



Extended Source-Filter Model For Harmonic Instruments For Expressive Control Of Sound Synthesis And Transformation

Henrik Hahn Axel Röbel

henrik.hahn@ircam.fr

IRCAM - CNRS - UMR 9912 - STMS, Paris, France

3rd September 2013



Introduction

Enhanced Source-Filter Model

Results

Demo



Introduction

Enhanced Source-Filter Model

Results

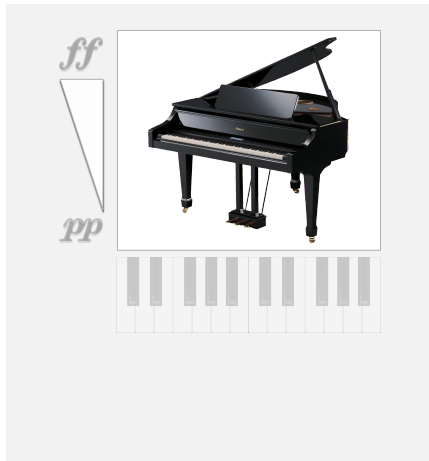
Demo

Musical Sampler (State of the Art)



- ▶ Based on recordings
- ▶ Along two sound dimensions (pitch, intensity)
- ▶ Instrument characteristics are *discretized*
- ▶ Synthesis sounds *static*

- ▶ Transformations are used to interpolate the sound space
- ▶ These are based on local models.

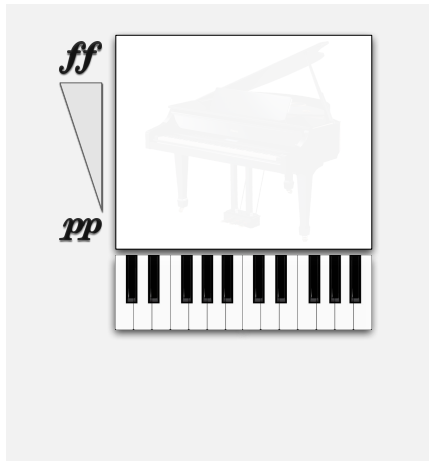


Musical Sampler (State of the Art)



- ▶ Based on recordings
- ▶ Along two sound dimensions (pitch, intensity)
- ▶ Instrument characteristics are *discretized*
- ▶ Synthesis sounds *static*

- ▶ Transformations are used to interpolate the sound space
- ▶ These are based on local models.

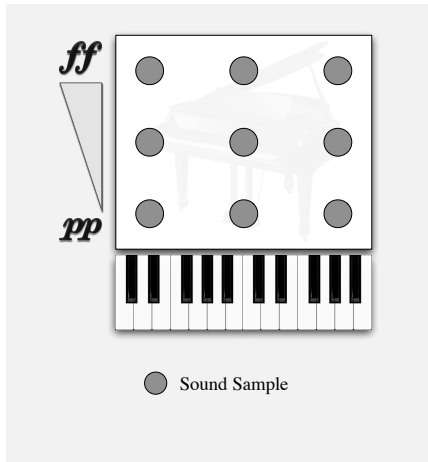


Musical Sampler (State of the Art)



- ▶ Based on recordings
- ▶ Along two sound dimensions (pitch, intensity)
- ▶ Instrument characteristics are *discretized*
- ▶ Synthesis sounds *static*

- ▶ Transformations are used to interpolate the sound space
- ▶ These are based on local models.

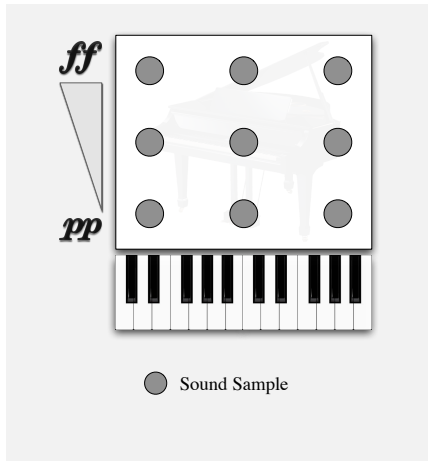


Musical Sampler (State of the Art)



- ▶ Based on recordings
- ▶ Along two sound dimensions (pitch, intensity)
- ▶ Instrument characteristics are *discretized*
- ▶ Synthesis sounds *static*

- ▶ Transformations are used to interpolate the sound space
- ▶ These are based on local models.

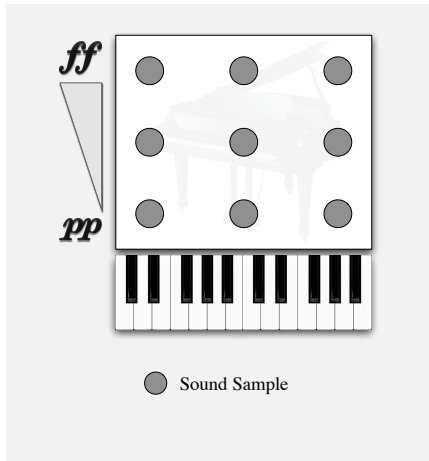


Musical Sampler (State of the Art)



- ▶ Based on recordings
- ▶ Along two sound dimensions (pitch, intensity)
- ▶ Instrument characteristics are *discretized*
- ▶ Synthesis sounds *static*

- ▶ Transformations are used to interpolate the sound space
- ▶ These are based on local models.

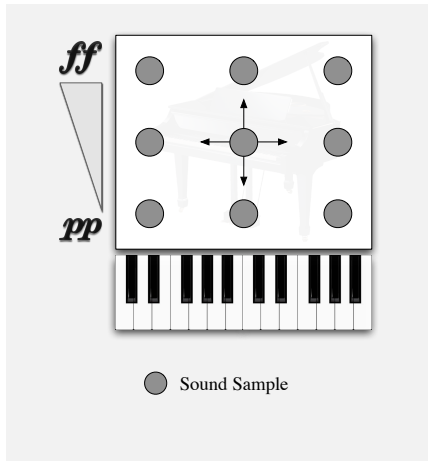


Musical Sampler (State of the Art)



- ▶ Based on recordings
- ▶ Along two sound dimensions (pitch, intensity)
- ▶ Instrument characteristics are *discretized*
- ▶ Synthesis sounds *static*

- ▶ Transformations are used to interpolate the sound space
- ▶ These are based on local models.

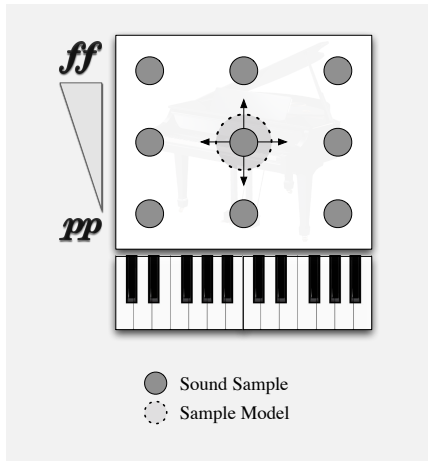


Musical Sampler (State of the Art)



- ▶ Based on recordings
- ▶ Along two sound dimensions (pitch, intensity)
- ▶ Instrument characteristics are *discretized*
- ▶ Synthesis sounds *static*

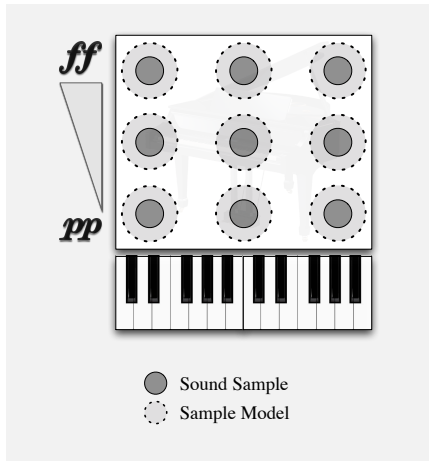
- ▶ Transformations are used to interpolate the sound space
- ▶ These are based on local models.



Musical Sampler (State of the Art)

- ▶ Trafos do not account for real instrument characteristics
- ▶ Interpolation is actually an extrapolation
- ▶ One model per sample means high model complexity
- ▶ Implicit overfitting

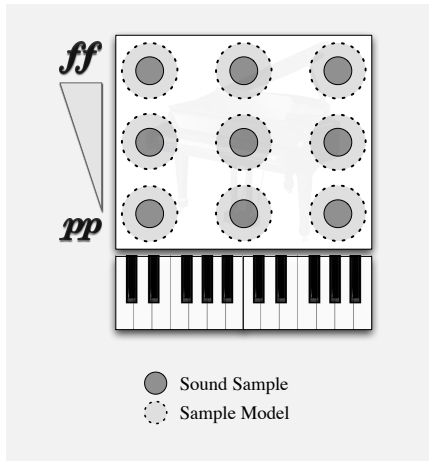
- ▶ Note repetitions lead to strange artifacts
- ▶ Audible jumps on switching samples



Musical Sampler (State of the Art)

- ▶ Trafos do not account for real instrument characteristics
- ▶ Interpolation is actually an extrapolation
 - ▶ One model per sample means high model complexity
 - ▶ Implicit overfitting

- ▶ Note repetition leads to strange timbre effects
- ▶ Audible jumps on switching samples

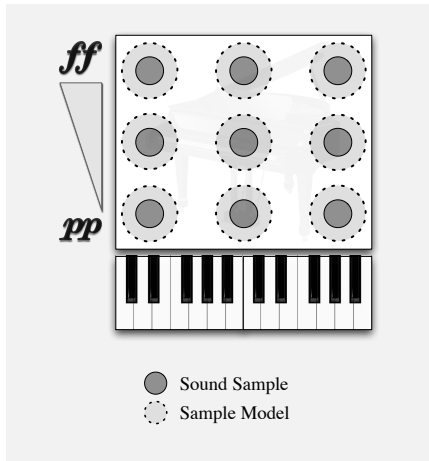


Musical Sampler (State of the Art)



- ▶ Trafos do not account for real instrument characteristics
- ▶ Interpolation is actually an extrapolation
- ▶ One model per sample means high model complexity
- ▶ Implicit overfitting

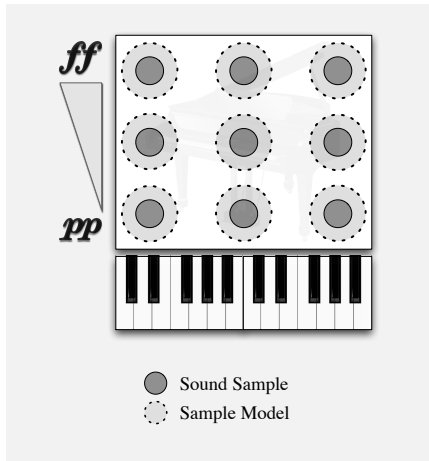
- ▶ Note repetition results in overlapping samples
- ▶ Multiple Samples per Note
- ▶ Available samples for switching samples



Musical Sampler (State of the Art)

- ▶ Trafos do not account for real instrument characteristics
- ▶ Interpolation is actually an extrapolation
- ▶ One model per sample means high model complexity
- ▶ Implicit overfitting

- ▶ Multiple samples for each instrument
- ▶ Multiple models for each sample
- ▶ Available samples for each instrument sample

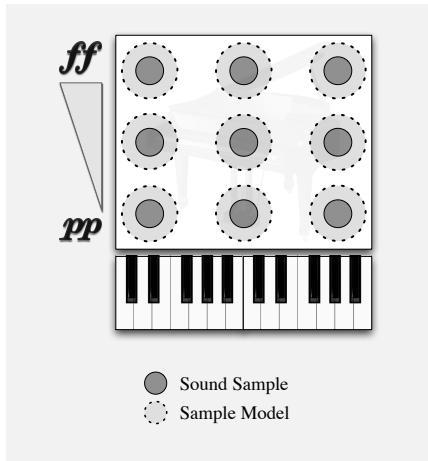


Musical Sampler (State of the Art)



- ▶ Trafos do not account for real instrument characteristics
- ▶ Interpolation is actually an extrapolation
- ▶ One model per sample means high model complexity
- ▶ Implicit overfitting

- ▶ Note repetitions lead to Machine-Gun-Effect
- ▶ Audible jumps on switching samples

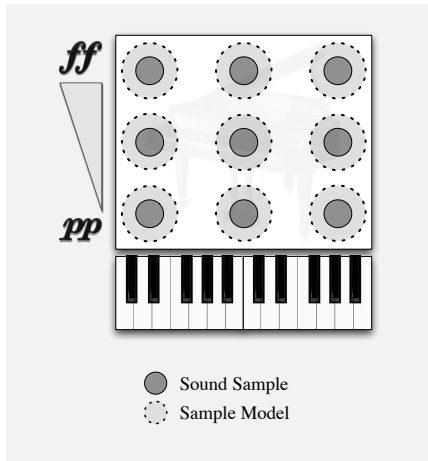


Musical Sampler (State of the Art)

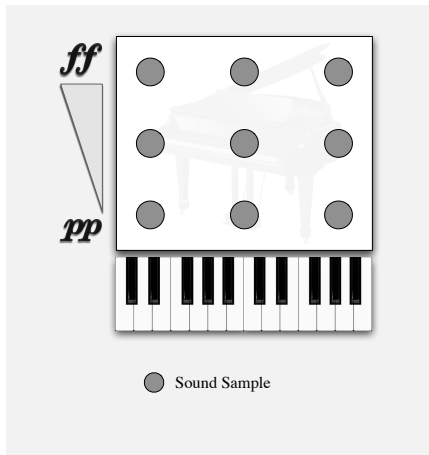


- ▶ Trafos do not account for real instrument characteristics
- ▶ Interpolation is actually an extrapolation
- ▶ One model per sample means high model complexity
- ▶ Implicit overfitting

- ▶ Note repetitions lead to Machine-Gun-Effect
- ▶ Audible jumps on switching samples

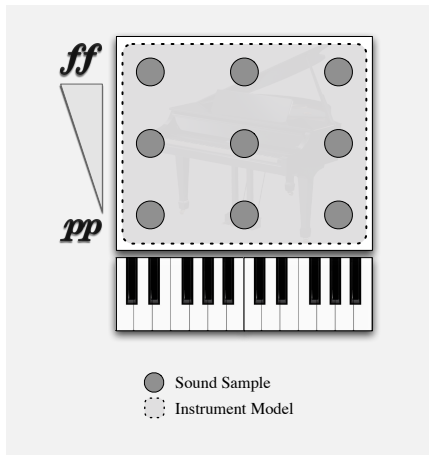


Musical Sampler (Proposition)



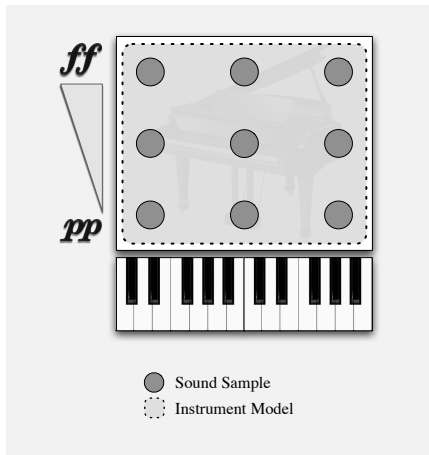
- ▶ Parametric model describing the sound space according to **pitch** and **global note intensity**
- ▶ and the **temporal evolution** of a sound (Attack-Sustain-Release)
- ▶ With these 3 parameters being the **Gestural Control Parameters**
- ▶ Process harmonic and noise content separately
- ▶ Model parameters shall be learned from standard databases
- ▶ Model shall be applicable for all harmonic instruments

Musical Sampler (Proposition)



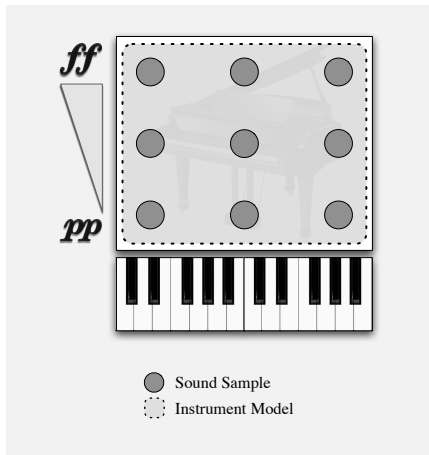
- ▶ Parametric model describing the sound space according to **pitch** and **global note intensity**
- ▶ and the **temporal evolution** of a sound (Attack-Sustain-Release)
- ▶ With these 3 parameters being the **Gestural Control Parameters**
- ▶ Process harmonic and noise content separately
- ▶ Model parameters shall be learned from standard databases
- ▶ Model shall be applicable for all harmonic instruments

Musical Sampler (Proposition)



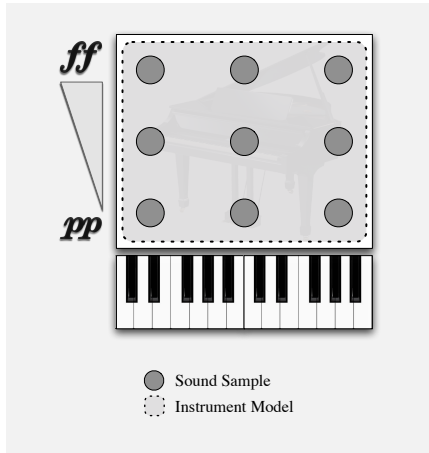
- ▶ Parametric model describing the sound space according to **pitch** and **global note intensity**
- ▶ and the **temporal evolution** of a sound (Attack-Sustain-Release)
- ▶ With these 3 parameters being the **Gestural Control Parameters**
 - ▶ Process harmonic and noise content separately
 - ▶ Model parameters shall be learned from standard databases
 - ▶ Model shall be applicable for all harmonic instruments

Musical Sampler (Proposition)



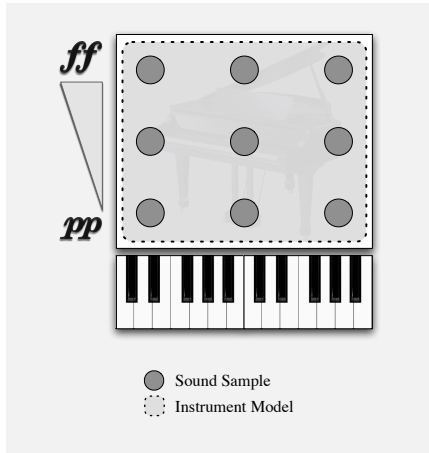
- ▶ Parametric model describing the sound space according to **pitch** and **global note intensity**
- ▶ and the **temporal evolution** of a sound (Attack-Sustain-Release)
- ▶ With these 3 parameters being the **Gestural Control Parameters**
- ▶ Process harmonic and noise content separately
- ▶ Model parameters shall be learned from standard databases
- ▶ Model shall be applicable for all harmonic instruments

Musical Sampler (Proposition)



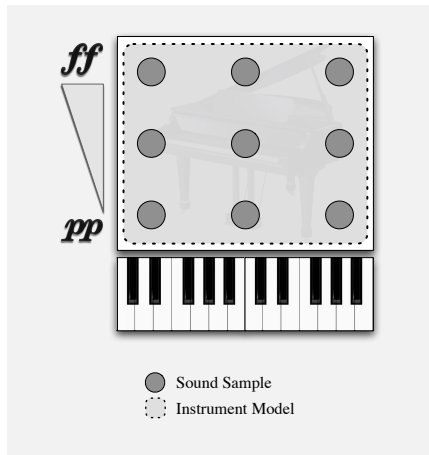
- ▶ Parametric model describing the sound space according to **pitch** and **global note intensity**
- ▶ and the **temporal evolution** of a sound (Attack-Sustain-Release)
- ▶ With these 3 parameters being the **Gestural Control Parameters**
- ▶ Process harmonic and noise content separately
- ▶ Model parameters shall be learned from standard databases
- ▶ Model shall be applicable for all harmonic instruments

Musical Sampler (Proposition)



- ▶ Parametric model describing the sound space according to **pitch** and **global note intensity**
- ▶ and the **temporal evolution** of a sound (Attack-Sustain-Release)
- ▶ With these 3 parameters being the **Gestural Control Parameters**
- ▶ Process harmonic and noise content separately
- ▶ Model parameters shall be learned from standard databases
- ▶ Model shall be applicable for all harmonic instruments

Musical Sampler with an Instrument Model

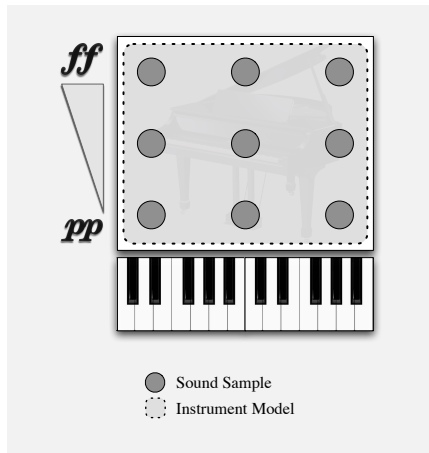


- ▶ Transformations based on real instrument characteristics
- ▶ Synthesis with continuous pitch and intensity values
- ▶ as well as arbitrary note durations

▶ Expressive sound synthesis with gestural control parameters

- ▶ Replacement of samples
- ▶ Creation of instrumental hybrids

Musical Sampler with an Instrument Model



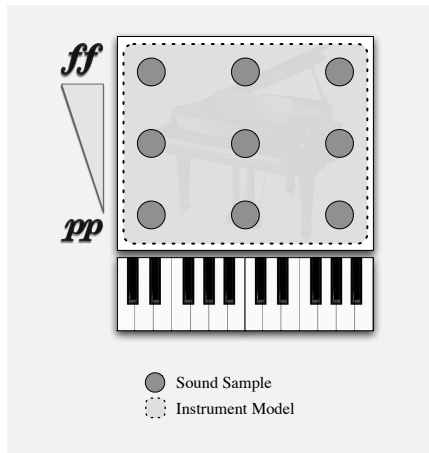
- ▶ Transformations based on real instrument characteristics
- ▶ Synthesis with continuous pitch and intensity values
- ▶ as well as arbitrary note durations

▶ Expressive sound synthesis with gestural control parameters

▶ Replacement of samples

▶ Creation of instrumental hybrids

Musical Sampler with an Instrument Model



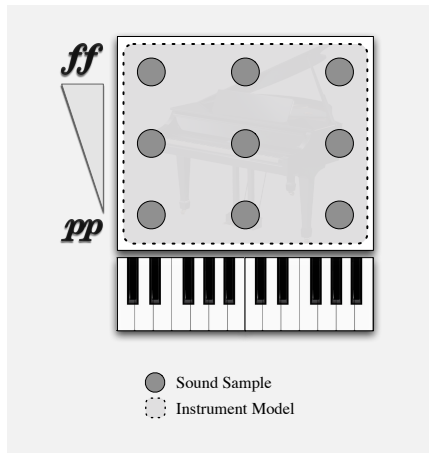
- ▶ Transformations based on real instrument characteristics
- ▶ Synthesis with continuous pitch and intensity values
- ▶ as well as arbitrary note durations

▶ Expressive sound synthesis with gestural control parameters

▶ Replacement of samples

▶ Creation of instrumental hybrids

Musical Sampler with an Instrument Model

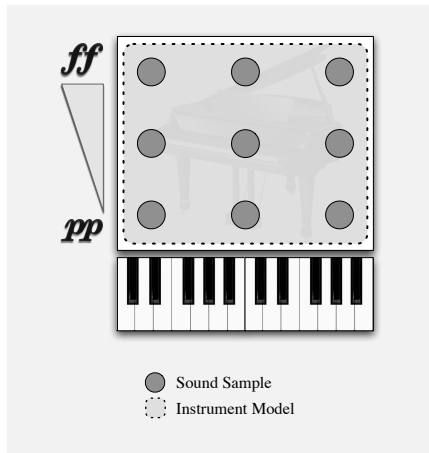


- ▶ Transformations based on real instrument characteristics
- ▶ Synthesis with continuous pitch and intensity values
- ▶ as well as arbitrary note durations

- ▶ Expressive sound synthesis with gestural control parameters

- ▶ Replacement of samples
- ▶ Creation of instrumental hybrids

Musical Sampler with an Instrument Model

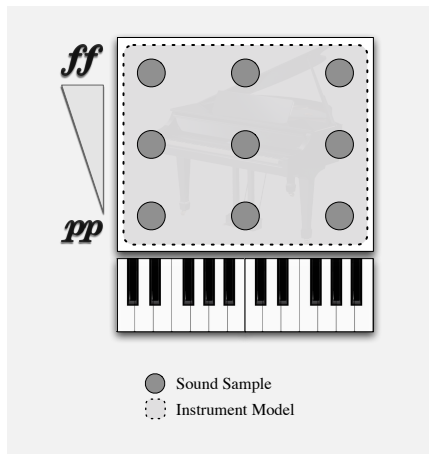


- ▶ Transformations based on real instrument characteristics
- ▶ Synthesis with continuous pitch and intensity values
- ▶ as well as arbitrary note durations

- ▶ Expressive sound synthesis with gestural control parameters

- ▶ Replacement of samples
- ▶ Creation of instrumental hybrids

Musical Sampler with an Instrument Model



- ▶ Transformations based on real instrument characteristics
- ▶ Synthesis with continuous pitch and intensity values
- ▶ as well as arbitrary note durations

- ▶ Expressive sound synthesis with gestural control parameters

- ▶ Replacement of samples
- ▶ Creation of instrumental hybrids



Introduction

Enhanced Source-Filter Model

Results

Demo

Overview



Analysis

Parameter Estimation

Flattening & Synthesis



Analysis

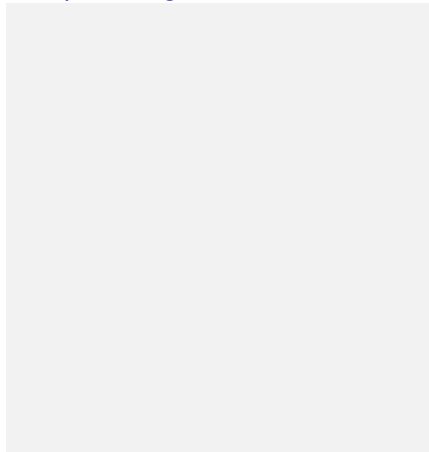
Sinusoidal/Noise segregation

- ▶ By means of partial tracking
- ▶ Residual obtained by subtraction of partials
- ▶ Smoothed cepstral envelope is obtained from residual

Gesture Analysis

- ▶ Constant pitch and global intensity
- ▶ Amplitude envelope over time

Temporal Segmentation





Analysis

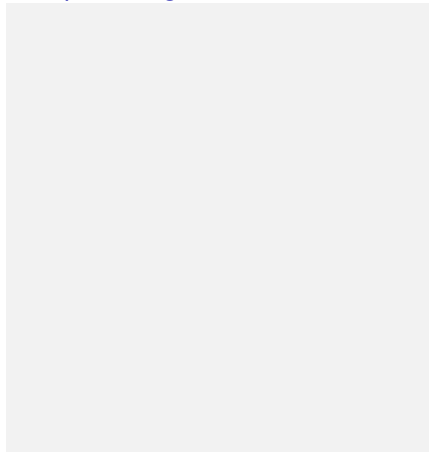
Sinusoidal/Noise segregation

- ▶ By means of partial tracking
- ▶ Residual obtained by subtraction of partials
- ▶ Smoothed cepstral envelope is obtained from residual

Gesture Analysis

- ▶ Constant pitch and global intensity
- ▶ Amplitude envelope over time

Temporal Segmentation





Analysis

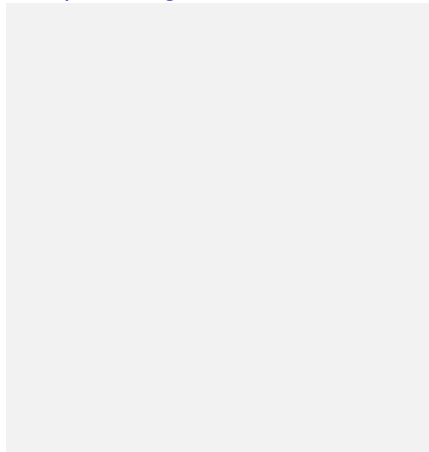
Sinusoidal/Noise segregation

- ▶ By means of partial tracking
- ▶ Residual obtained by subtraction of partials
- ▶ Smoothed cepstral envelope is obtained from residual

Gesture Analysis

- ▶ Constant pitch and global intensity
- ▶ Amplitude envelope over time

Temporal Segmentation





Analysis

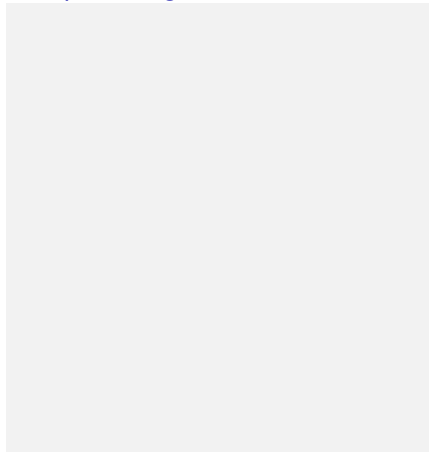
Sinusoidal/Noise segregation

- ▶ By means of partial tracking
- ▶ Residual obtained by subtraction of partials
- ▶ Smoothed cepstral envelope is obtained from residual

Gesture Analysis

- ▶ Constant pitch and global intensity
- ▶ Amplitude envelope over time

Temporal Segmentation





Analysis

Sinusoidal/Noise segregation

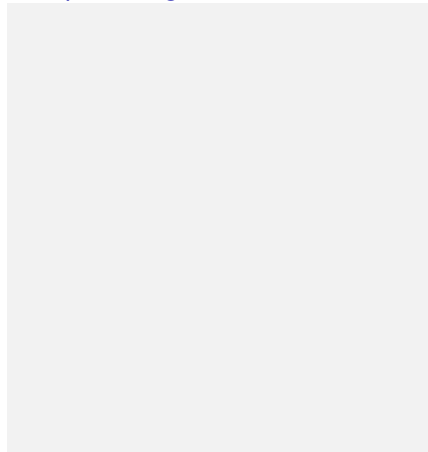
- ▶ By means of partial tracking
- ▶ Residual obtained by subtraction of partials
- ▶ Smoothed cepstral envelope is obtained from residual

$$A_k^{(\alpha)}(n), f_k^{(\alpha)}, C_l^{(\alpha)}(n)$$

Gesture Analysis

- ▶ Constant pitch and global intensity
- ▶ Amplitude envelope over time

Temporal Segmentation





Analysis

Sinusoidal/Noise segregation

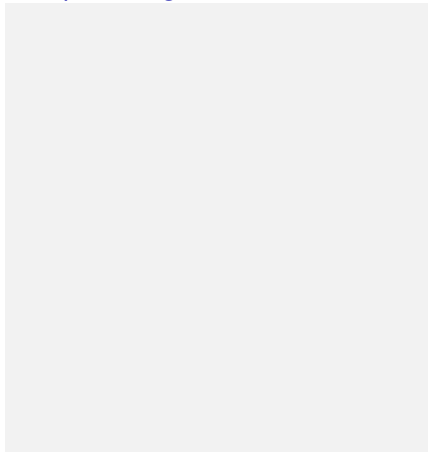
- ▶ By means of partial tracking
- ▶ Residual obtained by subtraction of partials
- ▶ Smoothed cepstral envelope is obtained from residual

$$A_k^{(\alpha)}(n), f_k^{(\alpha)}, C_l^{(\alpha)}(n)$$

Gesture Analysis

- ▶ Constant pitch and global intensity
- ▶ Amplitude envelope over time

Temporal Segmentation





Analysis

Sinusoidal/Noise segregation

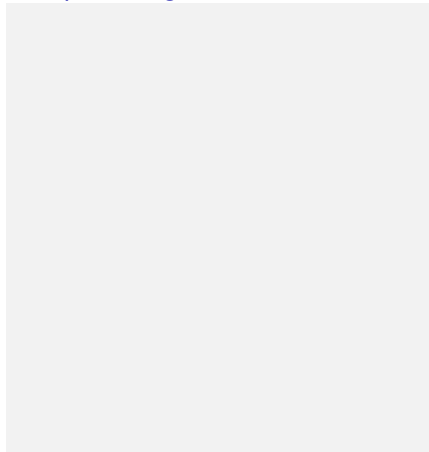
- ▶ By means of partial tracking
- ▶ Residual obtained by subtraction of partials
- ▶ Smoothed cepstral envelope is obtained from residual

$$A_k^{(\alpha)}(n), f_k^{(\alpha)}, C_l^{(\alpha)}(n)$$

Gesture Analysis

- ▶ Constant pitch and global intensity
- ▶ Amplitude envelope over time

Temporal Segmentation





Analysis

Sinusoidal/Noise segregation

- ▶ By means of partial tracking
- ▶ Residual obtained by subtraction of partials
- ▶ Smoothed cepstral envelope is obtained from residual

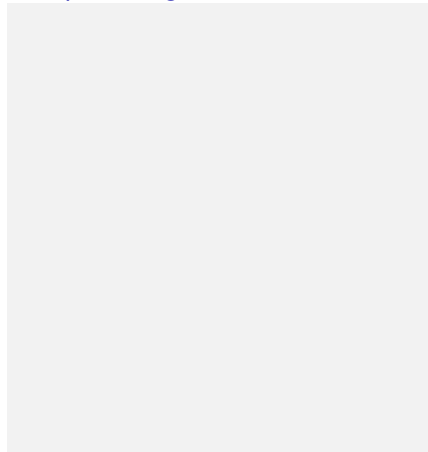
$$A_k^{(\alpha)}(n), f_k^{(\alpha)}, C_l^{(\alpha)}(n)$$

Gesture Analysis

- ▶ Constant pitch and global intensity
- ▶ Amplitude envelope over time

$$\Theta^{(\alpha)}(n) = \{I, E(n), P\}^{(\alpha)}$$

Temporal Segmentation



Analysis

Sinusoidal/Noise segregation

- ▶ By means of partial tracking
- ▶ Residual obtained by subtraction of partials
- ▶ Smoothed cepstral envelope is obtained from residual

$$A_k^{(\alpha)}(n), f_k^{(\alpha)}, C_l^{(\alpha)}(n)$$

Gesture Analysis

- ▶ Constant pitch and global intensity
- ▶ Amplitude envelope over time

$$\Theta^{(\alpha)}(n) = \{I, E(n), P\}^{(\alpha)}$$

Temporal Segmentation

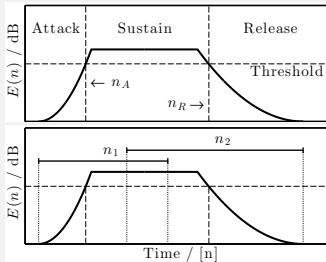


Figure: Temporal segmentation scheme for sustained (left) and impulsive (right) instrument signals.

Analysis

Sinusoidal/Noise segregation

- ▶ By means of partial tracking
- ▶ Residual obtained by subtraction of partials
- ▶ Smoothed cepstral envelope is obtained from residual

$$A_k^{(\alpha)}(n), f_k^{(\alpha)}, C_l^{(\alpha)}(n)$$

Gesture Analysis

- ▶ Constant pitch and global intensity
- ▶ Amplitude envelope over time

$$\Theta^{(\alpha)}(n) = \{I, E(n), P\}^{(\alpha)}$$

Temporal Segmentation

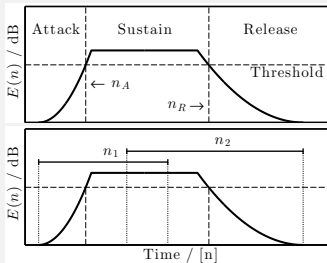


Figure: Temporal segmentation scheme for sustained (left) and impulsive (right) instrument signals.

$$n_s; s \in \{1, 2\}$$

Analysis



Harmonic signal representation

$$X_h^{(\alpha)} : \begin{cases} A_k^{(\alpha)}(n) \\ \Theta^{(\alpha)}(n) \\ f_k^{(\alpha)} \\ n_s \end{cases}$$

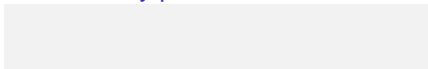
Noise signal representation

$$X_n^{(\alpha)} : \begin{cases} C_l^{(\alpha)}(n) \\ \Theta^{(\alpha)}(n) \\ n_s \end{cases}$$

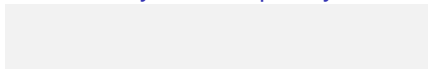


The Harmonic Model

Features by partial index k



Features by their frequency f_k





The Harmonic Model

Features by partial index k

$$S^{k,s}(I, E(n), P) = S^{k,s}(\Theta)$$

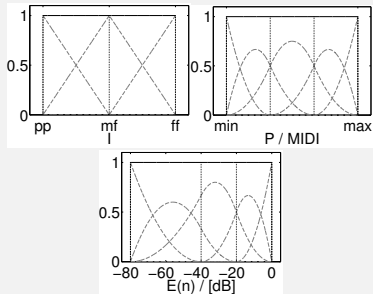
Features by their frequency f_k



The Harmonic Model

Features by partial index k

$$S^{k,s}(I, E(n), P) = S^{k,s}(\theta)$$



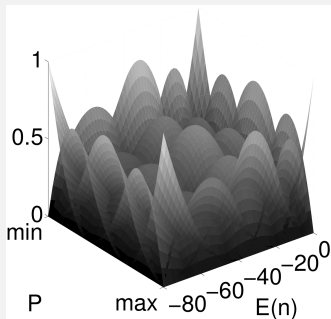
1D B-Spline models for I , $E(n)$, P

Features by their frequency f_k

The Harmonic Model

Features by partial index k

$$S^{k,s}(I, E(n), P) = S^{k,s}(\Theta)$$



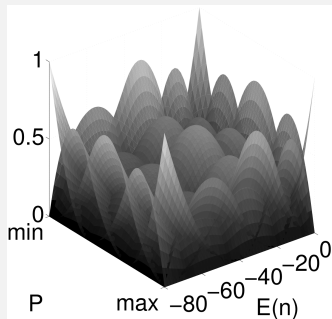
Tensor-Product B-Spline model for $E(n)$ and P

Features by their frequency f_k

The Harmonic Model

Features by partial index k

$$S^{k,s}(I, E(n), P) = S^{k,s}(\theta)$$



Tensor-Product B-Spline model for $E(n)$ and P

Features by their frequency f_k

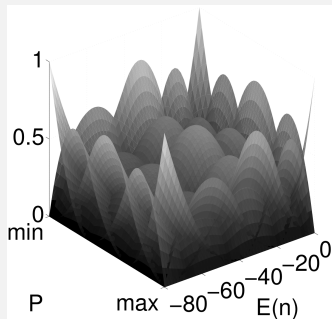
$$R(f)$$



The Harmonic Model

Features by partial index k

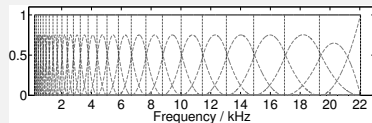
$$S^{k,s}(I, E(n), P) = S^{k,s}(\Theta)$$



Tensor-Product B-Spline model for $E(n)$ and P

Features by their frequency f_k

$$R(f)$$



B-Spline model for $R(f)$

Model Equations



Harmonic Model

$$\hat{A}_{k,s}(\Theta, f_k) = S_{k,s}(\Theta) + F(f_k)$$

Noise Model

$$\hat{C}_{l,s}(\Theta, f_k) = S_{l,s}(\Theta)$$

Model Equations



Harmonic Model

$$\hat{A}_{k,s}(\Theta, f_k) = S_{k,s}(\Theta) + F(f_k)$$

Noise Model

$$\hat{C}_{l,s}(\Theta, f_k) = S_{l,s}(\Theta)$$

Parameter Estimation



Least-Mean-Square Optimization

$$R_h^{(\alpha)} = \frac{1}{2} \sum_s \sum_{n_s} \sum_{k=1}^K \left(A_k^{(\alpha)}(n) - \hat{A}_{k,s}(\Theta^{(\alpha)}(n), f_k^{(\alpha)}) \right)^2$$

Problem Size

- ▶ The model itself is linear, closed form possible, but
- ▶ matrix sizes may exceeds several hundred TBs
- ▶ iterative approach using conjugate gradient

Parameter Estimation



Least-Mean-Square Optimization

$$R_h^{(\alpha)} = \frac{1}{2} \sum_s \sum_{n_s} \sum_{k=1}^K \left(A_k^{(\alpha)}(n) - \hat{A}_{k,s}(\Theta^{(\alpha)}(n), f_k^{(\alpha)}) \right)^2$$

Problem Size

- ▶ The model itself is linear, closed form possible, but
- ▶ matrix sizes may exceeds several hundred TBs
- ▶ iterative approach using conjugate gradient

Parameter Estimation



Least-Mean-Square Optimization

$$R_h^{(\alpha)} = \frac{1}{2} \sum_s \sum_{n_s} \sum_{k=1}^K \left(A_k^{(\alpha)}(n) - \hat{A}_{k,s}(\Theta^{(\alpha)}(n), f_k^{(\alpha)}) \right)^2$$

Problem Size

- ▶ The model itself is linear, closed form possible, but
- ▶ matrix sizes may exceeds several hundred TBs
- ▶ iterative approach using conjugate gradient

Parameter Estimation



Least-Mean-Square Optimization

$$R_h^{(\alpha)} = \frac{1}{2} \sum_s \sum_{n_s} \sum_{k=1}^K \left(A_k^{(\alpha)}(n) - \hat{A}_{k,s}(\Theta^{(\alpha)}(n), f_k^{(\alpha)}) \right)^2$$

Problem Size

- ▶ The model itself is linear, closed form possible, but
- ▶ matrix sizes may exceeds several hundred TBs
- ▶ iterative approach using conjugate gradient



A serious Problem

The Issue

- ▶ The data is extremely badly distributed

The cause

- ▶ Not all partials are present in the DB for all intensities and pitches

This results in

- ▶ Malformed model shapes in areas without any data
- ▶ Slow training speed

Our solutions

- ▶ Adding constraints to R for extrapolation into empty areas
- ▶ Preconditioning of the model parameters



A serious Problem

The Issue

- ▶ The data is extremely badly distributed

The cause

- ▶ Not all partials are present in the DB for all intensities and pitches

This results in

- ▶ Malformed model shapes in areas without any data
- ▶ Slow training speed

Our solutions

- ▶ Adding constraints to R for extrapolation into empty areas
- ▶ Preconditioning of the model parameters



A serious Problem

The Issue

- ▶ The data is extremely badly distributed

The cause

- ▶ Not all partials are present in the DB for all intensities and pitches

This results in

- ▶ Malformed model shapes in areas without any data
- ▶ Slow training speed

Our solutions

- ▶ Adding constraints to R for extrapolation into empty areas
- ▶ Preconditioning of the model parameters



A serious Problem

The Issue

- ▶ The data is extremely badly distributed

The cause

- ▶ Not all partials are present in the DB for all intensities and pitches

This results in

- ▶ Malformed model shapes in areas without any data
- ▶ Slow training speed

Our solutions

- ▶ Adding constraints to R for extrapolation into empty areas
- ▶ Preconditioning of the model parameters



A serious Problem

The Issue

- ▶ The data is extremely badly distributed

The cause

- ▶ Not all partials are present in the DB for all intensities and pitches

This results in

- ▶ Malformed model shapes in areas without any data
- ▶ Slow training speed

Our solutions

- ▶ Adding constraints to R for extrapolation into empty areas
- ▶ Preconditioning of the model parameters



A serious Problem

The Issue

- ▶ The data is extremely badly distributed

The cause

- ▶ Not all partials are present in the DB for all intensities and pitches

This results in

- ▶ Malformed model shapes in areas without any data
- ▶ Slow training speed

Our solutions

- ▶ Adding constraints to R for extrapolation into empty areas
- ▶ Preconditioning of the model parameters



A serious Problem

The Issue

- ▶ The data is extremely badly distributed

The cause

- ▶ Not all partials are present in the DB for all intensities and pitches

This results in

- ▶ Malformed model shapes in areas without any data
- ▶ Slow training speed

Our solutions

- ▶ Adding constraints to R for extrapolation into empty areas
- ▶ Preconditioning of the model parameters



A serious Problem

The Issue

- ▶ The data is extremely badly distributed

The cause

- ▶ Not all partials are present in the DB for all intensities and pitches

This results in

- ▶ Malformed model shapes in areas without any data
- ▶ Slow training speed

Our solutions

- ▶ Adding constraints to R for extrapolation into empty areas
- ▶ Preconditioning of the model parameters

Flattening & Synthesis



Envelope Generation

- ▶ Cepstral coefficients of noise model can directly transformed into spectral envelope
- ▶ Partial amplitudes from harmonic model are transformed using cepstral smoothing method.

Flattening

- ▶ Preprocessing of the whole data by applying inverted estimated envelopes from ALL harmonic and noise data
- ▶ Storing them as source data for the synthesizer

Synthesis

- ▶ Generate model envelopes using modified Gestural Control Parameters
- ▶ Filter source harmonic and noise data and downmix

Flattening & Synthesis



Envelope Generation

- ▶ Cepstral coefficients of noise model can directly transformed into spectral envelope
- ▶ Partial amplitudes from harmonic model are transformed using cepstral smoothing method.

Flattening

- ▶ Preprocessing of the whole data by applying inverted estimated envelopes from ALL harmonic and noise data
- ▶ Storing them as source data for the synthesizer

Synthesis

- ▶ Generate model envelopes using modified Gestural Control Parameters
- ▶ Filter source harmonic and noise data and downmix

Flattening & Synthesis



Envelope Generation

- ▶ Cepstral coefficients of noise model can directly transformed into spectral envelope
- ▶ Partial amplitudes from harmonic model are transformed using cepstral smoothing method.

Flattening

- ▶ Preprocessing of the whole data by applying inverted estimated envelopes from ALL harmonic and noise data
- ▶ Storing them as source data for the synthesizer

Synthesis

- ▶ Generate model envelopes using modified Gestural Control Parameters
- ▶ Filter source harmonic and noise data and downmix

Flattening & Synthesis



Envelope Generation

- ▶ Cepstral coefficients of noise model can directly transformed into spectral envelope
- ▶ Partial amplitudes from harmonic model are transformed using cepstral smoothing method.

Flattening

- ▶ Preprocessing of the whole data by applying inverted estimated envelopes from ALL harmonic and noise data
- ▶ Storing them as source data for the synthesizer

Synthesis

- ▶ Generate model envelopes using modified Gestural Control Parameters
- ▶ Filter source harmonic and noise data and downmix

Flattening & Synthesis



Envelope Generation

- ▶ Cepstral coefficients of noise model can directly transformed into spectral envelope
- ▶ Partial amplitudes from harmonic model are transformed using cepstral smoothing method.

Flattening

- ▶ Preprocessing of the whole data by applying inverted estimated envelopes from ALL harmonic and noise data
- ▶ Storing them as source data for the synthesizer

Synthesis

- ▶ Generate model envelopes using modified Gestural Control Parameters
- ▶ Filter source harmonic and noise data and downmix

Flattening & Synthesis



Envelope Generation

- ▶ Cepstral coefficients of noise model can directly transformed into spectral envelope
- ▶ Partial amplitudes from harmonic model are transformed using cepstral smoothing method.

Flattening

- ▶ Preprocessing of the whole data by applying inverted estimated envelopes from ALL harmonic and noise data
- ▶ Storing them as source data for the synthesizer

Synthesis

- ▶ Generate model envelopes using modified Gestural Control Parameters
- ▶ Filter source harmonic and noise data and downmix

Flattening & Synthesis



Envelope Generation

- ▶ Cepstral coefficients of noise model can directly transformed into spectral envelope
- ▶ Partial amplitudes from harmonic model are transformed using cepstral smoothing method.

Flattening

- ▶ Preprocessing of the whole data by applying inverted estimated envelopes from ALL harmonic and noise data
- ▶ Storing them as source data for the synthesizer

Synthesis

- ▶ Generate model envelopes using modified Gestural Control Parameters
- ▶ Filter source harmonic and noise data and downmix

Flattening & Synthesis



Envelope Generation

- ▶ Cepstral coefficients of noise model can directly transformed into spectral envelope
- ▶ Partial amplitudes from harmonic model are transformed using cepstral smoothing method.

Flattening

- ▶ Preprocessing of the whole data by applying inverted estimated envelopes from ALL harmonic and noise data
- ▶ Storing them as source data for the synthesizer

Synthesis

- ▶ Generate model envelopes using modified Gestural Control Parameters
- ▶ Filter source harmonic and noise data and downmix

Flattening & Synthesis



Envelope Generation

- ▶ Cepstral coefficients of noise model can directly transformed into spectral envelope
- ▶ Partial amplitudes from harmonic model are transformed using cepstral smoothing method.

Flattening

- ▶ Preprocessing of the whole data by applying inverted estimated envelopes from ALL harmonic and noise data
- ▶ Storing them as source data for the synthesizer

Synthesis

- ▶ Generate model envelopes using modified Gestural Control Parameters
- ▶ Filter source harmonic and noise data and downmix



Introduction

Enhanced Source-Filter Model

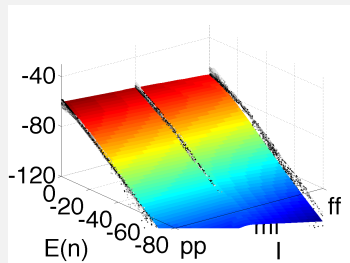
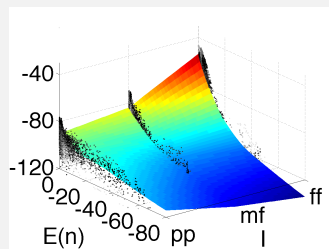
Results

Demo

Results



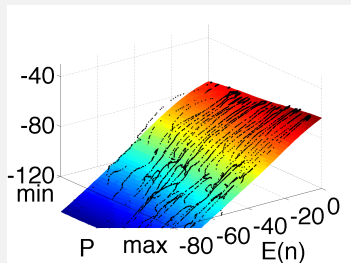
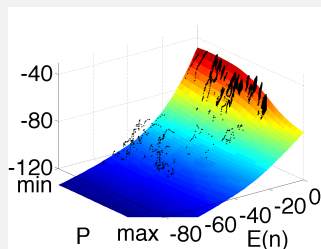
Trumpet

Estimated model for $S^{1,2}(\Theta)$ Estimated model for $S^{30,2}(\Theta)$

Results



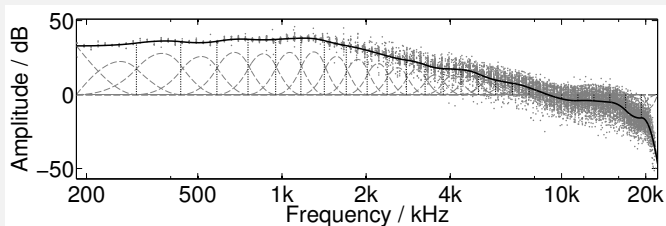
Trumpet

Estimated model for $S^{1,2}(\Theta)$ Estimated model for $S^{30,2}(\Theta)$

Results



Trumpet

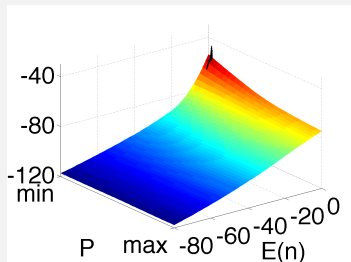
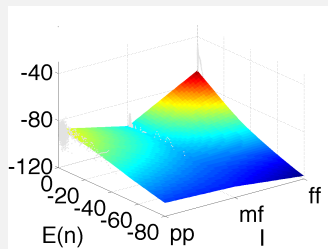


Estimated model for $R(f)$

Results



Trumpet

Estimated model for $S^{100,2}(\theta)$ Estimated model for $S^{100,2}(\theta)$

Demo



Demo

Thanks to Jean-Philippe Lambert and Norbert Schnell

Fin



Thanks for listening

Questions?