Learning Bimodal Structure in Audio-Visual Data

Gianluca Monaci, Pierre Vandergheynst and Friedrich T. Sommer

***Abstract*—A novel model is presented to learn bimodally informative structures from audio-visual signals. The signal is represented as a sparse sum of audio-visual kernels. Each kernel is a bimodal function consisting of synchronous snippets of an audio waveform and a spatio-temporal visual basis function. To represent an audio-visual signal, the kernels can be positioned independently and arbitrarily in space and time. The proposed algorithm uses unsupervised learning to form dictionaries of bimodal kernels from audio-visual material. The basis functions that emerge during learning capture salient audio-visual data structures. In addition it is demonstrated that the learned dictionary can be used to locate sources of sound in the movie frame. Specifically, in sequences containing two speakers the algorithm can robustly localize a speaker even in the presence of severe acoustic and visual distracters.**

I. Background and Significance

To smoothly interact with our environment we must be able to analyze and understand complex relationships between the inputs to different sensory modalities. Not surprisingly, this behavioral requirement of multimodal processing is reflected by corresponding observations in brain research. A fast growing body of experimental evidence suggests that different sensory modalities in the brain do not operate in isolation but exhibit interactions at various levels of sensory processing [1–8]. Also the fields of signal processing and computer vision have recently seen the development of perception-inspired audio-visual fusion algorithms. Examples include methods for speech-speaker recognition [9] and speaker detection aided by video [10, 11], audio filtering and separation based on video [12–16], or audio-visual sound source localization [17–26].

Typically, algorithms for audio-visual fusion exploit *synchronous co-occurrences of transient structures in the different modalities*. In their pioneering work, Hershey and Movellan [17] localized sound sources in the image frame by computing the correlation between acoustic energy and intensity change in single pixels. Recently, more sophisticated feature representations have been proposed, for example, audio features derived from audio energy [20, 21, 23] or cepstral representations [11, 18, 19, 22] and video features based on pixel intensities [19, 20, 23] or on temporal signal changes [11, 18, 19, 21, 22]. Another line of research relevant for this work is sparse coding of audio or video signals with overcomplete bases which has been shown to yield excellent results in signal compressing and de-noising [27–32]. Recently, these methods have been proposed for analyzing audio-visual signals [16, 24, 25].

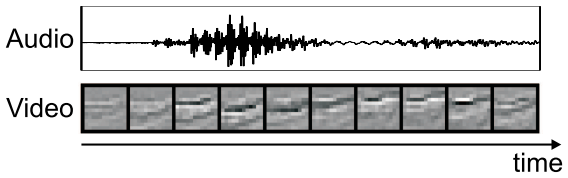


Fig. 1. An audio-visual function composed of an audio [Top] and a video part [Bottom] which are time locked. Video frames are represented as a succession of images.

The methods of audio-visual signal analysis mentioned so far can be characterized by the two following steps. First, fixed and predefined unimodal features are used to encode the essential structures in the audio and video stream separately. Second, correlations between the resulting feature representations of audio and video signal are analyzed, for example by estimating joint distributions of audio-visual features [11, 19, 20, 22, 23], using Canonical Correlation Analysis (CCA) [18, 21] or detecting temporal coincidences of audio-visual structures [16, 24, 25].

Alternatively, we have recently suggested a different approach to sensor fusion [26]. The idea is to analyze the audio-visual data jointly by extracting typical templates of audio-visual features, see Fig. 1 for an example. These templates represent synchronous transient structures that co-occur in both modalities. Simple template matching can then be used for solving sensor fusion tasks, such as speaker localization. The audio-visual template in Fig. 1 was extracted from a movie showing a speaker: the audio part is the waveform of a spoken digit in English, while the corresponding video part shows a moving edge that could represent the lower lip during the utterance of the digit. The direct extraction of audio-visual templates is interesting because it focuses on relevant bimodal structure rather than first computing the full representations in both modalities separately and then analyzing the joint statistics of features. However, the efficiency of the algorithm in [26] was limited because the template extraction and matching is brittle in the presence of accidental superpositions of separate transient structures.

Here we present a novel model of audio-visual fusion that combines the advantages of joint bimodal signal analysis [26] and sparse coding, e.g. [27–32]. To combine the two approaches we build on previous work that used unsupervised learning of efficient sparse codes to understand response properties of neurons in various sensory systems. Efficient coding (*redundancy reduction*) has served as an

Gianluca Monaci contributed to this work while at the Redwood Center for Theoretical Neuroscience, University of California, Berkeley, USA. Now, he is with the Video Processing and Analysis group, Philips Research, Eindhoven, the Netherlands. Email: gianluca.monaci@philips.com.

Pierre Vandergheynst is with the Institute of Electrical Engineering, Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland. Email: pierre.vandergheynst@epfl.ch.

Friedrich T. Sommer is with the Redwood Center for Theoretical Neuroscience, University of California, Berkeley, CA 94720-3190 USA. Email: fsommer@berkeley.edu.

important computational objective for unsupervised learning on sensory input [33]. This principle led to the design of learning algorithms capable of matching the responses of the visual system, e.g. [34, 35], and of the auditory system, e.g. [36]. Learning methods used in these approaches typically get their input from local data patches, and as a consequence the emerging features are usually redundant with respect to translation, rotation or scale. Recently, a family of sparse generative models have arisen, motivated by the observation that natural stimuli typically exhibit characteristics that are *shift-invariant*, that is, they can occur and re-occur at any spatio-temporal location. The original sparse coding models have been thus extended in many different ways to build shift-invariant sparse codes for sound [37–41], images [41–43] and video [44].

In the model we propose, the bimodal signal structure is captured by a shift-invariant sparse generative model. The *bimodal signal structure* is the audio-visual signal component that is informative for sensor fusion. Conversely, signal structure that is uncorrelated in both modalities is less informative and therefore only incompletely encoded. The new model uses unsupervised learning for forming an overcomplete dictionary adapted to efficiently and sparsely encode the informative signal component. It will be demonstrated that the new method avoids the problems of template matching used in [26] and thus has significantly improved performance for speaker localization in movies.

The paper is organized as follows: Section II describes the proposed audio-visual signal model. Section III presents the Audio-Visual Matching Pursuit algorithm for coding bimodal signals. Section IV introduces the algorithms for learning bimodal data structure. In Section V experimental results based on synthetic and natural audio-visual data are shown. Section VI concludes the paper with a summary of the achieved results and with the outline of future developments of this approach.

II. Convolutional Generative Model for Audio-Visual Signals

Audio-visual data is a quite unequal couple $s = (a, v)$ of signals. First, the dimensions differ: while the audio signal is a 1-D stream $a(t)$, the video sequence is a 3-D signal $v(x, y, t)$ with (x, y) the pixel position. Second, because the temporal resolution of auditory and visual perception differs by orders of magnitude, the audio signal is usually sampled at much higher rate (typically 6–60 kHz) than the video signal (typically 15–60 Hz).

Extending the sparse coding approach for movies [44], one can formulate a generative model for audio-visual signal as a linear sum of audio-visual kernels or *atoms* $\phi_k = (\phi_k^{(a)}(t), \phi_k^{(v)}(x, y, t))$ taken from a dictionary $\mathcal{D} = \{\phi_k\}$. Each atom consists of an audio and a video component with unitary ℓ_2 norm each. In the representation of the audio-visual signal an atom can be placed in any point in space and time. To place an audio-visual function ϕ at a spatio-temporal position (p, q, r) we introduce the shift operator $T_{(p,q,r)}$:

$$T_{(p,q,r)}\phi = \Big(\phi^{(a)}(t - r), \phi^{(v)}(x - p, y - q, t - r)\Big) \ . \quad (1)$$

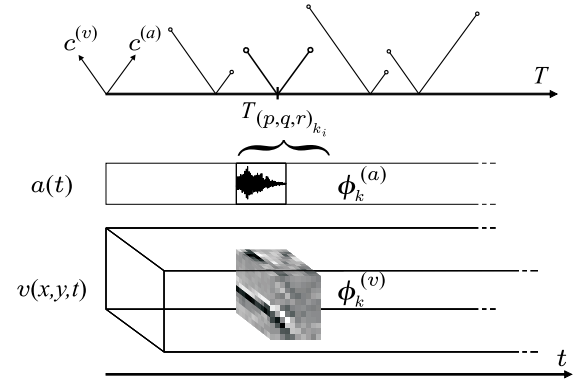


Fig. 2. Schematic representation of the audio-visual code. The signal $s = (a(t), v(x, y, t))$ [Bottom] is modeled as a sum of kernels $\phi_k = (\phi_k^{(a)}, \phi_k^{(v)})$, $\phi_k^{(a)}$ being a 1-D audio function and $\phi_k^{(v)}$ a 3-D video function. Each kernel is localized in space and time and may be applied at any spatio-temporal position T within the signal [Top].

Note that the shift operator shifts audio and visual component of ϕ by the same amount of time r and thus relative timing is preserved. Using the shift operator, an audio-visual signal can be expressed:

$$s \approx \sum_{k=1}^{K} \sum_{i=1}^{n_k} c_{k_i} T_{(p,q,r)_{k_i}} \phi_k \ , \quad (2)$$

where $T_{(p,q,r)_{k_i}}$ is used as compact notation for $T_{(p_{k_i}, q_{k_i}, r_{k_i})}$. The index n_k is the number of instances the kernel ϕ_k is used and the pair $c_{k_i} = (c_{k_i}^{(a)}, c_{k_i}^{(v)})$ specifies the weights for the audio and the visual components of ϕ_k at instance i. The use of two coefficients per instance allows us to use the same kernel function irrespective of the relative power of audio and visual signal. This invariance in the coding is important because audio-video patterns may be stereotyped although the relative intensities in the two modalities can vary.

Typically [34, 35, 37, 44, 45], the free parameters in Eq. (2) are adjusted by two interleaved optimization procedures: sparse coding and learning. **Sparse coding:** To represent a particular signal s with Eq. (2) the translation $T_{(p,q,r)_{k_i}}$ and the coefficients $c_{k_i}^{(a)}$ and $c_{k_i}^{(v)}$ have to be chosen in order to optimize the approximation of the signal. In addition, to provide a sparse code, the coefficients have also to satisfy a sparseness constraint, for example, have few non-zero entries or have a kurtotic, heavy-tailed distribution centered at zero [27–32, 34]. **Learning:** The efficiency of the described coding procedure with Eq. (2) can be optimized by adapting the dictionary of audio-visual kernels $\phi_k = (\phi_k^{(a)}(t), \phi_k^{(v)}(x, y, t))$ to the data.

The model is schematically illustrated in Fig. 2.

III. Sparse Coding

A. Simultaneous Matching Pursuit algorithm

In the coding procedure described by Eq. (2) the coefficients and spatio-temporal translations of dictionary elements have to be determined to approximate a given audio-visual input. It has been shown in general, that finding the optimal sparse representation of arbitrary signals is a NP-hard problem [46]. There are many approximate methods to encode a signal

given a certain dictionary [27, 29, 39, 44, 47]. Because of their computational complexity however, most of these techniques are too slow for high dimensional signals like audio-visual data.

Matching Pursuit (MP) algorithm [27] is a simple, relatively fast iterative method to build signal approximations in Eq. (2) by selecting at each step one atom from the dictionary and by using the selected atom to improve the signal approximation. More formally, the two steps involved in each iteration of convolutional MP can be described as follows:

1) ***Projection step***: For a selected atom ϕ_n taken from dictionary $\mathcal{D}$, coefficients c_n and position $T_{(p,q,r)_n}$ are determined and used to compute a signal approximation $s_n \in \text{span}(T_{(p,q,r)_n}\phi_n \ : \ n \in \{0,\dots,N-1\})$ and a residual $R^n s = s - s_n$.
2) ***Selection step***: Based on a similarity criterion $C(R^n s, \phi)$ between the current residual and dictionary elements, the best matching atom is selected for the next projection step.

Here we will use an extension to audio-visual signals of Matching Pursuit. MP has been successfully used to compute sparse codes for unimodal audio signals [37] and images [35]. Tropp et al. have recently proposed Simultaneous Orthogonal MP (S-OMP), an MP algorithm for jointly encoding multichannel signals [48]. However S-OMP was designed for signals of the same type, while for capturing the bimodally informative structure in audio-visual data the method has to be extended. To overcome S-OMP limitations we introduce here the Audio-Visual Matching Pursuit method (AV-MP).

B. Audio-Visual Matching Pursuit

Our motivation in this study is the question whether perceptual effects of sensor fusion could be modeled by joint encoding of audio-visual signals. The general idea is that if coding of both channels is not independent, one modality could influence and thereby alter and improve the encoding of the other modality. Such a crossmodal influence might explain effects of sensory fusion, such as crossmodal denoising, crossmodal alterations of perception (e.g. McGurk effect [6], bouncing illusion [5]), source localization, etc. In audio-visual signals some signal structures are more important for sensor fusion than other structures. Specifically, transient substructures that co-occur synchronously in both modalities are particularly indicative of common underlying physical causes, they are what Barlow coined "suspicious coincidences" [49]. As an example, think of a spoken syllable in the audio signal occurring in synchrony with a person's lip movement in the video. Conversely, transient signals that are uncorrelated across modalities are less informative for multimodal signal analysis. Thus, although coding and learning could be designed so that Eq. (2) captures the entire structure in the signal, the goal here is to design a generative model for simultaneously capturing the bimodal signal structure that is informative in sensor fusion.

Because audio and video signals have different dimensionality and different temporal sampling rate, plain S-OMP cannot encode them. Another extension that is required in order to capture the bimodally informative signal structure, is to introduce the concept of synchrony between audio-visual events in the coding. The next paragraph describes the core algorithm of Audio-Visual Matching Pursuit (AV-MP). Subsequently, in Sec. III-B2 we describe possible similarity measures to combine the audio and video projections for selecting audio-visual atoms in AV-MP.

*1) **The core algorithm**:* In MP the coding is based on the best match between the signal and the translated kernel function. Since in digital signals the different modalities are sampled at different rates over time, we define a discretized version of the translation operator T in Eq. (1) that temporally shifts the two modalities by different integer number of samples. The ***discrete*** *audio-visual translation* $\mathcal{T}$ is defined as

$$\mathcal{T}_{(p,q,r)}^{(\nu^{(a)},\nu^{(v)})} = \left(\mathcal{T}_{\alpha}, \mathcal{T}_{(p,q,\beta)}\right) = \mathcal{T}_{(p,q,\alpha,\beta)} \qquad (3)$$

with

$$\begin{aligned} \alpha &= \text{nint}(r/\nu^{(a)}) \in \mathbb{Z} \\ \beta &= \text{nint}(r/\nu^{(v)}) \in \mathbb{Z}. \end{aligned}$$

Here $\nu^{(a)}$ and $\nu^{(v)}$ denote the audio and video temporal sampling rates, respectively. $\mathcal{T}_{\alpha}$ and $\mathcal{T}_{(p,q,\beta)}$ are the translation operators for shifting the audio and visual signals by α and (p,q,β) samples respectively. The nearest integer function is denoted by $\text{nint}(\cdot)$. In practice the audio is sampled at higher rates than the video, i.e. $\nu^{(v)} > \nu^{(a)}$, and therefore every video frame corresponds to about $F = \text{nint}(\nu^{(v)}/\nu^{(a)})$ audio samples[1]. Thus, the shift operator in Eq. (3) is somewhat "sloppy" in preserving audio-visual synchrony since it shifts the audio kernel at much finer steps than the visual kernel. In fact the following relationship holds between audio translation α and video temporal translation β:

$$\alpha = F \cdot (\beta - 1) + \alpha_{\text{offset}}\,, \quad \text{with } 1 \leq \alpha_{\text{offset}} \leq F.$$

However, this sloppiness coincides well with human perception and thereby introduces a desired *quasi* invariance in the representation, as will be explained in the next section.

Audio-Visual Matching Pursuit (AV-MP) approximates a multimodal signal $s = (a, v)$ with successive projections onto the audio-visual dictionary $\mathcal{D}$. Let us initialize $R^0 s = s$; then the first step of AV-MP decomposes s as

$$R^0 s = \left(\hat{c}_0^{(a)}\, \mathcal{T}_{\alpha_0}\phi_0^{(a)}, \hat{c}_0^{(v)}\, \mathcal{T}_{(p,q,\beta)_0}\phi_0^{(v)}\right) + R^1 s \qquad (4)$$

with

$$\begin{aligned} \hat{c}_0^{(a)} &= \langle a, \mathcal{T}_{\alpha_0}\phi_0^{(a)}(t)\rangle \\ \hat{c}_0^{(v)} &= \langle v, \mathcal{T}_{(p,q,\beta)_0}\phi_0^{(v)}(x,y,t)\rangle. \end{aligned}$$

In Eq. (4) $R^1 s$ is the residual after projecting s in the subspace spanned by $\mathcal{T}_{(p,q,\alpha,\beta)_0}\phi_0$. The pair of values $(\langle a, \mathcal{T}_{\alpha_0}\phi_0^{(a)}\rangle, \langle v, \mathcal{T}_{(p,q,\beta)_0}\phi_0^{(v)}\rangle)$ represents the pair of coefficients $\hat{c}_0 = (\hat{c}_0^{(a)}, \hat{c}_0^{(v)})$. The function ϕ_0 and its spatio-temporal translation $\mathcal{T}_{(p,q,\alpha,\beta)_0}$ are chosen maximizing the *similarity measure* $C(R^0 s, \phi)$.

[1] In our experiments, values of the sampling rates are $\nu^{(a)} = 1/8000$ for audio signals at 8 kHz and $\nu^{(v)} = 1/29.97$ for videos at 29.97 frames per second (fps), and consequently $F = 267$.

Recursively applying this procedure, after N iterations we can approximate s with $\hat{s}$ as

$$\hat{s} = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \hat{c}_{k_i} \mathcal{T}_{(p,q,\alpha,\beta)_{k_i}} \phi_k \,, \tag{5}$$

where we split the sum over $n = 0, \ldots, N-1$, into two sums over k and i, with $N = \sum_{k=1}^{K} n_k$. The algorithm can be stopped either after a fixed number N of iterations or when the maximum value of the similarity measure C between residual and dictionary elements falls below a certain threshold. Note that the number of iterations is equal to the number of nonzero coefficients in the signal representation. Thus, a given ℓ_0 sparseness can be enforced simply by limiting the number of iterations.

*2) **Similarity measure for audio-visual coding**:* The critical question for processing audio-visual signals is how to define the similarity measure C in the selection step of AV-MP. It is important that the selection step reflects some basic properties of human perception. From psychophysics it is known that relative shifts between audio and visual signals that are smaller than the duration of a video frame are essentially imperceptible and do not seem to affect audio-visual integration [4, 50, 51]. Thus, the selection of audio-visual kernels should also be unaffected by small relative time shifts. Fortunately, the "sloppiness" of the shift operator in Eq. (3) we described earlier allows this perceptive invariance to be introduced in the selection step as follows. As described in Sec. III-B1, for each video frame there are F corresponding audio samples. The first video frame is associated with audio samples from 1 to F, the second with audio samples from $F+1$ to $2F$ and so on. Thus $\alpha \in [F \cdot (\beta - 1) + 1, F \cdot \beta]$.

We define then the similarity measure C_ρ for AV-MP as

$$C_\rho(R^n s, \phi) = \|\langle R^n a, \mathcal{T}_\alpha \phi^{(a)} \rangle\|^\rho + \|\langle R^n v, \mathcal{T}_{(p,q,\beta)} \phi^{(v)} \rangle\|^\rho \tag{6}$$
$$\text{subject to} \quad \alpha \in [F \cdot (\beta - 1) + 1, F \cdot \beta] \,.$$

At each iteration AV-MP selects the audio-visual kernel ϕ_n and its spatio-temporal translation $\mathcal{T}_{(p,q,\alpha,\beta)}$ that maximize Eq. (6). Note that the two addends in Eq. (6) are defined at different time resolutions but the time shifts α and β are linked by the simple constraint in Eq. (6). This constraint expresses the fact that for each video translation β there are F possible audio translations α associated. Thus, for each value of β we have to check the F corresponding values of $\alpha \in [F \cdot (\beta - 1) + 1, F \cdot \beta]$, and select the couple of translations that maximizes Eq. (6). More formally, translation indexes α and β are selected as:

$$\{\alpha, \beta\} = \underset{\beta \in \mathbb{Z},\, \alpha \in [F \cdot (\beta-1)+1, F \cdot \beta]}{\arg\max} C_\rho(R^n s, \phi) \,,$$

where C_ρ is expressed by Eq. (6). Interestingly, a similar constraint was introduced in the learning algorithm [38] to avoid the selection of slightly shifted audio features having high correlation with themselves.

The sum in Eq. (6) represents the ℓ_ρ norm of the matches between the audio and visual atoms and the residual. In the literature, different values of ρ have been used in simultaneous sparse approximation algorithms. For example, the ℓ_1 norm was used in [48, 52], while ℓ_2 norm was used in [53]. [54] proposed several algorithms that used ℓ_2 and ℓ_∞ norms. To understand the consequences of these different choices of ρ we represent the audio and video matches in a polar plane, the audio match along the $0°$ direction and video match along the $90°$ direction. Each pair of audio and video matches is a point on this plane. To assess how different ρ values affect the weighing between unimodal and bimodal matches, Fig. 3 shows the geodesic lines for different C_ρ, with ρ in the range from 0.1 to infinity, on the unit circle in the plane of audio and video matches. Three ρ values stand out: C_2 (dashed line in Fig. 3) is constant which means that this measure weighs unimodal matches ($0°$ and $90°$) and bimodal matches ($45°$) evenly. C_1 (continuous line) favors the selection of kernels that contribute energy to both audio and video signal over kernels that contribute energy exclusively to one modality. C_∞ (dotted-dashed) favors the selection of kernels that contribute mainly to a single modality. Values of ρ larger than 2 seem useful to encourage unimodal coding even more strongly than C_∞. However, values $\rho < 1$ cannot be used to put stronger emphasis on bimodal coding than C_1, for $\rho < 1$ the C_ρ curves become flatter and more resemblant to C_2. To summarize, the setting of ρ can either promote independent unimodal encoding or bimodal encoding of audio-visual structure. Since we want to model events that are essentially multimodal (i.e. that are reflected by relevant signal structures in both audio and video streams), we will use and compare the similarity measures C_1 and C_2.

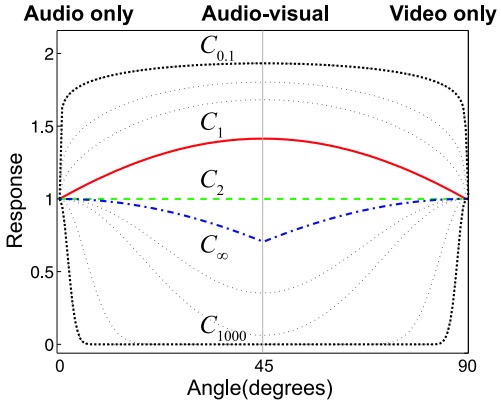


Fig. 3. C_ρ responses for values of p going from 0.1 to infinity. The plots are in polar coordinates on a plane whose axes represent audio and video projections as in Eq. (6). Audio and video projections vary defining a circular sector of unitary radius. C_1 (continuous red line) favors audio-visual kernels, C_∞ favors unimodal kernels (blue dotted-dashed), while C_2 attributes equal chances to unimodal and multimodal coding (green dashed).

IV. Learning

The AV-MP algorithm provides a way to encode signals given a set of audio-visual kernel functions. To optimize the kernel functions to a given set of audio-visual data we compare two algorithms that have been successful for unimodal data: gradient-based method [44] and the K-SVD algorithm [45]. The Gradient Ascent method has been used to demonstrate that biologically plausible codes can be learned from natural statistics, such as acoustic stimuli [37], static

natural images [34, 35] and time-varying visual stimuli [44]. The K-SVD algorithm, which is similar in principle, has been introduced more recently and has been reported to exhibit fast convergence [45].

A. Gradient Ascent Learning

Following [37, 44], one can rewrite Eq. (2) in probabilistic form as $p(s|\mathcal{D}) = \int p(s|\mathcal{D}, c)p(c)dc$, with $p(c)$ a sparse prior on the usage of dictionary elements. It is common to approximate the integral by the maximum of the integrant (its mode), i.e.

$$p(s|\mathcal{D}) = \int p(s|\mathcal{D}, c)p(c)dc \approx p(s|\mathcal{D}, c^\star)p(c^\star)\,. \quad (7)$$

Here the optimal code $c^\star$ is approximated by the AV-MP decomposition of the signal, $\hat{c}$. Note that in this case $p(\hat{c})$ is a prior on the ℓ_0 sparseness of the representation that is imposed by limiting the number of AV-MP iterations. Assuming the noise in the likelihood term, $p(s|\mathcal{D}, \hat{c})$, to be Gaussian with variance σ_N^2, the log probability can be expressed:

$$\log p(s|\mathcal{D}) \approx \frac{-1}{2\sigma_N^2}\left\| s - \sum_{k=1}^{K}\sum_{i=1}^{n_k} \hat{c}_{k_i}\mathcal{T}_{(p,q,\alpha,\beta)_{k_i}}\phi_k \right\|^2 . \quad (8)$$

The kernel functions can be updated through gradient ascent on Eq. (8):

$$\begin{aligned}\frac{\partial \log(p(s|\mathcal{D}))}{\partial \phi_k} &\approx \frac{-1}{2\sigma_N^2}\frac{\partial}{\partial \phi_k}\left\{ s - \sum_{k=1}^{K}\sum_{i=1}^{n_k} \hat{c}_{k_i}\left\{s-\hat{s}\right\}_{\mathcal{T}_{(p,q,\alpha,\beta)_{k_i}}} \right\}^2 \\ &= \frac{1}{\sigma_N^2}\sum_{i=1}^{n_k} \hat{c}_{k_i}\left\{s-\hat{s}\right\}_{\mathcal{T}_{(p,q,\alpha,\beta)_{k_i}}}\,, \quad (9)\end{aligned}$$

where $\{s-\hat{s}\}_{\mathcal{T}_{(p,q,\alpha,\beta)_{k_i}}}$ indicates the residual error over the extent of kernel ϕ_k at position $\mathcal{T}_{(p,q,\alpha,\beta)_{k_i}}$. Thus the functions ϕ_k are updated with a "delta" learning rule, that is, the product of neural activity and residual.

To summarize, the Gradient Ascent method (GA) suggests the following iterative update of the kernel functions:

$$\phi_k[j] = \phi_k[j-1] + \eta\Delta\phi_k\,,$$

where j indexes the learning algorithm iteration and η is a constant learning rate. $\Delta\phi_k$ is the update step:

$$\begin{aligned}\Delta\phi_k &= \left(\Delta\phi_k^{(a)}, \Delta\phi_k^{(v)}\right) \\ &= \left(\sum_{i=1}^{n_k}\hat{c}_{k_i}^{(a)}\{a-\hat{a}\}_{\mathcal{T}_{\alpha_{k_i}}}, \sum_{i=1}^{n_k}\hat{c}_{k_i}^{(v)}\{v-\hat{v}\}_{\mathcal{T}_{(p,q,\beta)_{k_i}}}\right). \quad (10)\end{aligned}$$

After each update step the ℓ_2 norm of the audio-visual kernels components are normalized to 1.

B. The K-SVD Algorithm

Like GA, K-SVD learns the basis functions maximizing the approximate log probability of Eq. (8) (actually it minimizes $-\log p(s|\mathcal{D})$). The idea here is to update only one atom ϕ_k at a time, together with its corresponding coefficient. Then the penalty term in Eq. (8) can be rewritten as:

$$\begin{aligned}\left\| s - \sum_k\sum_i \hat{c}_{k_i}\mathcal{T}_{k_i}\phi_k \right\|^2 &= \left\| s - \sum_{j\neq k}\sum_{i=1}^{n_j}\hat{c}_{j_i}\mathcal{T}_{j_i}\phi_j - \sum_{i=1}^{n_k}\hat{c}_{k_i}\mathcal{T}_{k_i}\phi_k \right\|^2 \\ &= \left\| E_k - \sum_{i=1}^{n_k}\hat{c}_{k_i}\mathcal{T}_{k_i}\phi_k \right\|^2 , \quad (11)\end{aligned}$$

where the subscript (p, q, α, β) of the translation operator $\mathcal{T}$ has been omitted to simplify the notation. In Eq. (11), E_k is the representation error when the k-th kernel is removed, while the second term is a weighed combination of function ϕ_k. K-SVD however does not minimize this function, since this would lead to a "non sparse" solution because no sparsity constraint is imposed on the coefficients at this dictionary update step [45]. Instead, K-SVD minimizes a penalty term that is estimated by taking into account only those signal portions over which the kernel ϕ_k is placed, so that at the update step the number of nonzero coefficients can only decrease. The K-SVD algorithm learns the kernel functions ϕ_k minimizing

$$\left\| \mathbf{R}_k^{(m)} - \Phi_k^{(m)}\hat{\mathbf{c}}_k^{(m)} \right\|^2 , \quad (12)$$

where m, $m = a, v$ denotes the modality. $\mathbf{R}_k^{(m)} \in \mathbb{R}^{L^{(m)}\times n_k}$ is the residual matrix whose columns are vectors of length $L^{(m)}$ obtained by reshaping the n_k residuals $\{m - \widehat{m}_k\}_{\mathcal{T}_{(p,q,\alpha,\beta)_{k_i}}}$, where the notation is the same of previous paragraph. $\hat{\mathbf{c}}_k^{(m)} \in \mathbb{R}^{1\times n_k}$ is a row vector of the coefficients and $\Phi_k^{(m)} \in \mathbb{R}^{L^{(m)}\times 1}$ is the column vector representing the k^{th} kernel in modality m.

Eq. (12) is easily minimized by computing the singular value decomposition (SVD) of $\mathbf{R}_k^{(m)}$, $\mathbf{R}_k^{(m)} = \mathbf{U}_k^{(m)}\mathbf{S}_k^{(m)}\mathbf{V}_k^{(m)^T}$, where $\mathbf{S}_k^{(m)}$ has the same dimension of $\mathbf{R}_k^{(m)}$, with nonnegative diagonal elements in decreasing order (i.e. $\mathbf{S}_k^{(m)}(1,1) > \mathbf{S}_k^{(m)}(2,2) > \ldots$). Eq. (12) is minimized by updating the coefficients $\hat{\mathbf{c}}_k^{(m)}$ with the first column of $\mathbf{V}_k^{(m)}$, $\mathbf{V}_k^{(m)}(:,1)^T$, multiplied by $\mathbf{S}_k^{(m)}(1,1)$, and the function $\Phi_k^{(m)}$ with the first column of $\mathbf{U}_k^{(m)}$, $\mathbf{U}_k^{(m)}(:,1)$.

To summarize, K-SVD iteratively updates the basis functions using the rule

$$\phi_k = \left(\text{reshape}(\mathbf{U}_k^{(a)}(:,1)), \text{reshape}(\mathbf{U}_k^{(v)}(:,1))\right)\,,$$

where the $\text{reshape}(\cdot)$ operator rearranges the column vectors in order to obtain the correct kernel dimensions. At the same time the coefficients corresponding to ϕ_k are also updated. Due to the form of the solution, each kernel component remains normalized.

Two major differences between GA and K-SVD algorithms should be emphasized. First, K-SVD updates each function with the principal component of the residual errors at position $\mathcal{T}_{(p,q,\alpha,\beta)_{k_i}}$ over the extent of ϕ_k (discarding the contribution of ϕ_k), while GA computes at each iteration an incremental update that is the weighed sum of the residuals. Second, the K-SVD algorithm sweeps through the kernels and uses always the updated coefficients as they emerge from preceding SVD

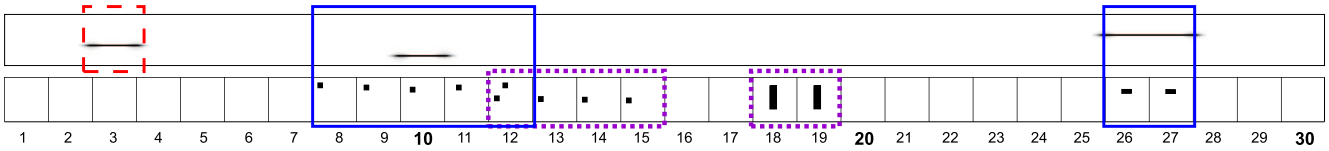


Fig. 4. Synthetic example. The top plot is the spectrogram of the audio part, consisting of three sine pulses at different frequencies. The bottom plot shows the video part consisting of 30 video frames. The sequence shows four black geometric shapes on a white background. There are five events embedded in this sequence, one audio-only structure (red dashed box), two visual-only structures (purple dotted) and two audio-visual structures (blue continuous).

steps, while GA updates the coefficients only at the successive coding steps. This should lead to a faster convergence of the algorithm [45].

V. Simulation Experiments

In this section we demonstrate the proposed framework on synthetic and natural sequences. To illustrate how the proposed audio-visual sparse coding model works we start with a simple synthetic example. In the second experiment we show that the learning algorithm is capable of discovering salient audio-visual patterns from training data. Finally, we will demonstrate that by detecting the learned multimodal patterns in audio-visual sequences exhibiting severe acoustic and visual distracters it is possible to robustly localize the audio-visual sources.

A. Experiment I: Synthetic Data

We build a 30 frames long audio-visual sequence: the soundtrack consists of three sine waves at different frequencies (Fig. 4 [Top]), while the video shows four simple black shapes, static or moving on a white background (Fig. 4 [Bottom]). The sequence represents three possible audio-visual patterns: audio-only structure (red dashed box), visual-only structures (purple dotted) and audio-visual structures (blue continuous).

The AV-MP algorithm is used to learn an audio-visual dictionary of 10 functions for this scene. The kernels have an audio component lasting 1602 samples and a video component of size 8×8 pixels and 6 frames in time. After few iterations, the algorithm yields to learn two audio-visual functions that are shown in Fig. 5 (the remaining 8 were not trained). For brevity, only the results are shown that were obtained with similarity measure C_1 and Gradient Ascent for learning.

It is obvious that the learned audio-visual bases shown in Fig. 5 represent the two crossmodal structures highlighted in blue in Fig. 4. Kernel 1 represents the audio-visual pattern on frames 26–27, with the static rectangle and the synchronous sine wave, while kernel 2 represents the moving square with the short sinusoidal pulse associated appearing on frames 8–12. This experiment demonstrates that our learning algorithm can extract meaningful bimodal structures from data. The algorithm focuses on audio-visual structures, suppressing audio-only and video-only components.

B. Experiment II: Audio-Visual Speech

The next experiment demonstrates the capability of AV-MP to recover audio-visual patterns in natural signals. The

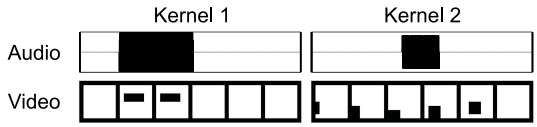
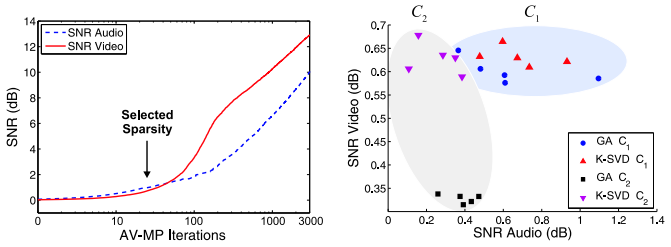


Fig. 5. The two audio-visual kernels learned for the synthetic sequence shown in Fig. 4. Audio components are on the top and video components on the bottom (each image is a video frame). Time is displayed on the horizontal axes.

performance is assessed using two different training sets. The first, $\mathcal{S}1$, consists of five audio-visual sequences representing the mouth of one speaker uttering the digits from zero to nine in English. The mouth region has been manually cropped form the first portion of sequence *s01m* of the *individuals* section of the CUAVE database [55]. Dataset $\mathcal{S}2$ is composed of six clips representing the mouth of six different persons pronouncing the digits from zero to nine. The mouths have been manually cropped from random sequences of the CUAVE database. Training audio tracks are sampled at 8 kHz and the gray-scale videos are at 29.97 fps and at a resolution of 35×55 pixels. The total length of the sequences is 1310 video frames (approximately 44 seconds) for $\mathcal{S}1$ and 1195 video frames (approximately 40 seconds) for $\mathcal{S}2$. The audio signal is directly encoded while the video is whitened using the procedure described in [44] to speed up the training.

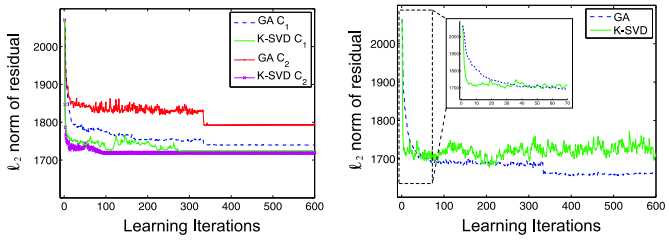
For each training set we learn four dictionaries using the similarity measures C_1 or C_2 for coding and GA or K-SVD for learning. The dictionaries learned on $\mathcal{S}1$, denoted as $\mathcal{D}1_{C_1,GA}$, $\mathcal{D}1_{C_2,GA}$, $\mathcal{D}1_{C_1,K}$, $\mathcal{D}1_{C_2,K}$, should represent collections of basis functions adapted to a particular speaker, while those learned on $\mathcal{S}2$, $\mathcal{D}2_{C_1,GA}$, $\mathcal{D}2_{C_2,GA}$, $\mathcal{D}2_{C_1,K}$, $\mathcal{D}2_{C_2,K}$, aim at being more "general" sets of audio-video atoms.

Dictionaries are initialized with thirty random audio-visual kernels with an audio component of 2670 samples and a video component of size $12 \times 12 \times 10$. Since all training and test sequences have the same spatial dimension of 35×55 pixels, we define the *sparsity* Σ of an audio-visual signal representation as the number of atoms N used to encode it divided by the duration in frames of its video component, i.e. $\Sigma = N/\#\text{Frames}$. For coding, the signal is decomposed with AV-MP using $N = 180$ audio-visual atoms for $\mathcal{S}1$ and $N = 160$ for $\mathcal{S}2$, so that for both datasets $\Sigma = 0.13$. Note that very few elements are used to represent the signals because we are interested in the audio-visual structure informative for sensor fusion. For learning, we fixed the maximum number of iterations to 1000 both for K-SVD and GA. For the GA algorithm, as suggested in [34], the learning rate η was set to



(a) (b)

Fig. 6. (a) Evolution of audio and video SNR with the number of functions used for the approximation. The x axis is in logarithmic scale to ease readability. The plot is for one sequence and one dictionary ($\mathcal{D}2_{C_1,K}$). The arrow indicates the sparsity level used for learning, $\Sigma = 0.13$. (b) Summary of audio-visual coding behavior. Points represent the five test sequence encoded with four different dictionaries. The SNR of the audio approximation is on the x axis and the SNR of the video approximation is on the y axis. Each dictionary is identified by a different marker. Similarity measure C_1 provides better audio-visual approximation results (points on the upper right part of the plot) than C_2 methods (points on the left of the figure).



(a) (b)

Fig. 7. Evaluation of different algorithmic settings for learning audio-visual codes. The plots show the evolution of the ℓ_2 norm of the residual versus the learning iteration number. (a) Results using 180 audio-visual functions for the decomposition and (b) results using 360 audio-visual functions (only curves for matching measure C_1 are shown).

5.0 for the first 333 iterations, then 2.5 for the successive 333 and finally 1 for the remainder.

Using a 2Ghz processor with 1Gb of RAM, our Matlab code takes about 150 hours to learn a dictionary on $\mathcal{S}2$ and slightly longer on $\mathcal{S}1$. However, we want to stress that learning is in general an offline procedure; hence it is not dramatic if the algorithm is complex. Furthermore, the computation can be considerably accelerated using multi-threading on a multi-core architecture. Matlab now supports multi-threading and every PC has several CPUs. The computational bottleneck of the algorithm is the projection of dictionary elements on the training signal at the coding step. Since these projections are computed as products of the Fourier transforms of atoms and signal, and multi-threading significantly speeds up the computation of the Fourier transform, the learning can be made much faster. On informal tests we have measured a speed-up factor close to 4 on a 4 CPUs architecture.

*1) **Coding quality and learning convergence**:* Here we investigate how the behavior of AV-MP depends on the choice of the similarity measure (C_1 versus C_2) and on the learning strategy (GA or K-SVD). First we measured the coding efficacy of the learned dictionaries. We use the four dictionaries learned on the more general dataset $\mathcal{S}2$ to encode five audio-visual sequences representing mouths uttering digits in English. These sequences have the same characteristics of those used for learning: resolution of 35×55 pixels and length between 150 and 195 frames.

Figure 6 (a) shows the audio and video SNR as a function of the AV-MP iterations (results for one test sequence and dictionary $\mathcal{D}2_{C_1,K}$). The arrow indicates the sparsity chosen for learning, $\Sigma = 0.13$. The sparseness level is chosen to focus on bimodally informative audio-visual structure. Obviously the SNR values are far from acceptable for encoding the entire audio-visual signal. In fact, the plot shows that it requires 3000 iterations to achieve a representation of the entire signal at moderate quality.

Each test sequence is approximated with AV-MP using a number of kernels such that for all decompositions the sparsity is $\Sigma = 0.13$, as shown in Fig. 6 (a). The scatter plot in Figure 6 (b) summarizes the SNR values for audio and video modalities for the five test clips and the four dictionaries. Each point in the plot represents one sequence. Different dictionaries are represented using different markers: circles for $\mathcal{D}2_{C_1,GA}$, triangles for $\mathcal{D}2_{C_1,K}$, squares for $\mathcal{D}2_{C_2,GA}$ and upside-down triangles for $\mathcal{D}2_{C_2,K}$.

Although the low SNR values would not allow complete signal reconstruction, they can be used to compare the different encoding methods. $\mathcal{D}2_{C_2,GA}$ has the lowest SNR values in both audio and visual components (squares grouped around the lower left corner of the plot). This low performance is presumably due to the considerably smaller number of functions constituting this dictionary (see Table I). Compared to $\mathcal{D}2_{C_2,GA}$ the dictionary $\mathcal{D}2_{C_2,K}$ achieves higher SNR for the video component but even lower SNR for the audio component (upside-down triangles on the upper left corner). Interestingly the dictionaries trained with the measure C_1 ($\mathcal{D}2_{C_1,GA}$ and $\mathcal{D}2_{C_1,K}$) have the best overall performance, they occupy the upper right corner in the scatter plot (circles and triangles). The relative performances depicted in Fig. 6 (b) are also representative for other sparseness levels (data not shown). This comparison suggests that the similarity measure C_1 encourages the encoding of joint audio-visual structures and provides better approximation results than the C_2 methods.

Next the learning convergence of the different algorithms is assessed by tracking the evolution of the ℓ_2 norm of the error between training signals and their reconstructions (Eq. (8)). Figure 7 (a) shows the error decrease during learning when dictionaries are learned on dataset $\mathcal{S}1$. In the coding step, the signal is decomposed with AV-MP using $N = 180$ audio-visual atoms ($\Sigma = 0.13$). The error decreases faster with K-SVD, no matter which similarity measure, C_1 or C_2, is used (this result also holds for $\mathcal{S}2$). Figure 7 (b) shows convergence results for GA and K-SVD (similarity measures C_1) in a regime of reduced sparseness, when the approximation uses $N = 360$ kernels ($\Sigma \approx 0.26$). In this regime the K-SVD error drops as quickly as in the case of higher sparseness, with $N = 180$, whereas GA reduces the error initially more slowly. However, after 50 learning iterations the GA error drops below the plateau of the K-SVD error and reaches error values that are significantly lower as K-SVD. Thus, these results confirm that K-SVD is a very fast and efficient learning method. Nevertheless, in some regime of sparseness and with

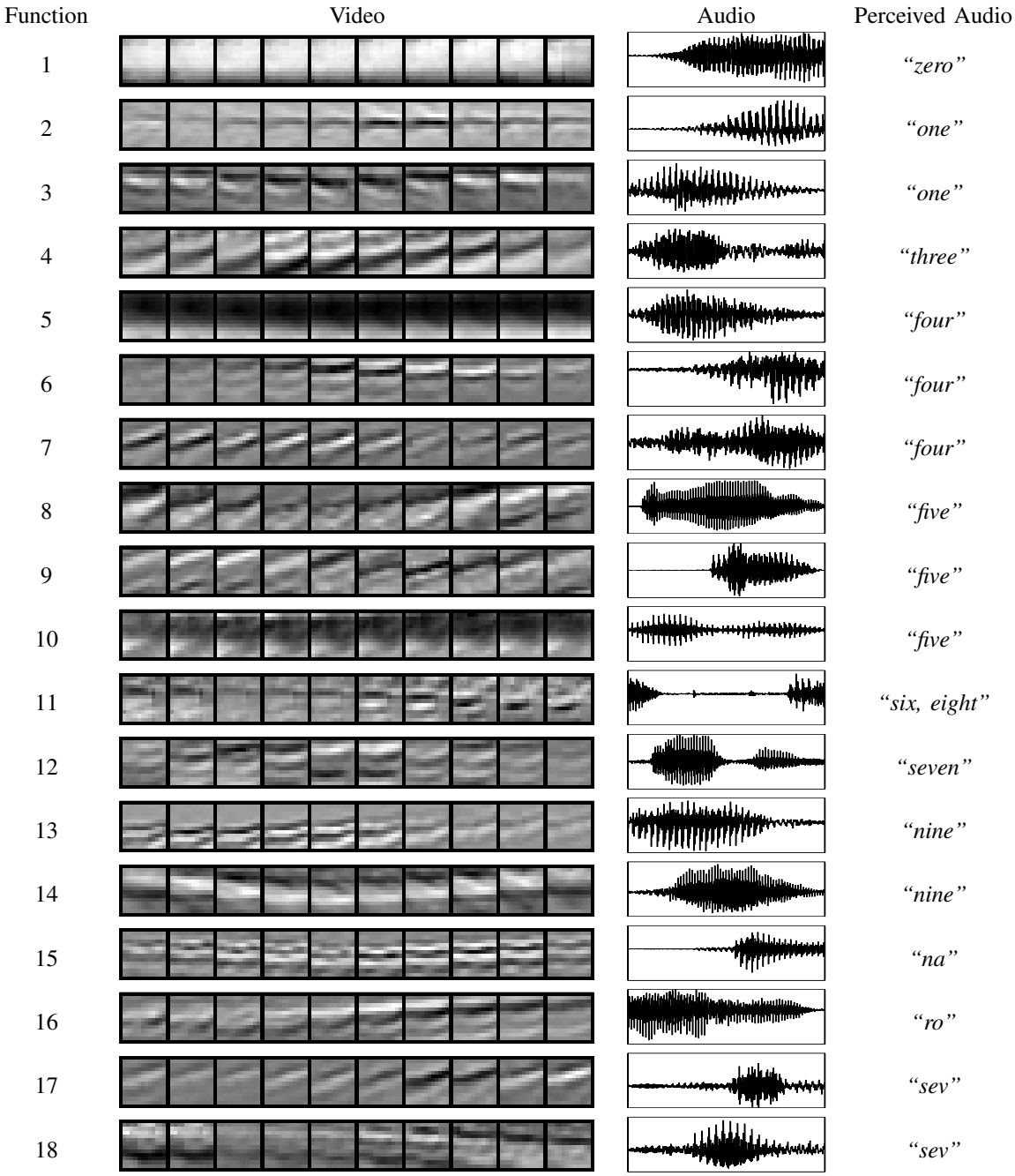
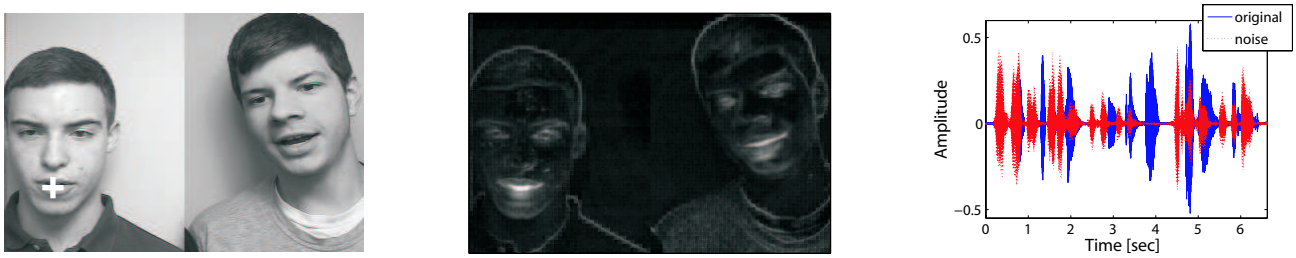


Fig. 8. Eighteen learned audio-visual kernels for $\mathcal{D}2_{C_1,GA}$. Video components are on the second column and are represented as a succession of video frames. Audio components are on the third column. Time is displayed on the horizontal axes. The meaning of the perceived audio component is given in the forth column.

enough learning iterations, the softer and less "aggressive" learning method GA can outperform K-SVD.

*2) **Learned structures in dictionaries**:* For all methods we started the training with a dictionary of 30 randomly initialized kernels. It depended on the method how many kernels were actually selected for coding and ultimately trained. Therefore a first important characterization of the methods is the effective dictionary size, that is, how many kernels were trained during learning, see Table I. Another indicator of the "goodness" of a dictionary is the number of recognizable structures in the data that are captured by dictionary elements. Here we consider only the audio part, and count the percentage of words present in the dataset (digits in English from zero to nine) that are recovered by the learning algorithm (Table I).

It is obvious that K-SVD yields generally larger dictionaries. Further, for any given training set and learning method the similarity measure C_1 yields larger dictionaries than C_2. All methods produce dictionaries with elements that represent intelligible digits or parts of digits and capture a high percentage of data structures (the ten digits). The percentage values of GA learning are somewhat higher than for K-SVD learning. As an example, Fig. 8 shows a selection of elements from dictionary $\mathcal{D}2_{C_1,GA}$. Visual basis functions are spatially localized and oriented edge or line detectors moving over



(a) (b) (c)

Fig. 9. (a) Sample frame of one test sequence. The white cross correctly pinpoints the position of the estimated audio-visual source. (b) Average motion on the clip in (a): gray-levels represent the temporal mean over the whole sequence of the absolute value of the difference between successive frames. Black pixels indicate thus no motion. Observing only the visual motion it is not possible to localize the sound source. (c) Audio signal with the speech of the real speaker (blue line) and noise signal with SNR = 0 dB (dashed red line). The test audio track is the sum of the two waveforms.

	GA		K-SVD	
	C_1	C_2	C_1	C_2
$\mathcal{S}1$	22 - 100%	13 - 80%	28 - 90%	26 - 80%
$\mathcal{S}2$	18 - 80%	10 - 60%	29 - 80%	21 - 70%

TABLE I
SUMMARY OF THE DICTIONARIES CHARACTERISTICS: NUMBER OF KERNELS (FIRST NUMBER) AND PERCENTAGE OF AUDIO DIGITS PRESENT IN THE DATA CAPTURED IN THE DICTIONARY.

time. They represent parts of the mouths making distinctive movements during the speech. The audio components can be perceived as intelligible speech signals, a few represent a part of a digit. If the same digit is captured by several kernels they usually correspond to different audio characteristics, like length or frequency content (e.g. functions 8, 9 and 10 all feature a *"five"*), or different associated video components (e.g. functions 13 and 14). Curiously, function 11 captures two digits, "six" and "eight", one after the other. This might be due to the fact that the audio-visual representation of number "six" has both low acoustic energy and small corresponding lip motion and thus it is associated with the number that re-occurs more often after it in the database, i.e. "eight".

It has to be emphasized that the set of functions shown in Fig. 8 is qualitatively different from the dictionary, learned with another method (MoTIF, see below) on the same dataset [26]. The audio-visual kernels that our AV-MP method produces are more heterogeneous and distributed in space and time. The algorithm in [26], due to de-correlation constraints between atoms, learns some spurious audio-visual kernels that do not represent any real data structure. It should be also emphasized that the kernels learned here are invariant under temporal *and* spatial shifts, while those learned in [26] are only time-invariant.

Overall, the AV-MP algorithm –unlike the older methods– seems to reflect the informative audio-visual structure in the data. The reason for this improvement is presumably because AV-MP integrates learning and coding in a way that is statistically more consistent and also biologically more plausible than in the previous model [26].

3) ***Audio-visual speaker localization****:* There is biological evidence that auditory-visual integration plays a major role in sound source localization [2]. Audio-visual source localization is also one of the primary objectives of crossmodal signal analysis and it has several practical applications [17–26]. In this experiment we show that by utilizing the learned kernels in audio-visual sequences exhibiting strong acoustic and visual distracters, it is possible to robustly localize the audio-visual source. This allows us to quantify the performances of the proposed approach and to compare them to those of our previous method [26].

For the localization task we build challenging clips using movie snippets from the *groups* section of the CUAVE dataset [55]. The test sequences consist of two persons in front of the camera arranged as in Fig. 9 (a). One person (the one on the left here) is uttering digits in English, while the other one is mouthing *exactly the same words*. As illustrated by Fig. 9 (b), both persons pronounce the same words at the same time, making it impossible to identify the sound source observing only visual motion (strong *visual distracter*). In addition, severe noise is mixed with the audio track, introducing a strong *acoustic distracter* (for an example see Fig. 9 (c)).

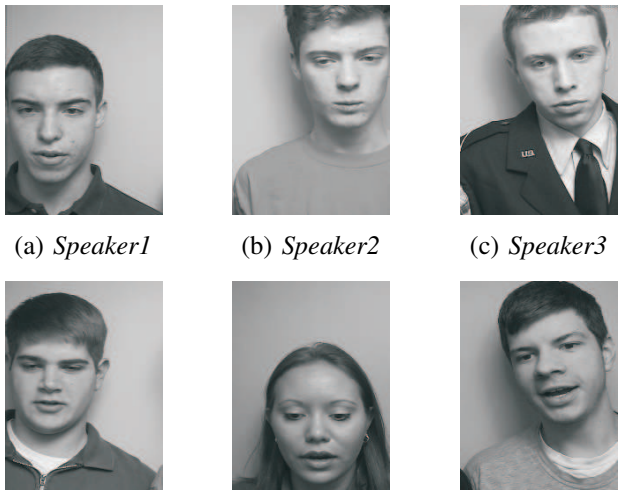
Audio-visual filtering for localization: The learned audio-visual kernels are detected on the test sequences to pinpoint the audio-visual sound source applying the procedure used in [26]. The audio track of the test clip is filtered with the audio component of each learned function. For each audio function the temporal position of the maximum projection is kept and a window of 31 frames around this time position is considered in the video. This restricted video patch is filtered with the corresponding video component and the spatio-temporal position of maximum projection between video signal and video kernel is kept. Thus, for each learned audio-visual function we obtain the location of the maximum projection over the image plane. The maxima locations on the video frames are grouped into clusters using a hierarchical clustering algorithm, as described in [26][2]. The centroid of the cluster containing the largest number of points is the estimated location of the sound source. The mouth center of the correct speaker has been manually annotated on the test sequences. The sound source location is considered to be correctly detected if it falls within a circle of radius 25 pixels centered in the labeled mouth.

[2]The MATLAB function `clusterdata.m` was used. Clusters are formed when the distance between groups of points is larger than 25 pixels. We tested several clustering thresholds and the results showed that localization performances do not critically depend on this parameter.

Audio-visual speech dictionaries: Localization is performed with the eight AV-MP dictionaries described in the previous section. Performances are compared with those of our previous algorithm, multimodal MoTIF [26]. The MoTIF algorithm extracts typical templates from audio-visual datasets representing synchronous co-occurring audio and video events. Although not a generative model (meaning that the coding is not taken into account during the learning process), the MoTIF algorithm demonstrated to achieve excellent localization results in challenging audio-visual sequences [26], out-performing previously proposed methods [24, 25]. The algorithms in [24, 25] have shown state-of-the-art localization results on the CUAVE database when compared to the work of Nock and colleagues [19] on the same data, and they have only recently been slightly outperformed by more complex methods that moreover required training [22] or face detection [23]. The MoTIF method represents thus a valid baseline for assessing the performances of the proposed framework.

Using the MoTIF algorithm we learn two audio-visual dictionaries, $\mathcal{D}1_{MoT}$ and $\mathcal{D}2_{MoT}$. $\mathcal{D}1_{MoT}$ and $\mathcal{D}2_{MoT}$ are learned on the datasets used in Sec. V-B, $\mathcal{S}1$ and $\mathcal{S}2$ respectively. Thus $\mathcal{D}1_{MoT}$ represents a set of functions adapted to one speaker, while $\mathcal{D}2_{MoT}$ is intended to be a more general audio-video dictionary. The dictionaries have the same characteristics of those learned here, that is, they are composed of the same number of audio-visual basis functions of size $12 \times 12 \times 10$ video samples and 2670 audio samples. Learning with MoTIF is faster than with the method proposed in this paper: it takes about 2 hours to build one of the dictionaries using a 2Ghz processor with 1Gb of RAM. There are two good reasons for that. First, in this paper we do not use small signal patches for training as it is done for MoTIF [26], but we consider the whole audio-visual dataset to learn temporal and position invariant basis functions. This clearly slows down the computation. Secondly, we learn here a whole audio-visual code at once, while MoTIF learns the basis functions one after the other imposing a de-correlation constraint on the learning objective. While being computationally efficient, this strategy produces artifacts in the resulting set of audio-visual functions [26].

Audio-visual test set: Test sequences contain audio tracks at 8 kHz and gray-level video components at 29.97 fps and at a resolution of 240×360 pixels. For testing we use nine different video sequences built employing clips taken from the *groups* section of the CUAVE database [55]. Three audio-visual clips show persons talking, the *Speakers* in Fig. 10 (a)-(c), and are extracted respectively from clips *g01* (first connected utterance), *g01* (second utterance) and *g04* (first utterance) of CUAVE. Three videos show persons only *mouthing* digits, the *Distracters* in Fig. 10 (d)-(f), and are extracted respectively from the first part of clips *g08*, *g17* and *g20* of CUAVE. In all clips *Speaker* and *Distracter* pronounce the same words, except for *Speaker2* who pronounces the same digits but in a different order. *Speaker1* is the same subject whose mouth was used to build dataset $\mathcal{S}1$; however, training and test sequences are different. Dataset $\mathcal{S}2$ is made of six clips, each one featuring the mouth of one subject in Fig. 10.



(a) *Speaker1* (b) *Speaker2* (c) *Speaker3*

(d) *Distracter1* (e) *Distracter2* (f) *Distracter3*

Fig. 10. The three speakers used for testing (a)-(c) and the three subjects used as video distracters (d)-(e).

Audio noise with average SNR of 0, -5 and -10 dB is mixed with the audio track. The SNR is calculated considering the signal as is, i.e. the speech with intervening silences. We use two types of noise: additive white Gaussian noise and the signal of a male voice pronouncing numbers in English (shown in Fig. 9 (c)). This second audio distracter has very similar characteristics to the target speech as it is the speech of the male speaker in sequence *g12* of the *groups* section of the CUAVE database. In addition, we test a no-noise condition for each video sequence, obtaining thus seven different audio test conditions. Considering all the possible combinations of audio and video distracters, we use a test-set of 63 sequences. We want to stress that no previous work in the field considers such a broad and challenging test set.

Localization results: Figure 9 (a) shows a sample frame of one test sequence where the white cross indicates the estimated position of the sound source over the image plane using $\mathcal{D}1_{C_1,GA}$. Indeed the found location coincides with the mouth of the actual speaker. Localization results are summarized in Table II. Values are in percentage of correct detection over the whole test set of 63 audio-visual sequences. Localization performances achieved by the dictionaries learned using AV-MP are clearly superior to those obtained using the audio-visual dictionaries learned with the MoTIF algorithm.

Gradient Ascent used with C_1 achieves the best performances with both $\mathcal{S}1$ and $\mathcal{S}2$ datasets. All methods proposed in this paper obtain perfect localization results when using the more general training set $\mathcal{S}2$. Overall, all combinations of matching measures and learning methods allow to obtain very accurate localization results, showing the robustness of the proposed framework. The learned codes can detect synchronous audio-visual patterns, allowing confident localization of sound source in complex multimodal sequences.

It is interesting to compare more in details the performances of the AV-MP algorithm and of the MoTIF method. For AV-MP we use the best settings, i.e. dictionaries $\mathcal{D}1_{C_1,GA}$ and

	AV-MP				MoTIF
	GA		K-SVD		
	C_1	C_2	C_1	C_2	
$\mathcal{S}1$	100 %	95.2 %	98.4 %	96.8 %	38.9 %
$\mathcal{S}2$	100 %	100 %	100 %	100 %	27.2 %

TABLE II
SUMMARY OF THE SOURCE LOCALIZATION RESULTS FOR AV-MP (ALL TESTED LEARNING SETTINGS) AND MOTIF. RESULTS IN PERCENTAGE OF CORRECT LOCALIZATION.

$\mathcal{D}2_{C_1,GA}$. Localization results expressed in terms of percentage of correct speaker localization for the two methods are shown in Fig. 11. Bars are grouped according to the speaker in the sequence. Bars express localization accuracy for the four dictionaries and for the two types of acoustic noise. Each bar is the average result over 12 sequences obtained using the three video distracters and the four audio noise levels. As already underlined, using $\mathcal{D}1_{C_1,GA}$ and $\mathcal{D}2_{C_1,GA}$ the speaker is correctly localized in all tested conditions. On the other hand, $\mathcal{D}1_{MoT}$ and $\mathcal{D}2_{MoT}$ exhibit poor localization performances on such a challenging database. The only exceptions are sequences involving *Speaker1* analyzed using $\mathcal{D}1_{MoT}$. This is not surprising since the audio-visual speech used for training $\mathcal{D}1_{MoT}$ is extracted from sequences of *Speaker1*. Sequences involving *Speaker2* can be better interpreted than those featuring *Speaker3*, which again is not surprising since *Speaker2* is not uttering the digits in the same order of the *Distracters*. These sequences have thus a lower degree of visual noise. The most challenging audio distracter is the added speech, which is very similar to the target audio signal. These results strongly indicate that using AV-MP, the algorithm learns audio-visual features that are more robust to strong acoustic and visual noise and that it is able to generalize better to different speakers.

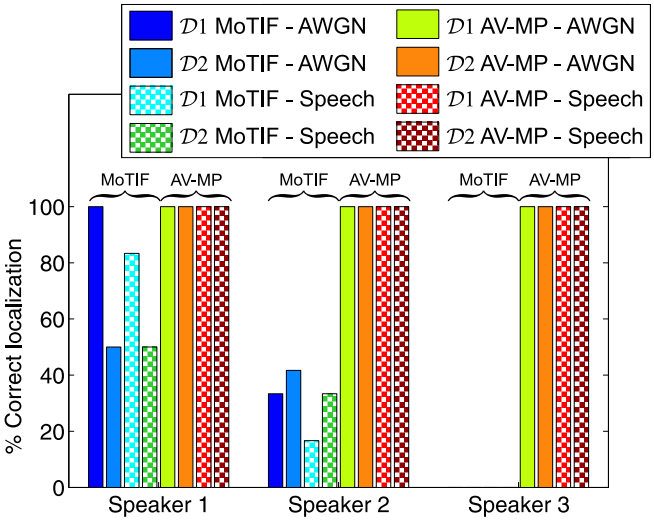


Fig. 11. Comparison between the average speaker localization performances using the dictionaries learned with the AV-MP method ($\mathcal{D}1_{C_1,GA}$ and $\mathcal{D}2_{C_1,GA}$) and with the MoTIF algorithm ($\mathcal{D}1_{MoT}$ and $\mathcal{D}2_{MoT}$). Bars are grouped according to the speaker present in the sequence. Bars express localization accuracy for the two audio noise conditions (uniformly colored bars –additive white Gaussian noise– and checked bars –added speech–) using the four learned dictionaries (first four bars –MoTIF– and last four bars –AV-MP–). Each bar is the average result over 12 sequences obtained using 3 video distracters and 4 audio noise levels (no noise, SNR = 0, -5, -10 dBs). Results are in percentage of correct localization. The improvement obtained with the proposed method is evident.

VI. SUMMARY

We have investigated algorithms to extract bimodally informative data structures from audio-visual training. The paper contains the following new results:

- *Audio-Visual Matching Pursuit (AV-MP) is described*, a method for coding audio-visual signals and learning bimodal structure in audio-visual data that is informative for tasks such as speaker localization and other fusion tasks.
- *Different audio-visual similarity measures and different learning algorithms are implemented in AV-MP and compared* in their ability to encode and learn characteristic audio-visual structure in synthetic and natural data.
- *AV-MP is tested in a challenging speaker localization task with audio and visual distracters and compared to the MoTIF algorithm.* All tested versions of AV-MP outperform MoTIF significantly.

Applications of the proposed approach can range from robust crossmodal source localization, to audio-visual source separation [16] or joint encoding of multimedia streams.

The presented model can be extended introducing the notion of scale invariance in the representation. If in the test sequences shown here the mouth regions had significantly different dimensions, or if the speech was pronounced at a different enough rate, the localization performance would probably degrade because of the fixed space-time scale of the audio-visual code. To account for spatial and temporal scale invariance a more complex architecture of the one presented here will be required. Such architecture will probably involve a multi-layer hierarchical model of audio-visual representation, in the line of recent studies on image [56, 57] and speech modelling [58]. Furthermore, a hierarchical framework seems appropriate to define a model with a slow-varying layer accounting for audio-visual synchrony and finer layers capturing audio and video details.

Interestingly, the framework developed here relies upon techniques that have been successfully employed for modeling unimodal perceptual mechanisms [35, 37, 44]. Thus, it is an intriguing possibility that our model might relate to mechanisms of audio-visual perception. It is unresolved what computation is performed by early audio-visual interactions that have been recently reported in different species [1–4]. The audio-visual learning model presented here can provide a starting point for biologically constraint models that study the computational function of early audio-visual interactions.

ACKNOWLEDGEMENTS

This work was supported by the Swiss National Science Foundation through the prospective researcher fellowship n° PBEL2-118742 to G. Monaci and by NSF through the grant n° 1-11863-26696-44-EUFTS to F. T. Sommer.

REFERENCES

[1] J. Driver and T. Noesselt, "Multisensory interplay reveals crossmodal influences on 'sensory-specific' brain regions, neural responses, and judgements," *Neuron*, vol. 57, no. 1, pp. 11–23, 2008.

[2] D. A. Bulkin and J. M. Groh, "Seeing sounds: visual and auditory interactions in the brain," *Current Opinion in Neurobiology*, vol. 16, no. 4, pp. 415–419, 2006.

[3] C. E. Schroeder and J. J. Foxe, "Multisensory contributions to low-level, unisensory processing," *Current Opinion in Neurobiology*, vol. 15, no. 4, pp. 454–458, 2005.

[4] S. Shimojo and L. Shams, "Sensory modalities are not separate modalities: plasticity and interactions," *Current Opinion in Neurobiology*, vol. 11, no. 4, pp. 505–509, 2001.

[5] R. Sekuler, A. Sekuler, and R. Lau, "Sound alters visual motion perception," *Nature*, vol. 385, no. 6614, pp. 308, 1997.

[6] H. McGurk and J. W. MacDonald, "Hearing lips and seeing voices," *Nature*, vol. 264, no. 5588, pp. 746–748, 1976.

[7] J. Driver, "Enhancement of selective listening by illusory mislocation of speech sounds due to lip-reading," *Nature*, vol. 381, pp. 66–68, 1996.

[8] J.-P. Bresciani, F. Dammeier, and M. Ernst, "Vision and touch are automatically integrated for the perception of sequences of events," *Journal of Vision*, vol. 6, no. 5, pp. 554–564, 2006.

[9] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. W. Senior, "Recent advances in the automatic recognition of audiovisual speech," *Proc. IEEE*, vol. 91, no. 9, pp. 1306–1326, 2003.

[10] C. Zhang, P. Yin, Y. Rui, R. Cutler, and P. Viola, "Boosting-based multimodal speaker detection for distributed meetings," in *Proc. IEEE MMSP*, 2006.

[11] P. Besson, V. Popovici, J.-M. Vesin, J.-Ph. Thiran, and M. Kunt, "Extraction of audio features specific to speech production for multimodal speaker detection," *IEEE Trans. Multimedia*, vol. 10, no. 1, pp. 63–73, 2008.

[12] R. Goecke, G. Potamianos, and C. Neti, "Noisy audio feature enhancement using audio-visual speech data," in *Proc. IEEE ICASSP*, 2002, pp. 2025–2028.

[13] W. Wang, D. Cosker, Y. Hicks, S. Saneit, and J. Chambers, "Video assisted speech source separation," in *Proc. IEEE ICASSP*, 2005, vol. 5, pp. 425–428.

[14] B. Rivet, L. Girin, and C. Jutten, "Mixing audiovisual speech processing and blind source separation for the extraction of speech signals from convolutive mixtures," *IEEE Trans. Audio, Speech and Language Proc.*, vol. 15, no. 1, pp. 96–108, 2007.

[15] B. Rivet, L. Girin, and C. Jutten, "Visual voice activity detection as a help for speech source separation from convolutive mixtures," *Speech Communication*, vol. 49, no. 7, pp. 667–677, 2007.

[16] A. Llagostera, G. Monaci, P. Vandergheynst, and R. Gribonval, "Blind audiovisual source separation using overcomplete dictionaries," in *Proc. IEEE ICASSP*, 2008.

[17] J. Hershey and J. Movellan, "Audio-vision: Using audio-visual synchrony to locate sounds," in *Proc. of NIPS*, 1999, vol. 12.

[18] M. Slaney and M. Covell, "FaceSync: A linear operator for measuring synchronization of video facial images and audio tracks," in *Proc. of NIPS*, 2000, vol. 13.

[19] H. J. Nock, G. Iyengar, and C. Neti, "Speaker localisation using audio-visual synchrony: an empirical study," in *Proc. CIVR*, 2003, pp. 488–499.

[20] J. W. Fisher III and T. Darrell, "Speaker association with signal-level audiovisual fusion," *IEEE Trans. Multimedia*, vol. 6, no. 3, pp. 406–413, June 2004.

[21] E. Kidron, Y. Schechner, and M. Elad, "Cross-modal localization via sparsity," *IEEE Trans. Signal Processing*, vol. 55, no. 4, pp. 1390–1404, 2007.

[22] M. Gurban and J.-Ph. Thiran, "Multimodal speaker localization in a probabilistic framework," in *Proc. EUSIPCO*, 2006.

[23] M. R. Siracusa and J. W. Fisher, "Dynamic dependency tests: Analysis and applications to multi-modal data association," in *Proc. AIStats*, 2007.

[24] G. Monaci, Ò. Divorra Escoda, and P. Vandergheynst, "Analysis of multimodal sequences using geometric video representations," *Signal Processing*, vol. 86, no. 12, pp. 3534–3548, 2006.

[25] G. Monaci and P. Vandergheynst, "Audiovisual gestalts," in *Proc. IEEE CVPR Workshop on Perceptual Organization in Computer Vision*, 2006.

[26] G. Monaci, P. Jost, P. Vandergheynst, B. Mailhe, S. Lesage, and R. Gribonval, "Learning Multi-Modal Dictionaries," *IEEE Trans. Image Process.*, vol. 16, no. 9, pp. 2272–2283, 2007.

[27] S. Mallat and Z. Zhang, "Matching Pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[28] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1998.

[29] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal of Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[30] P. Frossard, P. Vandergheynst, R. M. Figueras i Ventura, and M. Kunt, "A posterior quantization of progressive matching pursuit streams," *IEEE Trans. Signal Processing*, vol. 52, no. 2, pp. 525 – 535, 2004.

[31] D. L. Donoho and M. Elad, "Optimal sparse representation in general (nonorthogoinal) dictionaries via l1 minimization," *Proc. Nat. Aca. Sci.*, vol. 100, pp. 2197–2202, 2003.

[32] E. J. Candès and D. L. Donoho, "Curvelets - A surprisingly effective nonadaptive representation for objects with edges," in *Curve and Surface Fitting*, A. Cohen, C. Rabut, and L.L. Schmaker, Eds. Vanderbilt University Press, 1999.

[33] H. B. Barlow, "The coding of sensory messages," in *Current Problems in Animal Behaviour*, W. H. Thorpe and O. L. Zangwill, Eds. Cambridge University Press, 1961.

[34] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," *Vision Resesarch*, vol. 37, pp. 3311–3327, 1997.

[35] M. Rehn and F. T. Sommer, "A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields," *Journal of Computational Neuroscience*, vol. 22, no. 2, pp. 135–146, 2007.

[36] M.S. Lewicki and T.J. Sejnowski, "Learning overcomplete representations," *Neural computation*, vol. 12, no. 2, pp. 337–365, 2000.

[37] E. C. Smith and M. S. Lewicki, "Efficient auditory coding," *Nature*, vol. 439, no. 7079, pp. 978–982, 2006.

[38] T. Blumensath and M. Davies, "Sparse and shift-invariant representations of music," *IEEE Trans. Audio, Speech and Language Proc.*, vol. 14, no. 1, pp. 50–57, 2006.

[39] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng, "Shift-invariant sparse coding for audio classification," in *Proc. Conf. on Uncertainty in Artificial Intelligence*, 2007.

[40] B. Mailhé, S. Lesage, R. Gribonval, and F. Bimbot, "Shift-invariant dictionary learning for sparse representations: extending K-SVD," in *Proc. EUSIPCO*, 2008.

[41] P. Smaragdis, B. Raj, and M.V. Shashanka, "Sparse and shift-invariant feature extraction from non-negative data," in *Proc. ICASSP*, 2008.

[42] H. Wersing, J. Eggert, and E. Körner, "Sparse coding with invariance constraints," in *Proc. ICANN*, 2003, pp. 385–392.

[43] M. Elad and M. Aaron, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[44] B. A. Olshausen, "Sparse codes and spikes," in *Probabilistic Models of the Brain: Perception and Neural Function*, R. P. N. Rao, B. A. Olshausen, and M. S. Lewicki, Eds. MIT Press, 2002.

[45] M. Elad M. Aharon and A. M. Bruckstein, "The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Processing*, vol. 11, no. 54, pp. 4311–4322, 2006.

[46] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Journal of Constructive Approximations*, vol. 13, no. 1, pp. 57–98, 1997.

[47] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm," *IEEE Trans. Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.

[48] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006.

[49] H. B. Barlow, "What is the computational goal of the neocortex?," in *Large-scale neuronal theories of the brain*, C. Koch and J. L. Davis, Eds. MIT Press, 1994.

[50] M. McGrath and Q. Summerfield, "Intermodal timing relations and audiovisual speech recognition by normal-hearing adults," *J. Acoust. Soc. Am.*, vol. 77, no. 2, pp. 678–685, 1985.

[51] L.M. Miller and M. D'Esposito, "Perceptual fusion and stimulus coincidence in the cross-modal integration of speech," *J. Neurosci.*, vol. 25, no. 25, pp. 5884–5893, 2005.

[52] J. Chen and X. Huo, "Theoretical results on sparse representations of multiple-measurement vectors," *IEEE Trans. Signal Processing*, vol. 54, no. 12, pp. 4634–4643, 2006.

[53] S.F. Cotter, B.D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Processing*, vol. 53, no. 7, pp. 2477–2488, 2005.

[54] D. Leviatan and V. Temlyakov, "Simultaneous approximation by greedy algorithms," *Advances in Computational Mathematics*, vol. 25, no. 1-3, pp. 73–90, 2006.

[55] E. K. Patterson, S. Gurbuz, Z. Tufekci, and J. N. Gowdy, "Moving-talker, speaker-independent feature study, and baseline results using the CUAVE multimodal speech corpus," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 11, pp. 1189–1201, 2002.

[56] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neuroscience*, vol. 2, pp. 1019–1025, 1999.

[57] T. Masquelier and S. J. Thorpe, "Unsupervised learning of visual features through spike timing dependent plasticity," *PLoS Comput. Biol.*, vol. 3, no. 2, pp. e31, 2007.

[58] R. E. Turner and M. Sahani, "Modeling natural sounds with modulation cascade processes," in *Proc. of NIPS*, 2008, vol. 20.