Example-based Cross-Modal Denoising

Dana Segev and Yoav Y. Schechner
Dept. Electrical Engineering
Technion - Israel Inst. Technology
Haifa 32000, ISRAEL
sdanone@tx.technion.ac.il
yoav@ee.technion.ac.il

Michael Elad
Dept. Computer Science
Technion - Israel Inst. Technology
Haifa 32000, ISRAEL
elad@cs.technion.ac.il

Abstract

Widespread current cameras are part of multisensory systems with an integrated computer (smartphones). Computer vision thus starts evolving to cross-modal sensing, where vision and other sensors cooperate. This exists in humans and animals, reflecting nature, where visual events are often accompanied with sounds. Can vision assist in denoising *another modality? As a case study, we demonstrate this principle by using video to denoise audio. Unimodal (audio-only) denoising is very difficult when the noise source is non-stationary, complex (e.g., another speaker or music in the background), strong and not individually accessible in any modality (unseen). Cross-modal association can help: a clear video can direct the audio estimator. We show this using an example-based approach. A training movie having clear audio provides cross-modal examples. In testing, cross-modal input segments having noisy audio rely on the examples for denoising. The video channel drives the search for relevant training examples. We demonstrate this in speech and music experiments.*

1. Introduction

Smartphones, tablets and a range of other devices integrate cameras with a suite of other sensors, including microphone, accelerometer, magnetometer, etc., all accessible in synchrony through an integrated computer. The affordability and dramatic spread of these integrated systems revolutionizes computer vision. Vision becomes *cross-modal*. For example, accelerometers are used in conjunction to cameras for disambiguating structure from motion [7] and initializing image stabilization [22] and mosaicing. Cross-modal analysis is also biologically-motivated: human and animals integrate vision with other senses [12, 19]. We discuss the following cross-modal question: can one modality be used to denoise another? For example, can video be used to denoise accelerometer readings or a noisy mono soundtrack? This is a general question, thus much of the analysis in this paper is general. We focus on audio-visual (AV) cross-modal denoising as a case study (See Fig. 1). AV analysis is an emerging topic [8, 26, 30, 38, 40], prompting studies in a range of interesting tasks [2, 17, 24, 25].[1]

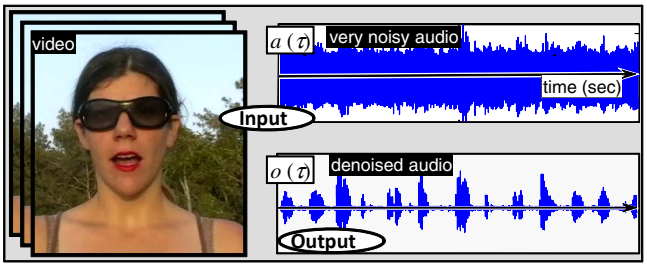

Figure 1. An input video accompanied by its soundtrack, which is highly corrupted by an unknown, unseen non-stationary noise source. The output soundtrack is denoised with the aid of the video. A 8sec section of a 240sec movie is plotted. The video with all the soundtracks can be linked through [35].

Unimodal denoising and source separation are difficult when the intensity of the noise is very high (overwhelming the signal) and non stationary (structured). This is the *cocktail party problem* [21], which is very challenging, especially when only a single sensor (microphone) is accessible [5, 34, 36]. In AV studies, *source separation* [2, 6, 17, 29] assumes that all the audio sources are visible in the field of view, e.g., a couple of speakers are seen while they speak. Here we seek more general *Denoising*: there may be no data about the auditory disturbance. The source of the noise may be in the *background*, inaccessible, unseen. In our problem, one modality suffers from strong noise which is non-stationary and unobserved directly. The modality is denoised using data from another, cleaner modality (video).[2]

[1] Some vision methods were adapted to unimodal audio analysis[23, 33].

[2] We aim to output cleaner audio, suitable for human and machine hearing. The aim is not computational word recognition [11].

Déjà Vu + Déjà Entendu

Our approach is: if "you already saw it *and* heard it," you can hear it well if you see it again. Our method uses training *examples*. A training movie has relatively clean audio. This enables prior learning of cross-modal association. Based on the learned association and clean training examples, it is possible in testing to use the clean modality (video) to help denoise the other (audio). For example, any smartphone has a microphone and a camera aiming at the user's face. Video calls from a quiet home create a clean example database. Later, calls are made in audio-noisy places such as a train station, bar or workshop. There, the clear audio example set can be used to denoise the voice. The examples are easily found since the video is relatively undisturbed. Another example is music: suppose undisturbed examples of audio-videos of a drum are obtained. Later they can be used to isolate a drum's sound in a rock show.

Example-based methods are used in various computer-vision tasks [4, 10, 15, 18]. This work builds on these contributions, extending them to cross-modal analysis.

2. Background

Unimodal single-channel audio denoising and source separation are long studied problems. In audio denoising, noise is commonly assumed to be stationary [9, 28, 37, 39]. Nevertheless, there are unimodal source separation techniques which successfully accomplish separating non-stationary sources [34, 36]. Music and speech signals have inherently different statistics. Thus, many algorithms are *distinct* for each, while some [16, 34, 36] are oriented to both. There, sparse representations of audio are used.

We also use a single microphone, and process music and speech using statistics applicable exclusively to each. The clear video enables audio denoising using simple mathematical operations. We cope with very low SNR, under overwhelming non-stationary noise, even when both the desired signal and noise originate from the same source.[3] In recent years, source separation algorithms *assisted by video* appeared. However, they [2, 6, 17] assume that *all* audio sources appear in the visual data.

3. Cross-Modal Representation

We generalize example-based denoising [10, 15, 18] to cross-modal processing, in the context of AV signals. The formulation involves the following main steps:

`1`. Defining multimodal signals.
`2`. Extracting multimodal features.
`3`. Learning feature statistics, based on training over natural signals (videos).
`4`. Performing cross-modal pattern recognition on multimodal feature vectors.

[3]For example, a xylophone melody suffering interference from a different xylophone melody.

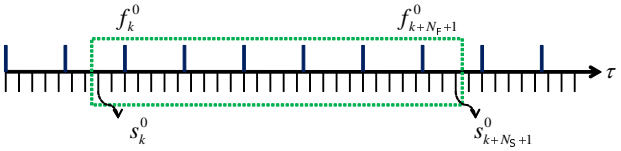

Figure 2. The initial and final frames and audio samples in temporal segment k, where τ is continues time. Here both audio and video segments have the same temporal length.

`5`. Rendering a denoised multimodal signal.

Here we focus on step `1`. Steps `3`, `4` and `5` are described in Secs. 4, 5 and 6, respectively. Step `2` is given in Sec. 7.

Joint Signals

AV signals simultaneously evolve *continuously*: at time τ, a camera senses an instantaneous object projection $\mathbf{v}(\tau)$ while a microphone senses instantaneous air-pressure, whose temporal change is the audio $a(\tau)$. The signals are sampled. The sampling periods of the audio and video are $\Delta\tau^{\mathrm{A}}$ and $\Delta\tau^{\mathrm{V}}$, respectively. Define $\rho = \Delta\tau^{\mathrm{V}}/\Delta\tau^{\mathrm{A}}$. Typically, $\mathcal{O}(\rho) \approx 800$.

A training video is divided into temporal segments, each N_{F} frames long. We define an *example* as a temporal segment composed of video ($\mathbf{v}_{\mathrm{e}}$) and audio (a_{e}) components. Consider k as an example index. The indices $[f_k^0, ..., (f_k^0 + N_{\mathrm{F}} - 1)]$ are the frames in segment k, with f_k^0 being its initial frame (See Fig. 2). The video data in this segment is a *visual-example*,

$$\mathbf{e}_k^{\mathrm{V}} = \left[\mathbf{v}_{\mathrm{e}}(f_k^0)\ \ \mathbf{v}_{\mathrm{e}}(f_k^0+1)\ \ \ldots\ \ \mathbf{v}_{\mathrm{e}}(f_k^0+N_{\mathrm{F}}-1)\right]. \quad (1)$$

The video segment is accompanied by an audio stream, containing N_{S} samples. The audio sample indices in segment k are $[s_k^0, ..., (s_k^0 + N_{\mathrm{S}} - 1)]$, where s_k^0 is the index of the first audio sample in this segment (Fig. 2). The audio data in this segment is an *audio-example*,

$$\mathbf{e}_k^{\mathrm{A}} = \left[a_{\mathrm{e}}(s_k^0),\ a_{\mathrm{e}}(s_k^0+1),\ \ldots\ a_{\mathrm{e}}(s_k^0+N_{\mathrm{S}}-1)\right]. \quad (2)$$

The corresponding examples measure the same event simultaneously in their respective modalities.

The k-th AV joint example is the row vector

$$\mathbf{e}_k \equiv \left[\mathbf{e}_k^{\mathrm{V}}\ \ \mathbf{e}_k^{\mathrm{A}}\right], \quad (3)$$

where $\mathbf{e}_k^{\mathrm{V}}$ and $\mathbf{e}_k^{\mathrm{A}}$ are given in Eqs. (1,2). The *example* set of AV signals constitutes

$$\mathbf{E} = \{\mathbf{e}_k\}_{k=1}^{N_{\mathrm{E}}}. \quad (4)$$

The examples can now be used for processing new AV test data, based on a pattern recognition system. The test set of raw measured *input test* signals is $\{\mathbf{i}_m\}_{m=1}^{M}$. Here, m indexes the input signal composed of video and audio components (Fig. 3). The input audio components are generally noisy and distorted, in contrast to signals obtained in a clutter-less environment during training. The input sequence is divided into temporal segments, each including

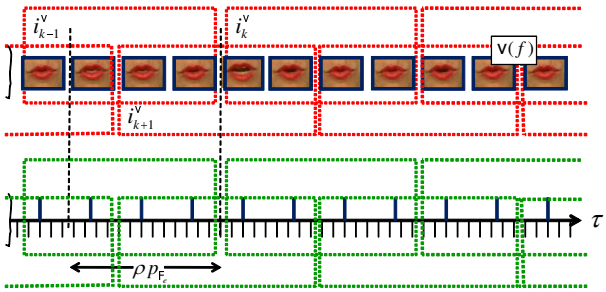

Figure 3. Periodic extraction of video and audio segments. Consecutive segments partially overlap in time.

N_{F} frames and N_{S} audio samples. The data in the m-th segment is a *visual-input* $\mathbf{i}^{\mathrm{V}}_m$ and an *audio-input* $\mathbf{i}^{\mathrm{A}}_m$. Thus, the m-th AV joint input signal is the row vector

$$\mathbf{i}_m \equiv \left[\mathbf{i}^{\mathrm{V}}_m \;\; \mathbf{i}^{\mathrm{A}}_m\right]. \tag{5}$$

There is a partial temporal overlap between input segments extracted from the raw sequence.

Each AV example and AV test input is pre-processed to yield a multimodal feature vector [6, 29]

$$\tilde{\mathbf{e}}_k = \mathcal{P}(\mathbf{e}_k) = \left[\tilde{\mathbf{e}}^{\mathrm{V}}_k \;\; \tilde{\mathbf{e}}^{\mathrm{A}}_k\right] \;\;,\;\; \tilde{\mathbf{i}}_m = \mathcal{P}(\mathbf{i}_m) = \left[\tilde{\mathbf{i}}^{\mathrm{V}}_m \;\; \tilde{\mathbf{i}}^{\mathrm{A}}_m\right]. \tag{6}$$

Here, $\tilde{\mathbf{e}}^{\mathrm{V}}_k$ and $\tilde{\mathbf{e}}^{\mathrm{A}}_k$ are respectively the visual and auditory feature row-vectors obtained from the k-th raw example. Similarly, $\tilde{\mathbf{i}}^{\mathrm{V}}_m$ and $\tilde{\mathbf{i}}^{\mathrm{A}}_m$ are respectively the visual and auditory feature vectors of the m-th raw input signal. The pre-process $\mathcal{P}$ is described in Sec. 7. Between a feature vector of the m-th input signal to that of the k-th example, $d_{\mathrm{V}}(\tilde{\mathbf{i}}^{\mathrm{V}}_m, \tilde{\mathbf{e}}^{\mathrm{V}}_k)$ and $d_{\mathrm{A}}(\tilde{\mathbf{i}}^{\mathrm{A}}_m, \tilde{\mathbf{e}}^{\mathrm{A}}_k)$ measure the distance between visual feature vectors or auditory feature vectors, respectively.[4] The distance measure can be the ℓ_2 norm.

4. Feature Statistics as a Prior

Before processing input segments, we establish the statistical nature of the signal, using training. The statistics then serve as prior knowledge, when processing a test sequence. As motivation, when listening to a familiar language, a strong prior is that some temporal sequences of syllables are highly probable (frequently appearing in words), while others much less so. The probability distribution of syllable temporal sequences is a prior, which can disambiguate speech under noise. Our work is motivated by language. However, we avoid a high-level approach that seeks division of the audio sequence into syllables. Instead, we use low-level audio features in example segments, and use training data to learn a probability distribution of temporally consecutive segments.

Our segments are 0.28 sec long, approximately the duration of a single syllable. Each example segment is turned into a feature vector $\tilde{\mathbf{e}}_k$. The set of example feature vectors $\mathbf{E}$ (Eq. 4) undergoes clustering into C clusters (we use K-means for this). The proper number for C is debatable, as there are $\mathcal{O}(10^4)$ potential syllable types. To reduce dimensionality in our experiments, we took as rule-of-thumb the number of vowel×consonant combinations (in any order), and then dictated $C = 350$. In this way, we obtain clusters of AV segments. Segments in each cluster sound/look rather similar. Segments across clusters can efficiently be used in consecutive order to render speech.

[4] To equalize the audio distance and the video distance, both feature-vectors are separately normalized.

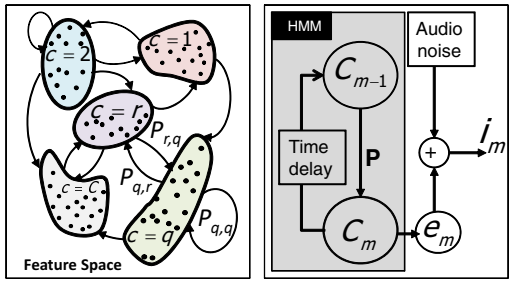

Figure 4. [Left] Feature vectors of segments are clustered. The probability of temporal transition from cluster q to cluster r is $P(q,r)$. [Right] Signals stem from a hidden Markov model (HMM): an underlying cluster index c changes in time m based on $\mathbf{P}$, yielding a clean segment $\mathbf{e}_m$ (example). Audio noise interferes, resulting in noisy raw segment $\mathbf{i}_m$.

Let segments have a fixed period of p_F frames (see Fig. 3). For the k'th example segment, the feature vector belongs to cluster $c_k = c(\tilde{\mathbf{e}}_k)$. The consecutive segment belongs to cluster $c_{k+P_F} = c(\tilde{\mathbf{e}}_{k+P_F})$. The set of all consecutive segments corresponding to fixed clusters $q, r \in [1, \ldots C]$ is

$$\Phi_{q,r} = \{k \;|\; c_k = r \;\text{ AND }\; c_{k+p_F} = q\}. \tag{7}$$

The probability for a transition from cluster q to r is estimated from the histogram of these sets,

$$P(q,r) = |\Phi_{q,r}|/N_{\mathrm{E}}. \tag{8}$$

The clusters and their transitions are illustrated in Fig. 4. In a $C \times C$ matrix $\mathbf{P}$, the (q,r) element is $P(q,r)$. This matrix is a statistical prior that expresses the joint probability for consecutive signal segments. The prior views signals as derived from a hidden Markov model (HMM) [13, 21, 31], as plotted in Fig. 4.

5. Cross-Modal Association

We seek association for each noisy input segment m to a single clean example whose index is k_m. A selected example k_m should roughly replace the input audio segment $\mathbf{i}^{\mathrm{A}}_m$. This choice should satisfy two requirements:
1. The feature vectors of example $\tilde{\mathbf{e}}_{k_m}$ and input $\tilde{\mathbf{i}}_m$ should be similar. This requirement is expressed by a *Data (fidelity) term* $\mathcal{D}$ in a cost function $\mathcal{C}$, defined next.
2. Consistency with prior knowledge. In our case, it is

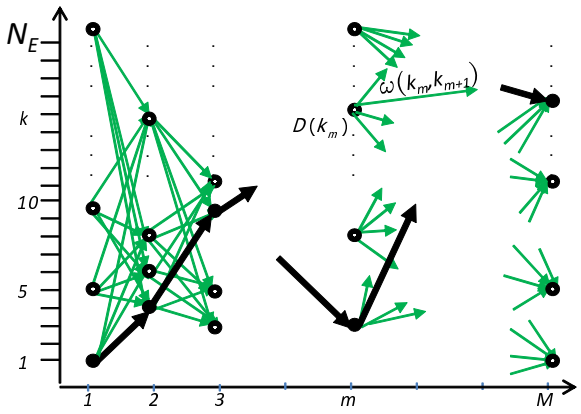

Figure 5. Each pair (m, k) is equivalent to a graph node. Directed graph edges represent transitions between examples selected for consecutive input segments. We seek the *optimal* path in the graph (thick black arrows). Out of $N_{\rm E}$ potential examples in the plot, only $K = 4$ are considered for each m in this illustration. This beneficial focus is obtained by the visual modality.

encapsulated in matrix $\mathbf{P}$ (Sec. 4), which expresses the probability that k_m is followed by k_{m+1}. This becomes a *Regularization term* $\mathcal{R}$ in $\mathcal{C}$.

One example is selected per input segment m. Concatenating the sequence of selected examples, yields a vector of indices $\mathbf{k} = [k_1, k_2, \ldots k_m \ldots k_M]$. The cost function is $\mathcal{C}(\mathbf{k}) = \mathcal{D}(\mathbf{k}) + \lambda\mathcal{R}(\mathbf{k})$, where λ weights[5] the regularization (prior) relative to the data term. We seek the overall $\mathbf{k}$ that simultaneously optimizes $\mathcal{C}$ across the entire temporal domain,

$$\hat{\mathbf{k}} = \arg\min_{\mathbf{k}} \; [\mathcal{D}(\mathbf{k}) + \lambda\mathcal{R}(\mathbf{k})]. \tag{9}$$

Once the data and regularization terms are defined, Eq. (9) can be solved. Eq. (9) is equivalent to finding a path in a graph, as illustrated in Fig. 5. A pair of input m and example k is a node in the graph. Directed edges in the graph represent transitions between examples selected for consecutive input segments. Graph node (m, k) carries a cost $\mathcal{D}(k_m)$, while an edge between (m, k_m) and $(m+1, k_{m+1})$ has a cost $\lambda\omega(k_m, k_{m+1})$, which we define in Sec. 5.1. As explained next, visual matching eliminates all examples except for $K \ll N_{\rm E}$ candidates considered per m. The graph reduces to $M \times K$ active nodes and $(M-1)K^2$ edges. Vector $\mathbf{k}$ is a path in the graph, and $\hat{\mathbf{k}}$ is the *optimal* path. The optimal path is efficiently found using *dynamic programming* [3] over this graph.

5.1. Regularization Term $\mathcal{R}$

At input segment m, the selected example is k_m. At the consecutive input segment, $m+1$, the selected example is k_{m+1}. These examples correspond to clusters c_{k_m} and $c_{k_{m+1}}$. This pair has prior probability $P(c_{k_m}, c_{k_{m+1}})$. We use it to induce a cost

[5]The value of λ was set to 1.5 in our experiments.

$$\omega(k_m, k_{m+1}) = -\log P(c_{k_m}, c_{k_{m+1}}). \tag{10}$$

A low probability transition between example segments induces a high cost, while a highly likely transition induces little or no cost. The cost $\omega(k_m, k_{m+1})$ is a *weight* corresponding to each directed edge in the graph of Fig. 5. The term $\mathcal{R}$ sums Eq. (10) over all temporal input segments:

$$\mathcal{R}(\mathbf{k}) = -\sum_{m=1}^{M-1} \log P(c_{k_m}, c_{k_{m+1}}). \tag{11}$$

5.2. Data Term $\mathcal{D}$

Data fitting in cross modal processing is challenging and interesting. This work does *not* aim to denoise *both* modalities using examples. The input video is relatively clean, with sufficient quality. Only the audio is considered as noisy, and needs to be estimated. Being of good quality, the video features $\tilde{\mathbf{i}}^{\rm V}_m$ and $\tilde{\mathbf{e}}^{\rm V}_k$ have critical importance. They have a prime role in eliminating from $\mathbf{E}$ examples that are unrelated to $\mathbf{i}_m$. In this way, visual features suggest candidate examples from $\mathbf{E}$ that are potentially close neighbors to $\mathbf{i}_m$. However, visual information often does not have a clear one-to-one correspondence to audio. In speech, different sounds may be created by similar lip movements. Hence, visual features provide a *coarse* fit in our audio denoising task, greatly reducing the number of relevant examples to $K \ll N_{\rm E}$, per input. Audio features finely discriminate among those examples.

For the m'th input segment, the set of K *visual* nearest-neighbors are found among the visual feature vectors:

$$\mathbf{K}_m = \left\{ k \mid d_{\rm V}(\tilde{\mathbf{i}}^{\rm V}_m, \tilde{\mathbf{e}}^{\rm V}_k) < d_{\rm V}(\tilde{\mathbf{i}}^{\rm V}_m, \tilde{\mathbf{e}}^{\rm V}_q),\ \forall q \notin \mathbf{K}_m \right\}. \tag{12}$$

Here, $\mathbf{K}_m \subset [1, .., N_{\rm E}]$ is of size $|\mathbf{K}_m| = K$. The subset $\{\tilde{\mathbf{e}}_k\}_{k\in\mathbf{K}_m}$ represents *candidate* example vectors, whose videos highly resemble the input video segment $\mathbf{i}^{\rm V}_m$. Among those candidates, finer discrimination is achieved by penalizing for a high distance $d_{\rm A}(\tilde{\mathbf{i}}^{\rm A}_m, \tilde{\mathbf{e}}^{\rm A}_k)$. Both criteria are compounded to a *single* data-term. Let T_m be a threshold over $d_{\rm V}(\tilde{\mathbf{i}}^{\rm V}_m, \tilde{\mathbf{e}}^{\rm V}_k)$ that sets $\mathbf{K}_m$, as in (12):

$$\begin{aligned} d_{\rm V}(\tilde{\mathbf{i}}^{\rm V}_m, \tilde{\mathbf{e}}^{\rm V}_k) &\le T_m, \quad \forall k \in \mathbf{K}_m, \\ d_{\rm V}(\tilde{\mathbf{i}}^{\rm V}_m, \tilde{\mathbf{e}}^{\rm V}_q) &> T_m, \quad \forall q \notin \mathbf{K}_m. \end{aligned} \tag{13}$$

For audio, define $d^{\max}_{\rm A} \equiv \max_{m,k} d_{\rm A}(\tilde{\mathbf{i}}^{\rm A}_m, \tilde{\mathbf{e}}^{\rm A}_k)$. All the audio vector-distances are normalized by $d^{\max}_{\rm A}$, yielding

$$\hat{d}_{\rm A}(\tilde{\mathbf{i}}^{\rm A}_m, \tilde{\mathbf{e}}^{\rm A}_k) = d_{\rm A}(\tilde{\mathbf{i}}^{\rm A}_m, \tilde{\mathbf{e}}^{\rm A}_k)/d^{\max}_{\rm A}\ , \tag{14}$$

where $0 \le \hat{d}_{\rm A} \le 1$. A data-fitting cost for a selected example k_m can then be posed as

$$\mathcal{D}(k_m) = \ [d_{\rm V}(\tilde{\mathbf{i}}^{\rm V}_m, \tilde{\mathbf{e}}^{\rm V}_{k_m}) \le T_m]\ [\hat{d}_{\rm A}(\tilde{\mathbf{i}}^{\rm A}_m, \tilde{\mathbf{e}}^{\rm A}_{k_m}) - 1]. \tag{15}$$

In Eq. (15), the left bracketed term is boolean, and it expresses the requirement that $k_m \in \mathbf{K}_m$. The right bracketed term is continuous-valued, and it expresses the requirement

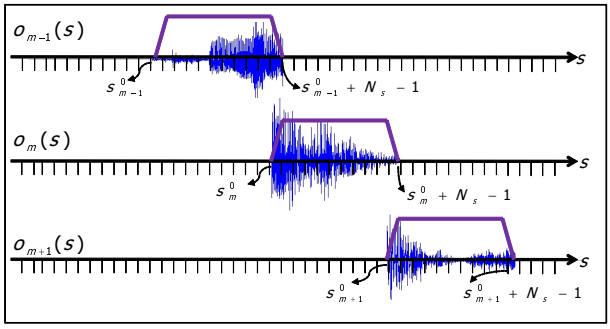

Figure 6. Output soundtrack m is silent, except for $[s^0_m, ..., (s^0_m + N_\mathrm{S} - 1)]$, which includes the denoised content $\mathbf{e}^\mathrm{A}_{\hat{k}_m}$. The trapezoidal windows illustrate a weighting function $w(s - s^0_m)$ used in audio mosaicing. Mosaicing output audio segments is done by fading in and out each segment, using the weighting function.

for low audio misfit cost. The lower bound of $\mathcal{D}(k_m)$ is -1, obtained when both the visual difference is low $(d_\mathrm{V} \leq T_m)$ *and* the audio perfectly fits $(\hat{d}_\mathrm{A} \to 0)$. This is the best we can strive for. The upper bound of $\mathcal{D}(k_m)$ is 0, obtained if the visual difference is high $(d_\mathrm{V} > T_m)$ *or* the audio fits poorly $(\hat{d}_\mathrm{A} \to 1)$. The cost $\mathcal{D}(k_m)$ is associated with node (m, k) in the graph of Fig. 5. The data term of $\mathcal{C}$ sums Eq. (15) over all temporal segments of the input sequence

$$\mathcal{D}(\mathbf{k}) = \sum_{m=1}^{M} [d_\mathrm{V}(\tilde{\mathbf{i}}^\mathrm{V}_m, \tilde{\mathbf{e}}^\mathrm{V}_{k_m}) \leq T_m][\hat{d}_\mathrm{A}(\tilde{\mathbf{i}}^\mathrm{A}_m, \tilde{\mathbf{e}}^\mathrm{A}_{k_m}) - 1]. \quad (16)$$

6. Rendering a Denoised Soundtrack

The selected digital audio track example $\mathbf{e}^\mathrm{A}_{\hat{k}_m}$ is a clean version of the noisy input $\mathbf{i}^\mathrm{A}_m$. A denoised output audio $\mathbf{a}^\mathrm{output}$ can apparently be created by concatenating the clear tracks corresponding to each consecutive input segment, $\mathbf{a}^\mathrm{output}_\mathrm{simplistic} = [\mathbf{e}^\mathrm{A}_{\hat{k}_1}\ \mathbf{e}^\mathrm{A}_{\hat{k}_2}\ \mathbf{e}^\mathrm{A}_{\hat{k}_3}\ \ldots\ \mathbf{e}^\mathrm{A}_{\hat{k}_M}]$. As in image mosaicing, a long soundtrack is created by stitching short audio segments. A temporal segment m partial overlaps with consecutive and preceding segments.

The initial audio sample in each input segment is

$$s^0_m = 1 + (m-1)\rho p_\mathrm{F}\ . \quad (17)$$

From (17), segment m is $[s^0_m, \ldots (s^0_m + N_\mathrm{S} - 1)]$. A denoised soundtrack $\mathbf{o}_m$ corresponding to segment m is silent (zero valued) at all times, except for the specific temporal samples $[s^0_m, \ldots (s^0_m + N_\mathrm{S} - 1)]$ as illustrated in Fig. 6. There, the optimized example corresponding to segment m is $\hat{k}_m$. Its corresponding audio is $\mathbf{e}^\mathrm{A}_{\hat{k}_m}$. This audio is finely aligned.[6] as explained in [35]. The sequence $\mathbf{o}_m$ is feathered using a weighting function $w_m(s) = w(s - s^0_m)$. The output of our system is therefore the audio

$$a^\mathrm{output}(s) = \Sigma_{m=1}^{M}\, o_m(s) w(s - s^0_m)\ . \quad (18)$$

[6]The temporal resolution of the video (upon which the examples $\hat{k}_m$ are primarily selected) is too coarse for audio. Thus, the audio undergoes a finer temporal alignment [35].

7. Auditory and Visual Features

7.1. Audio Features

Auditory perception is sensitive to far fewer degrees of freedom than those of a raw soundtrack. The known art determines the essential compact features of audio, such that a simple d_A measures the essential differences between perceived sounds. For stationary sounds in *speech*, such features are the mel-frequency Cepstral coefficients (MFCCs) [32]. Sound is generally not stationary throughout the temporal extent of an audio segment. Thus, each segment is divided into N_T brief consecutive tiles, each indexed by t. Per tile t, the MFCCs yield a feature row-vector $\mathbf{m}_t$. Thus, overall, the audio feature vector of the whole segment is $\tilde{\mathbf{e}}^\mathrm{A} = [\mathbf{m}_1,\ \mathbf{m}_2,\ \ldots,\ \mathbf{m}_{N_\mathrm{T}}]$, similarly for examples and input. In our speech experiments, we used $N_\mathrm{T} = 7$, and each $\mathbf{m}_t$ contains just 13 MFCCs.

In music experiments, we used a spectrogram summation over time as the audio feature vector. This implies the harmonic structure typical to musical instruments.

7.2. Visual Features

Extraction of visual features has three main steps:

`i.` Locking on the object of interest.

`ii.` Extracting global motion by tracking.

`iii.` Extracting features unrelated to global motion.

In speech, the object of interest is around the mouth. Step `ii` involves tracking the global location and orientation of the mouth. Other than image registration, we make no use of this global state here. Step `iii` extracts features associated with lip motion. We used low-level features: a stabilized region of interest around the mouth underwent spatio-temporal discrete cosine transformed (DCT). Based on the set $\mathbf{E}$, N_DCT DCT coefficients that have the highest variance are found. These N_DCT DCT coefficients form the visual feature vector.[7]

In musical instruments, the motion of interest depends on the kinetics of instrument operation. For a stationary xylophone, the interest is on the global motion of a hitting mallet. Training examples in $\mathbf{E}$ are sequences having exclusive xylophone sounds: example k corresponds to a hit on the k'th bar of the xylophone. A sound commences when the mallet hits an object projected to a pixel whose horizontal and vertical coordinates are x^e and y^e, respectively. The hit is a vertical *minimum* point. In the input sequence, we need to spot similar events. A local vertical minimum in the trajectory $\mathbf{x}^\mathrm{i}(f) = [x^\mathrm{i}(f), y^\mathrm{i}(f)]$ of input segment m is checked by the logical (binary) operator

$$\mathcal{M} \equiv \left[y^\mathrm{i}(f^0_m + 1) < \min\{y^\mathrm{i}(f^0_m), y^\mathrm{i}(f^0_m + 2)\}\right]\ , \quad (19)$$

where $N_\mathrm{F} = 3$. Being in the vicinity of the k'th bar is determined by the logical operator

[7]In our experiments, the mouth is bounded by a 71×91 window, $N_\mathrm{F} = 7$ and $N_\mathrm{DCT} = 1400$.

Noise Name	Input		
	Digits	*Barman*	*Xylophone*
Sweet	0.07	0.36	0.9
Phil	0.09	0.59	-
Female speech	1.05	1.1	-
Male speech	2.4	0.3	-
White Gaussian	1	0.38	0.001
Xylophone	-	-	1

Table 1. SNR values of each signal-noise combination. Added noises are: [Sweet] Music from the song *Sweet Child of Mine* by GNR. [Phil] Music from the song *I Wish It Would Rain Down* by Phil Collins. [Male speech] and [Female speech] from TIMIT database [20].

$$\mathcal{H}_k \equiv \left\{ \|\mathbf{x}^{\mathrm{i}}(f_m^0 + 1) - \mathbf{x}^{\mathrm{e}}(f_k^0 + 1)\|_2 < H \right\} . \quad (20)$$

Here H is a loose spatial tolerance for potentially being near a bar. It allows K bars to yield $\mathcal{H}_k = 1$ per frame, since the visual trajectory has ambiguities. The ambiguities stem from the xylophone being a 3D object (two levels) projected to a 2D video, and from a too coarse spatiotemporal resolution of the video, particularly for fast playing motion. Overall, the measure

$$d_{\mathrm{V}}(\tilde{\mathbf{i}}_m^{\mathrm{V}}, \tilde{\mathbf{e}}_k^{\mathrm{V}}) = \{\mathrm{NOT}\ [\mathcal{M}\ \mathrm{AND}\ \mathcal{H}_k]\} \quad (21)$$

has a minimum value (zero) only at input video segments i_m^{V} having spatial proximity to a sound-associated example $\mathbf{e}_k^{\mathrm{V}}$, while being at a minimum of the trajectory. Otherwise, $d_{\mathrm{V}} = 1$. If no sound-associated example $\mathbf{e}_k^{\mathrm{V}}$ matches i_m^{V} using these features, then the denoised audio prompted by segment m is *silence*. In other cases, Eqs. (12,21) yield K candidate examples, corresponding to different bars.

8. Experiments

We used a simple camcorder working at 25Hz video rate. Audio was sampled at 8kHz for speech and 16kHz for music. After the recordings, we added **strong** audio noise to the test sequences, making them difficult to comprehend (SNR can[8] be $\ll 1$). The noise types were varied and often *highly non-stationary*. They are listed in Table 1

We made music and two speech denoising experiments. As in [1, 6, 27, 29], we used a corpus of words, particularly digits $\{0, 1, ..., 9\}$. Our first speech experiments included randomly pronounced digits. Training lasted 60 sec, and testing was based on a different video lasting 240 sec. Our second experiment is of *bartender* speech, where a person says names of 30 beverages under strong noise from surrounding music. This is a much wider and more challenging corpus than digits. Training lasted 350 sec. The distinct testing video lasted 48 sec, corrupted by each noise type. Naturally, the sounds and appearances of lip motion varied during speech repetition.

[8]SNR is measured by the ratio of signal and noise energies.

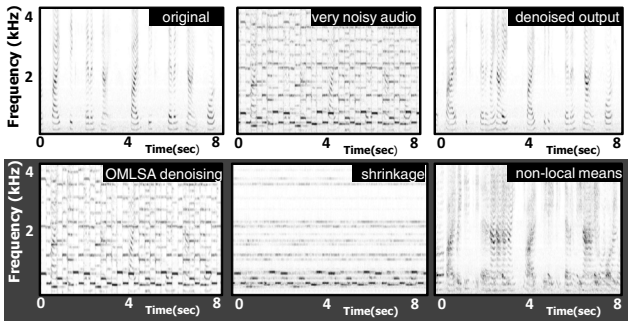

Figure 7. Spectrograms corresponding to the soundtracks described in Fig. 1 (8 out of 240 seconds). The noise is very intense (SNR$= 0.7$). Top-right: our result. Bottom: results of other methods. Our method successfully denoised the signal while the other methods failed.

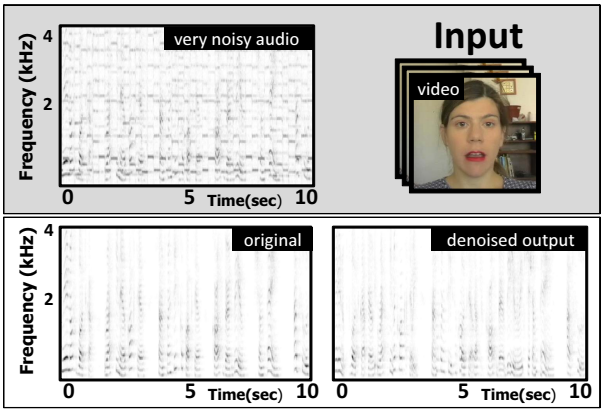

Figure 8. Spectrograms corresponding to the soundtracks of the *bartender* experiment (10 out of 48 seconds).

For speech, we used $N_{\mathrm{F}} = 7$ and $N_{\mathrm{S}} = 2240$, corresponding to 0.28 sec. We used $p_F = 6$ on the test input. Sample frames and an 8 sec section of the noisy *digits* input are shown in Fig. 1, as is the corresponding denoised result. The latter is very similar to the original plot (not shown, as there is hardly any difference). This is also seen in spectrograms[9] of the signals (Fig. 7).

The same applies throughout the long test sequences. As a consequence, the spoken digits are comprehendible, except for a few misses. This is acknowledged by watching (and hearing) the movies which can be linked through [35]. The *bartender* experiment shows that the method can also be applied on a richer domain of signals. Sample frames, a 10 sec section of a noisy *barman* input spectrogram and the corresponding denoised result are shown in Fig. 8.

For music, a xylophone was played. Training lasted 103 sec, and testing was based on a different video lasting 100 sec. We pruned $\mathbf{E}$: all examples were discarded, except for those having audio onsets [2]. The examples'

[9]For clarity, the contrast of all shown spectrograms was stretched in the same manner in the display. Furthermore, the display is negative (dark elements express high energy).

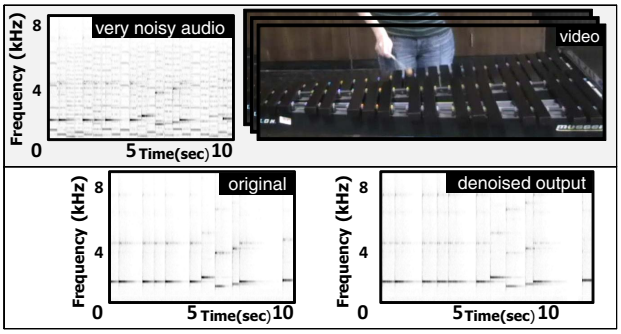

Figure 9. [Top] An input video of a person playing a xylophone. The accompanied soundtrack is very noisy. The noise source is unknown, unseen in the field of view and highly non-stationary. [Bottom] The audio is denoised, with the aid of the video, yielding clear sound and comprehension. The video with all the soundtracks can be linked through [35].

duration varies, $N_{\mathrm{S}} \in [12800 - 25600]$, according to the lingering of each note. We set $\mathbf{P}$ to be uniform here. The noisy test movies included playing several tunes, to which the mentioned strong noises were added. One of the added noises is another melody of this xylophone. This kind of separation (denoising) is very challenging to unimodal audio techniques. The spatial tolerance H (Eq. 20) was set to detect up to $K = 5$ candidates bars. During testing, the method handled well music that was played fast, in an arbitrary rhythm, since here $p_F = 1$. Fig. 9 shows sample frames, a spectrogram of a 10 sec section of the noisy input and the corresponding denoised spectrogram. The result is very similar to the clear original. Some differences stem from example segments having lower/higher sound intensities than the noisy input. Nevertheless, the resulting music completely got rid of the noise, and was recovered nicely. This is clearly demonstrated by watching (and hearing) the movies linked through [35].

Comparison to Other Methods

Consistently, cross-modal processing was by far superior to unimodal (audio-only or video-only) denoising:
$\{$Process 1$\}$ A process that was run on the examples and noisy inputs, unimodally on audio without video. The rendered results sound as a mess.
$\{$Process 2$\}$ Optimization of $\hat{\mathbf{k}}$ used only video. The results sound more coherent than audio-only results, but still unclear.
$\{$Process 3$\}$ We applied several unimodal audio denoising methods. State-of-the-art OMLSA [9], shrinkage [14] and non-local means denoising borrowed from image processing. In non-local means, the K examples are extracted by generalizing Eq. (12) to bimodal feature vectors, per input segment m.

$$\mathbf{K}_m = \left\{ k \;\middle|\; d(\tilde{\mathbf{i}}_m, \tilde{\mathbf{e}}_k) < d(\tilde{\mathbf{i}}_m, \tilde{\mathbf{e}}_q), \quad \forall q \notin \mathbf{K}_m \right\}. \quad (22)$$

Noise Name	Minimizing $\mathcal{D}$	Minimizing $\mathcal{C}$
Sweet	24%	67%
Phil	25%	68%
Female speech	30%	75%
Male Speech	29%	73%
White Noise	16%	64%

Table 2. Quantitative Evaluation. The correspondence rate of $\hat{\mathbf{k}}_{\mathrm{clear}}$ and $\hat{\mathbf{k}}$ in the *Barman* experiment.

All unimodal audio denoising results were very poor (hear in [35]).

There are unimodal denoising methods that cope with non-stationary noise [36]. However, we show a scenario that would truly challenge unimodal denoising. One xylophone melody serves as interfering noise overlayed on an another, desired, xylophone melody. Produced by the same instrument, both have the same sounds. Indeed, our method handles this scenario [35].

To quantify the performance in music, we counted the percentage of correctly played notes. On average, only 30% of the notes were correct in $\{$Process 1$\}$. Errors include missing notes, inserting notes at the wrong time and swapping notes. In cross-modal AV processing, 85% of the notes were correct. We used the following criterion for speech. First, an original sequence was "denoised" by the method. The selected example sequence in this case is $\hat{\mathbf{k}}_{\mathrm{clear}}$. When denoising a noisy version of the sequence, the result is $\hat{\mathbf{k}}$. The rate of correspondence between $\hat{\mathbf{k}}_{\mathrm{clear}}$ and $\hat{\mathbf{k}}$ is our criterion. The correspondence rate in $\{$Process 1$\}$ was zero. This rate was 19% in $\{$Process 2$\}$ and 64%-75% in cross-modal processing (Table 2).

9. Discussion

The features and the recovery algorithm should seek even better generalization, to treat movies that have a wider variety. It can be useful if training is done using ordinary, noisy examples. This paper relied on basic pattern recognition tools: nearest neighbors and HMM. However, highly elaborate tools have been developed for unimodal tasks. This work may motivate generalization of these advanced tools to cross-modal denoising.

Acknowledgments

We thank Israel Cohen and Lihu Berman for useful discussions, Marina Alterman for playing the xylophone and Sharon Gannot for enabling us to use his lab. Yoav Schechner is a Landau Fellow, supported by the Taub Foundation. This work was supported by the Israel Science Foundation (ISF) Grant 1031/08. It was conducted in the Ollendorff Minerva Center. Minerva is funded through the BMBF.

References

[1] Aharon M., Kimmel R.: Representation analysis and synthesis of lip images using dimensionality reduction. IJCV

67:297–312, 2006.

[2] Barzelay Z., Schechner Y.Y.: Harmony in motion. Proc. IEEE CVPR, 2007.

[3] Bellman R. *Dynamic Programming*. 1957, Princeton University Press

[4] Bregler C., Covell M., Slaney M. Video Rewrite: Driving Visual Speech with Audio. Proc. ACM SIGGRAPH. 353–360, 1997.

[5] Bimbot F., Benaroya L., Gribonval R.: Audio source separation with a single sensor. IEEE Trans. ASSP **14**:191–199, 2006.

[6] Casanovas A. L., Monaci G., Vandergheynst P.: Blind audio-visual source separation using sparse representations. IEEE ICIP 2007.

[7] Chang Y. J., Chen T: Multi-View 3D Reconstruction for Scenes under the Refractive Plane with Known Vertical Direction. In Proc. ICCV, 2011.

[8] Choudhury T., Rehg J., Pavlovic V., Pentland A.: Boosting and structure learning in dynamic bayesian networks for audio-visual speaker detection. In Proc. ICPR pp. 789–794, 2002.

[9] Cohen I., Berdugo B.: Speech enhancement for non-stationary noise environments. Signal Processing **81**:2403–2418, 2001.

[10] Criminisi A., Perez P., Toyama K.: Region filling and object removal by exemplar-based image inpainting. IEEE Trans. IP **13**:1200–1212, 2004.

[11] Deligne S., Potamianos G., Neti C.: Audio-visual speech enhancement with AVCDCN (audio-visual codebock dependent cepstral normalization). IEEE Worksh. Sensor Array & Multichannel SP, 68–71, 2002.

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

[13] Dupont S., Luettin J.:Audio-visual speech modeling for continuous speech recognition. IEEE Trans. Multimedia **2**:141-151, 2000.

[14] Elad M.: Sparse and Redundant Representations - From Theory to Applications in Signal and Image Processing. Springer New-York, 2010.

[15] Elad M., Datsenko D.: Example-based regularization deployed to super-resolution reconstruction of a single image. The Computer Journal, **50**:1-16, 2007.

[16] Fevotte C., Daudet L., Godsill S. J., Torresani B.: Sparse regression with structured priors: application to audio denoising. IEEE ICASSP, 2006.

[17] Fisher III J. W., Darrell T., Freeman W. T., Viola P.: Learning joint statistical models for audio-visual fusion and Segregation. in Proc. NIPS **13**, 772–778, 2001.

[18] Freeman W. T., Pasztor E. C., Carmichael O. T.: Learning low-level vision. IJCV **40**:25–47, 2000.

[19] Gutfreund Y., Zheng W., Knudsen E. I.: Gated visual input to the central auditory system. Science **297**:1556–1559, 2002.

[20] Garofolo J. S.: Getting Started With the DARPA TIMIT CD-ROM: An Acoustic-Phonetic Continuous Speech Database. Gaithersburg, MD: National Inst. of Standards and Technol. (NIST) 1993.

[21] Hershey J., Casey M.: Audio-visual sound separation via hidden markov models. in Proc. NIPS pp. 1173–1180, 2001.

[22] Karpenko A., Jacobs D. E.,Baek J., Levoy .M,: Digital Video Stabilization and Rolling Shutter Correction using Gyroscopes. Stanford CSTR pp. 2011–03, 2011.

[23] Ke Y., Hoiem D., Sukthankar R.: Computer vision for music identification. Proc. IEEE CVPR pp. 597– 604, 2005,

[24] Khalidov V., Forbes F., Hansard M., Arnaud E., Horaud R.: Audio-Visual clustering for 3D speaker localization. Proc. MLMI Workshop, 2008.

[25] Kidron E., Schechner Y. Y., Elad M.: Pixels that sound. Proc. IEEE CVPR pp. 88–95, 2005.

[26] Liu Y., Sato Y.: Visual localization of non-stationary sound sources. In Proc. Multimedia, 2009.

[27] Luettin J., Thacker N. A., Beet S. W.: Speechreading using Shape and Intensity Information. In ISCA, 1996.

[28] Martin R.: An efficient algorithm to estimate the instantaneous SNR of speech signals, Proc. EUROSPEECH:1093–1096, 1993.

[29] Monaci G., Sommer F., Vandergheynst P.: Learning bimodal structure in audio-visual data. IEEE Trans. NN, 2009.

[30] O'Donovan A., Duraiswami R., Neumann J.: Microphone arrays as generalized cameras for integrated audio visual processing. Proc. IEEE CVPR pp. :1–8, 2007.

[31] Potamianos G., Neti C., Gravier G., Garg A., Senior A.: Recent advances in the automatic recognition of audiovisual speech. Proc. IEEE, **91**:1306-1326, 2003.

[32] Quatieri T. F.: Discrete Time Speech Signal Processing, Principles and Practice. Prentice Hall, chapter 14, 2002.

[33] Sarel B., Irani M.: Separating transparent layers of repetitive dynamic behaviors. Proc. IEEE ICCV pp. 26-32, 2005.

[34] Schmidt M. N., Olsson R. K.: Single-channel speech separation using sparse non-negative matrix factorization. Conf. Spoken Language Processing, 2006.

[35] Segev D., Schechner Y. Y., Elad M.: Cross-modal denoising: supplemental online material. *www.ee.technion.ac.il/∼yoav/research/CM-denoising.html*

[36] Smaragdis P., Shashanka R. and Raj B.: A Sparse Non-Parameteric Approach for Single Channel Separation of Known Sounds. In Proc. NIPS pp. 1705–1713, 2009.

[37] Sohn J., Kim N.S, Sung W.: A statistical model-based voice activitydetector. IEEE SP Lett **6**:1–3, 1999.

[38] Song M., Bu J., Chen C., Li N.: Audio-visual based emotion recognition-a new approach. Proc. IEEE CVPR 2004.

[39] Stahl V., Fischer A., Bippus R.: Quantile based noise estimation for spectral subtraction and Wiener filtering. Proc. ICASSP pp. 1875-1878, 2000.

[40] Vajaria H., Islam T., Sarkar S., Sankar R., Kasturi R.: Audio segmentation and speaker localization in meeting videos. In ICPR pp. 1150–1153, 2006.