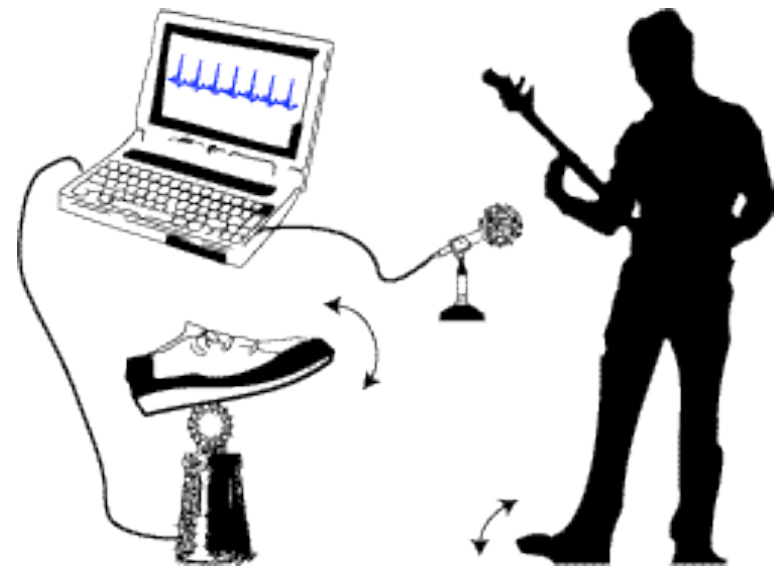


Beat Tracking of Musical Performances Using Low-Level Audio Features

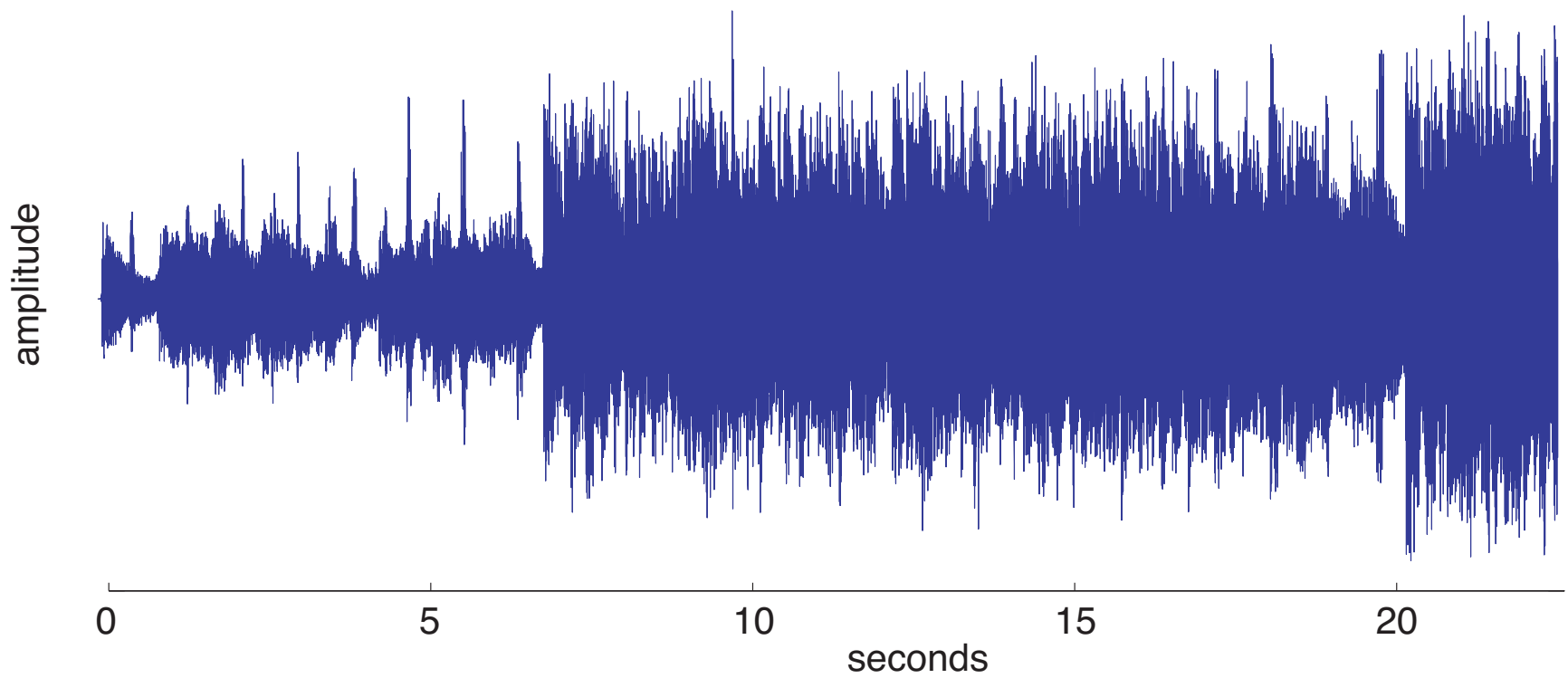
Beat tracking methods attempt to automatically synchronize to complex nonperiodic (yet repetitive) waveforms; to create an algorithm that can “tap its foot” in time with the music.

Beat tracking methods attempt to automatically synchronize to complex non-periodic (yet repetitive) waveforms; to create an algorithm that can “tap its foot” in time with the music. Important for

- musical signal processing
- information retrieval



First 1,000,000 samples of the *James Bond Theme*



Applications of Beat Tracking

- helping to understand how people process temporal information
- editing of audio data
- synchronization of visuals with audio
- audio information retrieval
- audio segmentation and signal processing
- a drum machine that “plays along with the band” rather than the band playing to the machine

Two Ideas:

The first is a method of data reduction that creates a collection of *rhythm tracks* (feature vectors) which represent the rhythmic structure of the piece. Each track uses a different method of (pre)processing the audio, and so provides a (somewhat) independent representation of the beat.

The second idea is to model the rhythm tracks (in simplified form) as a collection of random variables with changing variances: the variance is small when “between” the beats and large when “on” the beat. Exploiting this simple stochastic model of the rhythm tracks allows the beat detection to proceed using Bayesian methods.

What is a “beat” anyway?

Definition 1 An *auditory boundary* occurs at a time t when the sound stimulus in the interval $[t - \epsilon, t]$ is perceptibly different from the sound stimulus in the interval $[t, t + \epsilon]$.

Definition 2 A *beat* is a regular succession of auditory boundaries.

For example, a series of audio boundaries occurring at times

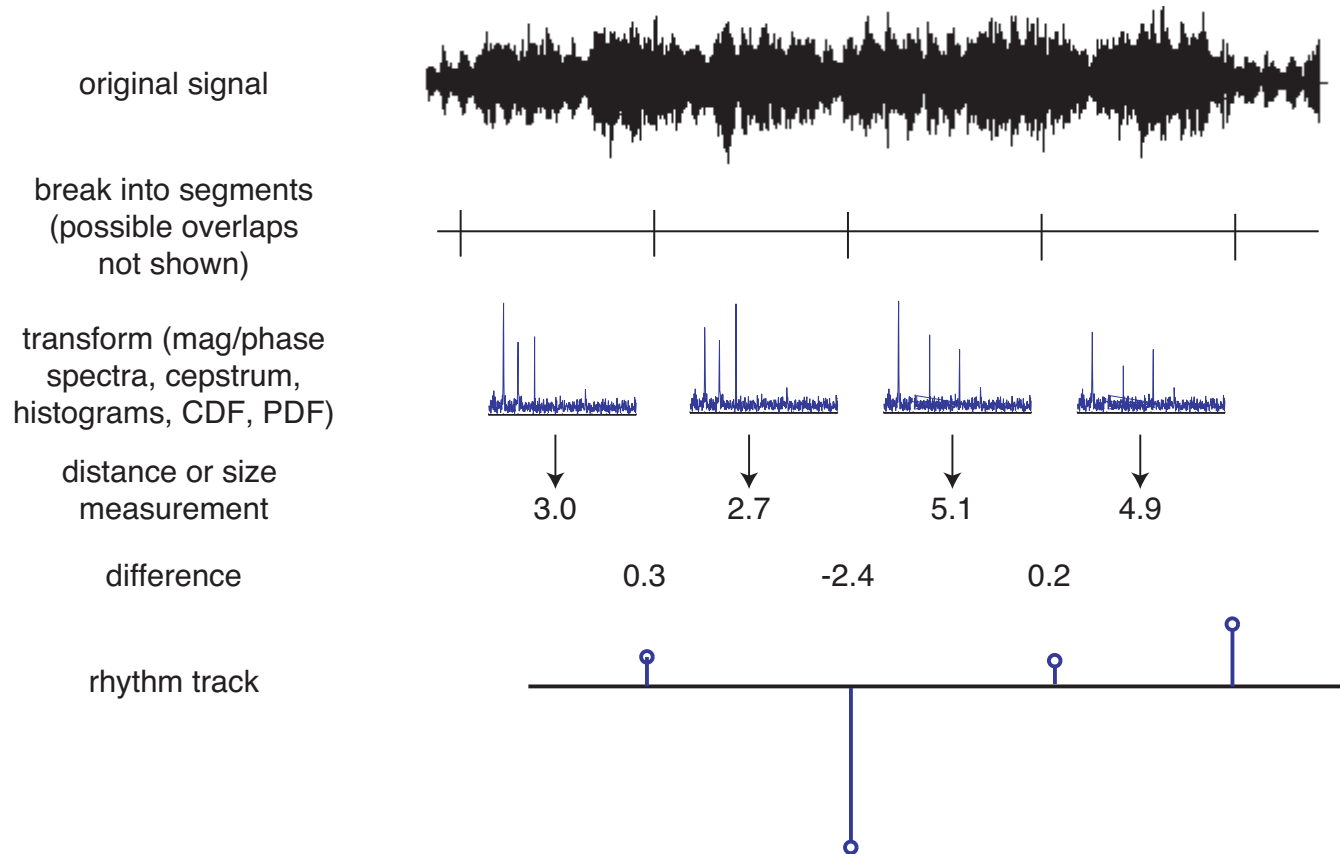
$$\tau, \tau + T, \tau + 2T, \tau + 3T, \tau + 4T, \dots$$

forms a beat of tempo T with a “phase” of τ .

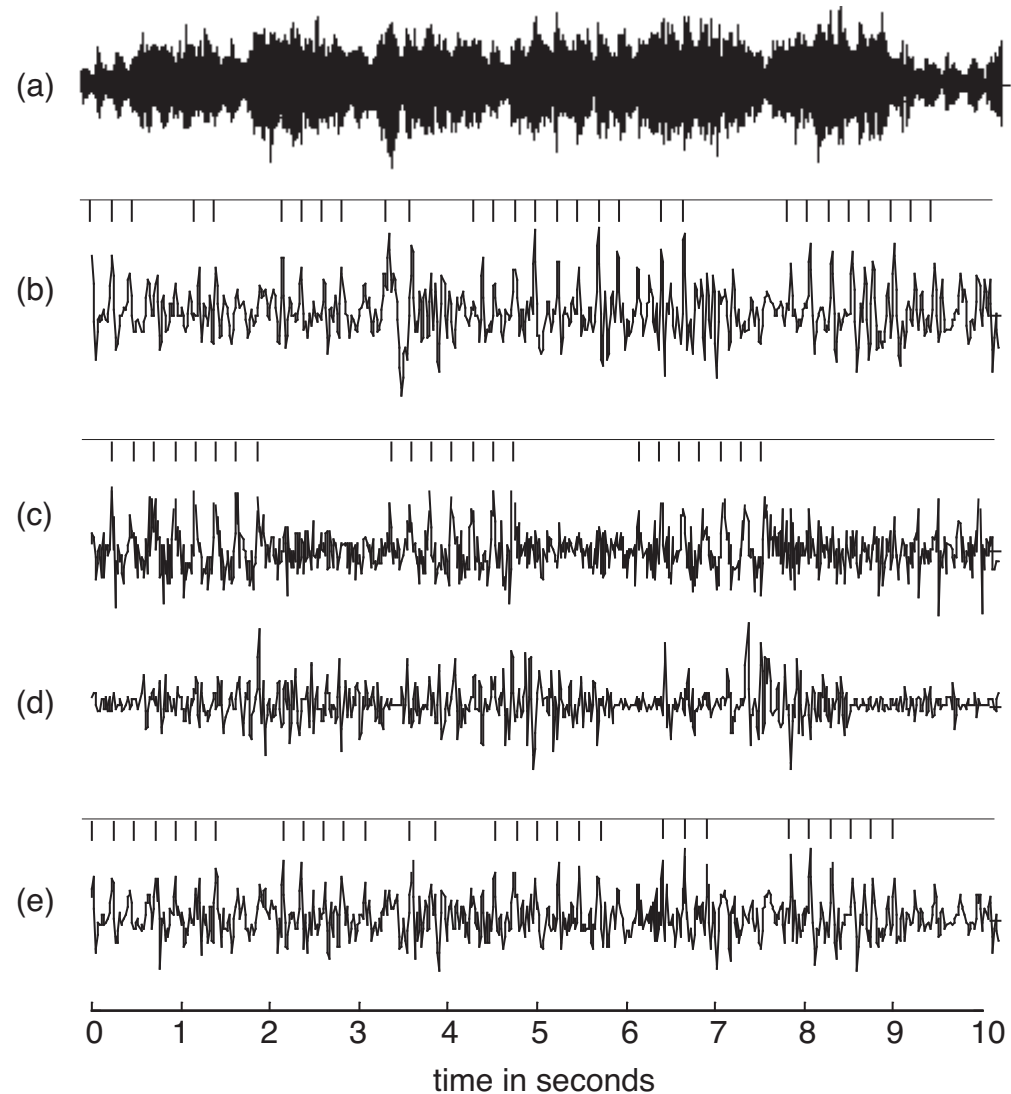
Idea of the rhythm track model

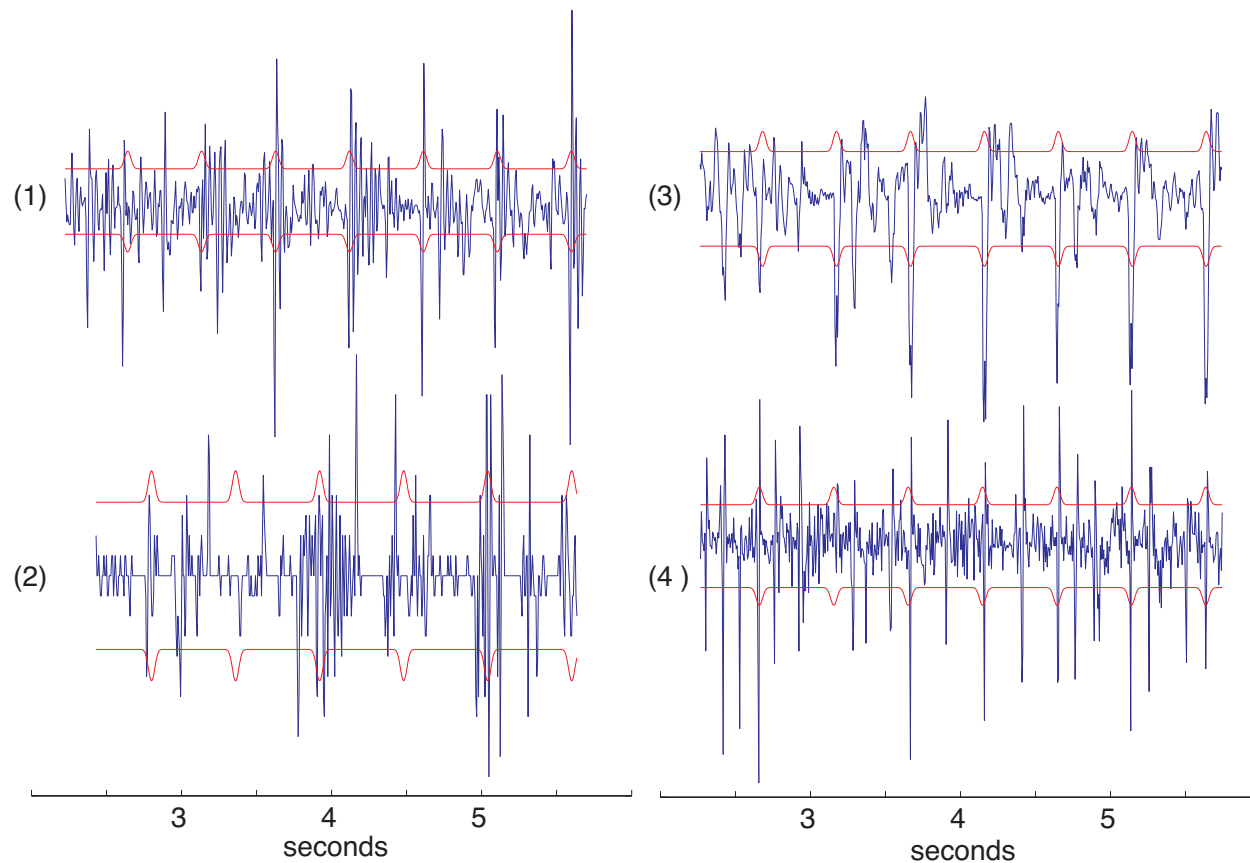
- T is on the order of 100ms-2s (hence want data reduction by a factor of ≈ 100).
- Beats tend to occur at auditory boundaries, when the sound changes.
- Many things may cause boundaries (amplitude changes, pitch/frequency changes, changes in timbre/spectrum, etc.) Will measure (initially) four different aspects.
- Because they look at different characteristics of the audio (each of which is related to the rhythmic aspects), they may be considered quasi-independent observations. Need way to combine these.

Building a rhythm track feature vector



Four rhythm tracks applied to the first 10 seconds of a recording of Handel's *Water Music*: part (a) the audio waveform, (b) the energy method, (c) group delay, (d) center of the spectrum, and (e) the dispersion. Tick marks emphasize beat locations that are visually prominent.





The four rhythm tracks of *Pieces of Africa* by the Kronos quartet between 2 and 6 seconds. The estimated beat times (which correctly locate the beat in cases (1), (3), and (4)) are superimposed over each track.

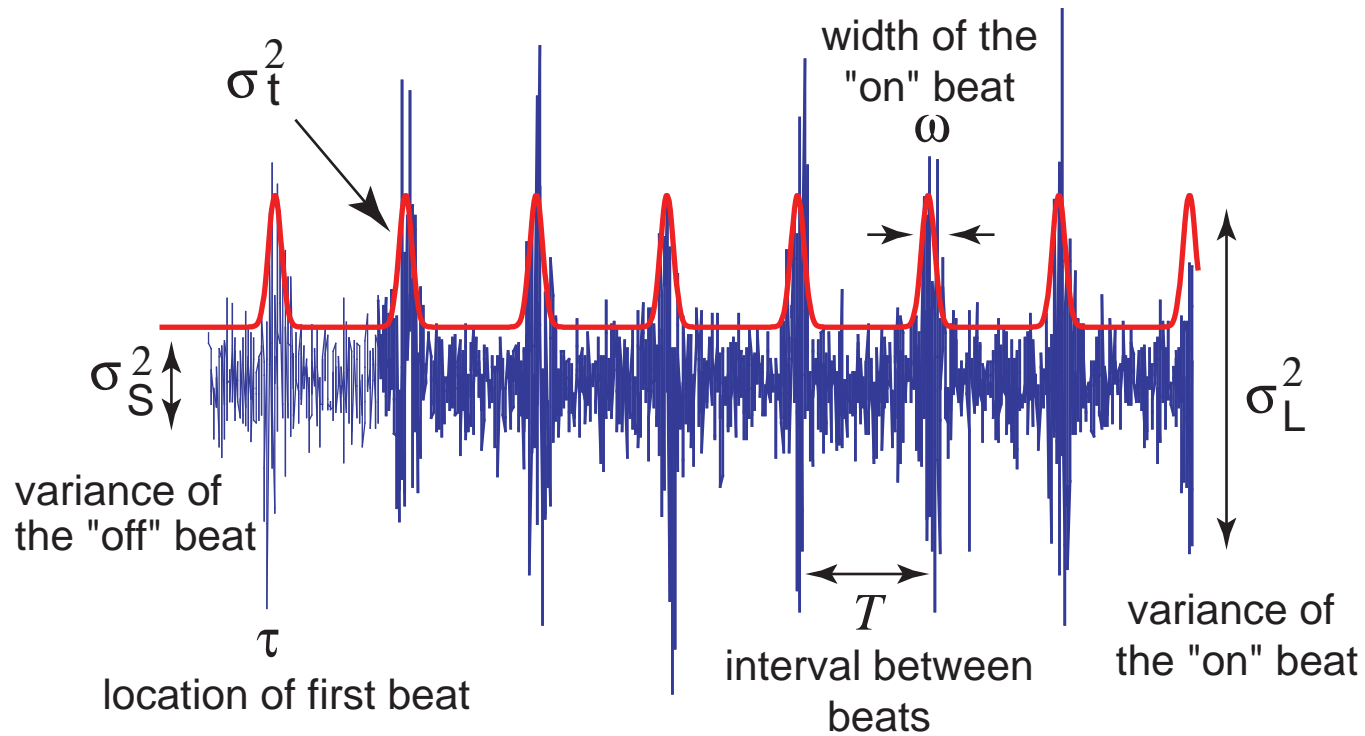
Parameters of the rhythm track model

Structural parameters:

- σ_1^2 is the “off the beat” variance,
- σ_2^2 is the “on the beat” variance, and
- ω is the beatwidth, the variance of the width of each set of “on the beat” events. For simplicity, this is assumed to have Gaussian shape.

Timing parameters:

- τ is the time of the first beat
- \mathcal{T} is the period of the beat, and
- $\delta\mathcal{T}$ is the rate of change of the beat period.

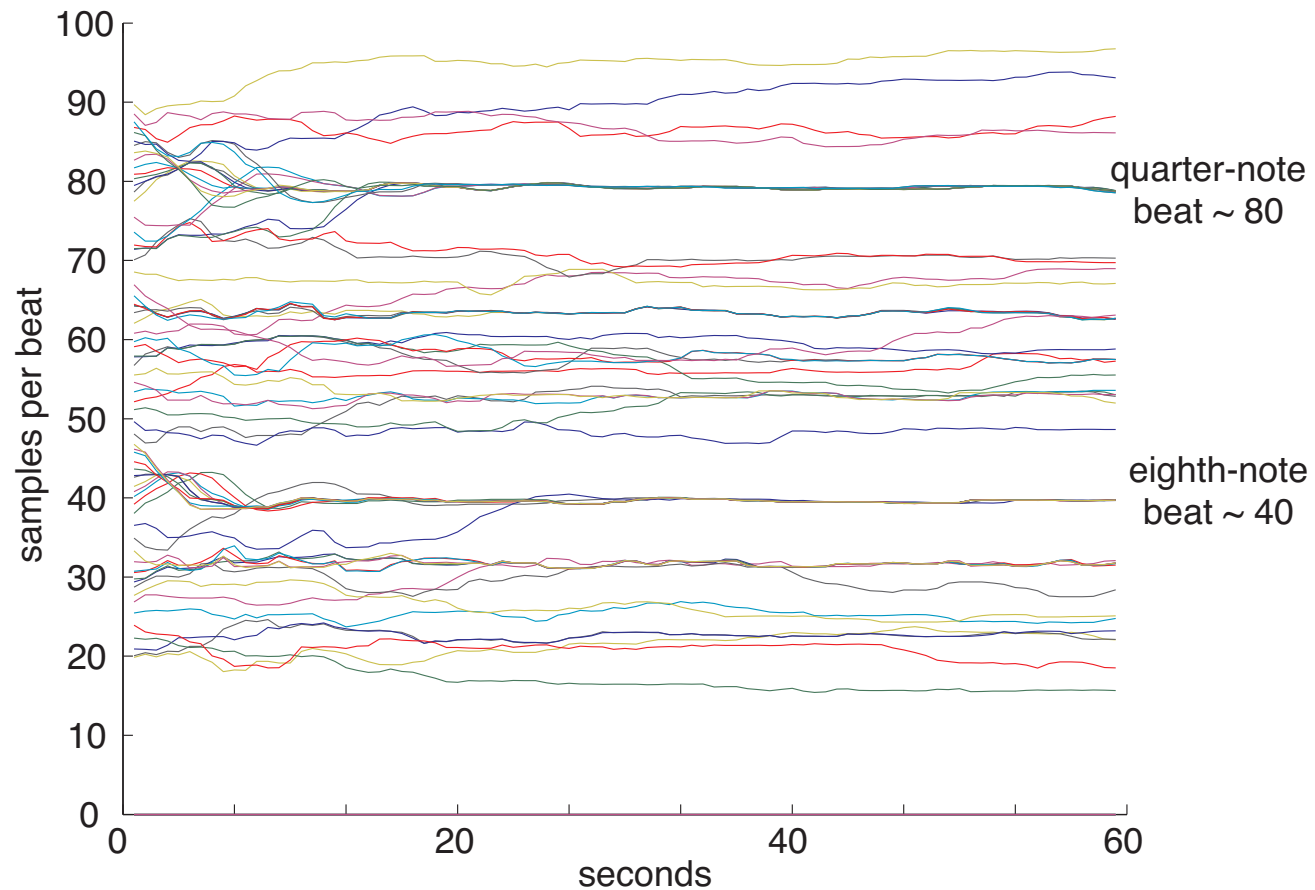


Parameters of the rhythm track model are \mathcal{T} , τ , ω , σ_1 , σ_2 and $\delta\mathcal{T}$ (not shown). Generative model assumes rhythm tracks composed of normal zero-mean random variables with variances defined by σ 's.

So now we have ways of transforming the raw audio into rhythm tracks. **Second idea** is to somehow parse the rhythm tracks in order to identify the parameters. Investigated two methods:

- A gradient method
- A Bayesian approach

Gradient is straightforward to implement, and computationally fast. But. . .



Estimates of the beat period for the *Theme from James Bond* using the gradient algorithm. Depending on initialization it may converge to a 1/8-note beat near 40 samples per period (0.23s) or to the 1/4-note beat near 80 (0.46s).

Applying a Bayesian model

Collect the timing parameters, τ , \mathcal{T} and $\delta\mathcal{T}$ into a state vector \mathbf{t} , and let $p(\mathbf{t}_{k-1}|\cdot)$ be the distribution over the parameters at block $k - 1$. The goal of the (recursive) particle filter is to update this to estimate the distribution over the parameters at block k , that is, to estimate $p(\mathbf{t}_k|\cdot)$.

The **predictive phase** details how \mathbf{t}_k is related to \mathbf{t}_{k-1} in the absence of new information: a diffusion model.

The **update phase** incorporates new information from the current time block.

Tracking using particle filters

- $p(\mathbf{t}_k | \text{current block of rhythm tracks})$ is proportional to $p(\text{current block of rhythm tracks} | \mathbf{t}_k)$
- Because the rhythm tracks are considered to be independent, the posterior is

$$\prod_i p(\text{rhythm track } i | \mathbf{t}_k) p(\mathbf{t}_k | \text{previous block of rhythm tracks})$$

- $p(\text{rhythm track } i | \mathbf{t}_k)$ is modeled as a product of Gaussians with the structured pattern of variances given above.

Particle Filters II

Applied to the beat tracking problem, the particle filter algorithm can be written in three steps. The particles are a set of N random samples, $\mathbf{t}_k(i)$, $i = 1 \dots N$ distributed as $p(\mathbf{t}_{k-1} | \mathbf{R}_{k-1})$.

1. **Prediction:** Each sample is passed through the system model to obtain samples of

$$\mathbf{t}_k^\dagger(i) = \mathbf{t}_{k-1}(i) + w_{k-1}(i) \text{ for } i = 1, 2, \dots, N,$$

which adds noise to each sample and simulates the diffusion portion of the procedure, where $w_{k-1}(i)$ is assumed to be a 3-dimensional Gaussian random variable with independent components. The variances of the three components depend on how much less certain the distribution becomes over the block.

2. **Update:** with the new block of rhythm track values, r_k , evaluate the likelihood for each particle. Compute the normalized weights for the samples

$$q_i = \frac{p(r_k | \mathbf{t}_k^\dagger(i))}{\sum_i p(r_k | \mathbf{t}_k^\dagger(i))}.$$

3. **Resample:** Resample N times from the discrete distribution over the $\mathbf{t}_k^\dagger(i)$'s defined by the q_i 's to give samples distributed as $p(\mathbf{t}_k | \mathbf{R}_k)$.

How well does it work?

OK.

Listen to a couple of examples. What we've done is to superimpose a short noise burst at each beat – hence it's easy to hear when things are working and when they're not.

- *Theme from James Bond* (bondtap)
- Handel's *Water Music* (watertap)
- Brubeck's *Take Five* (take5tap)
- Joplin's *Maple Leaf Rag* (maplecoolrobtap)
- Baltimore Consort's *Howell* (BChowelltap)

Examples of some things you can do when you know the beat structure of a piece of music:

- Manipulate on a per-beat basis (e.g., *Reversed Rag*) ([maplecoolreverse](#))
- An ideal situation for FFT-based analysis (e.g., *Rag Minus Noise*, *Rag Minus Signal*) ([maplecoolnoise](#)) ([maplecoolsig](#))
- Edit/add stuff (e.g., *Switched on Rag*) ([maplecoolrobdrums](#))
- Re-order a piece 1-2-3- . . . - 30-31-32-31-30-29 - . . . -3-2-1 ([friendneirf](#))
- Manipulate structure of piece (e.g., the *Maple Leaf Waltz*, *James Bond Waltz*, *Backwards Bond*, *Take 4*) ([maple34](#)) ([bond34](#)) ([bondback](#)) ([take4](#))

Beat-Based Signal Processing...

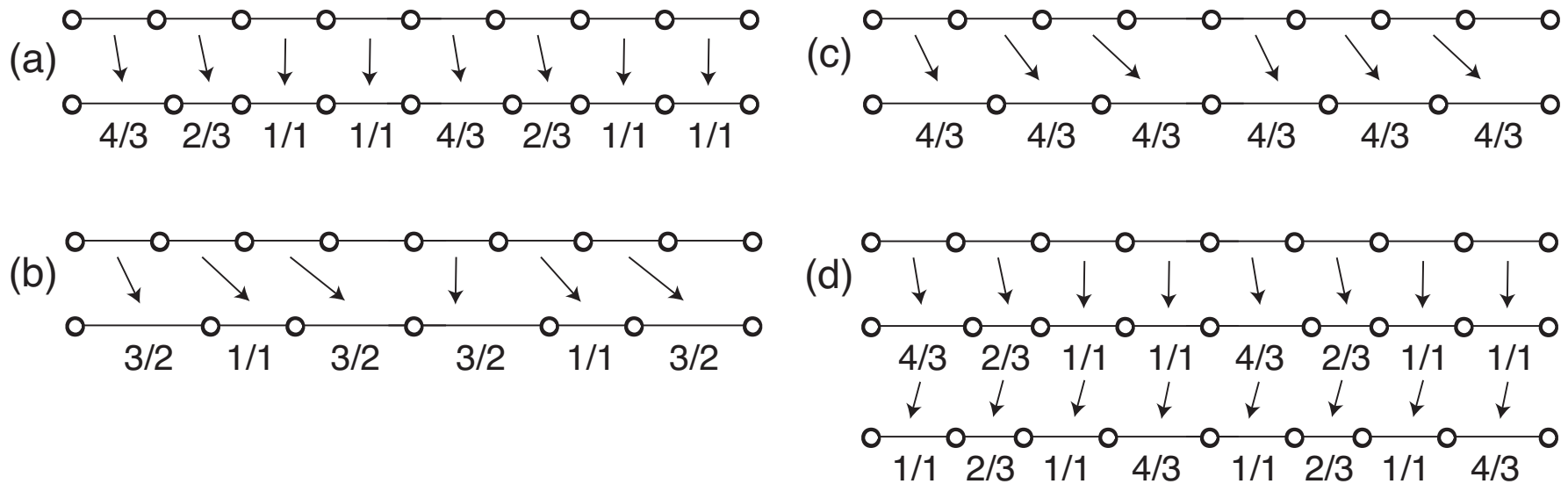
- David Bowie singing backwards... (New Star City)
- Map Maple to 5-tet (Pentatonic Rag)
- Map Maple to many n-tets (maplemanytet)

		n-tet destination																
<i>A</i>	:	3	3	3	4	4	4	4	5	5	5	5	7	7	7	7	3	:
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
		beat																

		n-tet destination																
<i>B</i>	:	4	4	4	5	4	4	4	5	5	5	4	5	5	5	4	4	:
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
		beat																

Beat-Based Signal Processing II...

Changing the duration of beat intervals can be used as a kind of beat-synchronized delay processing. Performing different versions simultaneously increases the density, often in a rhythmic way.



(MagicLeafRag, MakeItBriefRag)

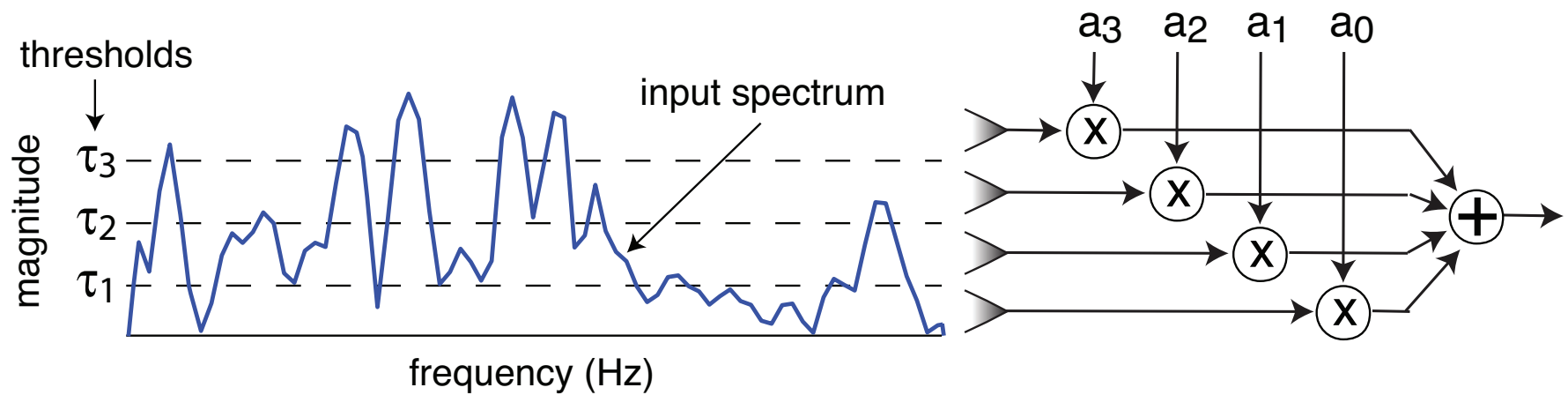
Beat-Based Signal Processing III...

Using complex waveforms (like the *Maple Leaf Rag*) as an input to a synthesizer, carrying out synthesis on a per-beat basis.

- *Beat Gated Rag* (BeatGatedRag)
- *Noisy Souls* and *Frozen Souls* (NoisySouls, FrozenSouls)

Beat-Based Signal Processing IV...

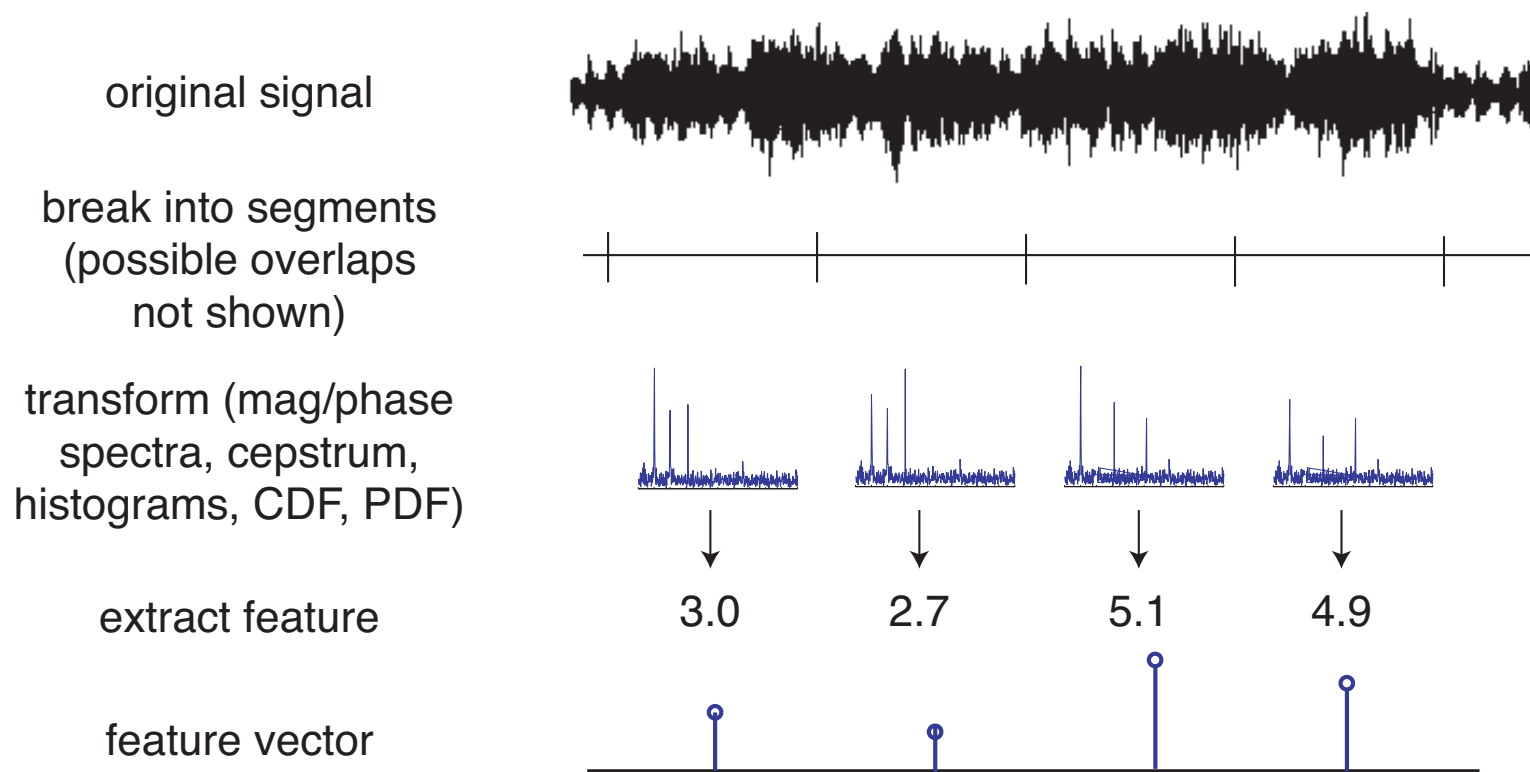
Spectral band filters sound radically different from any linear filter. *Local Variations* results from application of a fixed (eight band) spectral band filter to *Local Anomaly*. Within each beat, the relative sizes of the spectral peaks are rearranged, causing drastic timbral changes that nonetheless maintain the rhythmic feel.



(LocalAnomaly, LocalVariation)

	Phase Vocoder	Beat-Synchronized FFT
windows	small frames from 1K-4K with 2 to 8 times overlap	large beat-sized windows $\frac{1}{5}$ - $\frac{1}{2}$ sec, zero padded to a power of two
FFT resolution	40 Hz - 10 Hz (improved by phase adjustment)	3 Hz - 1.5 Hz (phase adjustment possible)
peak finding	all local max above median or threshold	plus distance parameter (forbidding peaks too close together)
spectral mapping	direct resynthesis: output frequencies placed in FFT vector with phase adjustment	resampling with identity window, no phase adjustment
beat detection	optional	required
examples	<p>Maple5tetPV</p> <p>Soul65HzPV</p>	<p>Maple5tetFFT</p> <p>Soul65HzFFT</p>

Feature Vectors attempt to extract relevant features of the sound from the waveform by reducing it to frames and deriving a single number from each frame



Some Interesting Features

Sensory dissonance (within each frame)

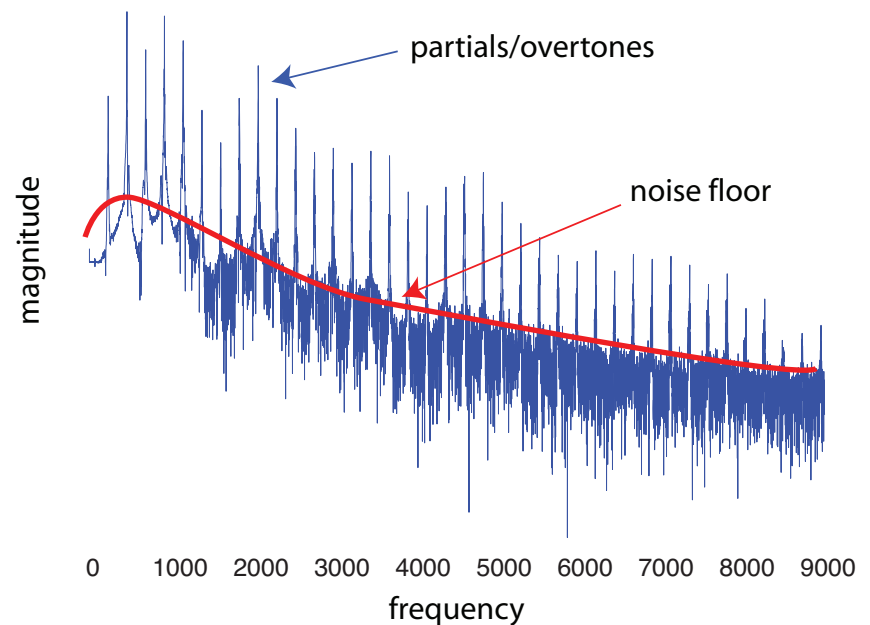
Centroid of magnitude spectrum

Dispersion about centroid

Signal to Noise ratio

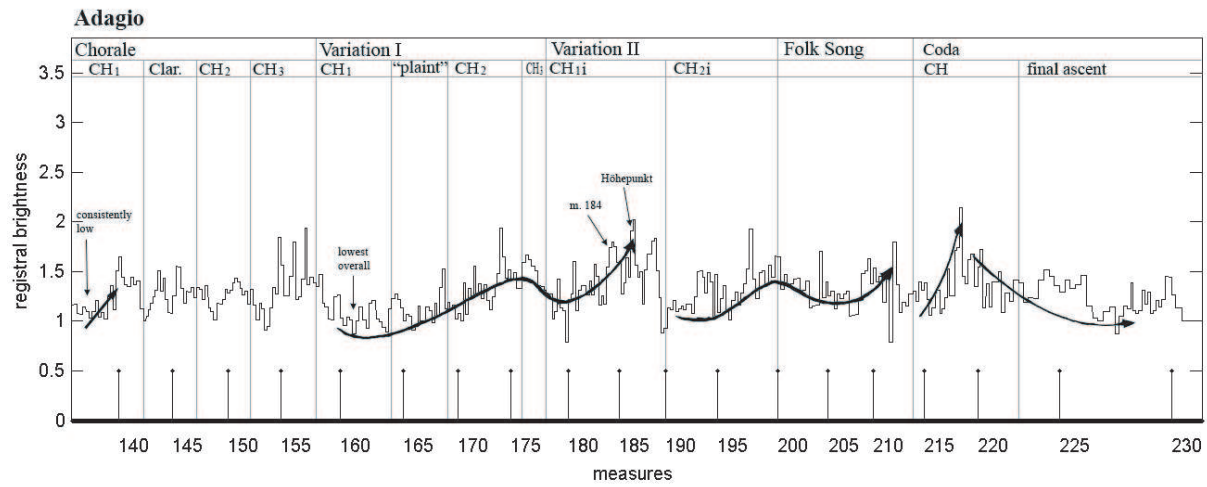
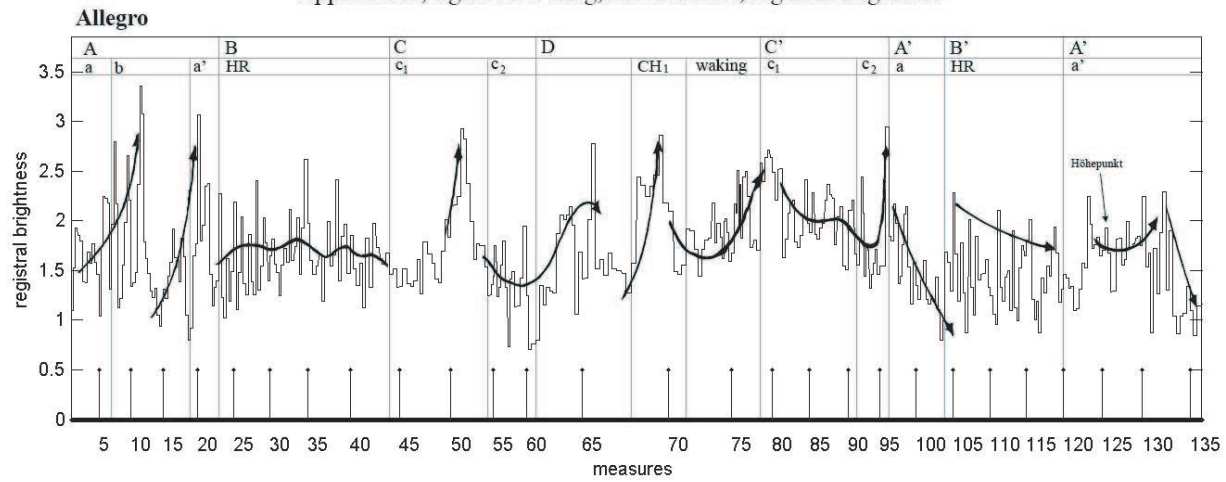
Number of significant partials

Slope of Noise Floor

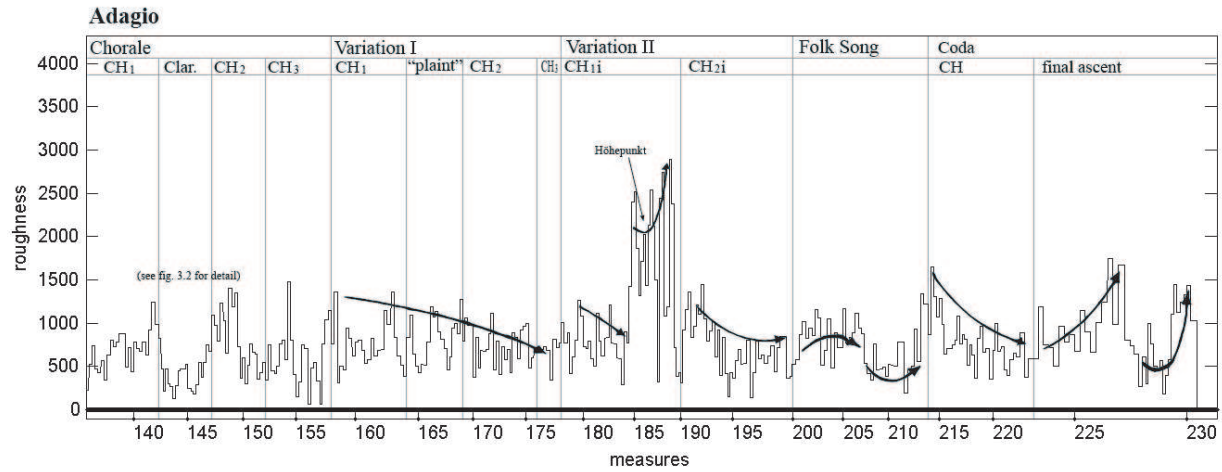
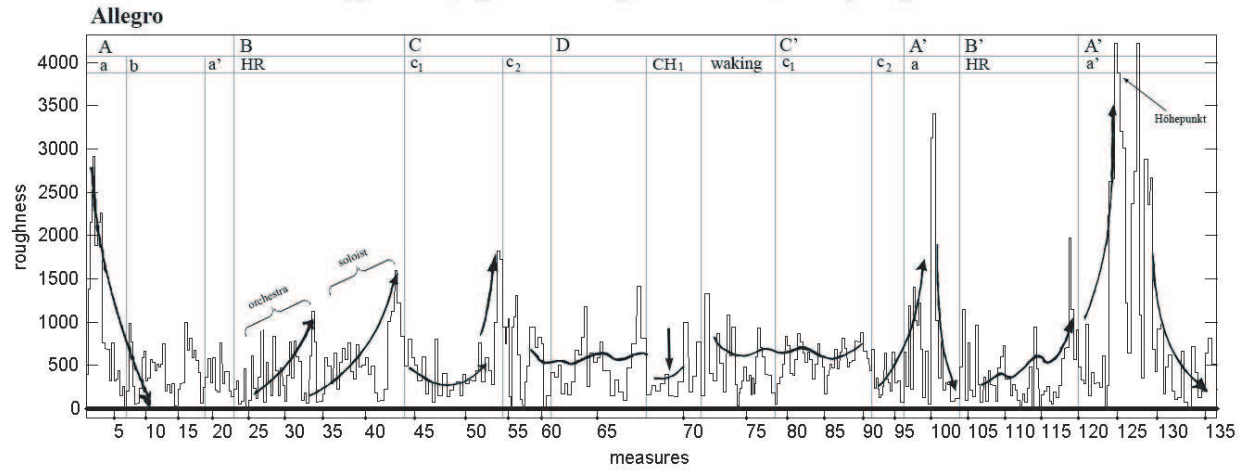


Some examples using Berg's Angel Concerto...

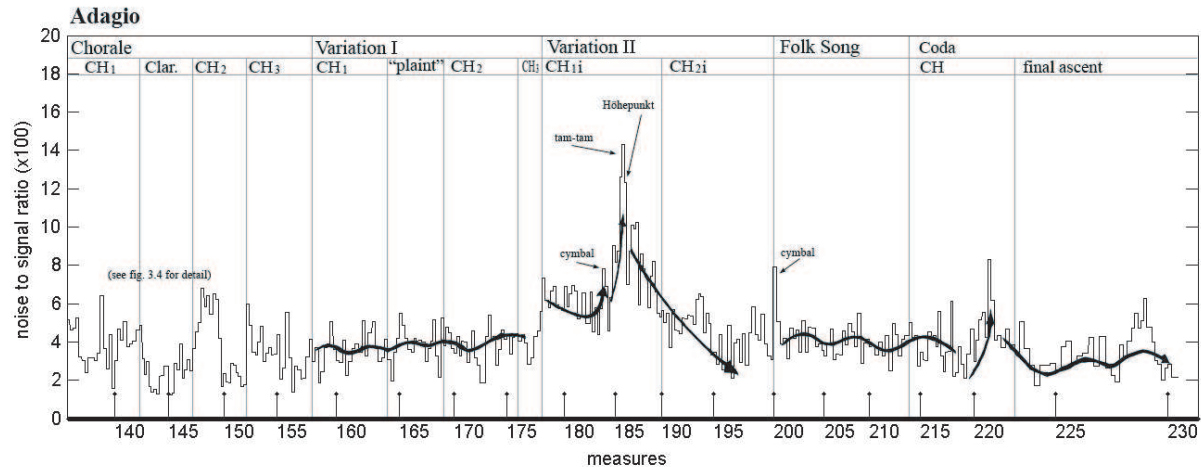
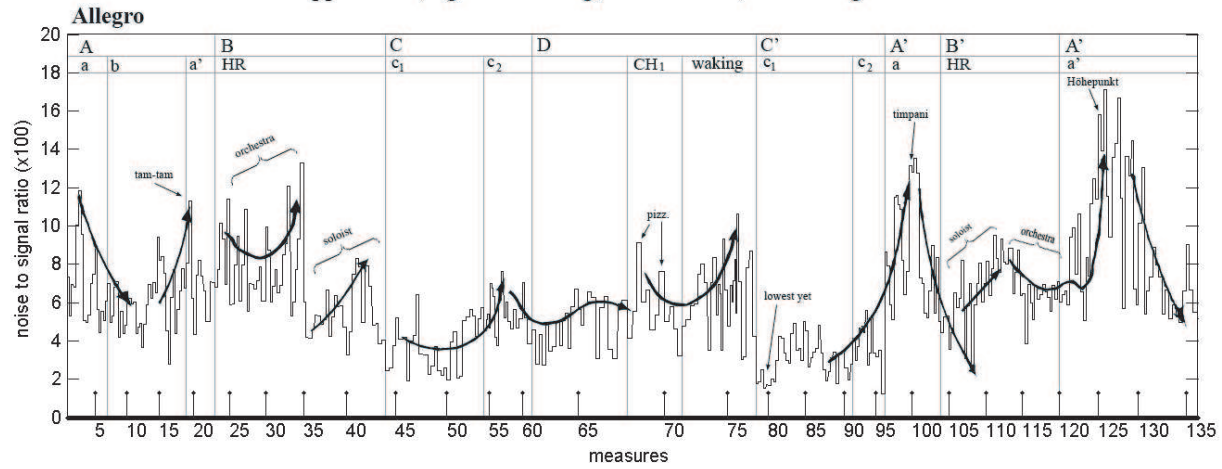
Appendix A, figure A.4. Berg, movement II, registral brightness



Appendix A, figure A.1. Berg, movement II, sensory roughness



Appendix A, figure A.3. Berg, movement II, noise to signal ratio



Of course, method can't possibly work when music is inappropriate, e.g., swirling undifferentiated sound masses with no discernable rhythm. But, when the music is appropriate, it still does not always work.

What are the failure modes?

When the algorithm cannot find the correct beat times, is it that

- the rhythm tracks fail to correctly locate beat boundaries?
- the Bayesian algorithm fails to find τ , T , and/or dT despite good rhythm tracks?

What other kinds of rhythm tracks can we think of?

other ways to measure distance

1	$\sum_i x_i^2$	l^2 energy
2	$\sum_i x_i $	l^1 norm
3	$\sum_i \log(x_i^2)$	log energy
4	$\sum_i x_i^2 \log(x_i^2)$	entropy
5	$\sum_i x_i \log(x_i)$	absolute entropy
6	$\operatorname{argmax} x_i $	location of maximum
7	$\max z_i - y_i $	KS test (for CDF)
8	$\#\{i : z_i > \eta\}, \eta = \operatorname{mean}(z_i)$	number of $ z_i $ larger than mean
9	$\eta = \sqrt{2 \log_e(n \log_2(n))}$	SURE threshold in measure 8
10	$\max(z_i) - \min(z_i)$	range of data
11	$\min_m \sum_i (z_i - mi - b)^2$	slope
12	$\min_m \left \sum_{j=1}^m z_j^2 - \frac{1}{2} \sum_{i=1}^n z_i^2 \right $	center
13	$\sum_i (m - z_i^2)^2$	dispersion about center m
14	$\sum_i z_{i+1} - z_i $	total absolute variation
15	$\sum_i (z_{i+1} - z_i)^2$	total square variation
16	$\sum_i \log\left(\frac{ z_i }{ y_i }\right)$	cross information
17	$\sum_i z_i \log\left(\frac{ z_i }{ y_i }\right) + y_i \log\left(\frac{ y_i }{ z_i }\right)$	symmetrized cross entropy
18	$\sum_i i z_i^2$	weighted energy

The partitioned data can be transformed into different domains.

label	domain
<i>A</i>	time signal
<i>B</i>	magnitude of FFT
<i>C</i>	phase of FFT
<i>D</i>	cepstrum
<i>E</i>	PDF of time signal
<i>F</i>	CDF of time signal
<i>G</i>	FFT of the PDF of time
<i>H</i>	PDF of FFT magnitude
<i>I</i>	CDF of FFT magnitude
<i>J</i>	PDF of cepstrum
<i>K</i>	CDF of cepstrum
<i>L</i>	various subbands

How many different ways of building rhythm tracks are there?

Approximately the product of:

$$\left\{ \begin{array}{l} \# \text{ ways to} \\ \text{choose partitions} \end{array} \right\} \times \left\{ \begin{array}{l} \# \text{ of} \\ \text{domains} \end{array} \right\} \times \left\{ \begin{array}{l} \# \text{ of distance} \\ \text{measures} \end{array} \right\} \times \left\{ \begin{array}{l} \# \text{ ways of} \\ \text{differencing} \end{array} \right\}$$

We found 7344 different rhythm tracks. Need a way of testing to see if these are good or bad.

Idea for a test

Since rhythm tracks may be modeled as a collection of normal random variables with changing variances, can measure the quality Q of a rhythm track by measuring the fidelity of the rhythm track to the model.

- (a) Choose a set of test pieces for which the beat boundaries are known.
- (b) For each piece and for each candidate rhythm track, calculate the quality measure Q .
- (c) Those rhythm tracks which score highest over the complete set of test pieces are the best rhythm tracks.
- (d) Independence: check that the rhythm tracks are truly independent of each other (e.g., SVD test).

The best rhythm tracks

- based on magnitude of FFT
- based on CDF/PDF (histograms) of FFT
- based on CDF/PDF (histograms) of cepstrum
- the only time-based measure remaining was energy
- none of standard stochastic tests (SURE, K-S, etc.) based on time signal, but some using CDFs



A new book focusing on the technologies of beat tracking. *Rhythm and Transforms* describes the impact of beat tracking on music theory and on the design of sound processing electronics such as musical synthesizers, drum machines, and special effects devices. Coming this summer!