CCIT Report 810 EE 1767
Example-based Cross-Modal Denoising: Supplementary Document

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

In Ref. [1] Example-based Cross-Modal Denoising *we refer to temporal cropping of audio-visual data and fine alignment during synthesis of a denoised audio. In this additional document we give more details.*

1. Temporal Density

The input segments are extracted from the input video. A simple way is to extract segments using a fixed period of $p_{\rm F}$ frames, as illustrated in Fig. 1. Then, the initial frame in segment m is

$$f_m^0 = 1 + (m-1)p_{\rm F} \ . \tag{1}$$

Segment m comprises frames $[f_m^0, ..., (f_m^0 + N_{\rm F} - 1)]$ of the input video, where $N_{\rm F}$ is the number of frames per segment. Note that $p_{\rm F} \leq N_{\rm F}$, so that each video frame is extracted in at least one segment. If $p_{\rm F} < N_{\rm F}$, there is temporal overlap between consecutive video segments. Similarly, the segment encapsulates $N_{\rm S}$ input audio samples, indexed by $[s_m^0, ..., (s_m^0 + N_{\rm S} - 1)]$. Here,

$$s_m^0 = 1 + (m-1)\rho p_{\rm F} \ , \tag{2}$$

where

$$\rho = \Delta\tau^{\rm V}/\Delta\tau^{\rm A} \tag{3}$$

is the ratio between the temporal sampling periods $\Delta\tau^{\rm A}$ and $\Delta\tau^{\rm V}$ of the continuous-time audio and inputs. We used $\rho p_{\rm F} < N_{\rm S}$ in order to have temporal overlap between audio segments.

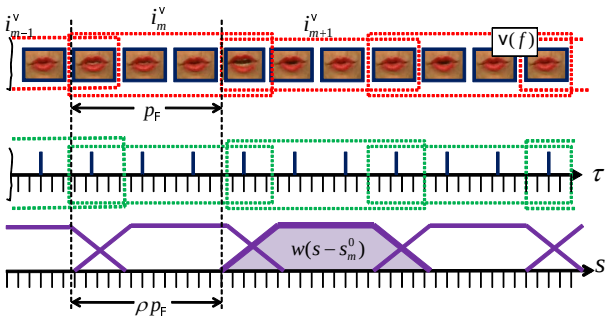

Figure 1. Periodic extraction of video and audio segments. Consecutive segments partially overlap in time. [Bottom]: Mosaicing output audio segments is done by fading in and out each segment, using a weighting function $w(s - s_m^0)$.

There is a difference between *input* and *example* cropping of data. To fully exploit the short training, it is sometimes useful to extract as many examples as possible from the training video. The highest yield of segments to the example set is achieved if a new example segment k starts after each frame (as if $p_{\rm F} = 1$):

$$f_k^0 = k, \qquad s_k^0 = 1 + (k-1)\rho \ . \tag{4}$$

The hidden Markov model, described in Section. 4 of [1] generally does *not* use $p_{\rm F} = 1$. Rather the value of $p_{\rm F}$ for the model is the one used in the *input* segments, as we describe now.

For an *input speech* sequence to be denoised, $\rho p_{\rm F}$ is in the order of $N_{\rm S}$: we used $p_{\rm F} = 7$, corresponding to 0.28sec. There are two benefits for this diluted extraction. First, this time span is comparable the duration of a syllable. Therefore, we can use directly the hidden Markov model, which we describe in Section. 4 of [1]. Had the segments been much shorter, capturing the relation between consecutive segments would have diminished. The second benefit is reduction of processing time.

This said, a periodic extraction as in Eqs. (1,2,4) is generally *not necessary*, in neither input nor training. It should often be more efficient to extract segments adaptively, focusing on audio onsets. Alternatively, the example set $\mathbf{E}$ can be automatically *pruned*, to eliminate useless example segments. This is what we do for music sequences: $p_{\rm F} = 1$ at the input, while examples correspond only to onsets and their lingering sound.

2. Tuning to Audio Resolution

As written in Section 6 of [1], the digital audio track example $\mathbf{e}^{\mathrm{A}}_{\hat{k}_m}$ is a clean version of the noisy input $\mathbf{i}^{\mathrm{A}}_m$. This is one example out of the example set $\{\mathbf{e}_k\}_{k=1}^{N_{\mathrm{E}}}$. As described above (Sec. 1), this set is extracted in increments that cannot be smaller than a single-frame, i.e, temporal lag of $\Delta\tau^{\mathrm{V}}$. This is coarse, relative to the audio dynamics. The audio we seek from $\mathbf{e}_{\hat{k}_m}$ might not be $\mathbf{e}^{\mathrm{A}}_{\hat{k}_m}$, but a slightly shifted track, with a temporal lag smaller than $\Delta\tau^{\mathrm{V}}$. Such a slight temporal shift in the training audio is not resolvable by the set of examples $\mathbf{E}$.

To handle this issue, we perform a finer association of inputs to examples. It proceeds after the cross-modal process described in [1]. However, in the temporally-fine association to examples, only the audio modality is used. Recall from [1] that the data in *audio-example* k is

$$\mathbf{e}^{\mathrm{A}}_k = \left[a_{\mathrm{e}}(s^0_k),\ a_{\mathrm{e}}(s^0_k+1),\ \ldots\ a_{\mathrm{e}}(s^0_k+N_{\mathrm{S}}-1)\right]. \quad (5)$$

Therefore, $\mathbf{e}^{\mathrm{A}}_{\hat{k}_m}$ is created by extracting samples $[s^0_{\hat{k}_m},...,(s^0_{\hat{k}_m}+N_{\mathrm{S}}-1)]$ from a long soundtrack. Similarly, a temporally shifted sequence $\mathbf{e}^{\mathrm{A}}_{\hat{k}_m,\delta}$ can be extracted:

$$\mathbf{e}^{\mathrm{A}}_{\hat{k}_m,\delta} = \left[a_{\mathrm{e}}(s^0_{\hat{k}_m}+\delta),\ \ldots\ a_{\mathrm{e}}(s^0_{\hat{k}_m}+\delta+N_{\mathrm{S}}-1)\right]. \quad (6)$$

Here $\delta \in [-\rho N_{\mathrm{F}},\ldots\rho N_{\mathrm{F}}]$, i.e, the shifts are much subtler than in $\mathbf{E}$. Each of these finely shifted audio examples yields a corresponding feature vector $\tilde{\mathbf{e}}^{\mathrm{A}}_{\hat{k}_m,\delta} = \mathcal{P}(\mathbf{e}^{\mathrm{A}}_{\hat{k}_m,\delta})$. The shift that optimizes the match of the audio example to the input audio is

$$\hat{\delta}_m = \arg\min_{\delta\in[-\rho N_{\mathrm{F}},\ldots\rho N_{\mathrm{F}}]} d_{\mathrm{A}}(\tilde{\mathbf{i}}^{\mathrm{A}}_m, \tilde{\mathbf{e}}^{\mathrm{A}}_{\hat{k}_m,\delta}). \quad (7)$$

Substituting $\hat{\delta}_m$ derived in Eq. (7) for δ in Eq. (6) results in the estimated clear audio segment m, $\mathbf{e}^{\mathrm{A}}_{\hat{k}_m,\hat{\delta}_m}$. This is the denoised version of $\mathbf{i}^{\mathrm{A}}_m$.

The vector $\mathbf{e}^{\mathrm{A}}_{\hat{k}_m,\hat{\delta}_m}$ is simply mapped to segment m of the output audio. Define $h_m = (s^0_{\hat{k}_m}+\hat{\delta}_m) - s^0_m$. Then, create an output soundtrack $\mathbf{o}_m$, whose samples are

$$o_m(s) = \begin{cases} a_{\mathrm{e}}(h_m+s), & s^0_m \le s \le s^0_m+N_{\mathrm{S}}-1 \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

This soundtrack is silent, except for $[s^0_m,...,(s^0_m+N_{\mathrm{S}}-1)]$, which includes the denoised content $\mathbf{e}^{\mathrm{A}}_{\hat{k}_m,\hat{\delta}_m}$. Finally, as we describe in Section 6 of [1], $\mathbf{o}_m$ is used to render the final audio output, by *mosaicing*.

References

[1] D. Segev, Y.Y. Schechner, M. Elad: "Example-based Cross-Modal Denoising" Proc. IEEE CVPR 2012.