CLICK REMOVAL IN DEGRADED AUDIO

Report for Module EEP55C22 Computational Methods

Darius I. Laherty, Student ID 123456
Trinity College Dublin
dariusl@tcd.ie

October 2022

This report is submitted in part fulfilment for the assessment required in EEP55C22 Computational Methods. I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year. These are found in Parts II and III at http://www.tcd.ie/calendar.

This report describes the algorithm designed for detection and removal of clicks in archived audio tracks. An example of a few clicks is shown in Figure XX. The sound manifests as a short sharp click or thump in the audio track. These degradation arise because of XXXX. Our problem is to XXXX.

1 Background

Let us define the observed degraded signal as G_k and the original, clean signal as y_k. We can model the degradation as follows.

$$G_k = \begin{cases} ccr_k & \text{when } b_k == 0 \\ y_k & \text{Otherwise} \end{cases} \tag{1}$$

where r_k is a random corruption ???? (could be something else that you decide on), and b_k is an indicator binary variable that is 1 at sites that are corrupted. Therefore the observed signal is created by corrupting specific sites indicated by $b_k == 1$, hence clicks.

Our problem is to XXXXX.

1.1 Detection

Following the work of XXXX [1], we assume that YYYY. Hence we generate the AR residual ZZZ in order to YYY etc etc

1.2 Interpolation

Having discovered b_k we now need to XXX. Again following our lectures we can use the AR model to interpolate the missing samples. Given $\mathbf{a}$ is our vector of AR coefficients and XXX and YYY.

2 The algorithm

The mathematical framework above only applies if the signal is statistically homogeneous. real signals are not, so we insted process the signal in blocks. Our algorithm is then as below.

1. For each block we process the inner XXXX samples
2. XXXModel the block using the normal equations with model order XXXXX
3. Detect ....
4. etc

3 Experiments and results

To test our algorithm, first an artificial signal was generated using XXX. Then the signal was artificially degraded .... by .... Figure **??** shows the ROC detection performance for different levels of degradation on this signal. As can be seen etc etc On the right we see MSE interpolation performance. defined as ....

A clean signal was then degraded using XXXXX. etc etc. Figure XXX shows ROC results of ... interpolation performance etc etc.

To test on a real signal the XXX signal was processed with XXX block size, model order P etc etc . An example of performance is shown in fig XXX. The cleaned up signal is uploaded in Bb for demonstration purposes.

4 Conclusions

This thing actually worked. But it was really hard to XX and YY. I wish we could have done A B C first. ts still bad here and here .. maybe we could try ZZZ instead to get better.

A Making good plots

There is an art to rendering good plots for your document regardless of the document processor being used. Remember to make all lines and text visible at the scale it will be rendered. That often means changing the size of fonts on axes and line thicknesses in plots. Beware of whitespace when rendering images. That causes your plot to occupy less area than you think it should. In Matlab for instance, when rendering an image for inclusion in a document. The following is recommended.

```
% Render an image in figure 1 (for instance)
figure(1);
image(pic);
% Make the picture occupy all the real estate in the figure window
axis off;
set(gca, 'position', [0.01 0.01 0.98 0.98]);
% Render the picture to an .eps file
print -depsc mypic.eps
```

Here is an example of some mathematics.

$$x_1 = \sum_{n=-4}^{N-1} \theta_n + 2\pi x_n - \int_{t=-\infty}^{\infty} \epsilon(t) dt \tag{2}$$

Equation 2 shows how a person can overcome his fear of marking.

Bibliography

[1] BOYKOV, Y., VEKSLER, O., AND ZABIH, R. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 11 (Nov 2001), 1222–1239.