Imperial College London

ACOUSTIC BLUR KERNEL WITH SLIDING WINDOW FOR BLIND ESTIMATION OF REVERBERATION TIME

Microsoft Research

Felicia Lim*, Mark R. P. Thomas†, Patrick A. Naylor*, Ivan J. Tashev†
*Dept. of Electrical and Electronic Engineering, Imperial College London, UK
†Microsoft Research, Redmond, USA

Abstract

- A single-channel blind T_{60} estimator is developed here employing spectral analysis in the modulation frequency domain.
- It was previously observed that window lengths used for transformation to the modulation domain crucially affects estimation accuracy.
- This work proposes the use of a sliding window length that is dynamically updated according to the length of the detected decay region.

Acoustic blur kernel

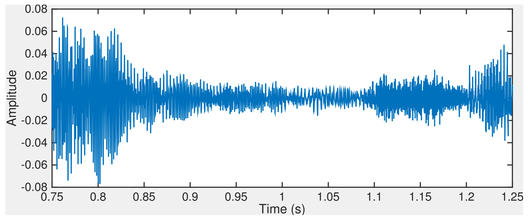

Fig. 1: Decaying segment of a speech signal due to reverberation.

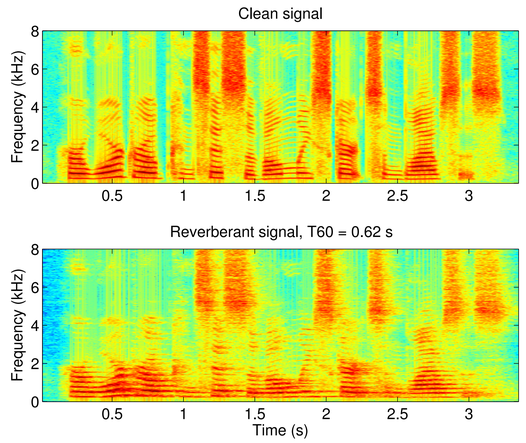

Fig. 2: Spectrograms of clean and reverberant speech signals, $T_{60} = 620$ ms.

- The effect of reverberation is analogous to motion blur in images.
- In image processing, a common method for blur kernel estimation employs spectral analysis to determine its direction and magnitude.
- We can apply the same principle for acoustic blur kernel estimation.
- Using Polack's room impulse response model, the acoustic blur kernel is approximated as a decaying exponential.
- We know that its direction is along the time axis towards $n = +\infty$.
- If its magnitude (governed by its decay rate) can be estimated, the corresponding T_{60} can be derived from it.

Blur Kernel Estimation

Consider a simplified reverberation model based on Polack's RIR model,

$$x[n] = \delta[n] * e^{-\alpha n}, \tag{1}$$

$$\alpha = 3\log(10)/T_{60}. \tag{2}$$

Its STFT magnitude spectrum can be approximated as

$$|X[m,k]| \approx e^{-\alpha p m}, \qquad \begin{matrix} m\text{: frame index} \\ k\text{: frequency bin} \\ p\text{: frame increment.} \end{matrix} \tag{3}$$

Apply a second STFT to examine the behaviour of $|X[m,k]|$ w.r.t. time,

$$|X[m,k]| \xrightarrow{STFT} \tilde{X}[m',k',k]. \qquad \begin{matrix} m'\text{: frame index in mod domain} \\ k'\text{: frequency bin in mod domain} \end{matrix} \tag{4}$$

Next, consider the case $L' \geq T_{60} f_s$ such that the signal decay is captured within the first modulation domain frame $m' = 0$. The expected result is

$$H_\alpha[k'] \triangleq \mathrm{DFT}\{e^{-\alpha p m}\}. \tag{5}$$

The broadband T_{60} can then be estimated by finding its corresponding α as

$$\hat{\alpha} = \arg\min_{\alpha} \left\{ \frac{1}{L'} \sum_{k'=0}^{L'-1} \left| H_\alpha[k'] - \tilde{X}[m',k'] \right|^2 \right\}, \tag{6}$$

and the $\hat{T}_{60}$ derived according to (2).

Variable Window Length

For accurate estimation of T_{60} from speech signals, long L and L' are desired for high T_{60}, while short L and L' are desired for low T_{60}.

We propose a dynamically updating variable window length.

1. Divide the time-domain signal into small frames.
2. Find sets of consecutive frames that contain speech decay by comparing the frame-by-frame statistics of the signal (max, min, var).
3. For each set of decaying frames, find the corresponding modulation domain frame to estimate α and its corresponding T_{60} using (6).
4. Finally, average the estimated T_{60} values across all frames.
5. Since the number of time-domain frames used in Step 2 is flexible, the final length of the modulation domain window is similarly flexible.

Experimental Results

Evaluation was carried out against 3 algorithms from the literature: 1) ML [Löllmann2010], 2) SRMRinv [Falk2010], 3) Blur Kernel [Lim2015].

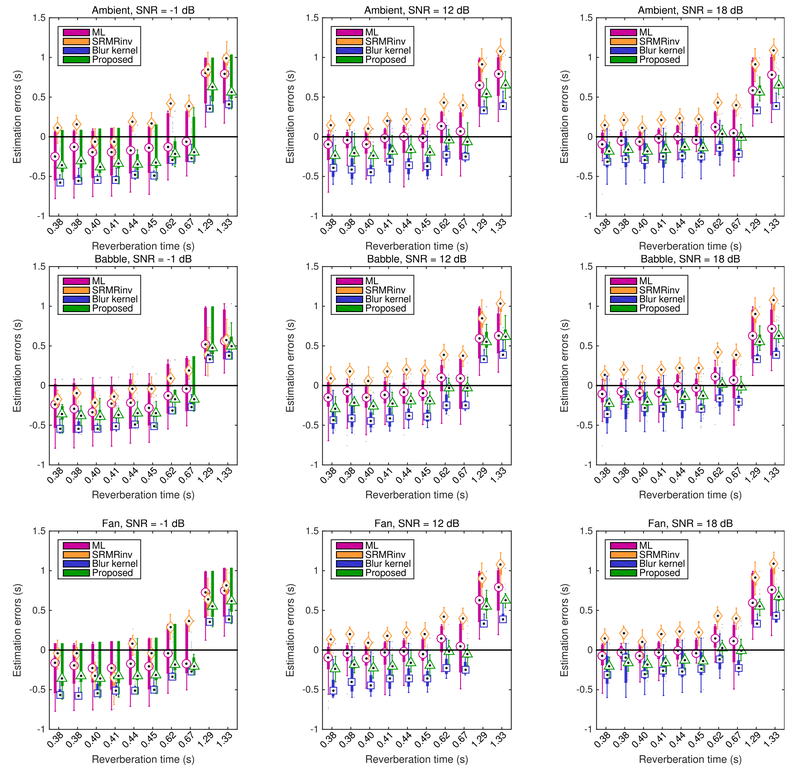

Fig. 3: ACE Challenge dataset

Algorithm	Averaged RTF
ML	0.02
SRMRinv	0.32
Blur Kernel	6.66
Blur Kernel with Sliding Window	0.23

Table 1: Averaged real time factors.

Conclusion

- Improved median estimates over Blur Kernel for $T_{60} \lesssim 700$ ms.
- At SNR > 12 dB, improved accuracy over SRMRinv for $T_{60} > 600$ ms and comparable performance at lower T_{60}.
- Smaller variances compared to ML with improved median estimates for $T_{60} \gtrsim 600$ ms but slightly worse median estimates for $T_{60} \lesssim 600$ ms.
- Low computational complexity.