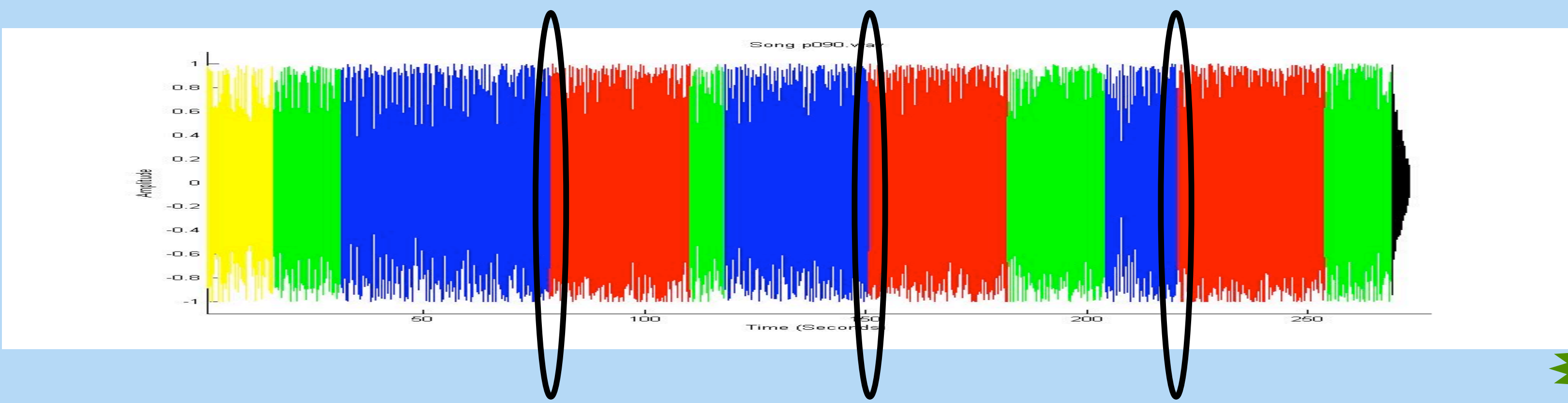
Music has structure

Consider the **structure** of a pop/rock song:

- Musical Segments
  - Introduction, Bridges, Verses, Choruses, Outro
- Musical Boundary
  - between two musical segments
  - e.g., the end of a verse and the beginning of a chorus



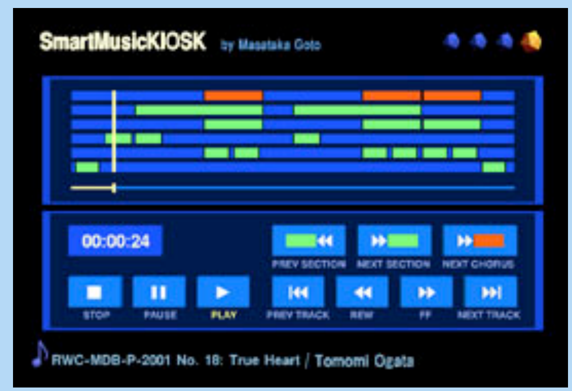
Automatic Music Segmentation

Our goal is to automatically segment music:

1. Represent a song as a digital signal
2. Extract **useful features** from that signal
3. Automatically detect '**musical boundaries**'

Purpose

1. More efficiently **scan through songs**
   - Smart Music Kiosk [Goto06] →



2. Generate better **musical thumbnails**
3. Develop novel **music information retrieval** applications.
   - Temporal dynamics of music
   - Structural comparison

Related Work

Two Approaches:

1. **Self-Similarity:** identify similar audio content within a song
   - Traditional Approach
   - **Unsupervised Approach**
     1. **Cluster** short-term features of a song [Abdallah 06, Lu 04]
     2. Find **repetitions** within a song [Goto 06, Foote 02,…]

2. **Edge Detection:** find '**changes**' in the audio content
   - **Difference Features** - designed to reflect acoustic changes
   - **Supervised Approach**
     - learn a model for musical boundaries using human segmentations
     - User defines 'boundary' through the training data

Outline

- **Difference Features**
- Supervised Music Boundary Detection
  - **Feature Generation**
  - **Boosted Decision Stumps**
- Concluding Remarks

Outline

- Difference Features
- Supervised Music Boundary Detection
  - Feature Generation
  - Boosted Decision Stumps
- Concluding Remarks

Difference Features

Auditory cues indicate the end of one segment and the beginning of the next segment.

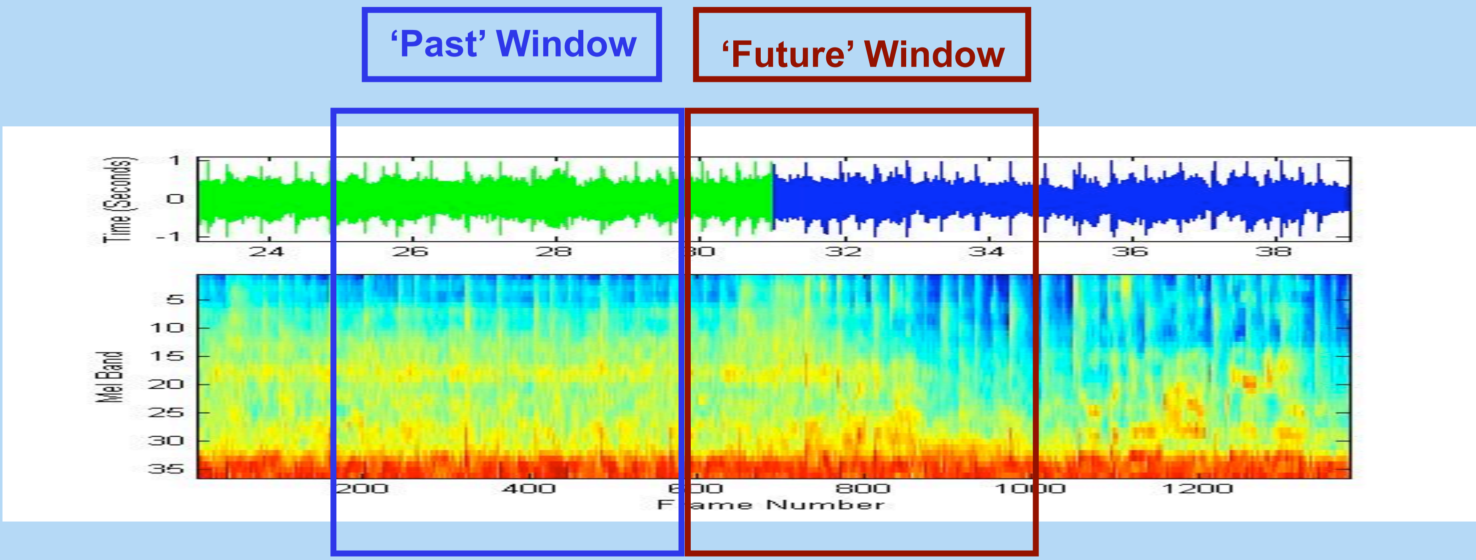
These cues are related to high-level musical notions:

1. Timbre - new instrumentation
2. Harmony - key change
3. Melody - decreased intensity in the singers voice
4. Rhythm - drum fill

We will design 'difference features' that attempt to model these cues.

Difference Features

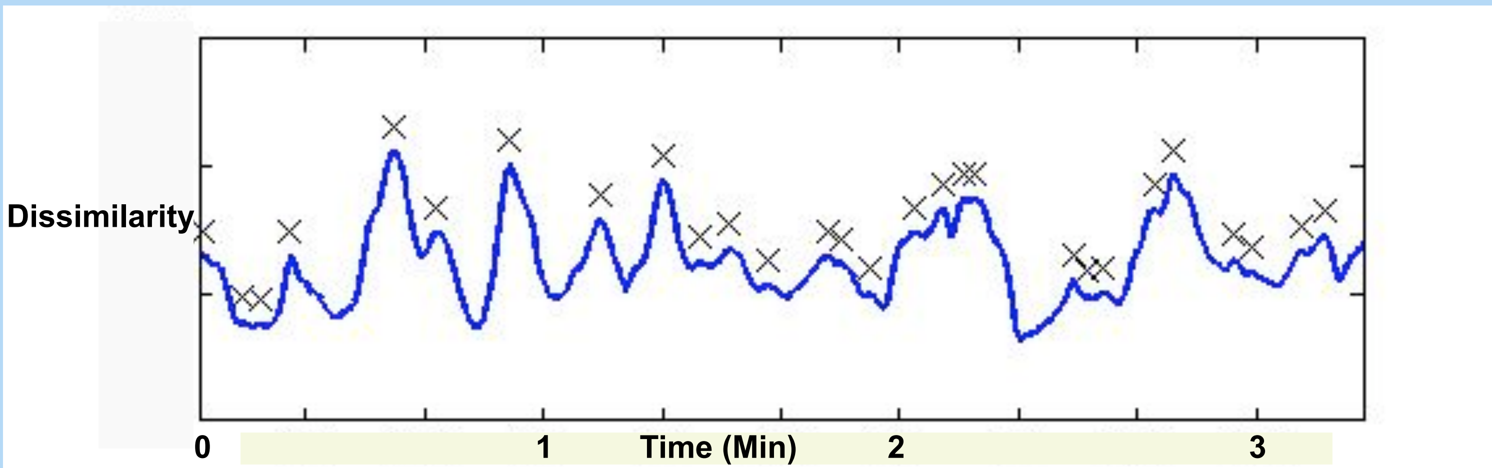
1. Using two adjacent windows
   - ‘Past’ & ‘Future’ Windows
2. Calculate a feature within each window
   - **Scalar:** RMS or BPM
   - **Matrix:** Fluctuation Pattern
   - **Time Series:** MFCC or Chroma
3. Calculate **dissimilarity** between features in each window
   - **Scalar, Matrix:** → Euclidean distance
   - **Time series:** → KL divergence between distributions of samples



Difference Features

Slide adjacent windows over a song to generate a time series of difference features

- Hop Size = resolution of boundary detection
- Peaks represent changes in the audio content



Outline

- **Difference Features**
- Supervised Music Boundary Detection
  - Feature Generation
  - Boosted Decision Stumps
- Concluding Remarks

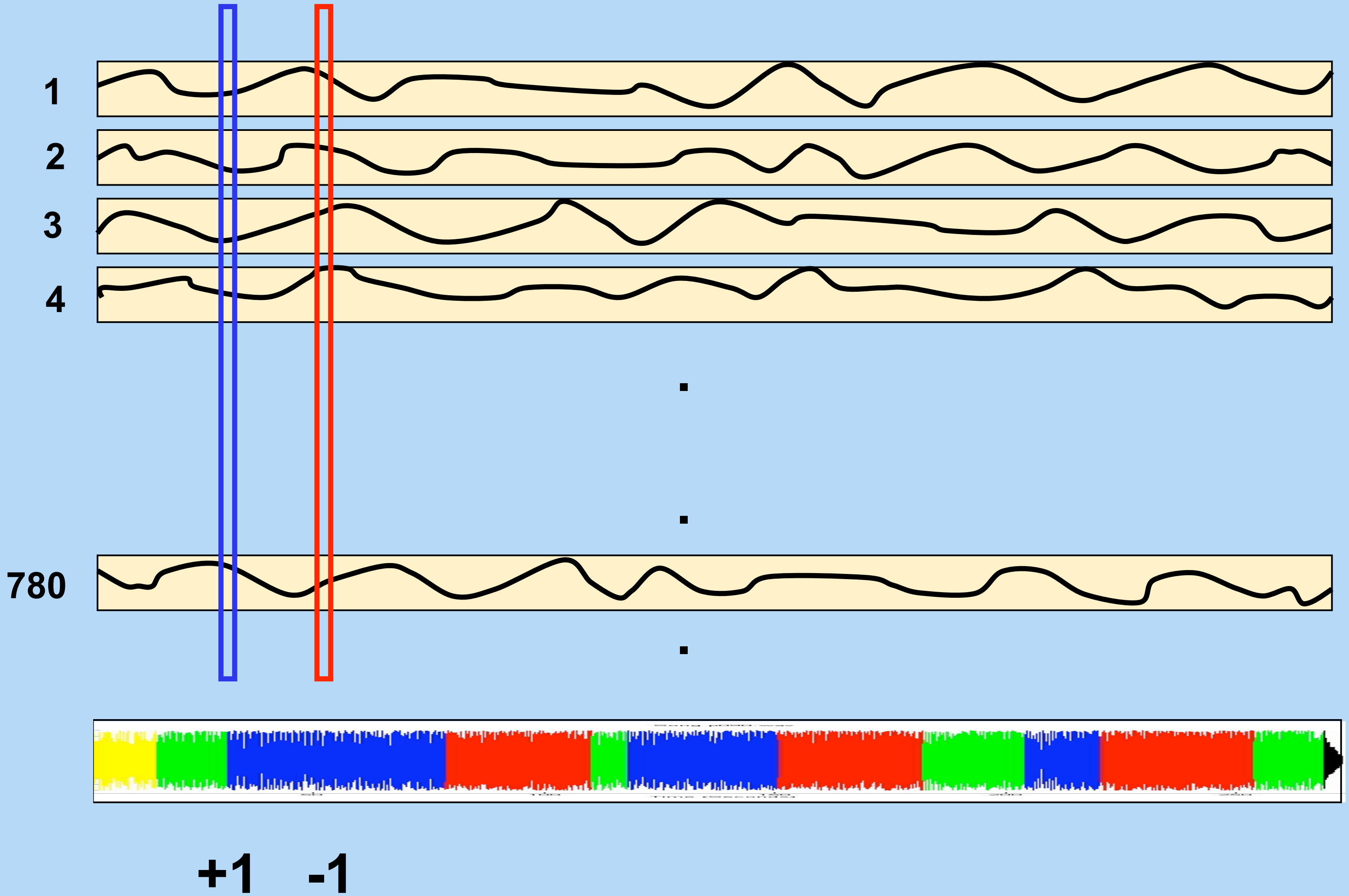
Feature Generation

1. Start with 52 time series
   - 37 difference features, 15 additional features
   - Each feature is a time series with a sampling rate of 0.1 seconds
2. Normalize each time series
   - Time series now is now mean = 0, and variance = 1
3. Generate multiple smoothed versions
   - 1.6 sec, 6.4 sec, 25.6 sec Gaussian kernels.
4. Calculate 1st and 2nd derivatives
   - include absolute value of derivatives

The result is a set of 780 time series

- 52 features x 3 smoothings x 5 derivatives
- Each sample is represented by a 780-dimensional feature vector

Training Data



Supervised Framework

We extract hundreds of 780-dimensional feature vectors per song for 100 human-segmented songs.

- 600 vectors per minute of audio content
- Positive Examples: 7-15 vectors per song will be labeled as 'boundary' vectors
- Negative Examples: randomly picked vectors far from boundaries

We learn a boosted decision stump (BDS) classifier.

- Popular classifier for object boundary detection in images [Dollar et al. 06, Viola & Jones 02].
- Powerful **discriminative** classifier based on Adaboost algorithm [Freund 95]
- Useful for efficient and effective **feature selection**

Boosted Decision Stump (BDS) Classifier

Decision Stump: a simple classifier that predicts one class if the value of an individual feature is above a threshold.

Boosted Decision Stumps: an ensemble of decision stumps and associated weights.

- **Learning**: The decision stump (feature and threshold) that reduces the training set error is added to the growing ensemble at each iteration of the algorithm.
- The weight given to the decision stump is determined by the boosting algorithm (e.g., AdaBoost).
- **Inference**: The prediction is based on a weighted ‘voted’ from the ensemble.

Feature Selection: The order in which features are added to the ensemble can be interpreted as ranking of features.

Evaluation - Hit Rate

Output of BDS Classifier is a time series of scores

- Score reflects confidence of sample being a ‘boundary’
- Boundary Estimates: smooth time series, pick the 10 highest peaks

Hit Rate: estimate is within a half second of a true boundary

- Precision: % of estimates hit a true boundary
- Recall: % boundaries are hit by estimates
- F: harmonic average

Framework	Classifier	Precision	Recall	F
Baseline	Uniform Placement	0.04	0.05	0.04
Unsupervised	Peak Picking (MFCC-Diff)	0.26	0.36	0.30
Supervised	BDS	0.33	0.46	0.38

Evaluation - Directional Hamming Distance

Directional Hamming Distance (F-score):

- 'goodness of overlap' between two segmentations
- Between 0 and 1, 1 being perfect

Framework	Classifier	DHD-F
Baseline	Uniform Placement	0.71
Unsupervised	Peak Picking (MFCC-Diff)	0.80
Supervised	BDS	0.82

- Rhodes et. al report DHD-F of 0.78 on a set of 14 songs using a unsupervised spectral clustering approach [ICASSP 06]
  - Hard to compare on different corpus with different segmentations

Outline

- **Difference Features**
- Supervised Music Boundary Detection
  - **Feature Generation**
  - **Boosted Decision Stumps**
- Concluding Remarks

Summary

1. Difference features attempt to model acoustic cues that indicate boundaries within a song.
2. A supervised approach allows a user to explicitly define their a notion of ‘musical segment’ through their training segmentations.
3. Boosted decision stumps are used to
   - quickly identify music boundaries
   - produce good music segmentations
   - implicitly perform feature selection

Future Work

Address problems with dissimilarity measures

- Euclidean Distance assumes Euclidean vector space
- Reducing time series to a bag-of-features ignores temporal info

Use additional features

- Information theoretic features
- Beat Onset features - e.g., Drum Fill Detector

Learn ‘segment-specific’ classifiers

- e.g., ‘Chorus-Onset’ classifier

Explore new applications

- ‘chorus-based’ music similarity and retrieval

A Supervised Approach for **Detecting Boundaries** in Music using **Difference Features** and **Boosting**

Douglas Turnbull
Computer Audition Lab
UC San Diego, USA

Gert Lanckriet, UC San Diego, USA

Elias Pampalk , AIST Japan

Masataka Goto, AIST, Japan

ISMIR
September 24, 2007

Summary of Difference Features

We create 37 difference features

- A features is a time series of scalars sampled at 0.1 sample/sec.

Cue	Feature	Difference
Timbre	**Spectral** (6) • RMS, ZCR, harm, sc, perc, loud **MFCC** (4x3) • 1-5, 1-20, 2-5, 2-20 coef • original, delta, delta2	Subtraction KL Distance
Harmony	**Chroma** (3x3) • 12, 24, 36 pitch classes • original, delta, delta2	KL Distance
Melody	**F0** (2) • F0 and F0-Power	KL Distance
Rhythm	**Fluctuation Pattern** (8) G, f, max, sum, base, non aggr, LFD	Frobenius Norm Subtraction

Evaluation - Median Time

Median Time: time between estimates and true boundaries

- Measurement in seconds
- Lower times are better

Framework	Classifier	Estimate-to-True	True-to-Estimate
Baseline	Uniform Placement	8.6	6.4
Unsupervised	Peak Picking (MFCC-Diff)	5.1	3.7
Supervised	BDS	4.3	1.8