

# Audio Event Detection for Automatic Scene Recognition

Xun Xu

Department of Computer Science and Engineering  
Shanghai Jiao Tong University

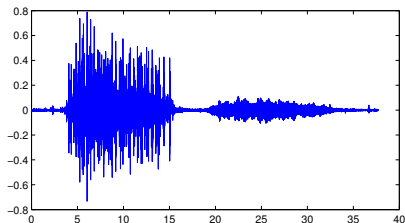
June 22, 2015

- 1 Introduction
- 2 Audio Event Detection
- 3 Scene Recognition
- 4 Evaluation
- 5 Demo

- 1 Introduction
  - Problem Description
  - Approach
- 2 Audio Event Detection
- 3 Scene Recognition
- 4 Evaluation
- 5 Demo

# Problem Description

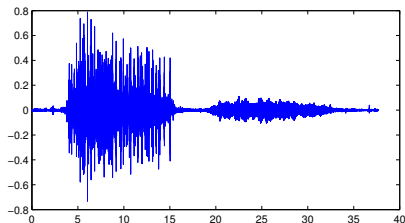
In this project, our problem is to recognize a scene where an audio is recorded.



*concert*

# Our Approach

Our approach is to detect the audible events in a clip.  
Then infer the scene from the detected events.



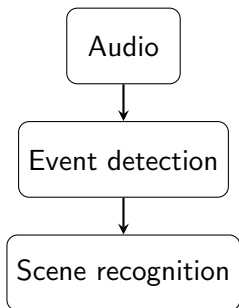
*applause,  
instrument*



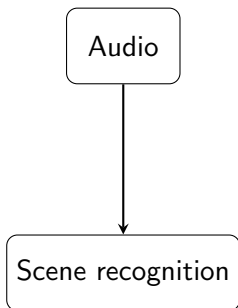
*concert*

# Our Approach vs. Other Approaches

Our approach:



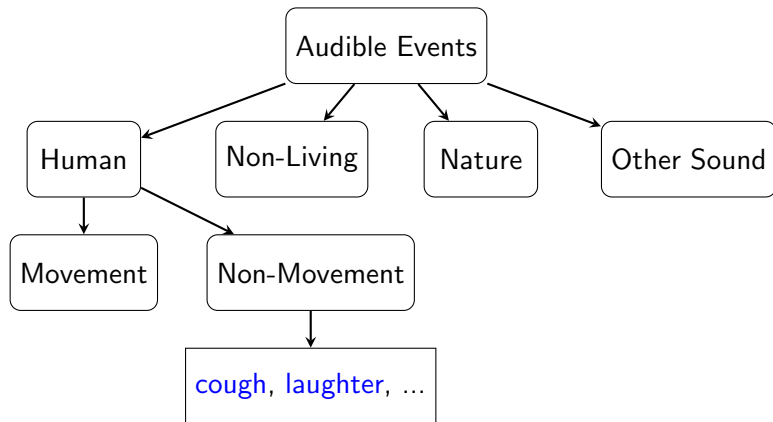
Other approaches:



- 1 Introduction
- 2 Audio Event Detection
  - Audible Event Taxonomy
  - Audio Data
  - Preprocess and Feature Extraction
  - Event Model
- 3 Scene Recognition
- 4 Evaluation
- 5 Demo

# Audible Event Taxonomy

We labelled common audible events into 4 classes.  
There are 120 events in total.





We download the audio data for events from Sound Search Engines (SSEs).  
For example, when we query “cough” in SSE:

	<b>cough9.aiff</b> cough environmental-sounds-research cough	★★★★★	<b>Fratz</b> May 24th, 2006 803 downloads 0 comments   
◆ 13 more results in the same pack "coughs"			
	<b>double_cough_01.wav</b> A man coughing twice in a row. ill coughing cough cold throat hack sick flu clear sickness foley	★★★★★	<b>Joedeshon</b> March 6th, 2015 526 downloads 1 comment   
	<b>Cough (2)</b> A standard cough. Could be used during an awkward silence in a concert hall. hack sick splutter cough clean	★★★★★	<b>OwlStorm</b> April 10th, 2012 471 downloads 1 comment   

We download clips from 1 second to 60 seconds.

# Preprocess and Feature Extraction

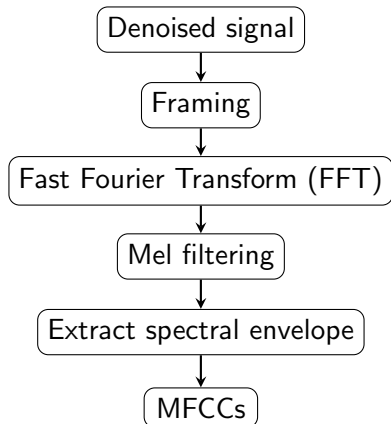
We first use Minimum Statistics to calculate the noise spectrum and subtract it from the input signal.

Then we extract Mel-Frequency Cepstral Coefficients (MFCCs) from denoised signal.

# Preprocess and Feature Extraction

We first use Minimum Statistics to calculate the noise spectrum and subtract it from the input signal.

Then we extract Mel-Frequency Cepstral Coefficients (MFCCs) from denoised signal.



# Preprocess and Feature Extraction

We first use Minimum Statistics to calculate the noise spectrum and subtract it from the input signal.

Then we extract Mel-Frequency Cepstral Coefficients (MFCCs) from denoised signal.

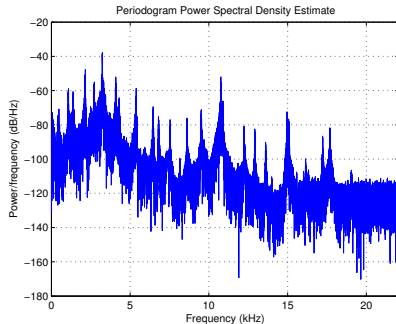
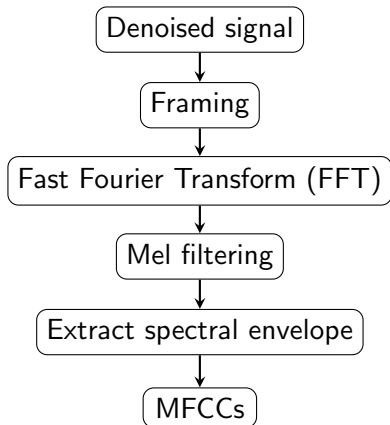


Figure: Audio in frequency domain

We use features to train Gaussian Mixture Models (GMMs).  
The training is done by Expectation-Maximization (EM) algorithm.

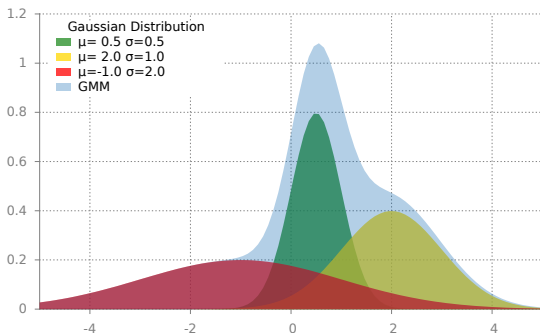


Figure: A GMM with three components

- 1 Introduction
- 2 Audio Event Detection
- 3 Scene Recognition
  - Scene Extraction
  - Scene-Event Relation Mining
  - Audio Segmentation
  - Scene Inference
- 4 Evaluation
- 5 Demo

# Scene Extraction

We use the scripts for movies, plays and TV series to extract the scenes.

Below is a script example. We call it a *context*, including a *scene-switching sentence*, and its content.

INT. LEONARD'S BATHROOM - Night

Leonard turns on the light, revealing a shower, toilet and sink.

He removes toiletries from the grocery bag and places them inside.

We use Natural Language Process (NLP) tools to process the *scene-switching sentence*, and eliminate the following type of words:

- Person names
- Time indicator
- Location names
- Adjective, determiner, number, ...

Table: Top 10 Occurred Scenes

Scene	Occurrence
house	3537
office	3259
apartment	2919
room	2580
bedroom	2257
car	1699
street	1622
kitchen	1431
living room	1374
studio	1250



# Scene-Event Relation Mining

To get the relation between scenes and audible events, we match the context in a script with our predefined audible events.

INT. LEONARD'S BATHROOM - Night

Leonard turns on the light, revealing a shower, toilet and sink.

He removes toiletries from the grocery bag and places them inside.

Based on the idea of Term-Frequency-Inverse Document Frequency (TFIDF), we calculate two scores of an event  $e$ , to a scene  $s$ .

①  $TF = \log(1 + f(e, s))$

$f(e, s)$  is the number of contexts  $e$  appears in all contexts under scene  $s$ .

②  $IDF = 1 + \log(\frac{N}{N_e})$

$N$  is the number of scenes.  $N_e$  is the number of scenes in which event  $e$  appears.

These two scores are then multiplied, and used as the importance of an event to a scene.

$$TFIDF = TF \times IDF \quad (1)$$

Table: An example of scene-event map

Scene	Top 10 events ranked by TF-IDf
bathroom	running+water, toilet, faucet, toothbrush, shower, drawer, drain, talk, paper, bowl
beach	seagull, sand, boat, talk, wave, sea, car, laughter, drink, wood, running
forest	tree, wood, dirt, talk, running, bird, river, car, leaf, grass, wind
kitchen	drawer, cutlery, microwave, dish, kettle, talk, bowl, phone, toaster, running+water
street	car, truck, subway, talk, traffic, engine, siren, phone, running, laughter

# Audio Segmentation

Scene-Event map is used when we have detected the events.  
We need to cut testing clips into segments for event detection.

# Audio Segmentation

Scene-Event map is used when we have detected the events.  
We need to cut testing clips into segments for event detection.

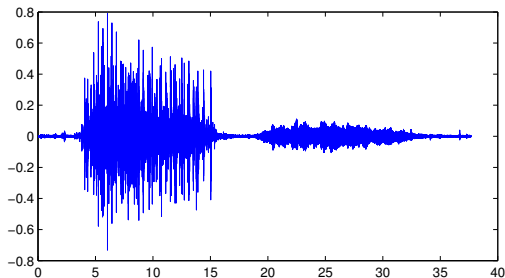


Figure: A example audio clip

We use frame energy and frequency to filter out silence and noise.

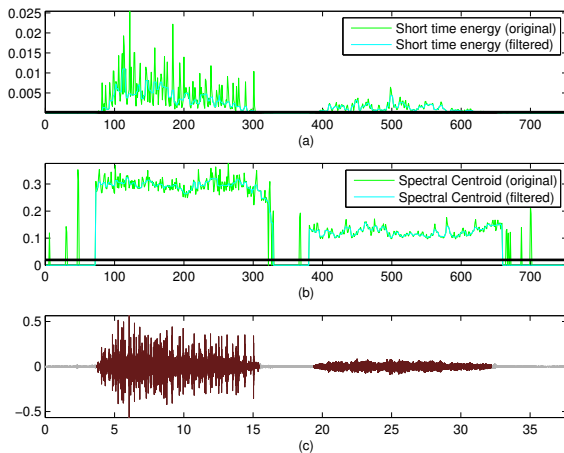
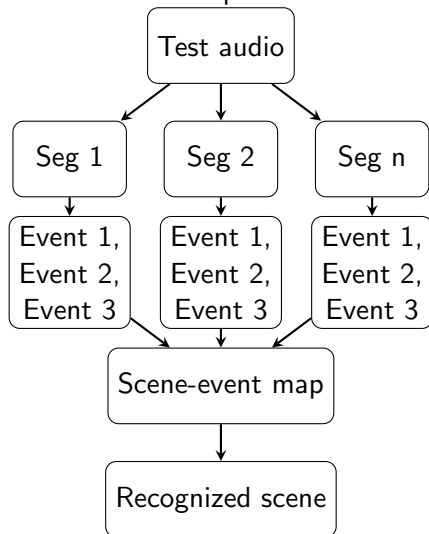


Figure: A segmentation example

# Scene Inference

For each segment, we evaluate it with our trained GMMs. We choose the top three detected events for scene voting.



- 1 Introduction
- 2 Audio Event Detection
- 3 Scene Recognition
- 4 Evaluation**
  - Event Detection Evaluation
  - Scene Recognition Evaluation
- 5 Demo



# Component Number Evaluation

Gaussian Mixture Model distribution:

$$P(\mathbf{x}|\pi, \mu, \Sigma) = \sum_{k=1}^M \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k), \quad (2)$$

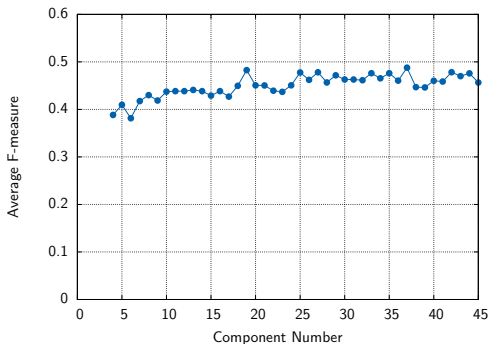


Figure: F-measure for different component number

# Component Number Evaluation

After comparing F-measure and running time, we choose 18 as our component number.

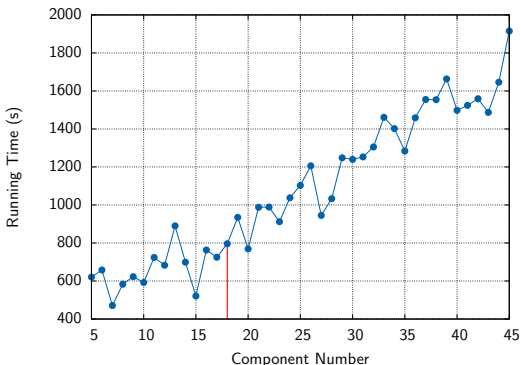


Figure: Running time for different component number

# Event Detection Evaluation

A event detection task of 16 events.

The other six systems are chosen from IEEE AASP Challenge.

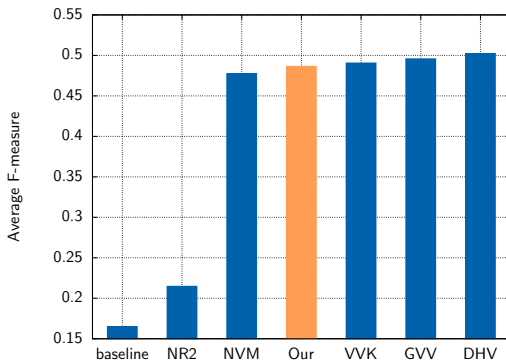


Figure: Event Detection F-Measure

# Scene Recognition Evaluation

In scene recognition, we choose 10 scenes, each scene has 10 clips. Accuracy for other 4 systems are calculated using 5-fold cross validation. Our system achieve an accuracy of 57%.

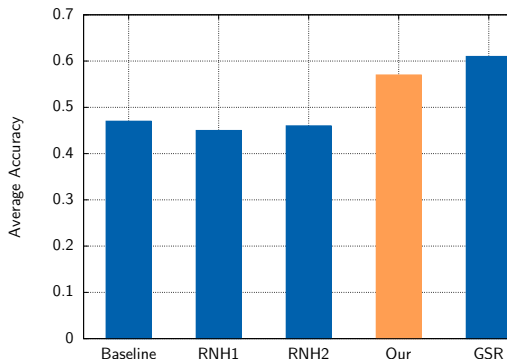


Figure: Recognition accuracy for 10 audio scenes

# Scene Recognition Evaluation

Detailed result of our system with the best system *GSR*.

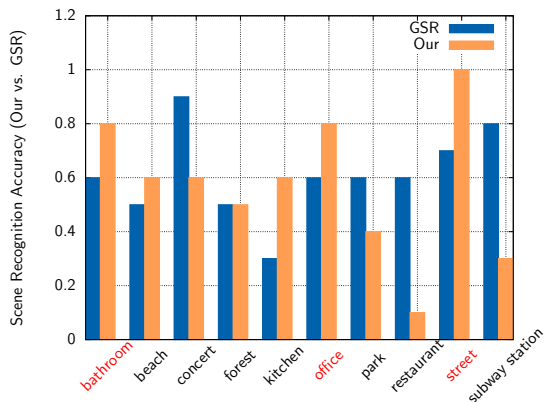


Figure: 10 scenes comparison

- 1 Introduction
- 2 Audio Event Detection
- 3 Scene Recognition
- 4 Evaluation
- 5 Demo

Live demo for our system.

# Acknowledgement

I would like to thank my advisor Kenny Q. Zhu, and my friend Xinyu Hua for their help in this project.



*Thank you!*

*Any Question?*