Detecting proximity from personal audio recordings

Dan Ellis, Hiroyuki Satoh, Zhuo Chen
LabROSA, Columbia Univ., NY USA
ICSI, Berkeley, CA, USA
Morikawa lab, University of Tokyo, Tokyo, Japan

dpwe@ee.columbia.edu  http://labrosa.ee.columbia.edu/

1. Detecting Proximity
2. Audio Similarity: Cross Correlation
3. Audio Similarity: Fingerprints
4. Evaluation & Conclusions
I. Detecting Proximity

• Easy for smartphones to “listen” to ambient audio
  • what can they do with the information?

• Ubiquitous Smartphones
  • opportunities from having everyone’s phones connected via the cloud?
Detecting Proximity

- **Application:** Who did I speak with?

- **Approaches:**
  - High-resolution indoor GPS walls?
  - Local wireless (NFC, Bluetooth)
    - what is the right range?
  - Ambient audio similarity
    - all phones have microphones
    - “radius” depends on noisiness
      - matches practical conversation radius
Data: Poster Sessions

- **Simultaneous recordings by multiple subjects in a real “poster session”**
  - two attempts: SANE 2013, NEMISIG 2014

- **Live subjects wore Red Hats**
  - warning others
  - for tracking in video

- **Final data set**
  - six subjects
  - 30 mins with at least 5 of 6
2. Audio Similarity: Cross Correlation

- Are two audio signals "proximal"?
  - recorded at slightly different places
  - .. different orientations, etc

- Expect differences in detail, but shared core

\[
M_A(e^{j\omega}) = H_A(e^{j\omega})C(e^{j\omega}) + N_A(e^{j\omega})
\]
\[
M_B(e^{j\omega}) = H_B(e^{j\omega})C(e^{j\omega}) + N_B(e^{j\omega})
\]

- Cross-correlation reveals common part

\[
S_{MA MB} = M_A M_B^* \\
= H_A H_B^* |C|^2 \\
+ H_A C N_B^* + H_B^* C^* N_A + N_A N_B^*
\]
Short-Time Cross Correlation

- Calculate cross-correlation between corresponding short windows
  - e.g. 2 s windows every 1 s

- Find peak in time domain correlation
  - lag at peak = best local time alignment
  - value at peak (normalized by energies)
    = degree of similarity between signals

- Plot best lag as vs. window time
  - threshold peak value to ignore chance correlation
• Compiled MATLAB application to calculate & plot short-time cross correlation of long-duration signals
  • raw cross-correlation plotted in grayscale
  • export peak lag times & values

http://labrosa.ee.columbia.edu/projects/skewview/
3. Audio Similarity: Fingerprints

- **Landmark-based audio fingerprinting:**
  - Represent audio as “constellation” of energy peaks
  - Index nearby pairs of peaks for rapid search
  - Match as multiple peaks in same relative positions

- **Robust to...**
  - channels - peak level not used
  - noise - only a few peaks need to match

- **Fast search** in large archives (Shazam)
**audfprint**

- Open source **audio fingerprinting** tool
- Matlab:
  
  http://labrosa.ee.columbia.edu/matlab/audfprint/

- Python:
  
  https://github.com/dpwe/audfprint

- Rapid retrieval of short noisy queries within large databases
  
  - 10 sec over-the-air queries within 100k+ reference items in ~1 s
4. Results

• Mutual proximity between all six channels:

  Cross-correlation

  Fingerprints

• Various “proximal episodes” between targets visible (dark)

• Good agreement between two methods
Evaluation

- **Ground truth?**
  - did not hand-mark video...

- **Cross-correlation is quite reliable ...**
  - use it as reference for fingerprinting

- **DET curve for thresholded proximity**
  - fprint vs. xcorr

- **Execution time [for 6 x 30 min tracks]:**
  - Cross-corr: \( \sim (0.006 \times t_{\text{dur}}) \times N^2 \) [427 s]
  - Fingerprint: \( \sim (0.030 \times t_{\text{dur}}) \times N \) [317 s, linear]
Conclusions

• Similarity between ambient audio e.g. from smartphone mics can be used to track **personal proximity**

• Similarity can be measured by:
  • cross-correlation (accurate but expensive)
  • landmark fingerprinting (fast, but adequate?)

• Experiments showed both approaches gave very similar results
  • fingerprinting suitable for **scaling** to very large datasets, e.g. across many users