Violin Audio Detection

Allan Tokuda
CS 395 - Machine Interpretation of Music
2006 March 10

Project Goal

Goal

To detect whether an input sound represents the violin.

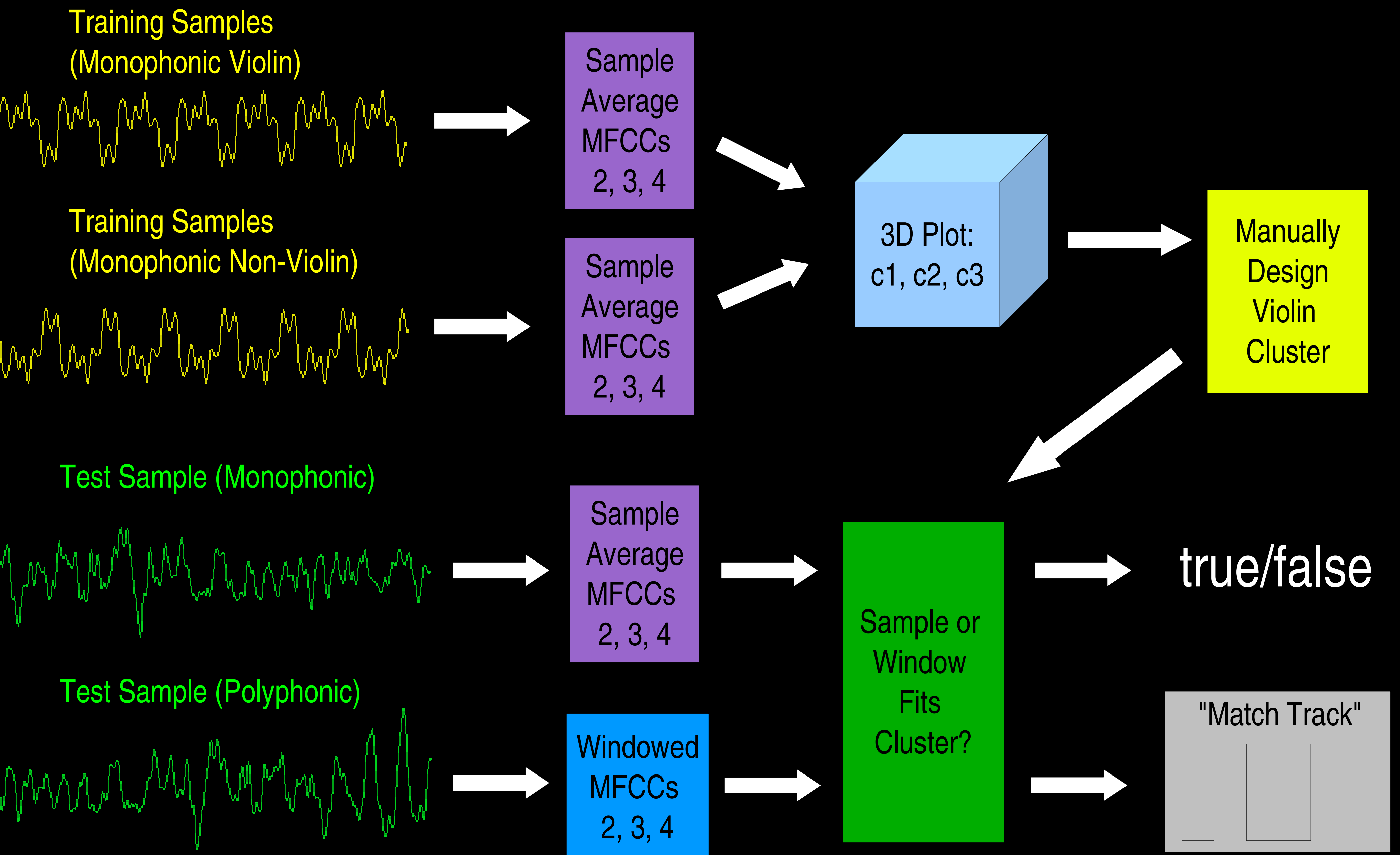
Rationale

- Step on the road to generalized instrument detection.
- Step on the road to polyphonic audio transcription.
- Excuse to learn how MFCCs work having not studied DSP.
- Pretty darn cool IMHO.

Method Overview

- Calculate mean MFCCs of violin and non-violin samples
- Generate 3D Plot of MFCCs 2 vs. 3 vs. 4 for all samples, coloring the violins differently as visual aid
- Manually design violin cluster space by defining planes
- For test samples, check that their mean MFCCs fall within the cluster (on the desired side of all the planes).
- Reuse cluster definition to indicate when violin is playing in polyphonic recordings.

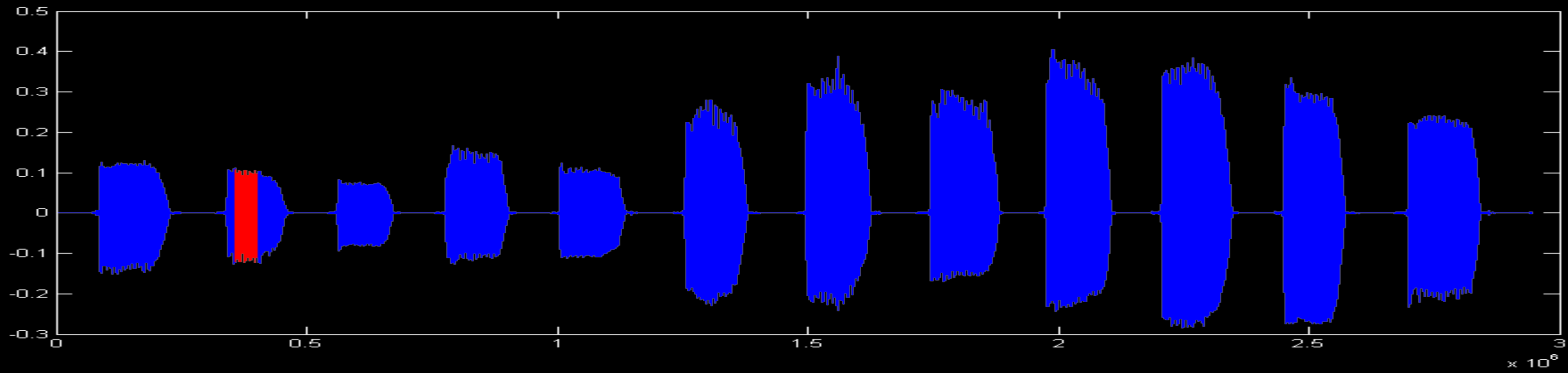
Method Diagram



Corpus Construction

- 32 samples of myself playing various notes on Violin:
  - four strings
  - three positions on each string
  - three bowing methods: slow/heavy, average, fast/light
- 255 samples of 18 instruments from U. of Iowa's website

samples were very long in time; picked random non-silent parts



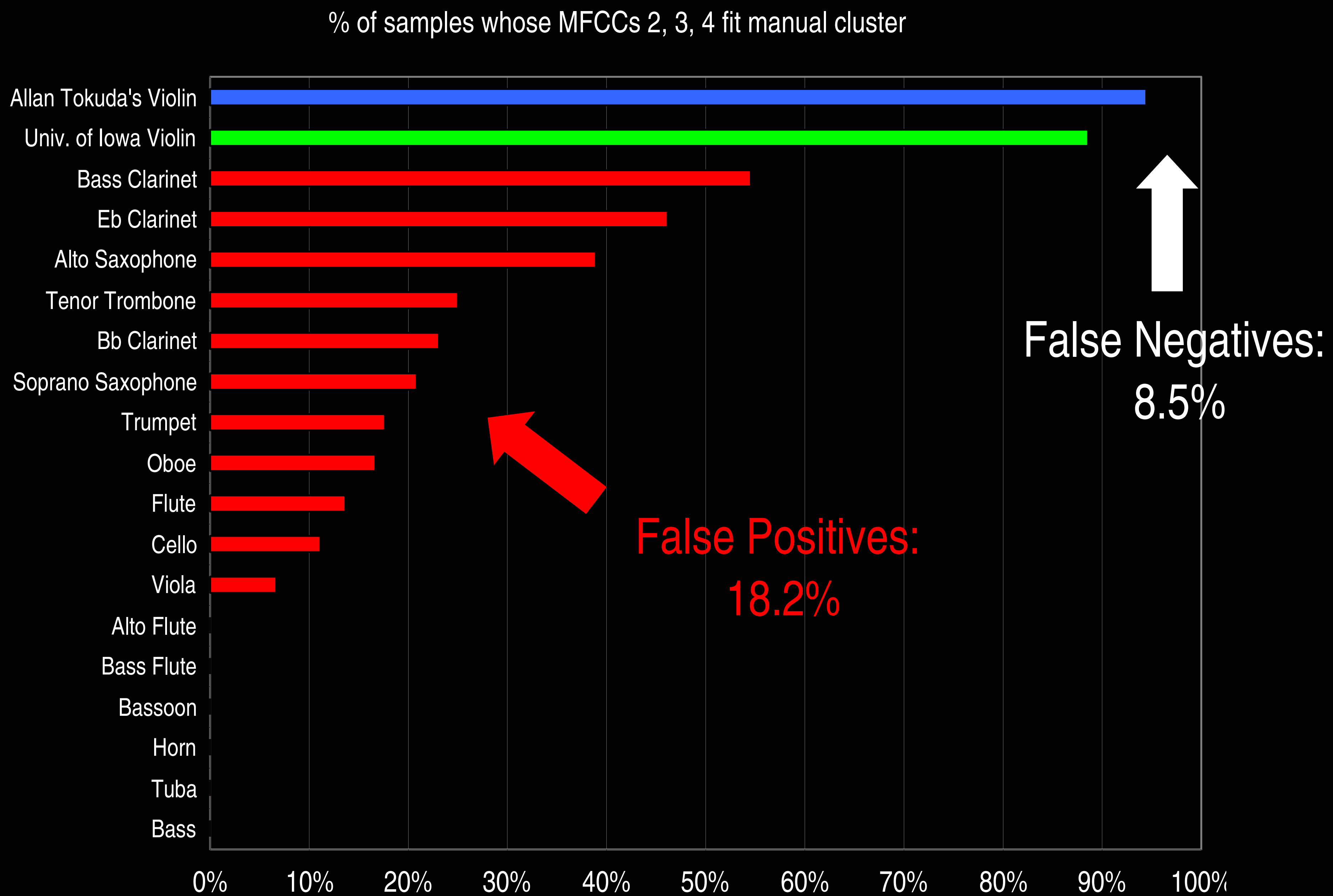
Reasons for Chosen Method

- I had faith in Bryan Pardo who spoke highly of the relevance of MFCCs to my application
- I found MFCCs 2, 3, and 4 were the only interesting ones:
  - MFCC 1 appeared totally random for Univ. of Iowa's data
  - MFCCs 5 and higher appeared to be convoluted by pitch
- I looked at the MFCCs and said "hey look at that, a cluster!"
- I had no faith in hill-climbing, and I didn't get to implementing K-means clustering; did manually for now

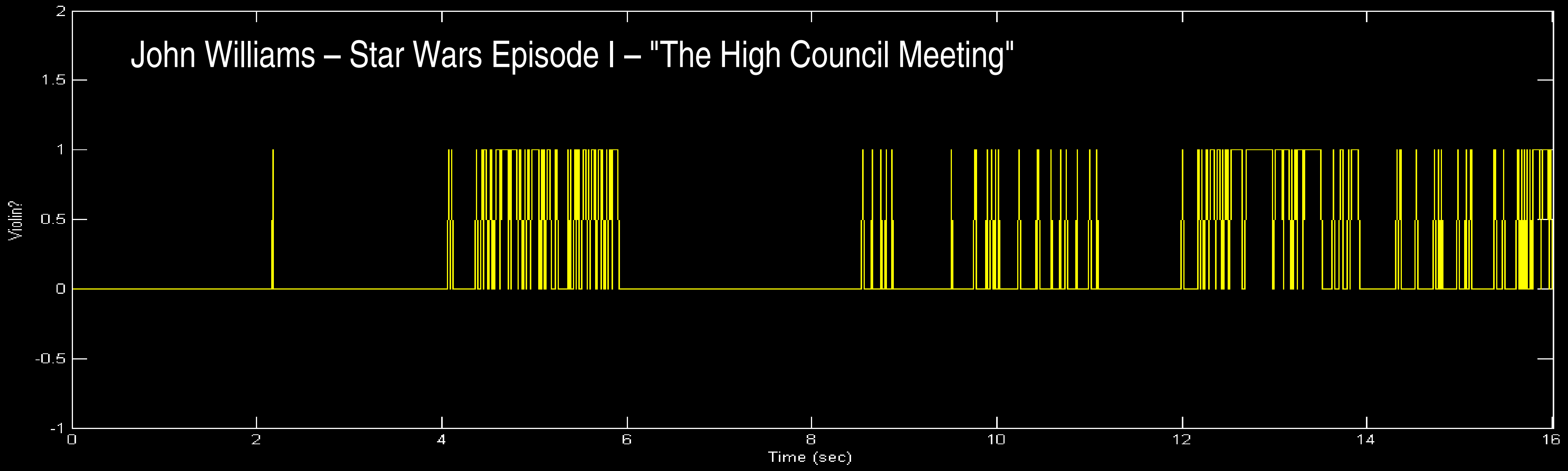
Measuring Success

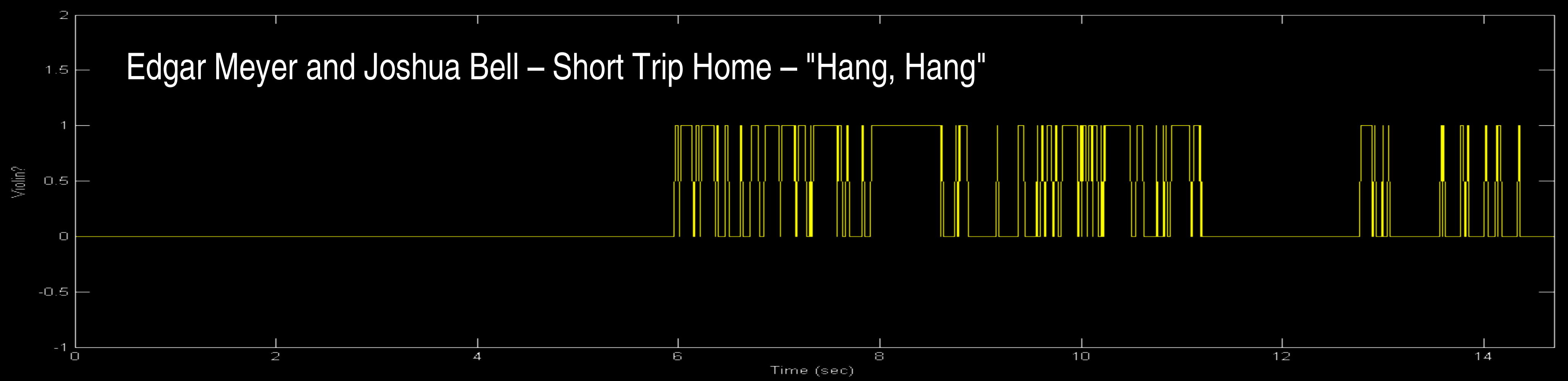
- Ideal system
  - monophonic audio: whole recordings classified as matches if and only if they are recordings of violins.
  - polyphonic audio: time windows of recordings classified as matches if and only if they take place when violins are playing.
- Success measures: classification of monophonic samples
  - % false positives (incorrectly classified as violins)
  - % false negatives (incorrectly classified as non-violins)

Results: Monophonic Input



Subjective Results: Polyphonic Input





Future Improvements

- Combine with pitch tracker: eliminate mistaking a violin and a bass clarinet playing below the range of a violin.
- Devise measurement system allowing the use of MFCCs 5 and 6 without penalizing high-pitch notes
- Add non-orchestral instruments and even non-instrumental sounds to corpus (piano, guitar, voice, synth)
- Implement K-means clustering to automate the training process and more easily generalize to other instruments

References

- The University of Iowa Electronic Music Studios, "Musical Instrument Samples", http://theremin.music.uiowa.edu/MIS.html, accessed Mar 2005.
- Ellis, D., "Matlab Audio Processing Examples", http://www.ee.columbia.edu/~dpwe/resources/matlab/, accessed Mar 2005.
- Chordia, Parag. "Segmentation and Recognition of Tabla Strokes." in Proc. of The 6th International Conference on Music Information Retrieval (ISMIR 2005), pp. 107-114, London, 2005.
- Mesaros, A., Astola, J., "The Mel-Frequency Cepstral Coefficients in the Context of Singer Identification.", in Proc. of 6th International Conference on Music Information Retrieval (ISMIR 2005), pp. 610-613, London, 2005.