DESIGNING RELEVANT FEATURES FOR VISUAL SPEECH RECOGNITION

Eric Benhaim[⋆†], *Hichem Sahbi* [⋆]

[⋆] Telecom ParisTech
CNRS-LTCI
46 rue Barrault, 75013 Paris, France

Guillaume Vitte[†]

[†] Parrot S.A.
174 quai de Jemmapes, 75010 Paris, France

ABSTRACT

Automatic speech analysis is currently evolving towards hybrid systems that combine both visual and acoustic information. This is due to limitations of existing acoustic-based approaches and the need for robust speech recognition systems working under extremely challenging conditions including noisy environments.

We introduce in this paper a novel visual speech recognition approach, based on string kernels and support vector machines. The main contributions of this work include (i) the design of a similarity function, based on string kernels, that models the dynamics as well as the appearance of visual features in talking faces and (ii) a kernel combination procedure based on multiple kernel learning, that makes visual feature selection effective and also more tractable. Experiments conducted, on a standard digit database, show that the proposed algorithm outperforms current state-of-the-art methods.

***Index Terms*—** Visual speech recognition, string kernels, support vector machines, visual feature selection, kernel combination.

1. INTRODUCTION

Speech perception is a multimodal process which integrates audio and visual information [1]. Over the past twenty years many authors have focused on improving automatic speech recognition (ASR) systems by incorporating visual features jointly with acoustic signals. Improvements are substantial especially in noisy environments, where conventional acoustic-based ASR systems perform badly [2]. The well known McGurk effect (shown in [3]) illustrates the importance and the interaction between the two modalities by conflicting visual stimulus example. For instance when a voice saying */ba/* was presented with a face articulating */ga/* most subjects heard */da/*.

The growing trend in this research area reflects the need to design robust systems for real-world applications including multimodal person identification, expression analysis, surveillance, and human machine interaction with multimodal remote control or speech enhancement. Related works can be divided into two research fields: automatic audio-visual speech recognition [4, 5, 2] (AVSR) which merges both modalities and visual speech recognition [6, 7, 8, 9] (VSR) also referred to as lip-reading.

Despite the increasing interest in this domain, performance of automatic lip-reading systems remain insufficient. The major difficulty stands in extracting relevant visual features. Moreover, unlike ASR systems, there is some inherent inter-speaker variability, in lip-motion and appearance, which causes significant drop in performance, when speaker utterances are classified with visual models trained on others. Our main effort is dedicated to extract speech-relevant visual features while being speaker-independent.

Several types of features, for visual speech recognition, have been proposed in the literature and are commonly grouped into two categories. Bottom-up (pixel-based) approaches compute mouth appearance directly from pixels within a region-of-interest (ROI). On the other side, top-down (model-based) approaches are based on geometric features and require mouth shape tracking. It is observed [2] that lip movements, tongue and facial muscles are more important than appearance, but they do not necessarily incorporate speech-relevant information. A model-based approach that combines shape and appearance aspects was first used for lip-reading in [8]. Authors of [6] compared the influence of both aspects and concluded that appearance is more informative than shape. However appearance-based features [2, 4] describe global mouth information and disregard local changes due to pronunciation. The work in [7] models these local changes using patch-based subdivision of the ROI. Other methods, including active appearance model (AAM), are also well suited to locate and track facial feature points during speech but they are also speaker dependent. Recently, a ROI-based AAM was introduced in [5] to extract speech-related information confined in lower face parts; a speaker-dependent normalization was also used in order to better encode speech unit variability.

In this paper, we introduce a novel visual speech recognition algorithm, based on support vector machines (SVMs). This work includes two main contributions

- We use string kernels in order to measure the similarity as well as the dynamics of visual (appearance and motion) feature sequences, in talking faces. The application of string kernels in visual speech recognition is novel and also natural in order to handle non vectorial data, i.e., sequences of different lengths.
- And we propose a novel training procedure, guided by the maximization of performance of SVMs, via multiple kernel learning (MKL). Our solution is based on an efficient and also effective greedy strategy for feature and string kernel combination.

Our choice was also motivated by the good generalization capability of SVMs in order to handle few training examples in high dimension data, in contrast to generative models, including Hidden Markov Models (HMMs). Indeed, discriminative machine learning methods, particularly kernel machines, are receiving increasing attention for ASR [10, 11, 12].

The rest of this paper is organized as follows: In Section 2 we first introduce our method for visual speech representation by an original design of structured features. Section 3 addresses the problem of visual speech classification and describes the proposed feature combination approaches. Experimental results demonstrating state-of-the-art performance are described in Section 4 before we conclude in Section 5.

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

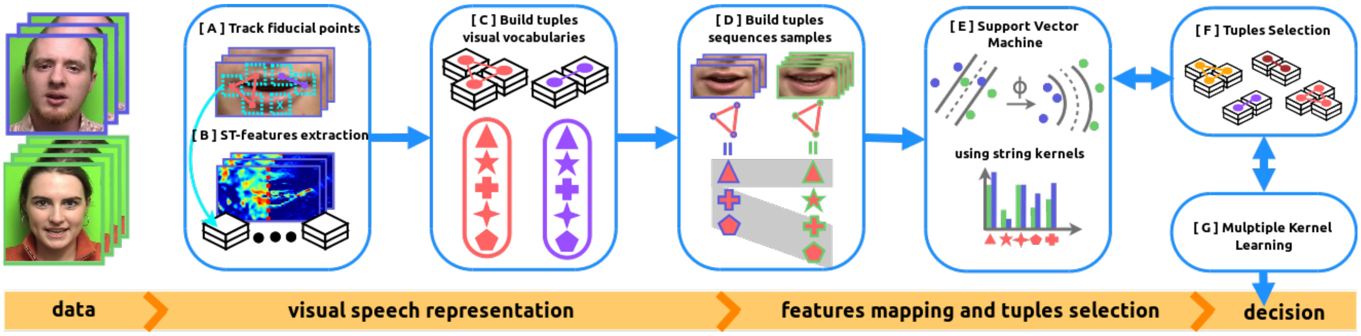

[A] Track fiducial points
[B] ST-features extraction
[C] Build tuples visual vocabularies
[D] Build tuples sequences samples
[E] Support Vector Machine
using string kernels
[F] Tuples Selection
[G] Mulptiple Kernel Learning
data
visual speech representation
features mapping and tuples selection
decision

Fig. 1. **System overview**: [A] Interest points tracking using deformable model, [B] Local features extraction from each point neighborhoods, [C] Tuple codebook learning, [D,E] Visual speech classification, [F] Tuple selection strategies, [G] MKL for kernel and structured feature selection.

2. VISUAL SPEECH REPRESENTATION

In this section, we present our method for visual speech representation, based on structured visual features that capture speech-induced variability. We compute histogram based descriptors in local neighborhoods of tracked interest points (see Fig. 1-AB and Section 2.1) and we describe visual speech using sequences of quantized structured features (see also Fig. 1-CD and Section 2.2).

2.1. Facial Interest Point Tracking and Description

Many approaches for detecting and tracking facial landmarks have been proposed in the past few years. A popular choice is the deformable template model called active appearance model [13] (AAM) which is built by applying PCA on a set of annotated training images to model their non-rigid shape and texture variations. Shape refers to the relative location of face landmarks including mouth and eye corners, whereas texture refers to the visual appearance of faces. During tracking, a fitting procedure [14] minimizes a squared loss between a new face image and the face model generated by AAM. In order to initialize the fitting procedure, state-of-the-art face detector, based on AdaBoost [15], is used and provides reliable performances and robustness to changes of illumination.

Appearance and motion are the key components of visual speech analysis. In order to characterize appearance and motion, we extract histogram-based descriptors in a local neighborhood around each tracked point. We compute Histograms of Oriented Gradient [16] (HOG) and Histograms of Oriented Optical Flow [17] (HOF); in practice, we use the integral image representation in order to efficiently extract gradient and disparity components.

Considering a frame taken from a given video at instant t, we define $\mathcal{P}_t = \{p_{t,i}\}_{i \in \{1,..,N\}}$ as the set of N tracked 2D points around the mouth region. We describe each point $p_{t,i} \in \mathcal{P}_t$ with a visual feature vector, denoted $f_{t,i}$, which corresponds to the concatenation of normalized HOF and HOG histograms extracted around $p_{t,i}$.

2.2. Structured Feature Quantization

Let $\mathcal{I} = \{1, .., N\}$ be the union of indices corresponding to the facial landmarks shown in Fig. 2. Let s be any subset of $\mathcal{I}$, referred to as *tuple* and $\mathcal{S}^{(n)} = \{s \subseteq \mathcal{I}, |s| = n\}$. Given a frame, at instant t, and a tuple $s \subseteq \mathcal{S}^{(n)}$, we define a feature vector $\mathcal{D}_{t,s} = [f_{t,i}]_{i \in s}$ as the concatenation of n distinct feature vectors $\{f_{t,i}\}_i$ with $i \in s$. Note that, by construction, each feature vector $\mathcal{D}_{t,s}$ captures local visual informations as well as spatial relationships between its underlying facial landmarks. In the remainder of this paper, $\mathcal{D}_{t,s}$ will be referred to as *structured feature* vector.

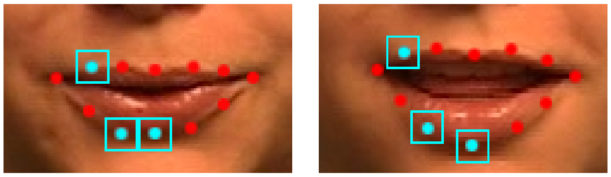


Fig. 2. This figure shows 12 tracked facial landmarks and a tuple $s \subseteq \mathcal{S}^{(3)}$.

Fix a tuple s and consider $\mathcal{D}_s$ as the union of all structured feature vectors[1] with indices in s. In order to map a given feature vector $\mathcal{D}_{t,s}$ to a unique codeword, we first partition $\mathcal{D}_s$ into k clusters using k-means, and then we assign $\mathcal{D}_{t,s}$ to the "cluster number" whose center is the closest to $\mathcal{D}_{t,s}$. The set of cluster numbers (or codewords), denoted $\mathcal{A}_s$ is referred to as codebook. In practice, we test codebooks of different sizes including 10, 50, 100, 256 and 1000. Now, given a video with ℓ frames and ℓ structured feature vectors $\{\mathcal{D}_{t,s}\}_{t \in \{1,...,\ell\}}$ taken from location indices in s; we map these vectors to an ordered sequence of codewords, denoted $\boldsymbol{X}_s$, with $\boldsymbol{X}_s \in \mathcal{A}_s^{\ell}$. We repeat this vector quantization and mapping process for different tuples in $\mathcal{S}^{(n)}$ and different values of n (see experiments).

3. TUPLE SELECTION AND COMBINATION

Although structured features, associated to different tuples, are expected to be complementary, they may share common – redundant – informations and only a subset of these features is discriminant. Therefore, it is necessary to define a procedure that selects and combines these structured features while optimizing the performance of visual speech recognition. In the subsequent section, we propose different tuple (and hence structured feature) selection and combination strategies, that best capture "visual speech"-induced variability (see Fig. 1-FG). Inspired by successful results in the neighboring field of text classification, we also introduce string kernel-based machines that effectively handle time varying "visual speech" sequences (see also Fig. 1-E and Section 3.3).

[1] These feature vectors are extracted from different frames of our training videos at location indices in s.

3.1. Tuple Combination using Multiple Kernel Learning

In this section, we use multiple kernel learning [18] (MKL) in order to combine tuples and the underlying structured features. MKL considers linear combination of multiple kernels, associated to different tuples, and finds "optimal" weights of this combination while training multi-class SVMs.

Let $\mathcal{X}_s = \cup_{\ell=1}^{L} \mathcal{A}_s^{\ell}$ be the union of all possible sequences of lengths up to (a fixed) L; again $\mathcal{A}_s$ is the codebook associated to a given tuple s. We consider $\mathcal{T} = \{\boldsymbol{X}^{(j)}\}_j$ as a training set of multi-sequences[2] with $\boldsymbol{X}^{(j)} = \{\boldsymbol{X}_s^{(j)} \in \mathcal{X}_s\}_s$ and $y_j \in \{1, \dots, M\}$ as the label (or class) of $\boldsymbol{X}^{(j)}$ taken from a well defined ground truth; in practice, $M = 10$.

Multi-class SVMs use a mapping Φ_s, that takes data from the input space $\mathcal{X}_s$ to a high (possibly infinite) dimensional space $\mathcal{H}_s$ and find the (unknown) label of a given test sequence $\boldsymbol{X}_s \in \mathcal{X}_s$ as

$$\arg\max_{y \in \mathcal{Y}} f_{y,s}(\boldsymbol{X}_s), \tag{1}$$

here $\mathcal{Y} = \{1, \dots, M\}$ and $f_{y,s}(\boldsymbol{X}_s) = \langle \boldsymbol{w}_{y,s}, \Phi_s(\boldsymbol{X}_s)\rangle + \boldsymbol{b}_{y,s}$, with $\boldsymbol{w}_{y,s}$, $\boldsymbol{b}_{y,s}$ being respectively hyperplane normal and bias associated to a given class $y \in \mathcal{Y}$ and tuple s.

In order to combine different tuples, we use multiple kernel learning that generalizes the above SVM framework. Its main idea consists in finding a kernel, denoted $\mathcal{K}$, as a linear combination of different elementary kernels $\{k_s\}_s$ associated to different tuples $\{s \subseteq \cup_n \mathcal{S}^{(n)}\}$. Thus, the kernel value between two multi-sequences $\boldsymbol{X} = \{\boldsymbol{X}_s\}_s$, $\boldsymbol{X}' = \{\boldsymbol{X}'_s\}_s$ is defined as

$$\mathcal{K}(\boldsymbol{X}, \boldsymbol{X}') = \sum_s \beta_s \langle \Phi_s(\boldsymbol{X}_s), \Phi_s(\boldsymbol{X}'_s)\rangle = \sum_s \beta_s k_s(\boldsymbol{X}_s, \boldsymbol{X}'_s), \tag{2}$$

here $\beta_s \geq 0$ and each kernel k_s operates using only the structured features of its underlying tuple s. Using a primal formulation, we predict the (unknown) label of a given multi-sequence $\boldsymbol{X}$ as

$$\arg\max_{y \in \mathcal{Y}} \quad f_y(\boldsymbol{X}), \tag{3}$$

with $f_y(\boldsymbol{X}) = \sum_s \beta_s \langle \boldsymbol{w}_{y,s}, \Phi_s(\boldsymbol{X}_s)\rangle + \boldsymbol{b}_y$ and $\boldsymbol{b}_y$, $\{\boldsymbol{w}_{y,s}\}_s$ being respectively the bias and the hyperplane normals associated to a given class y, for different tuples. We choose the parameters $\boldsymbol{\beta} = \{\beta_s\}_s$, $\boldsymbol{b} = \{\boldsymbol{b}_y\}_y$ and $\boldsymbol{w} = \{\boldsymbol{w}_{y,s}\}_{y,s}$, by solving the following constrained minimization problem

$$\begin{aligned} \min_{\boldsymbol{\beta}, \boldsymbol{w}, \boldsymbol{b}, \xi} \ & \frac{1}{2} \sum_s \sum_{y \in \mathcal{Y}} \beta_s \langle \boldsymbol{w}_{y,s}, \boldsymbol{w}_{y,s}\rangle + \sum_{j=1}^{|\mathcal{T}|} \xi_j \\ \text{s.t } \ & \xi_j = \max_{u \in \mathcal{Y} \backslash y_j} l\big(f_{y_j}(\boldsymbol{X}^{(j)}) - f_u(\boldsymbol{X}^{(j)})\big), \end{aligned} \tag{4}$$

here $y_j \in \mathcal{Y}$ is the actual label of $\boldsymbol{X}^{(j)}$, $\xi = \{\xi_j\}_j$ and $l(.)$ is a convex loss function.

3.2. Greedy Tuple Selection and Aggregation

If one considers high order tuples $\mathcal{S}^{(n)}$ (with $n > 2$) then $|\mathcal{S}^{(n)}|$ increases rapidly for some values of n and the tuple selection process becomes computationally prohibitive. In order to make this selection process more tractable, we propose in this section strategies that efficiently select tuples of increasing orders.

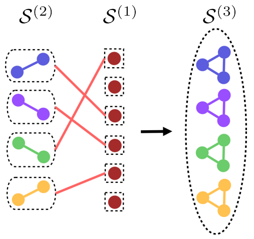


Figure 3. Diagram of the bipartite assignment procedure used in order to select tuples in $\mathcal{S}^{(3)}$ by augmenting those previously selected in $\mathcal{S}^{(2)}$.

3.2.1. Geometry-based Tuple Aggregation

Lip motion variability is due to various articulations and also to the different "visual speech" classes. In order to characterize this variability, we build a set of tuples using velocity statistics of the tracked facial landmarks around the lips. We start this selection process with the smallest order, i.e. with singletons in $\mathcal{S}^{(1)}$, and we follow a greedy approach that incrementally selects tuples in $\mathcal{S}^{(n+1)}$ by augmenting those selected in $\mathcal{S}^{(n)}$ using a variance maximization criterion (VMC) described below.

Our VMC strategy computes the Hausdorff distance, through different video frames, between (i) the 2D points related to selected tuples in $\mathcal{S}^{(n)}$ and (ii) the 2D points of singletons in $\mathcal{S}^{(1)}$. We select tuples in $\mathcal{S}^{(n+1)}$ by finding the best assignment between tuples in $\mathcal{S}^{(n)}$ and singletons in $\mathcal{S}^{(1)}$. Let's consider $\mathcal{G}^{(n)} = \langle \mathcal{V}^{(n)} \cup \mathcal{V}^{(1)}, \mathcal{V}^{(n)} \times \mathcal{V}^{(1)} \rangle$ as a bipartite graph where nodes in $\mathcal{V}^{(n)}$ (resp. $\mathcal{V}^{(1)}$) are associated to tuples in $\mathcal{S}^{(n)}$ (resp. in $\mathcal{S}^{(1)}$) and edges in $\mathcal{V}^{(n)} \times \mathcal{V}^{(1)}$ are weighted inversely proportional to the Hausdorff distance between the underlying nodes (see Fig. 3) . We find an assignment between nodes in $\mathcal{G}^{(n)}$ by solving a bipartite graph-matching problem; the Kuhn-Munkres algorithm provides an assignment with a minimum cost equal to the sum of the weights of the selected edges in $\mathcal{G}^{(n)}$.

We repeat this selection process for increasing values of n (taken from $\{1, 2, 3, 4\}$ in practice). At the end of this selection process, we only keep the tuples with high variances in order to achieve visual speech recognition.

3.2.2. MKL-based Tuple Aggregation

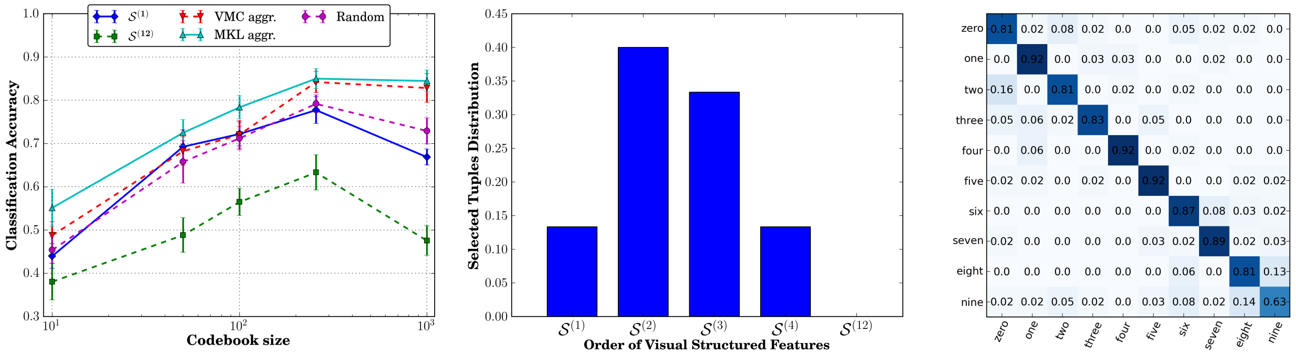
Similarly, this second procedure proceeds using aggregation. Initially, we start the process by learning a linear combination of elementary kernels each one assigned to a singleton tuple, then we keep only the singleton tuples with the highest MKL weights $\{\beta_s\}_s$. We repeat this process, for increasing values of n, by taking kernels (and hence tuples) selected at iteration $n-1$, and linearly combining them with the elementary kernels associated to tuples in $\mathcal{S}^{(n)}$. Again, we only keep tuples with the highest MKL weights. At the final stage of this process, the obtained linear combination of kernels corresponds to a set of discriminant tuples of different orders.

3.3. Elementary Kernels

We aim at classifying finite sequences, corresponding to visual speech units (codewords). As these sequences may vary in time, we map them to fixed length (high dimensional) representations using string kernels. Each kernel map captures local transitions between visual speech units along a sequence.

Given a sequence $\boldsymbol{X}_s \in \mathcal{X}_s$ (associated to a tuple $s \subseteq \cup_n \mathcal{S}^{(n)}$) with codewords in $\mathcal{A}_s$. The (g, m)-mismatch kernel [19] induces the following $|\mathcal{A}_s|^g$- dimensional representation for that sequence: if α is a g-mer (i.e., $\alpha \in \mathcal{A}_s^g$), then $\Phi_s^{(g,m)}(\alpha) = (\phi_\gamma(\alpha))_{\gamma \in \mathcal{A}_s^g}$, where $\phi_\gamma(\alpha) = 1_{\{\gamma \in \mathcal{N}_{(g,m)}(\alpha)\}}$ and $\mathcal{N}_{(g,m)}(\alpha)$ denotes the set of all g-length sequences (taken from $\mathcal{A}_s$) that differ from α by at most m mismatches. For a sequence $\boldsymbol{X}_s \in \mathcal{X}_s$ of any length, we extend

[2] A multi-sequence is a set of "sequences of codewords" extracted, from the same video frames, as described in Section 2.2. Each "sequence of codewords" is associated to a tuple $s \subseteq \mathcal{I}$.



(a) (b) (c)

Fig. 4. These figures show experiments on the CUAVE set. The classification accuracies w.r.t different tuple selection strategies and codebook sizes are shown in **(a)**. The distribution of tuple orders (of selected structured features) is shown in **(b)**. The confusion matrix for 10-digit classification is shown in **(c)** .

by summing the maps for all the g-mers in $\boldsymbol{X}_s$, leading to

$$\Phi_s(\boldsymbol{X}_s) = \sum_{g\text{-mers } \alpha \text{ in } \boldsymbol{X}_s} \Phi_s^{(g,m)}(\alpha). \tag{5}$$

Hence, the elementary mismatch kernel, associated to a tuple s, is defined as $k_s(\boldsymbol{X}_s, \boldsymbol{X}'_s) = \langle \Phi_s(\boldsymbol{X}_s), \Phi_s(\boldsymbol{X}'_s) \rangle, \ \ \boldsymbol{X}_s, \boldsymbol{X}'_s \in \mathcal{X}_s$. As described in [20], we take weighted sums of (g, m)-mismatch kernels for different values of g.

4. EXPERIMENTS

4.1. Evaluation Set and Settings

In order to evaluate the performance of our visual speech recognition method, we use the CUAVE database [21] including videos recorded at a frame rate of 29.97 *fps* and a resolution of 740×480 pixels. We consider a subset of videos including (nearly frontal) talking faces belonging to 36 different speakers. These speakers pronounce digits between *zero* and *nine* in American English.

Each video in the CUAVE set is processed in order to extract HOG and HOF descriptors. The latters correspond to orientation histograms (of 6-bins and 7-bins respectively) extracted, in local neighborhoods of 29×29 pixels, around twelve lip landmarks. These local descriptors are used to learn codebooks of different sizes, in $\{10, 50, 100, 256, 1000\}$, and also to build the elementary (g, m)-mismatch kernels (with $g \in \{1, 2, 3\}$ and $m = 1$). Elementary kernels, associated to tuples of different orders in $\{1, 2, 3, 4\}$, are linearly combined using MKL (as discussed in Section 3) and used to learn the 10-digit SVM classifiers. Performances of these SVM classifiers are evaluated, in a speaker independent setting, using 9-fold cross-validation with each fold including 4 speakers.

4.2. Results and Comparison

Fig. 4a shows the evolution of the overall digit classification performances with respect to different tuple selection strategies discussed earlier. In these results, the baseline corresponds to random tuple selection (referred to as RDM). According to these results, the best performances are obtained using MKL-based tuple aggregation strategy with a codebook of size 256. Fig. 4c shows the underlying confusion matrix.

Table 1 shows a comparison of MKL-based strategy against RDM (baseline) and VMC as well as related state-of-the-art visual speech recognition techniques. From these results, it is clear that VMC and MKL provide the two best performances; VMC is more efficient while MKL is slightly more effective. These results are also consistent with those provided in the literature. Indeed, Table 1 and Fig. 4a show that digit classification results obtained using a single kernel associated to a simple concatenation of all the local descriptors – i.e., by concatenating all the descriptors associated to tuples in $\mathcal{S}^{(N)}$ (with $N = 12$ in practice) – are consistent with those obtained in [4]. Similarly, the results obtained by learning a linear combination of elementary kernels associated to singletons in $\mathcal{S}^{(1)}$ are consistent with those in [7]. Finally, Fig. 4b shows the distribution of orders (of selected tuples) at the end of the MKL-based selection process discussed in Section 3.2.2. This distribution, which corresponds to 20 selected tuples with the highest MKL weights, is highly concentrated around orders 2 and 3. This clearly shows that the most discriminant structured features correspond to tuples in $\mathcal{S}^{(2)}$ and $\mathcal{S}^{(3)}$.

Method	Accuracy
Discrete Cosine Transform [4]	64%
Fused Holistic+Patch [7]	77.08%
ROI-AAM [5]	83%
Proposed - RDM selection	79.3% ± 3.1%
Proposed - VMC aggregation	84.2% ± 2.46%
Proposed - MKL aggregation	85% ± 2.29%

Table 1. This table shows the average accuracy of visual speech recognition on the CUAVE database using 9-fold cross validation.

5. CONCLUSION

We introduced in this paper a novel visual speech recognition algorithm, based on string kernels and support vector machines. The proposed method selects and combines discriminant (appearance and motion) structured features of different orders using multiple kernel learning.

Experiments show that the proposed feature design procedure is effective and also efficient, and achieves state of the art results in visual speech recognition. As a future work, we are currently investigating the application of our method to continuous speech using viseme models in order to improve the quality of lip-reading. We are also investigating the combination of visual and audio features in order to handle visual speech recognition in more challenging conditions including noisy car environments.

6. REFERENCES

[1] A. Q. Summerfield, B. Dodd, and R. Campbell, "Some preliminaries to a comprehensive account of audio-visual speech perception," Number Hillsdale, NJ, pp. 3–51. Erlbaum, 1987.

[2] G. Potamianos, C. Neti, J. Luettin, and I. Matthews, "Audio-visual automatic speech recognition: An overview," *Issues in Visual and Audio-Visual Speech Processing*, pp. 356–396, 2004.

[3] H. McGurk and J. MacDonald, "Hearing lips and seeing voices," *Nature*, pp. 746–748, 1976.

[4] M. Gurban and J.P. Thiran, "Information theoretic feature extraction for audio-visual speech recognition," *Signal Processing, IEEE Transactions on*, vol. 57, no. 12, pp. 4765–4776, 2009.

[5] G. Papandreou, A. Katsamanis, V. Pitsikalis, and P. Maragos, "Adaptive multimodal fusion by uncertainty compensation with application to audiovisual speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 3, pp. 423–435, 2009.

[6] Y. Lan, R. Harvey, B. Theobald, E.J. Ong, and R. Bowden, "Comparing visual features for lipreading," in *International Conference on Auditory-Visual Speech Processing 2009*, 2009, pp. 102–106.

[7] P. Lucey and S. Sridharan, "Patch-based representation of visual speech," in *Proceedings of the HCSNet workshop on Use of vision in human-computer interaction-Volume 56*. Australian Computer Society, Inc., 2006, pp. 79–85.

[8] I. Matthews, T.F. Cootes, J.A. Bangham, S. Cox, and R. Harvey, "Extraction of visual features for lipreading," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 2, pp. 198–213, 2002.

[9] G. Zhao, M. Barnard, and M. Pietikainen, "Lipreading with local spatiotemporal descriptors," *Multimedia, IEEE Transactions on*, vol. 11, no. 7, pp. 1254–1265, 2009.

[10] R. Solera-Urena, D. Martín-Iglesias, A. Gallardo-Antolín, C. Peláez-Moreno, and F. Díaz-de María, "Robust asr using support vector machines," *Speech Communication*, vol. 49, no. 4, pp. 253–267, 2007.

[11] J. Yousafzai, P. Sollich, Z. Cvetkovic, and B. Yu, "Combined features and kernel design for noise robust phoneme classification using support vector machines," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 5, pp. 1396–1407, 2011.

[12] R. Solera-Urena, A.I. García-Moral, C. Peláez-Moreno, M. Martínez-Ramón, and F. Diaz-de Maria, "Real-time robust automatic speech recognition using compact support vector machines," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 4, pp. 1347–1361, 2012.

[13] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 6, pp. 681–685, 2001.

[14] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.

[15] P. Viola and M.J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 1, pp. 886–893.

[17] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," *Computer Vision–ECCV 2006*, pp. 428–441, 2006.

[18] A. Zien and C.S. Ong, "Multiclass multiple kernel learning," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 1191–1198.

[19] C.S. Leslie, E. Eskin, A. Cohen, J. Weston, and W.S. Noble, "Mismatch string kernels for discriminative protein classification," *Bioinformatics*, vol. 20, no. 4, pp. 467–476, 2004.

[20] SVN Vishwanathan and A.J. Smola, "Fast kernels for string and tree matching," *Kernel methods in computational biology*, pp. 113–130, 2004.

[21] E. K. Patterson, S. Gurbuz, Z. Tufekci, and J. N. Gowdy, "Cuave: A new audio-visual database for multimodal human-computer interface research," in *ICASSP*. 2002, pp. 2017–2020, IEEE.