Recent developments in automated lip-reading

Richard Bowden[a], Stephen Cox[b], Richard Harvey[b], Yuxuan Lan[b], Eng-Jon Ong[a] , Gari Owen[c] and Barry-John Theobald[b]

[a]University of Surrey, Guildford, GU2 7XH, UK.
[b]University of East Anglia, Norwich, NR4 7TJ, UK.
[c]Annwvyn Solutions, Bromley, Kent, BR1 3DW, UK.

ABSTRACT

Human lip-readers are increasingly being presented as useful in the gathering of forensic evidence but, like all humans, suffer from unreliability. Here we report the results of a long-term study in automatic lip-reading with the objective of converting video-to-text (V2T). The V2T problem is surprising in that some aspects that look tricky, such as real-time tracking of the lips on poor-quality interlaced video from hand-held cameras, but prove to be relatively tractable. Whereas the problem of speaker independent lip-reading is very demanding due to unpredictable variations between people. Here we review the problem of automatic lip-reading for crime fighting and identify the critical parts of the problem.

Keywords: Lip-reading, speech recognition, pattern recognition

1. INTRODUCTION

Despite promising results in laboratory conditions[1] there remains natural scepticism about the usefulness of lip-reading when used in the sub-optimal conditions associated with crime-fighting. Partly this is because of the known variability associated with human lip-readers[2] where, for example reported word-error rates vary between 0 and 70% on a 1000-word dataset measured from professional lip-readers and, partly, as with any new field, people are unsure as to which problems are challenging and which ones are solved or within-sight of being solved.

To examine these concerns in more detail, we recorded a new dataset outdoors. In Part 1 of the data a speaker was recorded with three Sony Xacti FH1 cameras at a resolution of 1920 × 1080 pixels, video frame rate of 59.94 and interlaced scan. The relative angle of each camera to the speakers face is shown in Figure 1. The three cameras were mounted on tripods at angles of roughly 0°, 30° and 45° degrees to the speaker. The speaker was given no instructions other than to repeat the sentence, so head pose is unconstrained. The videos were recorded interlaced so, as in the right of Figure 1 there was considerable interlace 'zippering" due to the different sampling times for the two fields. The task was identify the US cities: Chicago, Dallas, Houston, Los Angeles, New York, Philadelphia, Phoenix, San Antonio, San Diego, and San Jose; which are embedded in the

Further author information: (Send correspondence to RWH)
RWH.: E-mail: r.w.harvey@uea.ac.uk

Figure 1. Left Relative angle of each camera to the speakers face. From left to right are cropped-out frames recorded by cameras placed on each of the 0, 30, and 45°angles related to the speakers face. On the right are zooms of the mouth region showing motion blur and interlace "zippering"

Figure 2. Sample frames from Part 2 of the dataset recorded. This video involves two subjects engaged in conversation filmed using a hand-held camera.

carrier sentence: "I am going to (see) XXX soon / again"*. There is a total of 120 sentences in the dataset. Some examples sentences are: "I am going to see San Antonio soon;" "I am going to Houston again".

In Part 2 of the data, we recorded two people having a conversation lasting 637 seconds (approximately 10 minutes). Both subjects have incorporated words from the 10 cities in Part 1 into the conversation. The locations in the conversation where the city names are uttered are unconstrained and both subjects are left to decide the most appropriate time for mentioning a city name. Examples of different image frames are shown in Figure 2.

The video was also recorded using the Sanyo Xacti FH1 used in Part 1. In this part, the camera was hand-held as opposed to being attached on a tripod. Consequently, there are frequent camera shakes, causing potentially large and rapid displacements of the subjects locations on the image. The fast movements caused by the camera held in an unstable manner introduces motion bluring of the subject as well. Additionally, the video sequence is made more challenging due to the significant head-pose variations that is present as both subjects turn to face each other (away from the camera) during the course of their conversation.

This two-part database is the most challenging lip-reading task to date since it consists of poor quality video recorded in highly non-optimal conditions. Our aim was to use this most challenging data to determine which components of a lip-reading system were likely to be robust or brittle. We are also able to tie-up a few loose ends from earlier work[1] in which the possibility of multi-angle lip-reading was explored but with one component, pose-detection, omitted.

2. MULTI-ANGLE LIP-READING

Previously we had shown[1] that it is possible to lip-read across five different angles: 0°, 30°, 45°, 60° and 90° (full profile). It was concluded that the best angle was 30°. This is consistent with anecdotal evidence from human lip-readers but it raised an important caveat on computer lip-reading experiments which are almost invariably carried out at 0°. A technique was developed to map other camera angles to the optimal with only small loss on the systems performance, which, given sufficient training for the system on the optimal angle, can later be used for sequences recorded on other angles. However, a key missing element was a module that identified the camera angle of a new sequence, so that the corresponding mapping function can be applied.

To keep in line with previous experiments,[1] we use the same dataset, the LiLIR dataset, for training and testing the camera angle detector. The LiLIR dataset contains multi-camera, multi-angle recordings of a speaker reciting 200 sentences from the RM (Resource Management) Corpus. The dataset contains two HD cameras recording at 0° and 90°. In addition, there are three SD cameras each recording at 30°, 45° and 60° view points. All cameras were sync-locked during recording. Figure 3[1] shows sample frames from each pose in the LiLIR dataset.

SD camera angles 30, 45, 60°are chosen for the development and testing of the angle detector, due to their high confusability compared to the other two angles. We attempt to solve the problem on two separate approaches,

*The sentence grammar is also illustrated in Figure 5.

Figure 3. In the three left images we show key points tracked by AAM on each of the 30°(left), 45°(middle), 60°(right) angles. On the right are three 30° frames tracked using AAMs trained on each of the three angle. The RMSE value for each of the model is 6.77 (30°(left)), 24.75 (45°(middle)), and 31.13 (60°(right)).

one based on AAM (active appearance model) fitting error, and the other on LGO (local gradient orientation) histogram.

2.1 AAM based camera angle detection

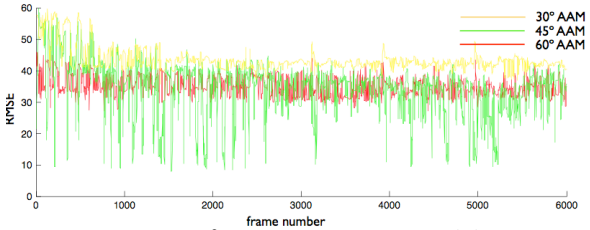
The Active Appearance Model (AAM) has been used to produce visual features for many computer lip-reading systems. It is also a face tracker which tracks key points on a face via an iterative search. Key points refer to a set of landmarks that lies along important elements on a face, for example eyes, nose, brows, lips, hairline and face contour. The aim of the search is to find the most suitable set of points, where the image it defines provides the closest match to the template image of the AAM. We denote I the warped images controlled by the key points, and T the template image. The search criterion can be described as a minimisation on root mean square error (RMSE) of $T^{(m)}$ and $I^{(m)}$ where (m) denotes the mth colour plane of the image.

$$\text{RMSE}^2 = \sum_{m=}^{r,g,b} (I(m) - T(m))^2 \tag{1}$$

where $I(r)$ is the red plane of I for example. RMSE indicates how well the tracked key points and the model fit the image.

An incoming video frame is tracked by three AAMs, one on each of the 30°, 45°, 60°angles (left of Figure 3). The frame is assigned with the angle of which its AAM model produces the smallest RMSE value. In the example on the right of Figure 3 the frame is assigned to an angle of 30°. A drawback on such an approach is that the system has to count on the reliability of the AAM tracker, which is prone to lose track if the object takes a sudden position change, or the face appearance is new to the tracker. For example, in the left of Figure 4, three trackers are used to track a video sequence on 45°angle, on which 45°AAM tracker should have the best fit (lowest RMSE). In reality, 45°AAM is unreliable and often loses track. This results in a large RMSE value and can cause the pose detector to produce erroneous decisions. Examples of the loss of tracking are available on video separately.

The angle detection system is evaluated on a frame by frame basis using the measurement of accuracy, i.e., the percentage of predicted angle being correct. The results are presented in a form of confusion matrix to



True class \ predicted class	30°	45°	60°
30°	0.9928	0.0006	0.0012
45°	0.0069	0.9890	0.1581
60°	0.0003	0.0104	0.8407

Figure 4. Left: RMSE on a 45° sequence tracked by AAMs trained on various angles. Each line colour denotes a different AAM. Right: Confusion matrix for the tracker on a frame-by-frame basis.

demonstrate how likely the system can confuse one angle with another. For example, on the right of Figure 4, the entry on first row and first column is the proportion of frames that are detected as 30° and are actually 30°; the entry on first row and second column is the proportion of frames that are detected as 30° but should be 45°, and so on. It is noticed that the system is at its lowest performance on frames of 60°, due to the challenge of tracking on this pose.

2.2 3.2 LGO based camera angle detection

The Local Gradient Orientation histogram (LGO), is a different approach to camera angle identification. The principle is to compute an LGO feature which is encoded with orientation information for the key region of each frame. The feature is then compared with other frames with known orientation to decide on its camera angle.

LGO is a type of SIFT[3] feature that has been used for head pose estimation.[4] Unlike a standard SIFT feature, the feature-patch is not rotation invariant. An LGO is computed by, first, localising the face in the image via a standard face detector (although we use an LP tracker). The face patch is then equally divided into 4×4 grids. On each grid the orientation is computed on each pixel and is quantised into 8 bins. An orientation histogram is then built for all the pixels inside the grid. This results in a $4 \times 4 \times 8$ matrix which is then smoothed and normalised to unit length. The final feature is a 128-dimensional vector encoded with orientation information. For each camera angle, a class centre is computed using the first 1000 frames of the recording. For a new frame with unknown angle, its LGO feature is computed, and is assigned to the angle of the nearest neighbour class centre. The performance of the LGO based system is reported in the form of confusion matrix and can be found in Table 1.

Table 1. Confusion matrix of LGO face tracker

		predicted class		
		30°	45°	60°
True class	30°	0.998775	0.0004	0.001648
	45°	0.0008611	0.9996	0.002131
	60°	0.000364	0	0.996221

The LGO method has higher accuracy. This is partly because the method does not require a very accurate face tracker - a rough estimate on the face location is sufficient; and partly due to the design of the LGO feature which itself is an orientation-based feature. A natural question is the performance of the face tracker on the Outdoor dataset (Part 1). This is shown in Table 2. The performance is quite acceptable which leads us to conclude that tracking an agile face even in poor conditions is not a problem that is likely to limit performance of a practical lip-reading system.

Table 2. Confusion matrix of the LGO based camera angle detection system tested on the outdoor dataset (Part 1)

		predicted class		
		30°	45°	60°
True class	30°	0.9945	0.0057	0.0007
	45°	0.0039	0.9828	0.0332
	60°	0.0017	0.0115	0.9661

3. KEYWORD SPOTTING RESULTS ON THE OUTDOOR DATASET, PART 1

The lip-reading system was trained and tested as a speaker-dependent system. The system consists of a set of Hidden Markov Models that are manipulated via HTK.[5] A fixed grammar (fg) model consistent with the data as shown on the left of Figure 5 was applied to constrain the recognition outputs to be valid. A more generalised model called the keyword spotting (ks) model was also applied to the system. Compared to the fg model, the latter is more flexible and can accommodate any words that fall outside the pre-defined vocabulary. As illustrated on the right of Figure 5, this grammar makes the assumption that a speech utterance consists of background noise, other speech (or extraneous words) and keyword. The ks grammar model is expected to give a lower performance to the fg model as a sacrifice for being closer to real application sentences.

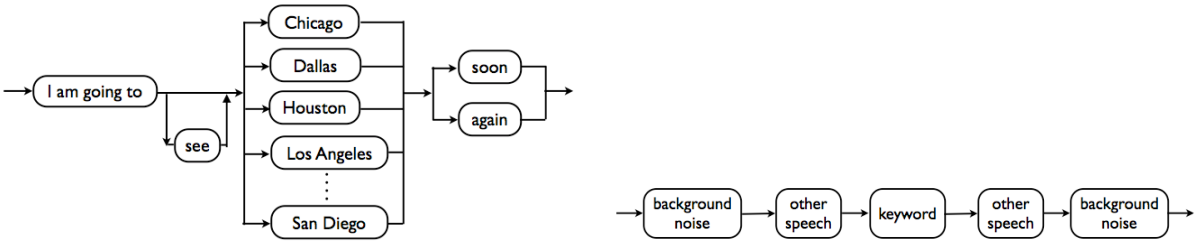


Figure 5. Network of the fixed grammar (fg) model used by our lip-reading system (left) and unconstrained grammar (right) used in the keyword-sootting configuration.

The system was evaluated using five-fold cross-validation, where the system is trained using data from four folds, and tested on the held-out fold. The procedure was repeated five times, alternating the held-out fold at each time.

Various visual features was tested to study their robustness in outdoor conditions. These features include AAM shape features (shape), AAM appearance features (app), and PCA-combined shape and appearance features (csam). We report system word accuracy rates across all words including both keywords and non-keywords, denoted overall *word acc*. We also report word accuracy rate on only the keywords, denoted *keyword acc*.

Firstly the system was tested as a single camera angle system, i.e., training and testing data are both from the same camera angle. Both the fg and ws grammar model were applied and the performances are plotted in Figure 6.

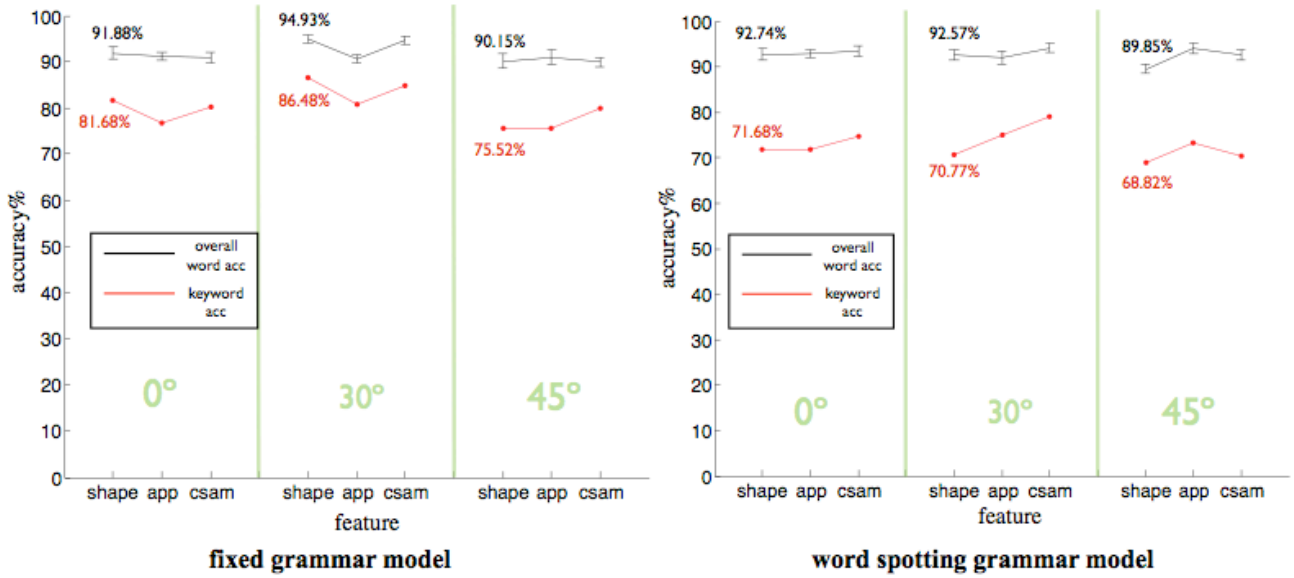


Figure 6. System performance when trained and tested on the same camera angle. Here we show the word accuracy on all words (overall *word acc*) and on keywords only (*keyword acc*), when adopting a fixed grammar model (left graph) and a word spotting grammar model (right graph) respectively.

We have also tested the system when trained on one camera angle and tested on a different angle. We utilised a linear mapping[1] to learn a correspondence between features of two angles. There is a significant drop in accuracy when the training angle and testing angle are different. This is likely to be caused by the head pose changes frequently occurring during speech. The head pose change introduces non-linearity and makes a linear mapping not appropriate for the situation.

4. DISCRIMINATIVE TRAINING

The classifier used in our lip-reading system, hidden Markov models (HMMs), are trained using a conventional generative training approach. This training, called maximum likelihood (ML) training, produces a generative Gaussian model, which maximises the likelihood of training data being output from the model. ML is a non-discriminative learning method because it tries to model the data distribution instead of directly separating classes. ML makes some assumptions that often do not hold for speech, for example, data are from a family of known distributions (often Gaussian), and unlimited training data. These assumptions could lead to failure to produce an optimal result.

In discriminative training, the step of modeling the distribution is avoided. To optimise a model, an objective function based on some criteria is formulated to reward parameters leading to correct classification, and penalise those liable to confuse classes. Some training criteria for HMMs include the maximum mutual information (MMI), minimum word error (MWE) and minimum phone error (MPE). We chose the MPE criterion here, the objective function of which is to minimise:

$$\sum_{r=1}^{R} \sum_{H} \Pr(H|\mathbf{O}_r, M) L(H, H_{\text{ref}}) \tag{2}$$

where H_{ref} is the reference hypothesis (the correct transcription) for rth training data, and H is a set of confusable hypotheses (or incorrect recognitions) that the classifier is likely to produce. $L(H, H_{\text{ref}})$ is the Levenshtein edit-distance between a correct and an incorrect recognition. $\Pr(H|\mathbf{O}_r, M)$ denotes the probability of an incorrect hypothesis H being generated from the model M with observation $\mathbf{O}_r$. A learning algorithm searches parameter space in model M and finds a set of values that minimises the above objective function.

Table 3. System performance with conventional ML generative training and MPE discriminative training.

	ML	MPE
word-pair (word acc)	19.66%	30.20%
bigram viseme (viseme acc)	45.55%	49.90%

For this experiment, speech from speaker 1 of the LiLIR audio-visual dataset is used and the performance is shown in Table 3. Two training strategies, one using ML generative training and other using MPE discriminative training, are tested. A language model is required during both the MPE training and later the recognition stage. Two language models, word-pair grammar and bigram viseme grammar model are applied. For the former model, the system outputs sequences of words, and the performance is measured by word accuracy. The bigram viseme model is built on training data, on which the system outputs a viseme sequence, and here we report viseme accuracy.

When a word-pair grammar is in use, MPE training outperforms ML training with a 53.6% relative increase in word accuracy. When using a bigram viseme grammar, we notice there is little improvement regardless of which training strategy is applied. Tables 4 and 5 are the viseme confusion matrix for ML and MPE training respectively. They show that the MPE training cannot consistently increase the recognition rate across all class categories: the accuracy of some visemes drop with the training. The difference in word error rate is attributed to the difference in error metrics: MPE includes a Levenshtein distance so may be said to include a primitive word model - the learning method with the word model is the one that performs best on words.

5. LANGUAGE MODELING

We know that a language model (LM) can strongly effect what is output from a speech recognisor. An over-generalised language model leads to an underperforming system. On the other hand, if the language model overfits the data, the resulting system will generalise poorly. We have studied this issue by applying three language models obtained under different situations. The data are the same as those in Section 1: there are 160 training sentences and 40 test sentences. The vocabulary size is just under 1000 words. Two word bigram language models were derived from these data. The first one was computed using only the training data (160 sentences) and is denoted

Table 4. Viseme confusion matrix when using ML training. The first column denotes true viseme classes, and the first row denotes predicted classes. The overall viseme accuracy is 45.55%.

	ah	eh	f	ao	t	uh	w	k	p	iy	aa	ch	oo	delete	c%
ah	28	4	1	3	6	0	1	3	0	10	2	1	1	85	44.40
eh	2	87	0	0	1	0	0	2	0	13	2	0	2	35	78.40
f	0	0	55	0	1	0	1	1	0	0	0	1	0	4	91.70
ao	2	0	0	16	1	0	1	1	0	1	0	0	0	11	72.70
t	1	4	1	4	241	0	0	3	0	4	0	8	1	88	87.30
uh	3	0	0	1	2	10	0	0	1	0	1	0	0	9	52.60
w	1	0	4	2	1	0	44	2	1	0	0	2	0	36	77.20
k	2	3	1	4	13	2	1	91	0	7	2	2	2	119	67.40
p	1	0	0	0	0	0	0	0	93	2	0	0	0	12	96.90
iy	6	4	0	1	2	1	0	2	0	93	1	2	3	55	78.80
aa	0	0	0	1	1	0	0	0	0	0	5	0	1	4	62.50
ch	0	0	0	0	3	0	1	0	0	0	0	28	0	3	84.80
oo	4	0	0	0	0	2	0	0	0	0	0	0	8	8	57.10
Insert		5	4	2	0	21	1	2	10	3	14	0	6	3	

Table 5. Viseme confusion matrix using MPE training. The first column denotes true viseme classes, and the first row denotes predicted classes. The overall viseme accuracy is 48.89%.

ah	eh	f	ao	t	uh	w	k	p	iy	aa	ch	oo	delete	c%	
ah	39	4	0	2	6	1	4	11	1	7	4	6	7	56	42.40
eh	4	90	0	2	0	0	1	1	2	13	0	2	3	28	76.30
f	0	0	60	0	0	0	2	1	0	0	0	0	0	1	95.20
ao	2	0	0	21	0	1	1	0	0	1	0	0	1	6	77.80
t	4	6	2	3	238	5	3	6	0	10	2	9	1	75	82.40
uh	0	0	0	3	2	8	2	1	1	1	0	1	0	9	42.10
w	3	0	1	2	1	0	65	3	0	1	0	0	0	17	85.50
k	5	4	4	6	10	1	5	104	1	9	2	3	7	93	64.60
p	0	1	0	2	1	0	1	1	93	2	0	0	0	7	92.10
iy	10	6	0	1	2	0	1	5	2	92	1	3	3	47	73.00
aa	0	1	0	0	1	0	0	0	0	0	4	0	0	6	66.70
ch	0	0	0	1	1	0	0	0	0	0	0	31	0	3	93.90
oo	0	0	0	0	0	1	0	0	0	2	0	0	15	4	83.30
Insert	13	13	3	10	15	3	13	22	3	21	5	9	6		

LM_tr, the second one is built on both the training and testing sentences (200 sentences), and is denoted LM_all. The third word bigram model, word-pair grammar model which was also used in Section 1, was derived from the sentence patterns that were used to generate the 2800 sentences in the Resource Management corpus. On average each word in the vocabulary can be followed by 60 other words. We denote this model as LM_wp. Table 6 reports word accuracy on each language model while testing the system on the test data.

Bigram statistics estimated by both LM_tr and LM_all model are sparse and insufficient due to the limited number of sentences used to train the model. LM_tr contains 1075 bigram pairs and LM_all with 1309 pairs. Furthermore, because LM_all also learns the test sentences (20% of the data), this model overfits the test set, and it is reflected by its high accuracy rate (66.38%). The model, however, does not generalise and its performance will plunge if unseen data is used in the test. In comparison, LM_wp is the best of three models and is based on a large dataset. It should be noted that in large speech recognition systems, a language model is often trained

]

Table 6. Word accuracy with three different language models.

	LM_tr	LM_all	LM_wp
word accuracy%	17.95	66.38	19.66

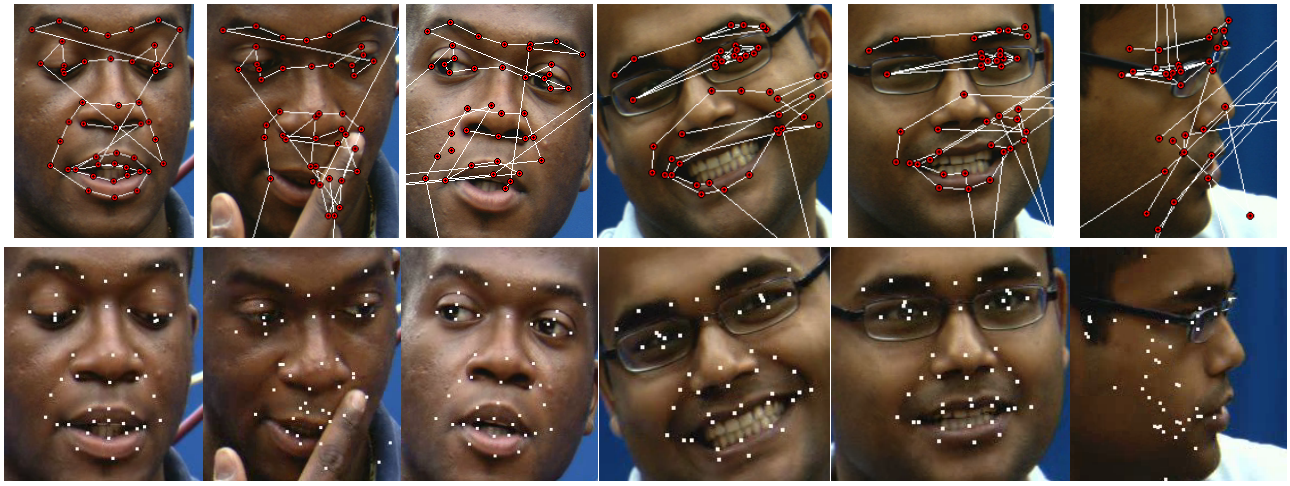


Figure 7. Tracking results at different frames for sequences with subjects engaged in naturalistic conversation, where time progresses from top to bottom (i.e. top frame is from the earlier part of the sequence and bottom frame from the latter part of the video sequence). Shown are the performance of the previous LP tracking method (top) and the new method (bottom).

using a sufficiently large amount of text in order to gain a good estimation of statistics.

Compared to word bigram models, viseme bigrams are less likely to encounter the problems mentioned above. As a result of a small vocabulary size, 14 visemes in our case, there is always sufficient viseme pairs even with a small dataset. In some situations, for example when evaluating a new feature against a test set, a viseme bigram model could be a more reliable choice than word bigram models.

6. ROBUST TRACKING OVER POSE AND EXPRESSIONS

In the past, the tracking of facial features were done using the method of Linear Predictors (LP)[6] which provide a linear mapping from support pixel (i.e a sparse template) differences to the displacement vector of a tracked facial feature. To overcome the non-linearities due to facial deformations, a set of relevant template pixels called support pixels were automatically selected. However, this only works when pose changes are limited. When pose variations become significant, it is not possible to find the above subset of "linear" support pixels hence there will be tracking failure when large pose variations are present in the target subject.

To overcome this, a non-linear mapping based on regression forests can be used. Separate regression forests are then organised in a hierarchical configuration to provide accurate and robust tracking. Both the forests and hierarchical structure are learnt in an automated way. Importantly, the resulting learnt hierarchical forest tracker is able to track facial features across large changes in pose and expression without the need for explicit 3D models, pose-specific models or pose estimators. Experimental results on the tracking are shown below and demonstrate that the system is capable of coping with large pose and expression changes.

The tracking results can be seen in Figure 7, where the tracking results of the previously used linear predictors and currently proposed method are compared. The new method based on the non-linear regression forests is far more robust to pose changes than the previous approach. Importantly, the mechanism providing this increase in robustness is not specific to only pose changes. This was also carried out on a number of long sequences of subjects with naturalistic conversation. The method is robust to occlusions and pose changes whilst the linear predictor method suffers catastrophic failure of multiple facial feature points. Any other highly non-linear variations can, in practice, also be modeled and accounted for using the new method. Consequently, in the future, we will attempt to quantify the performance of the system due to directional lighting changes that can occur in both indoor and outdoor environments using exactly the same mechanism.

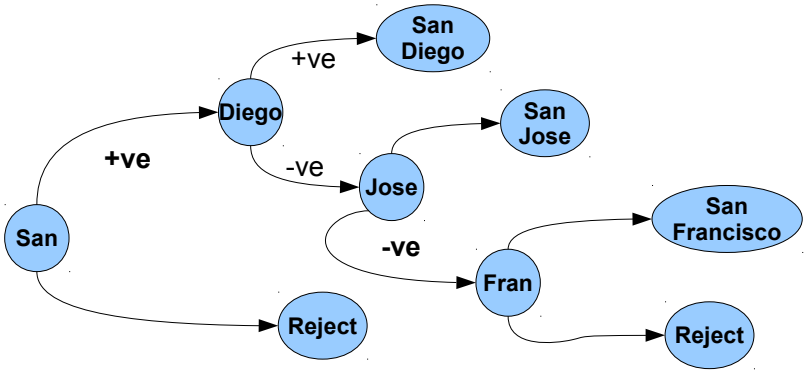


Figure 8. An example of an SP-Tree for discriminating between San Diego, San Jose and San Francisco.

7. SEQUENTIAL PATTERN HYPERTREES WITH INTERVALS

In this section, recent developments on the use of sequential pattern hypertrees (SP-hypertrees) will be described. A sequential pattern is an ordered sequence of feature subsets (called itemsets) that can be used to capture a unique spatio-temporal signature for a particular visual word that we want to recognise or spot. Previously, different sequential patterns from a set of classes were merged together into a sequential pattern tree (SP-Tree) structure. This allows initial common subsequences across different sequential patterns to be shared, allowing for a more efficient computational structure whilst reducing the complexity of the learning process. Here, sequential patterns that have common initial subsequences can be merged together into a initial branch, before separating into different branches to represent their latter subsequences. An example is the words: "San Diego", "San Jose" and "San Francisco". Their respective sequential patterns will share a common branch of "San" before branching into sub-branches of "Diego", "Jose" and "Francisco" respectively (Figure 8).

However, one disadvantage is that sequential patterns can only share initial common subsequences. Should the common subsequence occur in the middle of a set of sequential patterns (e.g. "ton" shared by "Kingston", "Wintonbury", "Tonville"), no sharing can occur. To tackle this, the sequential pattern hypertree (SP-Hypertree) is needed. In the SP-Hypertree model, a set of sequential patterns are again structured into a tree-like structure. However, the difference here is that each node of the tree represents a sequential pattern (as opposed to a just a component of the sequential pattern in the SP-Tree). Crucially, the node's sequential pattern is an arbitrary extension of its parent's sequential pattern. Such extensions will then allow subsequence sharing over a set of SPs to occur at any part of the member sequential patterns. Using the example above of "Kingston", "Wintonbury" and "Tonville", all three words' sequential patterns can be merged into a tree structure, where the parent node will encode the subsequence for ton, before branching into its child nodes with SPs for "Wintonbury", "Tonville" and "Kingston" respectively (see Figure 9).

Additionally, we further improve the sequential patterns within the SP-Hypertrees by including time intervals between the different itemsets of a particular SP. We define this form of sequential patterns as interval sequential patterns or interval-SPs. To detect an interval sequential pattern, we require that the time between the itemsets to be constrained to lie between a particular period. One consequence is that the entire SP is only valid for a particular period. This time-constraint mechanism in the interval SP allows us to exploit the fact that different words of interest may have varying duration (i.e. some words are longer than others). The SP-Hypertree with intervals described above can then be learnt in a similar way to that of the SP-Tree. An algorithm firstly learns SP-Hypertrees in a similar manner to decision trees. Additionally, whilst structuring the hypertree, the algorithm also determines the optimal time intervals between successive itemsets in the sequential patterns of each node in the hypertree model. To increase the robustness and accuracy of the word classifiers/spotters, multiple hypertrees are learnt and combined into a forest of interval SP-Hypertrees within a Boosting framework.

To evaluate the effectiveness of the new interval SP-Hypertrees, initial experiments were run on performing visual word classification using the OuluVS dataset. The OuluVS dataset is a lip-reading database that consists of 20 subjects, each repeating ten different phrases five times. The phrases are as follows: "Excuse Me", "Goodbye", "Hello", "How are you", "Nice to see you", "See you", "Sorry", "Thank you", "Time" and "Welcome". The goal of the experiment is then to learn a forest of interval SP-Hypertrees that is able to classify visually the above 10 phrases. To achieve this, a grid of pixels around the mouth of each subject is initially tracked. Following this, simple comparative features are used to form a feature-vector that models the shape and appearance of the mouth in a single frame. Each video sequence is then represented by a sequence of the above binary feature vectors. Following this, leave-one-out cross validation subject dependent experiments were performed for visually classifying between 10 phrases. This was achieved by learning a multi-phrase classifier as a forest of interval SP-Hypertrees. This forest, given a novel feature vector sequence from an input video will provide a phrase label. For comparison, we have also trained the previously used SP-Tree model that does not have the interval or arbitrary sharing mechanisms. We find that the interval SP-Hypertree model is significantly more accurate than the original SP-Trees. The average classification rate for the SP-Tree model is 60.73%, whilst the average classification rate for the new SP-Hypertrees is 83.5%, yielding 22.7% increase in accuracy. We find that the accuracy for the SP-Hypertree method is higher for all subjects, as can be seen in Figure 10.

8. WORD-SPOTTING RESULTS

Using the tracking method in Section 6 and the classification method in Section 7, the conversation video sequence known as the Outdoor Part 2 data was processed. Firstly, the mouth region of both subjects were automatically located by tracking the mouth corners. In total, 38220 image frames were tracked, with a minimal number of manual re-initialisation (approximately 5 per subject). Example frames of the tracking can be seen in Figure 11.

Having tracked the mouth region for both subjects, a grid of 20×20 pixels distributed evenly within the mouth region was defined. Following this, a set of pairs of random grid pixels were extracted. Using these pairs, intensity binary comparisons (i.e. is one pixel brighter than the other) are performed, and the results collated into a binary feature vector. This feature vector is similar to that used in the experiments described in Section 8. Following this, a set of example utterances of the city names were extracted. Additionally, a small random number of negative example sequences were also extracted from where the subject was talking but did not utter any of the cities of interest. Using these example sequences, a set of 100 SP-Hypertrees were trained to perform word-spotting of the ten city names of interest. The task of the system is then to use the trained SP-Hypertrees to detect and recognise and label these sequences in a continuous video stream. It was found that the system was able to correctly recall all instances when the subject uttered a particular city name in the continuous video sequence. Example frames of the detections can be seen in Figure 11.

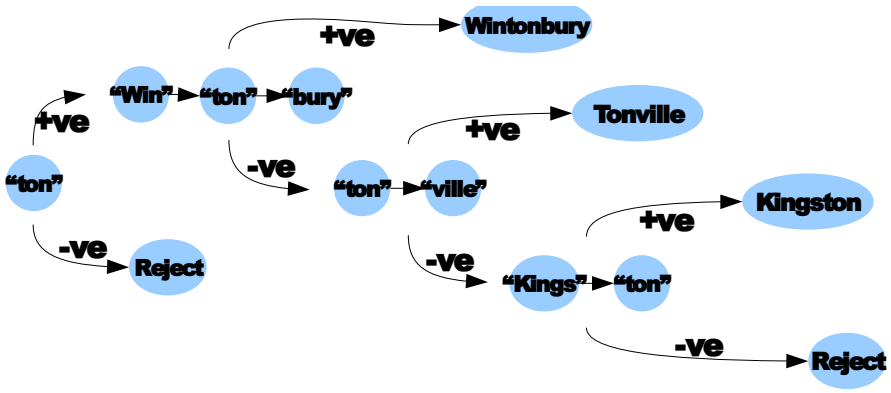


Figure 9. An example of an SP-Hypertree for discirminating between "Wintonbury", "Tonville" and "Kingston"

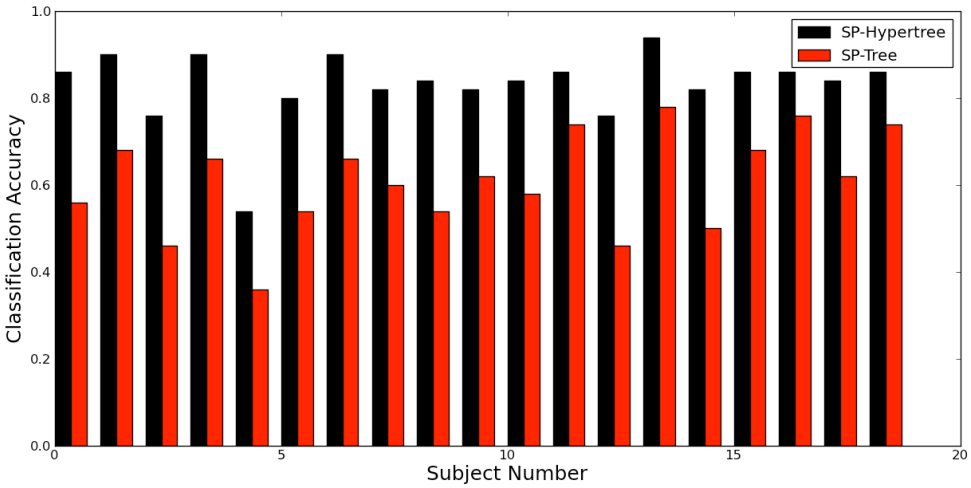


Figure 10. Subject dependent classification accuracy for the OuluVS dataset using the SP-Hypertrees and SP-Trees across 19 different subjects.

Figure 11. Example frames showing different city names that were visually spotted by the SP-Hypertree system.

9. CONCLUSIONS AND FUTURE WORK

A new two-part outdoor audio-visual dataset has been recorded. The first part involves a multi-camera dataset of a subject reading a set of 120 different carrier sentences. Embedded within each sentence is the name of one out of ten possible cities. Here, a total of three different view angles were recorded using Sanyo Xacti FH1 cameras mounted on tripods. The second part of the dataset is a video sequence of two subjects engaged in unconstrained conversation that lasts for approximately ten minutes. Within this conversation, both subjects have mentioned the names of the ten cities multiple times. This sequence was recorded using the same camera as before, but with the camera being handheld instead of mounted on a tripod.

The purpose of the first part of the dataset is to evaluate our lip-reading system for its property on view-independency, under challenging outdoor environments, and with less constrains on head pose changes. To tackle this, the lip-reading system has been further extended with the addition of a camera angle detector.

Two different approaches, one based on AAM tracker fitting error, and the other based on the local gradient orientation (LGO) histogram, were both examined. The result indicates that the LGO based method is much more reliable on detecting the camera angle to a speaker, due to the features sensitivity to pose change, and a low requirement on the trackers performance. Results showed that the system can reliably detect pre-defined camera angles, and across angles the system retains a similar level of performance. Experimental results also reveal that when head pose variation is introduced, the assumption of a linear relationship between corresponding features of two angles is no longer valid.

Experiments were also conducted to study the effect of language models. These models were constructed from corpora of different size. It has revealed that the word-pair grammar model is the most suitable out of three models tested. The viseme grammar model, on the other hand, is less affected by the corpus size. We can now extract a viseme level lattice formed during the recognition. We call this an H lattice. A step following will be to model the recognition errors, which will also be in a form of lattice and be composed with the H lattice. The resulting lattice will be constrained by the dictionary and language model.

We have also introduced an improved tracking method that is able to cope with a set of challenging conditions ranging from large expression changes to head pose variations. This method was then demonstrated to be superior to the original tracking method using linear prediction (LP method). Example tracking sequences have shown that the newer method can successfully track across sequences with large head pose variations whilst the original method suffered from catastrophic tracking failures.

We have also detailed an improved model for visual word classification based on interval SP-Hypertrees that have incorporated temporal-intervals within sequential patterns. Additionally, the method allows sequential pattern-based signatures that share common subsequences at arbitrary locations to be grouped together in an efficient manner. This is in constrast to the earlier SP-Tree model, where only SP patterns that share common initial subsequences can be combined together. Experimental results using the OuluVS dataset has shown that the new method gives an improvement in accuracy of 22.7% over the original SP-Tree model.

Both the improved tracking method and the SP-Hypertrees were then used to tackle the problem of word-spotting in a video sequence involving two subjects in unconstrained conversation. The new tracking method was shown to be able to efficiently track the mouth region of both subjects in the video sequence, despite the presence of significant camera motion that resulted in severe motion blur and large head pose variations. Using the tracking results, subsequently extracted feature vectors, and the SP-Hypertrees, we demonstrated the ability to automatically perform word-spotting of ten cities for both subjects within this challenging video sequence.

In conclusion, we perceive that many of the problems associated with lip-reading are not insurmountable. However the critical problem of person dependence remains and we look to forward to tackling this future work.

ACKNOWLEDGMENTS

The work described in this paper has been funded by a number of sources included the Engineering and Physical Research Council (EPSRC).

REFERENCES

[1] Bowden, R., Cox, S. J., Harvey, R. W., Lan, Y., Ong, E.-J., Owen, G., and Theobald, B.-J., "Is automated conversion of video to text a reality?," in [*Optics and Photonics for Counterterrorism, Crime Fighting Defence VIII*], Lewis, C. and Burgess, D., eds., **8546**(85460U), SPIE, Bellingham, WA (2012).

[2] Lan, Y., Harvey, R., and Theobald, B.-J., "Insights into machine lip reading," in [*Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*], 4825 –4828 (March 2012).

[3] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision* **60**(2), 91–110 (2004).

[4] Murphy-Chutorian, E., Doshi, A., and Trivedi, M., "Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation," in [*Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*], 709–714 (2007).

[5] Young, S., Evenmann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Valtchev, V., and Woodland, P., *The HTK Book (version 3.2.1)* (2002).

[6] Ong, E., Lan, Y., Theobald, B., Harvey, R., and Bowden, R., "Robust facial feature tracking using selected multi-resolution linear predictors," in [*In Proceedings of the International Conference Computer Vision (ICCV)*], (2009).