is there in a visual speech signal? We make some attempt to provide an answer to this question by ‘discovering’ matching segments of phoneme sequences that represent recurring words and phrases in audio and visual representations of the same speech. We use a modified version of the technique of segmental dynamic programming that was introduced by Park and Glass. Comparison of the results shows that visual speech displays rather less matching content than the audio, and reveals some interesting differences in the phonetic content of the information recovered by the two modalities.

Index Terms: automatic lip reading, visual speech processing, speech recognition

1. Introduction

Visual speech recognition is gaining in importance, both as a way of making audio speech recognition more robust and also as a technology in its own right where it is often known as automatic lip-reading, ALR. However, performance of ALR is poor compared with automatic speech recognition (ASR) [1]. This is not surprising, because speech is primarily an audio form of communication, and a considerable amount of information about speech sounds is missing from the visual speech signal [2]. Although there is anecdotal evidence that deaf people can function well using lip-reading, it is often assumed that much of their information comes from combining the information from the visual speech signal with external contextual information (such as estimating what the speaker is likely to be saying) and non-verbal cues provided by the speaker. Our own (limited) studies show that there is a wide variation in performance amongst professional lip-readers when the context of the material given to them is unfamiliar and restricted [3].

It is therefore interesting to ask the question: how does the information about the speech content that can be obtained from a visual speech signal, independent of any external guiding context, compare with that available from an audio signal? This question has been implicitly answered to a certain extent by comparing performance of ALR and ASR on similar material and tasks. But this approach is problematic, because our recognition systems have been tuned over many years to audio ASR, whereas we have little experience of tuning ALR systems for optimum performance. Indeed, some fundamental aspects of current recognition systems are predicated on audio, such as decision-tree clustering, where the decision tree questions are phonetic in nature and mostly not suitable for a visual signal. Furthermore, both ALR and ASR systems are usually run with strong language models, to the extent that it is not possible to discern how much of the performance is due to the features and how much the language model.

In 2006, Park and Glass described experiments in which they attempted to discover automatically (i.e. in an unsupervised fashion) recurring patterns in speech. They found that the patterns they discovered correlated well with words and phrases [4]. Their work is the inspiration for this paper. Here, we attempt to find similar signal segments in both audio and visual representations of speech in order to compare the effectiveness of these two modalities in discovering patterns. Our interest is to see to what extent the patterns that are evident in the audio signal can also be discovered in the visual signal, and how much complementary information there is between the two modalities. Although Park and Glass’s paper inspired this work, our approach is somewhat different. Their motivation was to find recurring patterns in their audio material and to demonstrate that these corresponded to words and phrases: our motivation is to quantify and compare the power of the audio and visual modalities in discovering words and fragments of phrases in speech.

2. Related Work

It has been documented since the 17th century that there is useful information conveyed about speech in the facial movements of a speaker [5]. Early work on visual speech attempted to quantify the benefit of being able to see as well as hear the speaker when listening conditions are noisy e.g. [6, 7]. Since then, there has been considerable work on incorporating visual features into speech recognition (audio-visual recognition) [8], automatic lip-reading [1] and visual speech synthesis [9]. The visual feature vectors used here build on work in developing active appearance models (AAMs) [10, 11]. The idea of discovering automatically units of speech is based on previous work in acquiring language automatically which has been explored in [12, 13, 14].

3. Finding Matching Fragments

In this work, we use dynamic programming (DP) to identify segments of either audio or visual feature vectors that are similar to each other. These segments come from different utterances, and hence may represent the same word or phrase occurring in each utterance. Utterances are represented as a sequence of either audio or visual feature vectors—the generation of these vectors from the audio or video signal is described in Section 4. Denote the frame sequences of a pair of utterances $U_1$ and $U_2$ as $F_1, F_2, \ldots, F_{N_1}$ and $F_{1'}, F_{2'}, \ldots, F_{N_2'}$, respectively. To compare $U_1$ and $U_2$, we construct an $N_1 \times N_2$ distance-matrix $D$ in which $D(i,j)$ is the Euclidean distance between frames $F_i$ and $F_j$.

An example distance-matrix, formed from the audio representation of two utterances, is shown in Figure 1. The blue end of the spectrum represents low distances, the red end, high, and the sequence of words in each utterance is shown along the top (utterance one) and on the right (utterance two), with red lines indicating their positions in the utterances. These two utterances have three words (LETTERS, ONE and SIZE) and $F_{1'}, F_{2'}, \ldots, F_{N_2'}$ respectively. To compare $U_1$ and $U_2$, we construct an $N_1 \times N_2$ distance-matrix $D$ in which $D(i,j)$ is the Euclidean distance between frames $F_i$ and $F_j$.

An example distance-matrix, formed from the audio representation of two utterances, is shown in Figure 1. The blue end of the spectrum represents low distances, the red end, high, and the sequence of words in each utterance is shown along the top (utterance one) and on the right (utterance two), with red lines indicating their positions in the utterances. These two utterances have three words (LETTERS, ONE and SIZE) in common, and the corresponding matching sections in the distance-matrix are ringed. These three sections show distinct blue near-diagonal paths in these areas that correspond to two sequences of frames in each utterance that match well to each other. It is the task of...
The DP algorithm to discover these alignments. Aligning $U_1$ and $U_2$ using conventional DP would produce a single alignment path that joined the top-left and bottom-right corners of the distance-matrix. This path could clearly not coincide with all three of the diagonal paths mentioned above because the three common words occur in different positions and in a different order in the two utterances. What is required is a technique that compares all different sections of the two utterances to each other so that any matching sections can be discovered, regardless of their position in the utterances. Such a technique was described by Parks and Glass, and they termed it Segmental DP. In his early work on DP, Sakoe pointed out that DP must be subject to some constraints to prevent aligning sections of two utterances that are far apart in time. He suggested that a warp path originating at $(i_1, j_1)$ must satisfy the constraint

$$| (i_k, j_k) - (i_1, j_1) | \leq R,$$

where $i_k$ and $j_k$ are the $k$'th $x$ and $y$ co-ordinates of the path and $R$ is a positive integer that limits the extent of the matching. If we apply this constraint to a conventional DP match, the optimal alignment path (OAP) is contained within a diagonal band (running from the top-left to the bottom-right corner of the distance-matrix) that is restricted to a width of $2R + 1$ frames. We can add more diagonal bands of the same width above and below this band and perform DP in each band in order to match different parts of the utterances to each other. It can be shown [4] that the start co-ordinates of the bands will be

$$((2R + 1)k + 1, 1), \quad 0 \leq k \leq \left\lfloor \frac{N_1 - 1}{2R + 1} \right\rfloor$$

$$((2R + 1)k + 1, 1), \quad 0 \leq k \leq \left\lfloor \frac{N_2 - 1}{2R + 1} \right\rfloor$$

The bands are illustrated in Figure 2. The two rectangles in this figure represent a distance matrix, here $30 \times 20$ frames, and show the bands formed when $R = 4$ (left-hand side), giving 6 regions, and when $R = 2$ (right-hand side) giving 11 regions. Clearly, using more regions means that there is less possibility of a match being missed, but at an extra computational cost. We experimented with different values of $R$ in this work.

We used a standard DP algorithm with a symmetrical local constraint spanning a single frame. For a given region, a “cumulative” matrix $F$ is formed in which $F(i, j)$ holds the lowest cost of reaching cell $(i, j)$ under the local constraint. $F(i, j)$ is calculated using the following recursion:

$$F(i, j) = D(i, j) + \min \begin{cases} F(i - 1, j) + P \\ F(i - 1, j - 1) \\ F(i, j - 1) + P \end{cases}$$

Figure 1: An example distance-matrix: the blue end of the spectrum represents low distance, the red end, high. These two utterances have the three words LETTERS, ONE and SIZE in common, and the correspondence of these is shown by blue diagonal lines, which are ringed.

Figure 2: The two rectangles here represent a $30 \times 20$ frames distance matrix, and show the bands formed when $R = 4$ (left-hand side), giving 6 regions and when $R = 2$ (right-hand side) giving 11 regions.
The penalty $P$ is added to discourage off-diagonal matching, and we experimented with varying its value in our experiments. The OAP for the region is then estimated by back-tracing thorough the cells selected in equation 4, which produces the optimal alignment of the utterance frames in the region. The associated cost sequence for the $l$th region is

$$CS_l = \{ F(A_1(1), A_2(1)), F(A_1(2), A_2(2)), \ldots, F(A_1(N_a), A_2(N_a)) \}$$

where $A_1(k)$ and $A_2(k)$ are the frame-numbers of the $k$th pair of aligned frames in utterances one and two respectively and $N_a$ the number of alignments made in this OAP.

4. Data and Data Pre-processing

The dataset used here consists of audio-visual recordings of 3000 sentences spoken by a single native English-speaking male speaker. The sentences were randomly selected from the 8000 sentences in the Resource Management (RM) Corpus [15]. It is ideal for the work described in this paper because (a) it is spoken by a single speaker, eliminating inter-speaker variability, and (b) it has a medium-size vocabulary (~1000 words) and many quite long phrases that occur regularly, which aids the discovery of matching words and phrases. The audio features were captured in a specialised recording environment using a Sanyo Xacti camera in portrait orientation at 1080 x 1920 pixel resolution using progressive scan at a sampling frequency of 59.94 frames per second. Audio was captured using a clip microphone at a sampling frequency of 48 kHz.

The visual features we used were active appearance models (AAMs). These are a compact encoding of the shape and appearance information of the area around the lips. A full description of AAMs, refer to [10]. The shape, $s$, of an AAM is described by the $x$ and $y$-coordinates of a set of $n$ vertices that delineate the lips: $s = \{ x_1, y_1, \ldots, x_n, y_n \}$. All videos were down-sampled to a third of their original resolution to 360 x 640 pixels and between 20 and 30 frames from each recording session were selected for hand-labelling. In each selected frame, 111 points were labelled over the whole face to ensure stability when tracking, which was done using the inverse compositional project-out AAM algorithm [16].

Although separate shape and the appearance components of an AAM can be used as features for lipreading, combined AAM features [10] are more discriminative [17], and we used these. Velocity ($\Delta$) and acceleration ($\Delta \Delta$) features were added, and we applied a per-utterance $z$-score normalisation to remove the mean and normalize the standard deviation.

The audio features were standard mel frequency cepstral coefficients (MFCCs) with delta and acceleration coefficients appended. As with the visual features, they were $z$-score normalised. The audio features were estimated at 100 frames per second.

5. Experiments

5.1. Matching utterance selection

3000 utterances are available in the database which means that there are $4 \times 500 \times 3000$ possible pairings of utterances that could be investigated. This is computationally unfeasible because the segmental DP algorithm is expensive. However, many of these pairings do not have any words or phrases in common and so are not useful. Hence, we took the text of each of the first 1500 utterances of the set and paired it with an utterance from the second 1500 that had some words and/or phrases in common with it. We then found the set of matching phoneme sequences in the two utterances. Phoneme matching can occur across word boundaries: for instance, in the word sequences ‘SUFFICIENT’ and ‘HIGHEST FUEL’, the matching phoneme transcription is ‘t l u w a x l t’ to incorporate the final ‘t’ of the words before ‘FUEL’. This matching yields many fragments that are only two phonemes long, such as the common function words ‘THE’, ‘OF’, ‘IN’, ‘ON’ etc., so we decided to ignore matching phrases of fewer than three phonemes. We term the set of matching phoneme sequences for a pair of utterances the ground-truth sequences.

5.2. Identifying matching fragments

We experimented with values of $R$ of 15, 20 and 25. These values typically produced 10–50 regions, depending on the length of the two utterances being compared. For each of these regions, we compute the cost sequence, $CS$, as described in section 3. $CS$ enables us to estimate the cost of aligning any fragment (subsequence of matching fragments) contained within the OAP. In [4], the authors found the length-constrained minimum average (LCMA) for each region. The LCMA fragment is the cost-sequence fragment of length at least $L$ frames that has the lowest mean cost. We found there were two problems with this approach:

1. Our experience was that after defining $L$, the LCMA was almost always exactly $L$ frames in length i.e. the lowest possible number of frames, regardless of the words or phrases that actually matched in that region.
2. It was sometimes the case that there was more than one valid matching fragment in the same region. Although it was possible to exclude the LCMA fragment that had been found in the region and then re-run the LCMA algorithm to discover potential other matches, this was cumbersome.

Our solution was to adopt a more flexible approach. We find fragments in the cost-sequence for which all sequential values lie below a threshold $T = \min(\bar{\mu}_\text{global}, \bar{\mu}_\text{local})$, where $\bar{\mu}_\text{global}$ is the mean of the complete set of cost sequences (over all regions) and $\mu_\text{local}$ the mean for the region being processed. The inclusion of $\bar{\mu}_\text{global}$ means that fragments that are low cost only in relation to an OAP with a relatively high overall cost (i.e. in a region with no matches) are unlikely to be identified. Because the cost sequences are somewhat ‘noisy’, we allow two fragments that are below the threshold but which are separated by a small number ($J$) of points above the threshold to be joined together into a single fragment. We experimented with different values of $J$.

It was clear that some fragments were more likely to be genuine matches than others: we observed that the longer the matching sequence and the lower its cost, the more likely that the fragment was a genuine match to a fragment of the same phonetic transcription in the other utterance. Hence a ‘quality’ measure for each fragment discovered was defined as

$$Q = \frac{\text{Fragment length}}{\text{Fragment cost}}.$$

By rejecting fragments whose $Q$ is lower than a threshold, we can limit the number of potential matching fragments. However, in doing so, we inevitably reject some genuine matches, and this point is taken up in Section 5.3.

5.3. Matching Fragment Analysis

For each fragment in the OAP that we identify as ‘matching’, there are corresponding phoneme sequences in the two utterances. For example, in Figure 1, the leftmost ringer fragment aligns (approximately) frames 40–80 of utterance 0965 to frames 320–360 of utterance 0955. We use a time-aligned phoneme transcription of the two utterances to estimate the phoneme sequence in each utterance that best corresponds to the sequences of frames identified as matching. Suppose that this phoneme sequence for utterance one is $\Pi = \{ p_1, p_2, \ldots, p_{N_1} \}$ and for utterance two
\[ \Pi^2 = \{p_1^2, p_2^2, \ldots, p_{n+2}^2\} \]. For a perfect matching fragment, \( \Pi^1 \) and \( \Pi^2 \) would be identical and would correspond to one of the ‘ground-truth’ sequences for this utterance pair. We apply two measures to \( \Pi^1 \) and \( \Pi^2 \).

1. The Common Sequence Ratio, \( CSR \). We measure the length of the intersection of the sequences \( \Pi^1 \) and \( \Pi^2 \) (i.e. the length of any subsequence that \( \Pi^1 \) and \( \Pi^2 \) have in common) and divide by the length of the union of \( \Pi^1 \) and \( \Pi^2 \). For instance, if \( \Pi^1 = 'i y b a y n e h' \) and \( \Pi^2 = 'b a y n e h k' \), the common subsequence is 'b a y n e h', of length four, the combined sequence is 'i y b a y n e h k' of length six, hence \( CSR = 4/6 \). A perfect matching fragment has a \( CSR \) of 1, and fragments with no sequences in common have \( CSR = 0 \). We insist on measuring sequences rather than disregarding order and simply measuring phonemes common to both sequences, which means that, for instance, \( \Pi^1 = 'i y t' \) and \( \Pi^2 = 'i h t' \) have \( CSR = 0 \).

2. Precision and Recall. An ideally matched pair of utterances would return a set of fragments that were identical to the complete set of ground-truth phrases for this pair. However, if we set \( Q \) high, we tend to recover only the phrases that are ‘easier’ to match—generally the longer ones. Setting \( Q \) lower recovers some of the ‘harder’ ground-truth phrases, but also allows some spurious (usually short) matches. In addition, \( \Pi^1 \) and \( \Pi^2 \) are often inaccurate in that they consist of the ground-truth phrase but with phonemes missing or inserted at the start or the end of the phrase. Hence we use Precision (\( P \)) and Recall (\( R \)) [18] to measure the quality of the matching. For each matching fragment recovered, we find the corresponding ground-truth phrase i.e the ground-truth phrase that has the longest sequence in common with \( \Pi^1 \) and \( \Pi^2 \). We then measure the \( P \) and \( R \) of \( \Pi^1 \) and \( \Pi^2 \) using this phrase as the reference. Taking the example in 1. above, we find that the ground-truth phrase to which \( \Pi^1 \) and \( \Pi^2 \) match best is the eleven phoneme sequence ‘b a y n e h k s t m a h n e h’ (‘BY NEXT MONTH’).

The longest subsequence in \( \Pi^1 \) that matches this sequence we term \( \Omega_1 \), and so here, \( \Omega_1 = 'b a y n e h' \) and is four phonemes long (the first phoneme ‘i y’ does not match). Hence for \( \Pi^1 \), \( P = 4/5 \) and \( R = 4/11 \). Similarly, for \( \Pi^2 \), \( \Omega_2 = 'b a y n e h k' \) so \( P = 5/5 \) and \( R = 5/11 \). If we find that any ground-truth phrase has not been identified at all in the utterance matching process, this is recorded as two phrases with \( P = 0/0 \) and \( R = 0/M \) for each, where \( M \) is the number of phonemes in the unidentified ground-truth sequence. We accumulate the numerators and denominators for \( P \) and \( R \) over all fragments and utterances before finding the final mean values of \( P \) and \( R \).

Finally, we use \( P \) and \( R \) to form the balanced \( F \)-ratio: 
\[ F = 2PR/(P + R). \]

6. Results

6.1. Baseline: random segments

The two utterances we compared were selected to have words and phrases in common, and so we would expect to see some matching behaviour even if the position of a fragment within the OAP were chosen at random. Hence, as a baseline, we measure performance when the position and length of the fragments in the OAP are randomly selected. The length of a fragment depends critically on the threshold value \( Q \), and the starting point of the fragment is correlated with the length. Therefore, we modelled the starting-point and length as a different bivariate normal distribution for each value of \( Q \) and sampled from this distribution to produce a random fragment. The distributions are slightly different for audio and video OAPs, and we have used the audio OAPs here.

6.2. Comparison of audio, video and random performance

We experimented with different values of \( R \) (15, 20 and 25), different values of diagonal penalty \( P \) (0, 2, \ldots, 10) different values of the number of joining points, \( J \) (2, 3, 4 and 5), and also with the features z-score normalised and un-normalised, a total of \( 3 \times 6 \times 4 \times 2 = 144 \) conditions. The differences in \( F \)-ratio produced were very small, and it was concluded that none of these parameters had a significant effect on matching performance.

Figure 3 shows the Common Sequence Ratio (\( CSR \)) for audio and video features and for randomly selected fragments, averaged over all the 144 conditions above. We see that for audio features, the \( CSR \) increases monotonically with \( Q \) to about 0.9 for \( Q = 24 \) i.e. when \( Q = 24 \), about 90% of the phoneme sequences found in the two utterances are the same. For video, the pattern is the same but the value is 20–30% lower than the audio at higher values of \( Q \), reaching 63% for \( Q = 24 \). The \( CSR \) for randomly selected fragments also increases with \( Q \) to about 7%, presumably because the longer the sequences are, the more chance there is of some overlap of ground-truth phrases in the two utterances.

Next, we examine the retrieval performance of the three modes. Figure 4 plots the Precision (\( P \)), Recall (\( R \)) and \( F \)-ratio (\( F \)) as described in section 5.3 against the quality-threshold \( Q \) when averaged over all 144 conditions. Figure 4 shows that for the audio features, \( P \) (which has a strong correlation with \( CSR \)) increases monotonically with \( Q \), and \( R \) increases by 10% over the range of \( Q \), so that \( F \) also increases monotonically. This behaviour appears to be anomalous: normally, in information retrieval (IR) applications, as \( P \) increases, \( R \) decreases, and vice-versa. However, the number of ground-truth items (here phoneme-sequences) is not fixed in these experiments as it is in classical IR experiments. For low \( Q \) values, there are many postulated matching fragments, and each fragment is assigned a ground-truth phrase, so that there can be several fragments assigned to a single phrase. Many of these fragments are poor or spurious matches and so give low values of both \( P \) and \( R \). As \( Q \) increases, matches become better and hence both \( P \) and \( R \) tend to increase. At \( Q = 24 \), the audio achieves its best \( F \)-score of 57.1%. For video, the pattern is similar but \( P \) is lower and \( R \) slightly higher than audio. The resulting curve for \( F \) also peaks at \( Q = 24 \) with a value of 52.2%. \( P \) for the randomly chosen fragments rises slightly with \( Q \) for the same reason as \( CSR \) rises, but the \( F \)-ratio value is fairly constant over all thresholds at about 35%.
6.3. Analysis of phrases found

We were interested in the content of the matching phrases $\Omega_1$ and $\Omega_2$ identified by the audio and video features. Both modalities discovered approximately equal numbers of phrases at a threshold value of $Q = 6$ and we noted the number of different phrases found by each mode, and the number of times that each phrase was identified. Figure 5 depicts the number of phrases discovered by the audio and video modes as a Venn diagram. It shows that about 30% of the 4109 different phrases found were common to both modes. The most frequently occurring phrases discovered in both modes were two-phoneme words that often occurred in the utterances such as ‘dh ax’ (THE), ‘s iy’ (SEE/SEA) ‘f ao’ (FOR) etc., although longer words such as ‘p r oh b l ax m z’ (PROBLEMS) and ‘d ih s p l ey’ (DISPLAY) occurred often in both modes. 72.6% of the 17 400 counts of the phrases in $\Omega_1$ and $\Omega_2$ were generated by these common phrases.

An analysis of the phonetic content of the phrases that occurred only in the audio mode and only in the video mode is shown in Figure 6. We see from Figure 6 that the probability of the phonemes ‘ay’, ‘b’, ‘m’, ‘p’ and ‘s’ occurring is significantly higher in the audio than the video mode. This is interesting, because ‘b’, ‘m’ and ‘p’ are the bi-labial phonemes and are often deemed to be amongst the easiest to identify visually because they require lip-closure, which is an unambiguous gesture that is easy to recognise! ‘s’ is well-known to be one of the easiest audio phonemes to distinguish because of its pure noise characteristic but would be harder visually. Conversely, ‘ax’, ‘d’, ‘dh’ ‘l’ and ‘t’ occur more often in video than audio mode. In the last four of these phonemes, the mouth is open and the tongue is to the front, characteristics that would be well modelled by the AAM features, and this may account for the prominence of these phonemes in the visual mode. There was clear evidence that the modes behaved in a complementary fashion here, as some phrases that did not occur at all in the audio mode occurred tens of times in the video mode, and vice versa.

7. Discussion

These experiments have shown that patterns that correspond to recurring words and phrases can be discovered without supervision in visual speech features. We devised two different ways of measuring the quality of the patterns discovered: the Common Sequence Ratio $CSR$ measures the consistency of the phonetic content of two utterance fragments that match, and the $F$-ratio the ability of the audio and video modes to return accurately all potential matching fragments in two utterances. The evidence from the measurements of these two statistics is that matching patterns are (as expected) not as common in video features as in audio features. However, the video matching performs well above the random baseline we established, and in fact, the difference between the two modes is not that large. We also found clear evidence of complementary behaviour between the modes. All of this increases our belief that there are excellent prospects of improving the derivation of useful and complementary information from visual speech signals for both audio-visual speech recognition and automatic lip-reading.

There are several ways in which we want to take this work forward. Firstly, our initial results in comparing the phrases discovered by one mode but not the other are counter-intuitive, and we wish to investigate this further and also to explore what the nature of the complementary information in the two signals is, and how it can be exploited. An obvious use of this work is to compare the quality of different video front-end parameterisations for discovering information. Finally, this work was done (like that of Park and Glass) on speech data supplied by a single speaker, and we wish to explore if our findings hold up when several speakers are used to generate the features.
8. References


9. Acknowledgements

Thanks to Dominic Howell and Barry Theobald for recording the database.