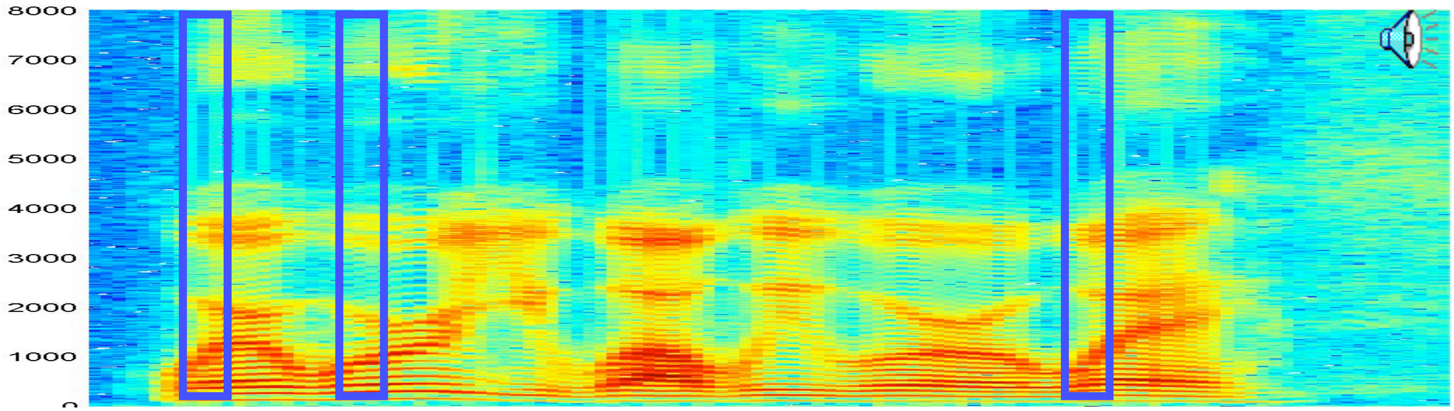
Shift- and Transform-Invariant Representations
Denoising Speech Signals

Class 18. 22 Oct 2009

Summary So Far

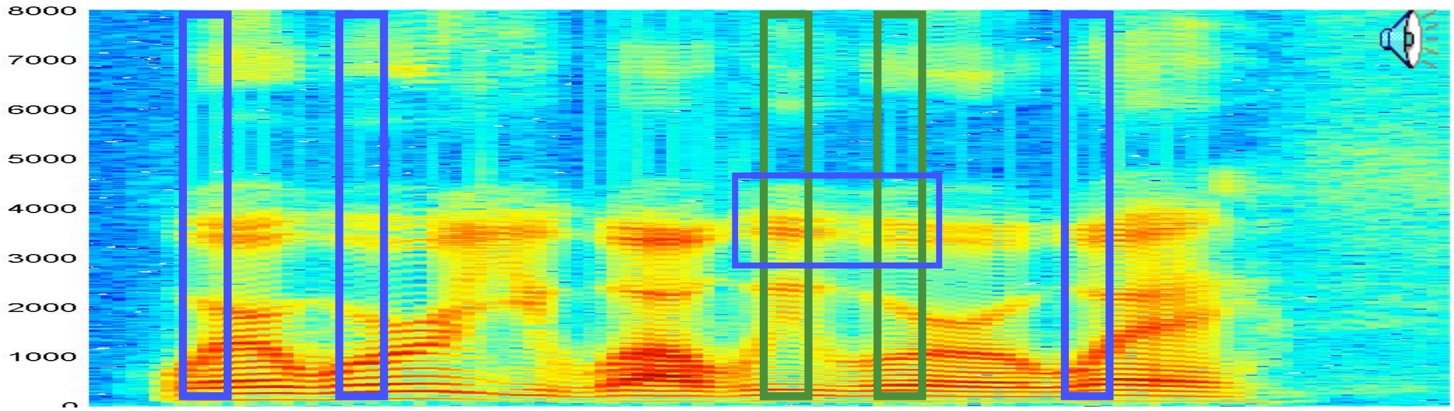
- PLCA:
  - The basic mixture-multinomial model for audio (and other data)
- Sparse Decomposition:
  - The notion of sparsity and how it can be imposed on learning
- Sparse Overcomplete Decomposition:
  - The notion of *overcomplete* basis set
- Example-based representations
  - Using the training data itself as our representation

Next up: Shift/Transform Invariance



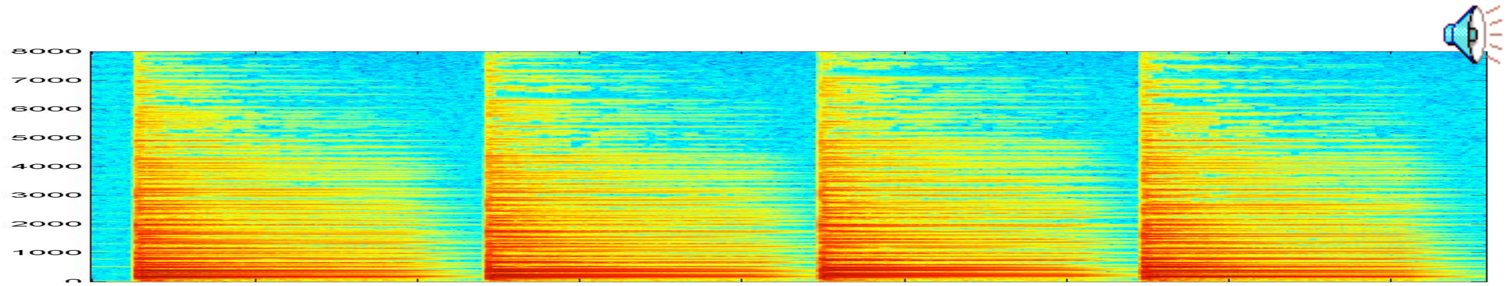
- Sometimes the "typical" structures that compose a sound are wider than one spectral frame
  - E.g. in the above example we note multiple examples of a pattern that spans several frames

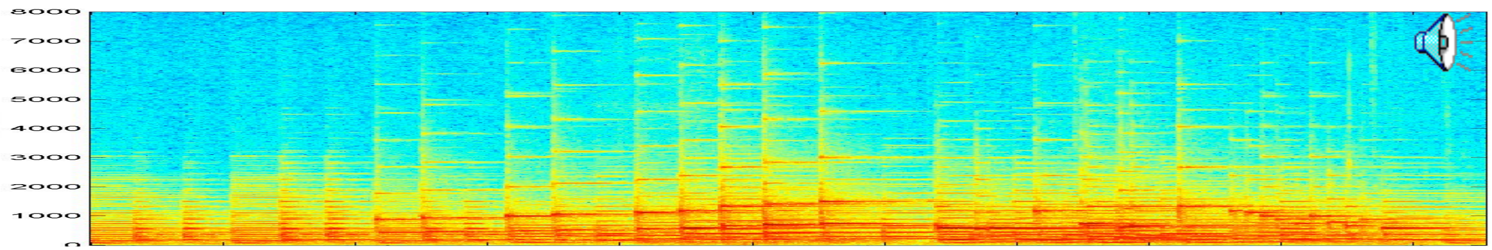
Next up: Shift/Transform Invariance



- Sometimes the “typical” structures that compose a sound are wider than one spectral frame
  - E.g. in the above example we note multiple examples of a pattern that spans several frames
- Multiframe patterns may also be local in frequency
  - E.g. the two green patches are similar only in the region enclosed by the blue box

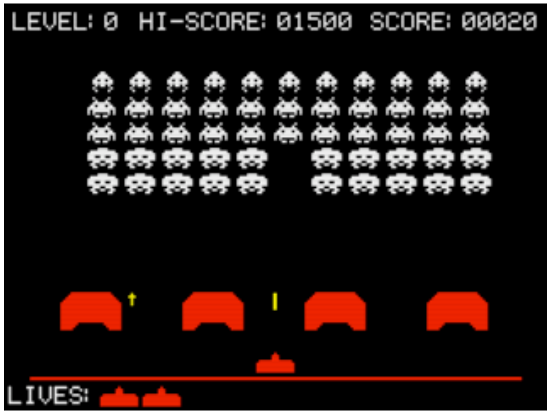
Patches are more representative than frames

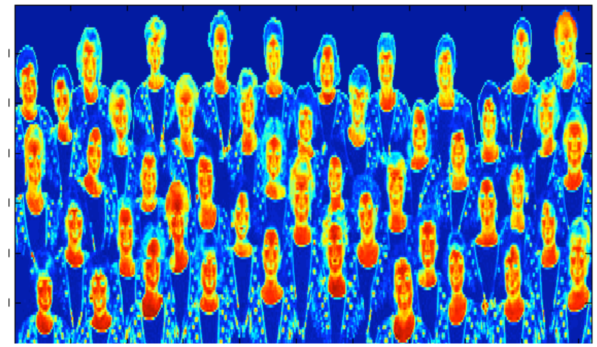




- Four bars from a music example
- The spectral patterns are actually patches
  - Not all frequencies fall off in time at the same rate
- The basic unit is a spectral patch, not a spectrum

Images: Patches often form the image



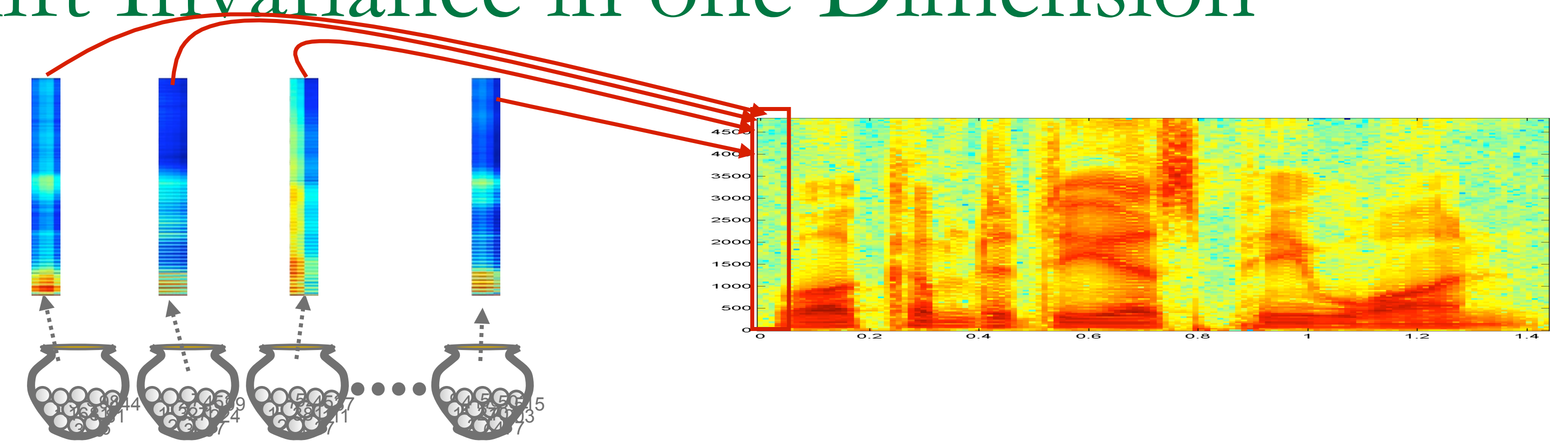


- A typical image component may be viewed as a patch
  - The alien invaders
  - Face like patches
  - A car like patch
    - overlaid on itself many times..

Shift-invariant modelling

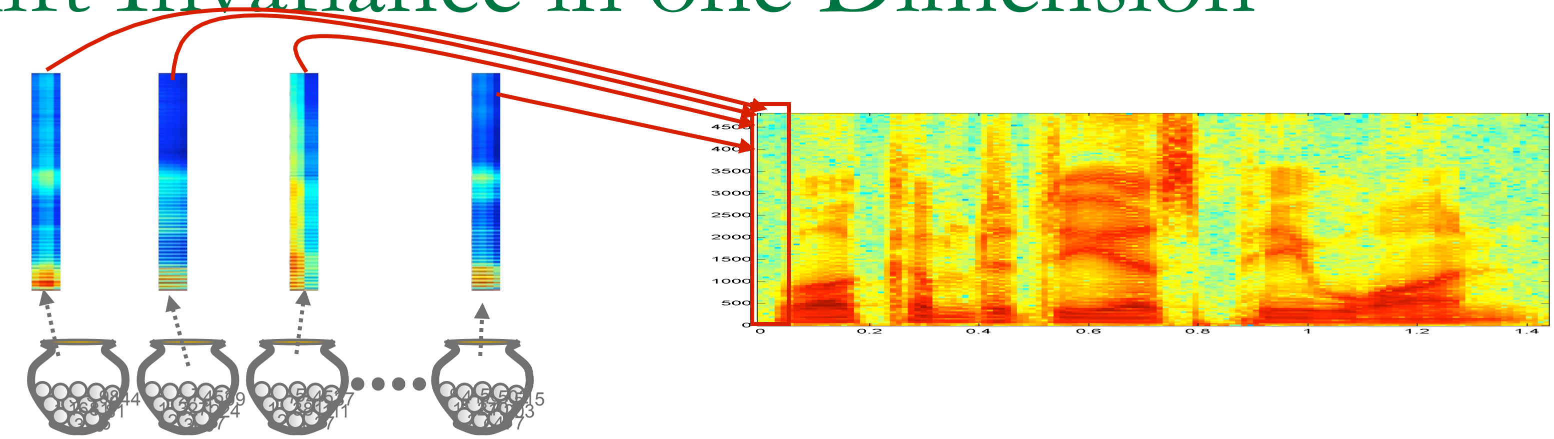
- A shift-invariant model permits individual bases to be *patches*
- Each patch composes the entire image.
- The data is a sum of the compositions from individual patches

Shift Invariance in one Dimension



- Our bases are now "patches"
  - Typical *spectro-temporal* structures
- The urns now represent patches
  - Each draw results in a (t,f) pair, rather than only f
  - *Also associated with each urn: A shift probability distribution P(T|z)*
- The overall drawing process is slightly more complex
- Repeat the following process:
  - Select an urn Z with a probability P(Z)
  - Draw a value T from P(t|Z)
  - Draw (t,f) pair from the urn
  - Add to the histogram at (t+T, f)

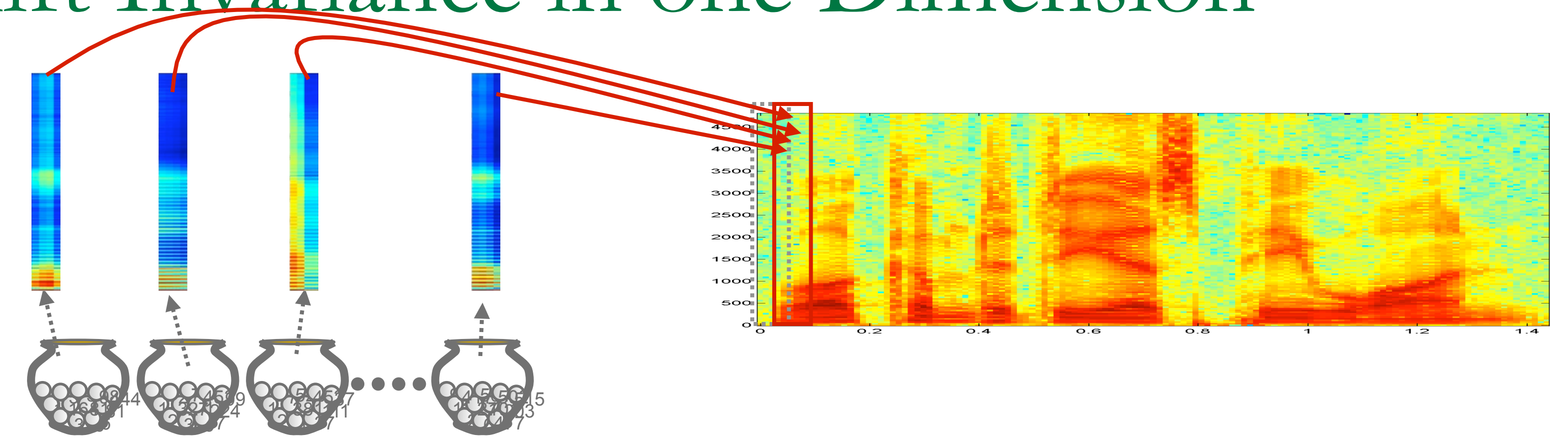
Shift Invariance in one Dimension



- The process is *shift-invariant* because the probability of drawing a shift P(T|Z) does not affect the probability of selecting urn Z

- Every location in the spectrogram has contributions from every urn patch

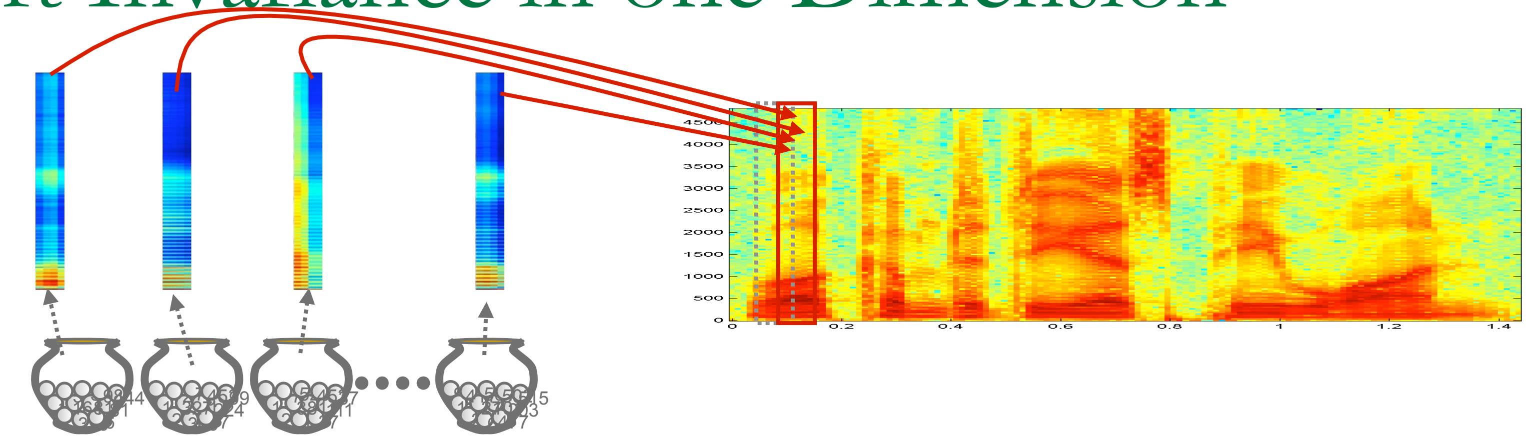
Shift Invariance in one Dimension



- The process is *shift-invariant* because the probability of drawing a shift P(T|Z) does not affect the probability of selecting urn Z

- Every location in the spectrogram has contributions from every urn patch

Shift Invariance in one Dimension



- The process is *shift-invariant* because the probability of drawing a shift P(T|Z) does not affect the probability of selecting urn Z

- Every location in the spectrogram has contributions from every urn patch

Probability of drawing a particular (t,f) combination

$$P(t,f) = \sum_z P(z) \sum_\tau P(\tau \mid z) P(t-\tau, f \mid z)$$

- The parameters of the model:
  - P(t,f|z) – the urns
  - P(T|z) – the *urn-specific* shift distribution
  - P(z) – probability of selecting an urn
- The ways in which (t,f) can be drawn:
  - Select any urn z
  - Draw T from the urn-specific shift distribution
  - Draw (t-T,f) from the urn
- The actual probability sums this over all shifts and urns

Learning the Model

- The parameters of the model are learned analogously to the manner in which mixture multinomials are learned

- Given observation of (t,f), it we knew which urn it came from and the shift, we could compute all probabilities by counting!
  - If shift is T and urn is Z
    - Count(Z) = Count(Z) + 1
    - For shift probability: Count(T|Z) = Count(T|Z)+1
    - For urn: Count(t-T,f | Z) = Count(t-T,f|Z) + 1
      - Since the value drawn from the urn was t-T,f

  - After all observations are counted:
    - Normalize Count(Z) to get P(Z)
    - Normalize Count(T|Z) to get P(T|Z)
    - Normalize Count(t,f|Z) to get P(t,f|Z)

- Problem: When learning the urns and shift distributions from a histogram, the urn (Z) and shift (T) for any draw of (t,f) is not known
  - These are unseen variables

Learning the Model

- Urn Z and shift T are unknown
  - So (t,f) contributes partial counts to *every* value of T and Z
  - Contributions are proportional to the *a posteriori* probability of Z and T,Z

$$P(t,f,Z) = P(Z)\sum_{T} P(T\,|\,Z)P(t-T,f\,|\,Z) \qquad P(T,t,f\,|\,Z) = P(T\,|\,Z)P(t-T,f\,|\,Z)$$

$$P(Z\,|\,t,f) = \frac{P(t,f,Z)}{\sum_{Z'} P(t,f,Z')} \qquad P(T\,|\,Z,t,f) = \frac{P(T,t-T,f\,|\,Z)}{\sum_{T'} P(T',t-T',f\,|\,Z)}$$

- Each observation of (t,f)
  - P(z|t,f) to the count of the total number of draws from the urn
    - Count(Z) = Count(Z) + P(z | t,f)
  - P(z|t,f)P(T | z,t,f) to the count of the shift T for the shift distribution
    - Count(T | Z) = Count(T | Z) + P(z|t,f)P(T | Z, t, f)
  - P(z|t,f)P(T | z,t,f) to the count of (t-T, f) for the urn
    - Count(t-T,f | Z) = Count(t-T,f | Z) + P(z|t,f)P(T | z,t,f)

Shift invariant model: Update Rules

- Given data (spectrogram) S(t,f)
- Initialize P(Z), P(T|Z), P(t,f | Z)
- Iterate

$$P(t,f,Z) = P(Z)\sum_{T} P(T \mid Z)P(t-T,f \mid Z) \qquad P(T,t,f \mid Z) = P(T \mid Z)P(t-T,f \mid Z)$$

$$P(Z \mid t,f) = \frac{P(t,f,Z)}{\sum_{Z'} P(t,f,Z')} \qquad P(T \mid Z,t,f) = \frac{P(T,t-T,f \mid Z)}{\sum_{T'} P(T',t-T',f \mid Z)}$$

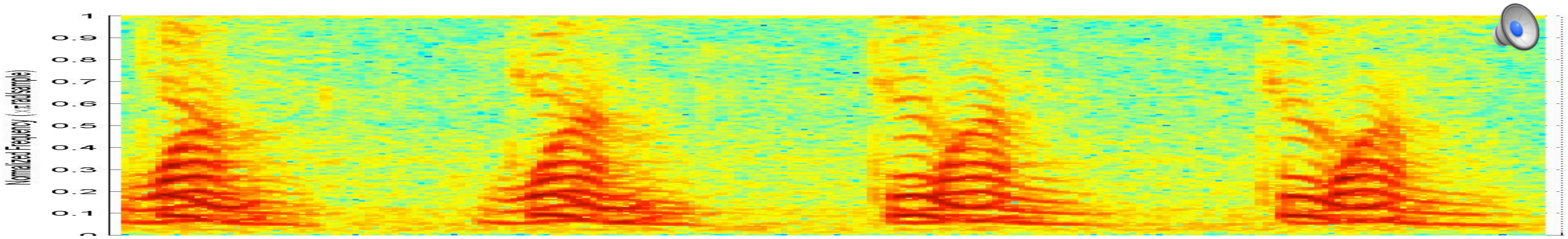
$$P(Z) = \frac{\sum_{t}\sum_{f} P(Z \mid t,f)S(t,f)}{\sum_{Z'}\sum_{t}\sum_{f} P(Z' \mid t,f)S(t,f)} \qquad P(T \mid Z) = \frac{\sum_{t}\sum_{f} P(Z \mid t,f)P(T \mid Z,t,f)S(t,f)}{\sum_{T'}\sum_{t}\sum_{f} P(Z \mid t,f)P(T' \mid Z,t,f)S(t,f)}$$

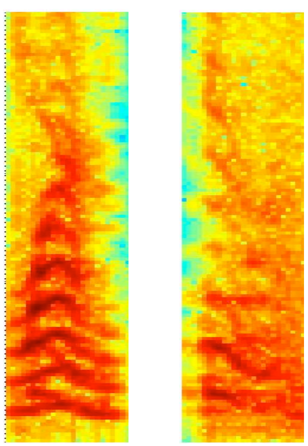
$$P(t,f \mid Z) = \frac{\sum_{T} P(Z \mid T,f)P(T-t \mid Z,T,f)S(T,f)}{\sum_{t'}\sum_{T} P(Z \mid T,f)P(T-t' \mid Z,T,f)S(T,f)}$$

Shift-invariance in one time: example

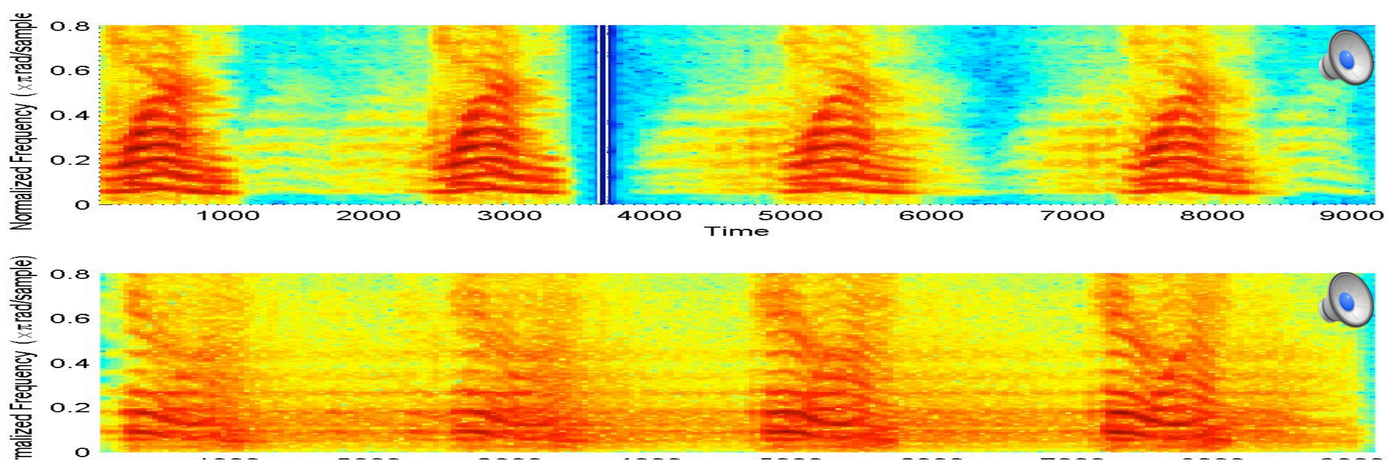
- An Example: Two distinct sounds occuring with different repetition rates within a signal
  - Modelled as being composed from two time-frequency bases
  - NOTE: Width of patches must be specified

INPUT SPECTROGRAM



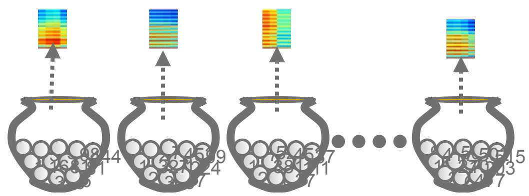


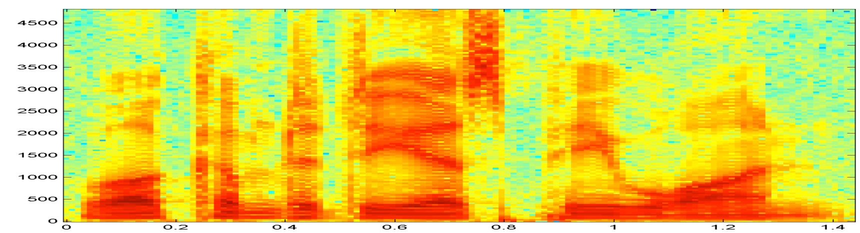
Discovered time-frequency "patch" bases (urns)



Contribution of individual bases to the recording

Shift Invariance in Two Dimensions





- We now have urn-specific shifts along both T and F
- The Drawing Process
  - Select an urn Z with a probability P(Z)
  - Draw SHIFT values (T,F) from $P_s(T,F|Z)$
  - Draw (t,f) pair from the urn
  - Add to the histogram at (t+T, f+F)
- This is a two-dimensional shift-invariant model
  - We have shifts in both time and frequency
    - Or, more generically, along both axes

Learning the Model

- Learning is analogous to the 1-D case

- Given observation of (t,f), it we knew which urn it came from and the shift, we could compute all probabilities by counting!
  - If shift is T,F and urn is Z
    - Count(Z) = Count(Z) + 1
    - For shift probability: ShiftCount(T,F|Z) = ShiftCount(T,F|Z)+1
    - For urn: Count(t-T,f-F | Z) = Count(t-T,f-F|Z) + 1
      - Since the value drawn from the urn was t-T,f-F

  - After all observations are counted:
    - Normalize Count(Z) to get P(Z)
    - Normalize ShiftCount(T,F|Z) to get P_s(T,F|Z)
    - Normalize Count(t,f|Z) to get P(t,f|Z)

- Problem: Shift and Urn are unknown

Learning the Model

- Urn Z and shift T,F are unknown
  - So (t,f) contributes partial counts to *every* value of T,F and Z
  - Contributions are proportional to the *a posteriori* probability of Z and T,F|Z

$$P(t,f,Z) = P(Z)\sum_{T,F} P(T,F \mid Z)P(t-T,f-F \mid Z)$$

$$P(T,F,t,f \mid Z) = P(T,F \mid Z)P(t-T,f-F \mid Z)$$

$$P(Z \mid t,f) = \frac{P(t,f,Z)}{\sum_{Z'} P(t,f,Z')}$$

$$P(T,F \mid Z,t,f) = \frac{P(T,F,t-T,f-F \mid Z)}{\sum_{T',F'} P(T',F',t-T',f-F' \mid Z)}$$

- Each observation of (t,f)
  - P(z|t,f) to the count of the total number of draws from the urn
    - Count(Z) = Count(Z) + P(z | t,f)
  - P(z|t,f)P(T,F | z,t,f) to the count of the shift T,F for the shift distribution
    - ShiftCount(T,F | Z) = ShiftCount(T,F | Z) + P(z|t,f)P(T | Z, t, f)
  - P(T | z,t,f) to the count of (t-T, f-F) for the urn
    - Count(t-T,f-F | Z) = Count(t-T,f-F | Z) + P(z|t,f)P(t-T,f-F | z,t,f)

Shift invariant model: Update Rules

- Given data (spectrogram) S(t,f)
- Initialize P(Z), P_s(T,F|Z), P(t,f | Z)
- Iterate

$$P(t,f,Z) = P(Z)\sum_{T,F} P(T,F\mid Z)P(t-T,f-F\mid Z) \qquad P(T,F,t,f\mid Z) = P(T,F\mid Z)P(t-T,f-F\mid Z)$$

$$P(Z\mid t,f) = \frac{P(t,f,Z)}{\sum_{Z'} P(t,f,Z')} \qquad P(T,F\mid Z,t,f) = \frac{P(T,F,t-T,f-F\mid Z)}{\sum_{T',F'} P(T',F',t-T',f-F'\mid Z)}$$

$$P(Z) = \frac{\sum_t \sum_f P(Z\mid t,f)S(t,f)}{\sum_{Z'}\sum_t \sum_f P(Z'\mid t,f)S(t,f)} \qquad P(T,F\mid Z) = \frac{\sum_t \sum_f P(Z\mid t,f)P(T,F\mid Z,t,f)S(t,f)}{\sum_{T'}\sum_{F'}\sum_t \sum_f P(Z\mid t,f)P(T',F'\mid Z,t,f)S(t,f)}$$

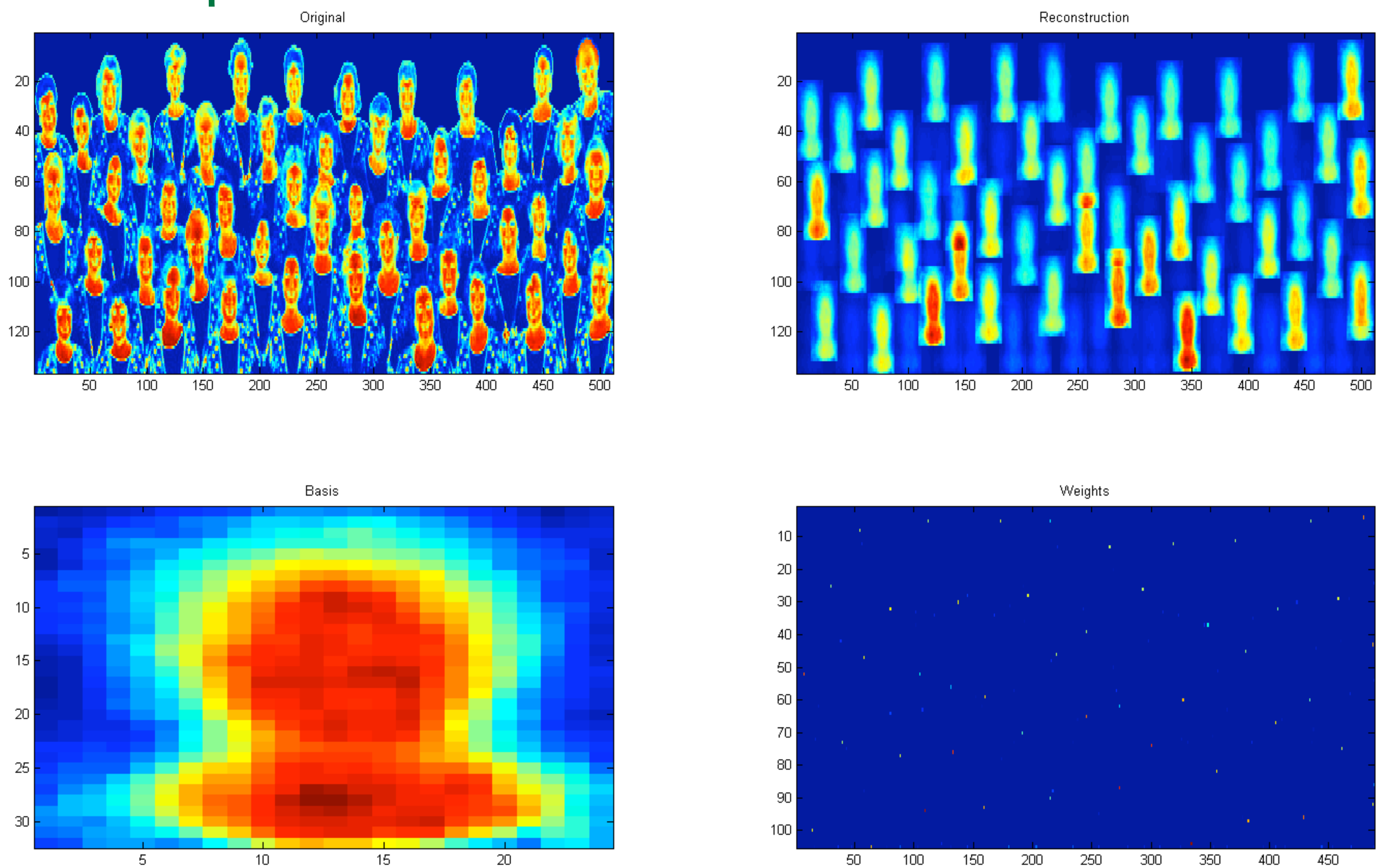
$$P(t,f\mid Z) = \frac{\sum_{T,F} P(Z\mid T,F)P(T-t,F-f\mid Z,T,F)S(T,F)}{\sum_{t',f'}\sum_{T,F} P(Z\mid T,F)P(T-t',F-f'\mid Z,T,F)S(T,F)}$$

2D Shift Invariance: The problem of indeterminacy

- P(t,f|Z) and $P_s(T,F|Z)$ are analogous
  - Difficult to specify which will be the "urn" and which the "shift"

- Additional constraints required to ensure that one of them is clearly the shift and the other the urn
- Typical solution: Enforce sparsity on $P_s(T,F|Z)$
  - The patch represented by the urn occurs only in a few locations in the data

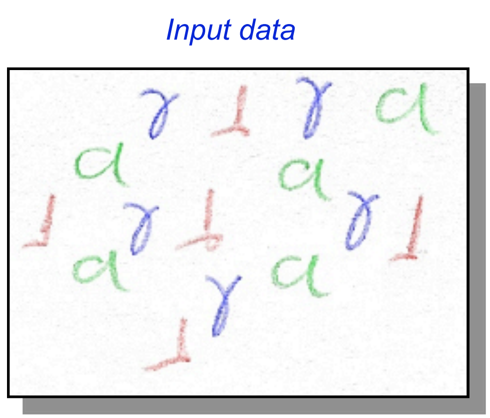
Example: 2-D shift invariance

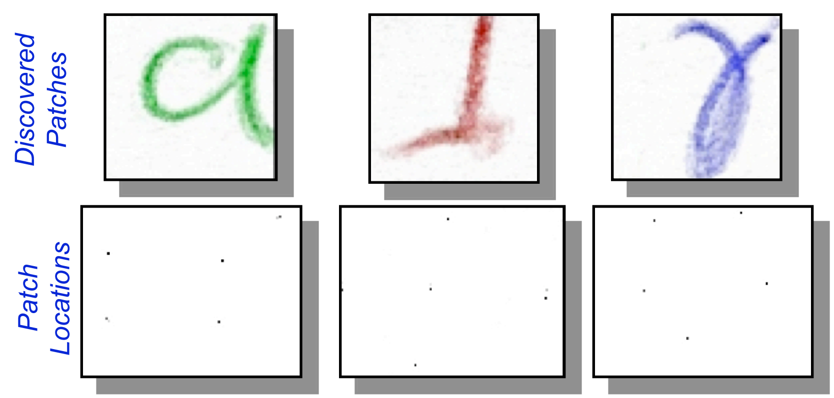


- Only one "patch" used to model the image (i.e. a single urn)
  - The learnt urn is an "average" face, the learned shifts show the locations of faces

Example: 2-D shift invarince

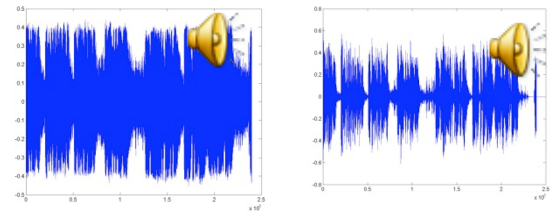
- The original figure has multiple handwritten renderings of three characters
  - In different colours
- The algorithm learns the three characters and identifies their locations in the figure



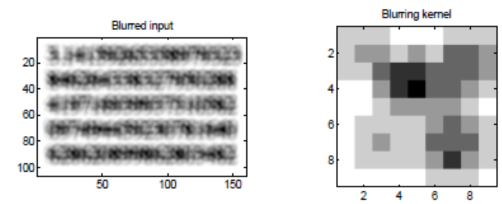


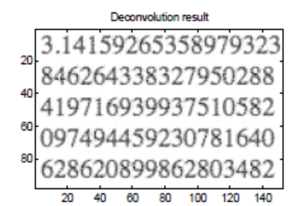
Shift-Invariant Decomposition – Uses

- Signal separation
  - The arithmetic is the same as before
  - Learn shift-invariant bases for each source
  - Use these to separate signals



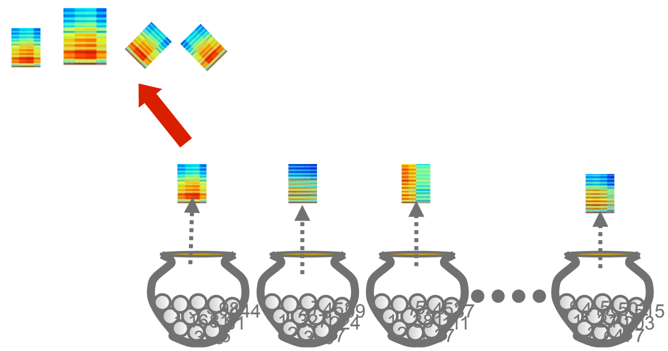
- Dereverberation
  - The spectrogram of the reverberant signal is simply the sum several shifted copies of the spectrogram of the original signal
    - 1-D shift invariance





- Image Deblurring
  - The blurred image is the sum of several shifted copies of the clean image
    - 2-D shift invariance

Beyond shift-invariance: transform invariance



- The draws from the urns may not only be shifted, but also transformed
- The arithmetic remains very similar to the shift-invariant model
  - We must now impose one of an enumerated set of transforms to (t,f), after shifting them by (T,F)
  - In the estimation, the precise transform applied is an unseen variable

Transform invariance: Generation

- The set of transforms is enumerable
  - E.g. scaling by 0.9, scaling by 1.1, rotation right by 90degrees, rotation left by 90 degrees, rotation by 180 degrees, reflection
  - Transformations can be chosen by draws from a distribution over transforms
    - E.g. P(rotation by 90 degrees) = 0.2..
    - Distributions are URN SPECIFIC

- The drawing process:
  - Select an urn Z (patch)
  - Select a shift (T,F) from P_s(T, F| Z)
  - Select a transform from P(txfm | Z)
  - Select a (t,f) pair from P(t,f | Z)
  - *Transform* (t,f) to txfm(t,f)
  - Increment the histogram at txfm(t,f) + (T,F)

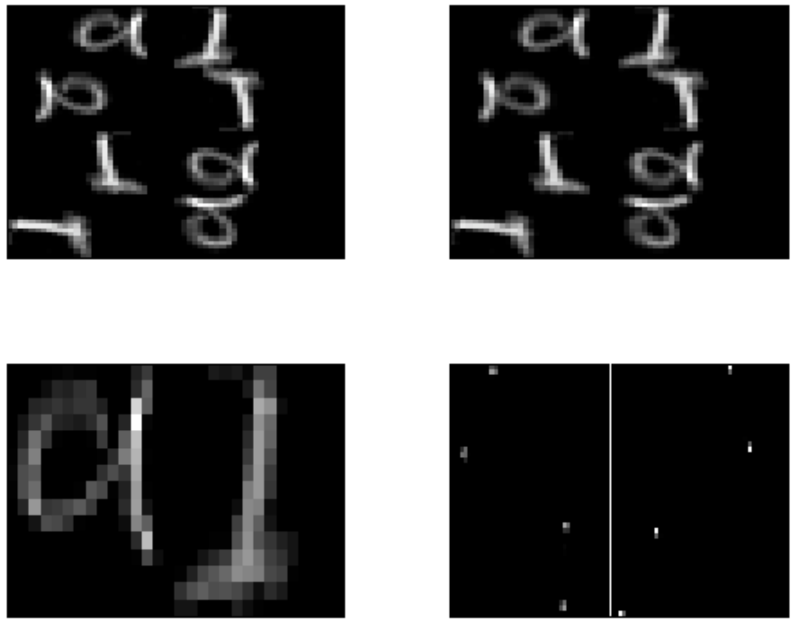
Transform invariance

- The learning algorithm must now estimate
  - P(Z) – probability of selecting urn/patch in any draw
  - P(t,f|Z) – the urns / patches
  - P(txfm | Z) – the urn specific distribution over transforms
  - P_s(T,F|Z) – the urn-specific shift distribution

- Essentially determines what the basic shapes are, where they occur in the data and how they are transformed
- The mathematics for learning are similar to the maths for shift invariance
  - With the addition that each instance of a draw must be fractured into urns, shifts AND transforms

- Details of learning are left as an exercise
  - Alternately, refer to Madhusudana Shashanka's PhD thesis at BU

Example: Transform Invariance



- Top left: Original figure
- Bottom left – the two bases discovered
- Bottom right –
  - Left panel, positions of “a”
  - Right panel, positions of “l”
- Top right: estimated distribution underlying original figure

Transform invariance: model limitations and extensions

- The current model only allows *one* transform to be applied at any draw
  - E.g. a basis may be rotated or scaled, but not scaled *and* rotated
- An obvious extension is to permit combinations of transformations
  - Model must be extended to draw the combination from some distribution
- Data dimensionality: All examples so far assume only *two* dimensions (e.g. in spectrogram or image)
- The models are trivially extended to higher-dimensional data

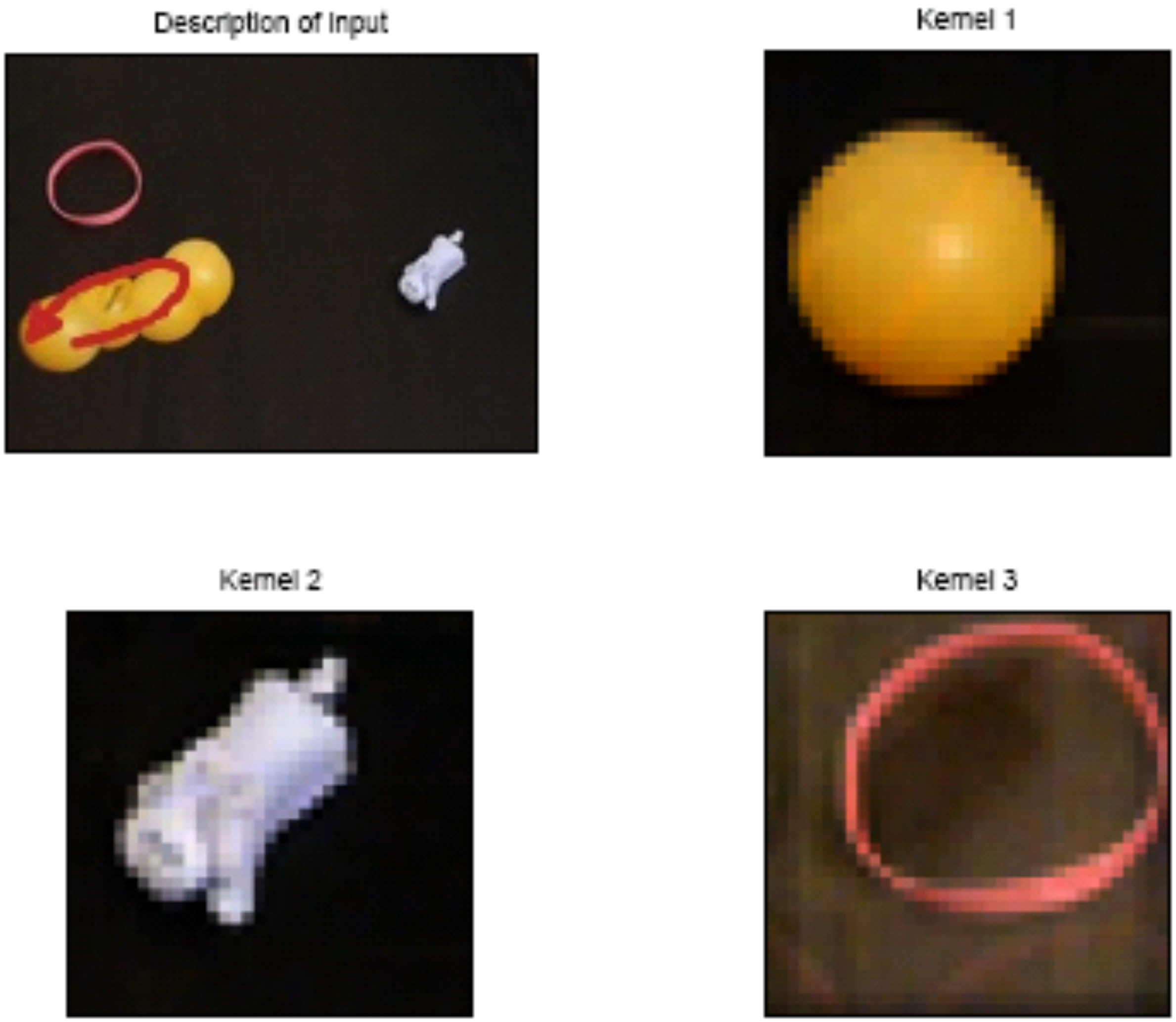
Transform Invariance: Uses and Limitations

- Not very useful to analyze audio
- May be used to analyze images and video

- Main restriction: Computational complexity
  - Requires unreasonable amounts of memory and CPU
  - Efficient implementation an open issue

Example: Higher dimensional data

- Video example



Summary

- Shift invariance
  - Multinomial bases can be “patches”
    - Representing time-frequency events in audio or other larger patterns in images

- Transform invariance
  - The patches may further be transformed to compose an image
    - Not useful for audio

De-noising Audio Signals

De-noising

- Multifaceted problem
  - Removal of unwanted artifacts
  - Clicks, hiss, warps, interfering sounds, …
- For now
  - Constant noise removal
    - Wiener filters, spectral/power subtraction
  - Click detection and restoration
    - AR models for abnormality detection
    - AR models for making up missing data

The problem with audio recordings

- Recordings are inherently messy!!
- Recordings capture room resonances, air conditioners, street ambience, etc …
  - Resulting in low frequency rumbling sounds (the signature quality of a low-budget recording!)
- Magnetic recording media get demagnetized
  - Results in high frequency hissing sounds (old tapes)
- Mechanical recording media are littered with debris
  - Results in clicking and crackling sounds (ancient vinyl disks, optical film soundtracks)
- Digital media feature sample drop-outs
  - Results in gaps in audio which when short are perceived as clicks, otherwise it is an audible gap (damaged CDs, poor internet streaming, bad bluetooth headsets)

Restoration of audio

- People don’t like noisy recordings!!
  - There is a need for audio restoration work
- Early restoration work was an art form
  - Experienced engineers would design filters to best cover defects, cut and splice tapes to remove unwanted parts, etc.
  - Results were marginally acceptable
- Recent restoration work is a science
  - Extensive use of signal processing and machine learning
  - Results are quite impressive!

Audio Restoration I
Constant noise removal

- Noise is often inherent in a recording or slowly creeps in the recording media

- Hiss, rumbling, ambience, …
- Approach
  - Figure out noise characteristics
  - Spectral processing to make up for noise

Describing additive noise

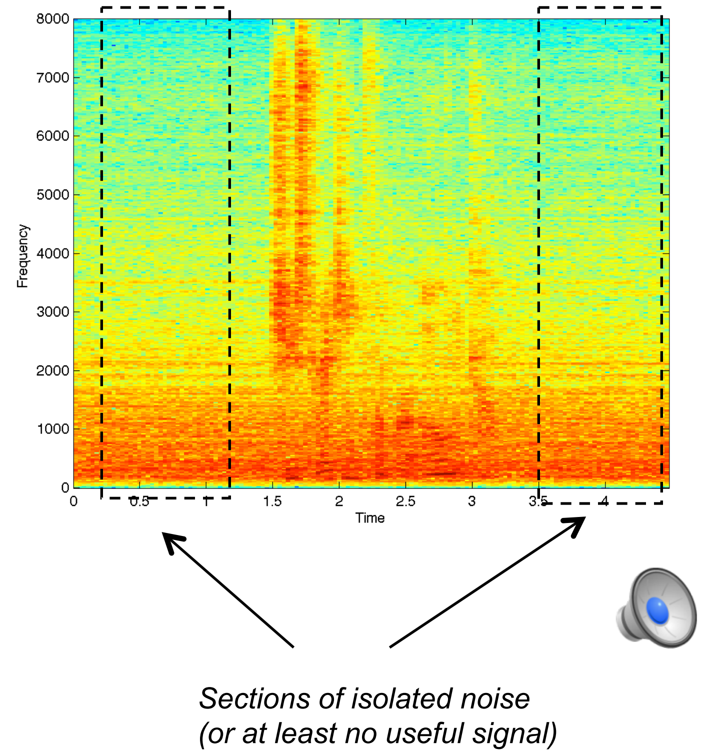
- Assume additive noise

  $x(t) = s(t) + n(t)$
- In the frequency domain

  $$X(t,f) = S(t,f) + N(t,f)$$
- Find the spots where we have only isolated noise
  - Average them and get noise spectrum

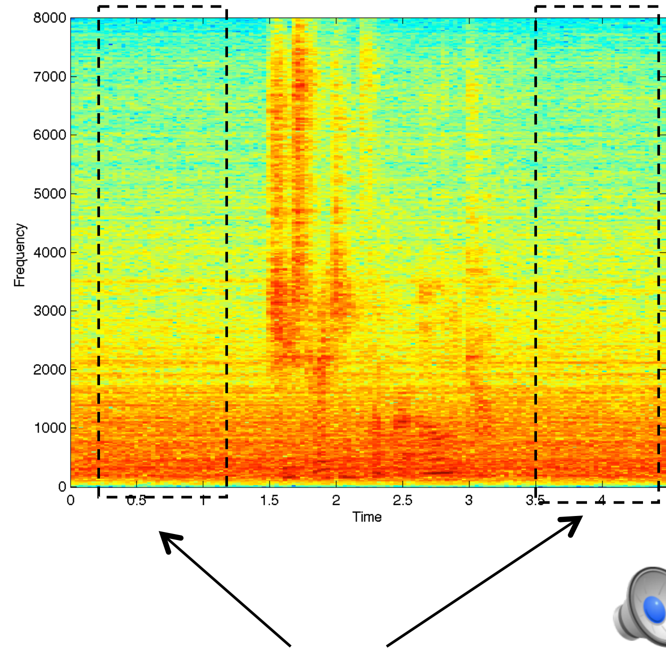
$$\mu(f) = \frac{1}{M} \sum_{\forall t, S(t,f) \approx 0} \left\| X(t,f) \right\|$$

$$M = \text{number of noise frames}$$



Spectral subtraction methods

- We can now (perhaps) estimate the clean sound
  - We know the characteristics of the noise (as described from the spectrum $\mu(f)$)
- But, we will assume:
  - The noise source is constant
    - If the noise spectrum changes $\mu(f)$ is not a valid noise description anymore
  - The noise is additive



Sections of isolated noise (or at least no useful signal)

Spectral subtraction

- Magnitude subtraction
  - Subtract the noise magnitude spectrum from the recording's

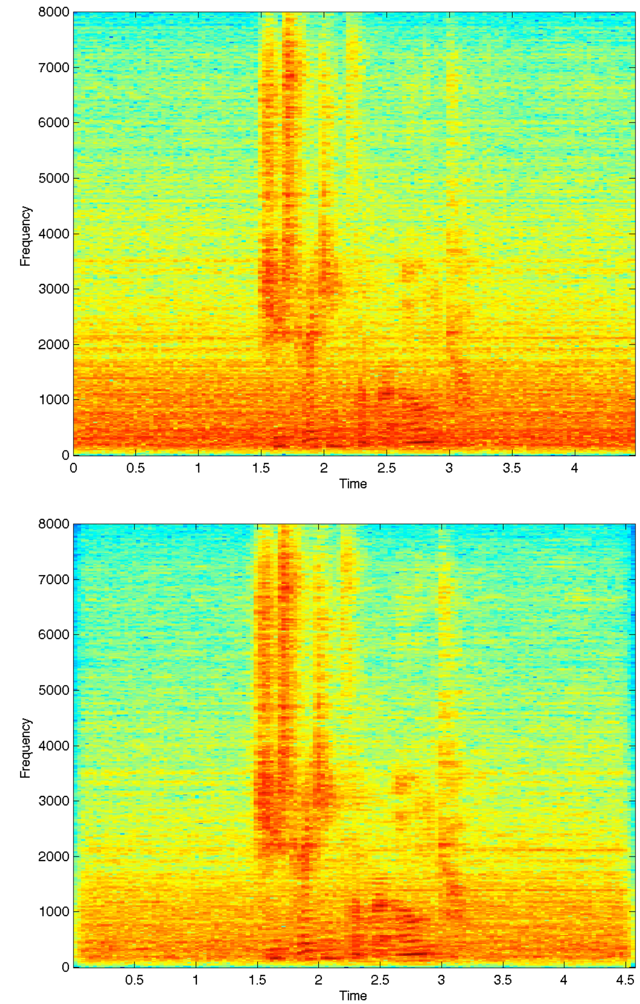
$$X(t,f) = S(t,f) + N(t,f) \Rightarrow$$

$$\left\|\hat{S}(t,f)\right\| = \left\|X(t,f)\right\| - \mu(f)$$

- We can then modulate the magnitude of the original input to reconstruct

$$\hat{S}(t,f) = \left[\left\|X(t,f)\right\| - \mu(f)\right] \angle X(t,f)$$

- Sounds pretty good …



Original input

After spectral subtraction

Estimating the noise spectrum

- Noise is usually not stationary
  - Although the rate of change with time may be slow
- A running estimate of noise is required
  - Update noise estimates at every frame of the audio
- The exact location of “noise-only” segments is never known
  - For speech signals we use an important characteristic of speech to discover speech segments (and, consequently noise-only segments) in the audio
  - The onset of speech is always indicated by a sudden increase in the energy level in the signal

A running estimate of noise

- The initial T frames in any recording are assumed to be free of the speech signal
  - Typically T = 10

- The noise estimate $N(\mathrm{T,f})$ is estimated as

  $N(\mathrm{T,f}) = (1/\mathrm{T})\ \Sigma_{\mathrm{t}}\ |X(\mathrm{t,f})|$

- Subsequent estimates are obtained as follows
  - Assumption: The magnitude spectrum increases suddenly in value at the onset of speech

$$|N(t,f)|^p \approx \begin{cases} (1-\lambda)\,|N(t-1,f)|^p + \lambda\,|X(t,f)|^p & \text{if } |X(t,f)| < \beta\,|N(t-1,f)| \\ |N(t-1,f)|^p & \text{otherwise} \end{cases}$$

A running estimate of noise

$$|N(t,f)|^p = \begin{cases} (1-\lambda)|N(t-1,f)|^p + \lambda|X(t,f)|^p & \text{if } |X(t,f)| < \beta|N(t-1,f)| \\ |N(t-1,f)|^p & \text{otherwise} \end{cases}$$

- p is an exponent term that is typically set to either 2 or 1
  - p = 2 : power spectrum; p = 1 : magnitude spectrum
- λ is a noise update factor
  - Typically set in the range 0.1 – 0.5
  - Accounts for time-varying noise
- β is a thresholding term
  - A typical value of β is 5.0
  - If the signal energy jumps by a factor of β, speech onset has occurred

- Other more complex rules may be applied to detect speech offset

Cancelling the Noise

- Simple Magnitude Subtraction
  - $|S(t,f)| = |X(t,f)| - |N(t,f)|$
- Power subtraction
  - $|S(t,f)|^2 = |X(t,f)|^2 - |N(t,f)|^2$
- Filtering methods: $S(t,f) = H(t,f)X(t,f)$
  - Weiner Filtering: build an optimal filter to remove the estimated noise
  - Maximum-likelihood estimation..

The Filter Functions

- We have a source plus noise spectrum

$$X(t,f) = S(t,f) + N(t,f)$$

- The desired output is some function of the input and the noise spectrum

$$\hat{S}(t,f) = g\big(X(t,f), N(t,f)\big)$$

- Let's make it a "gain function"

$$H(t,f) = f\big(X(t,f), N(t,f)\big)$$

$$\hat{S}(t,f) = H(t,f)X(t,f)$$

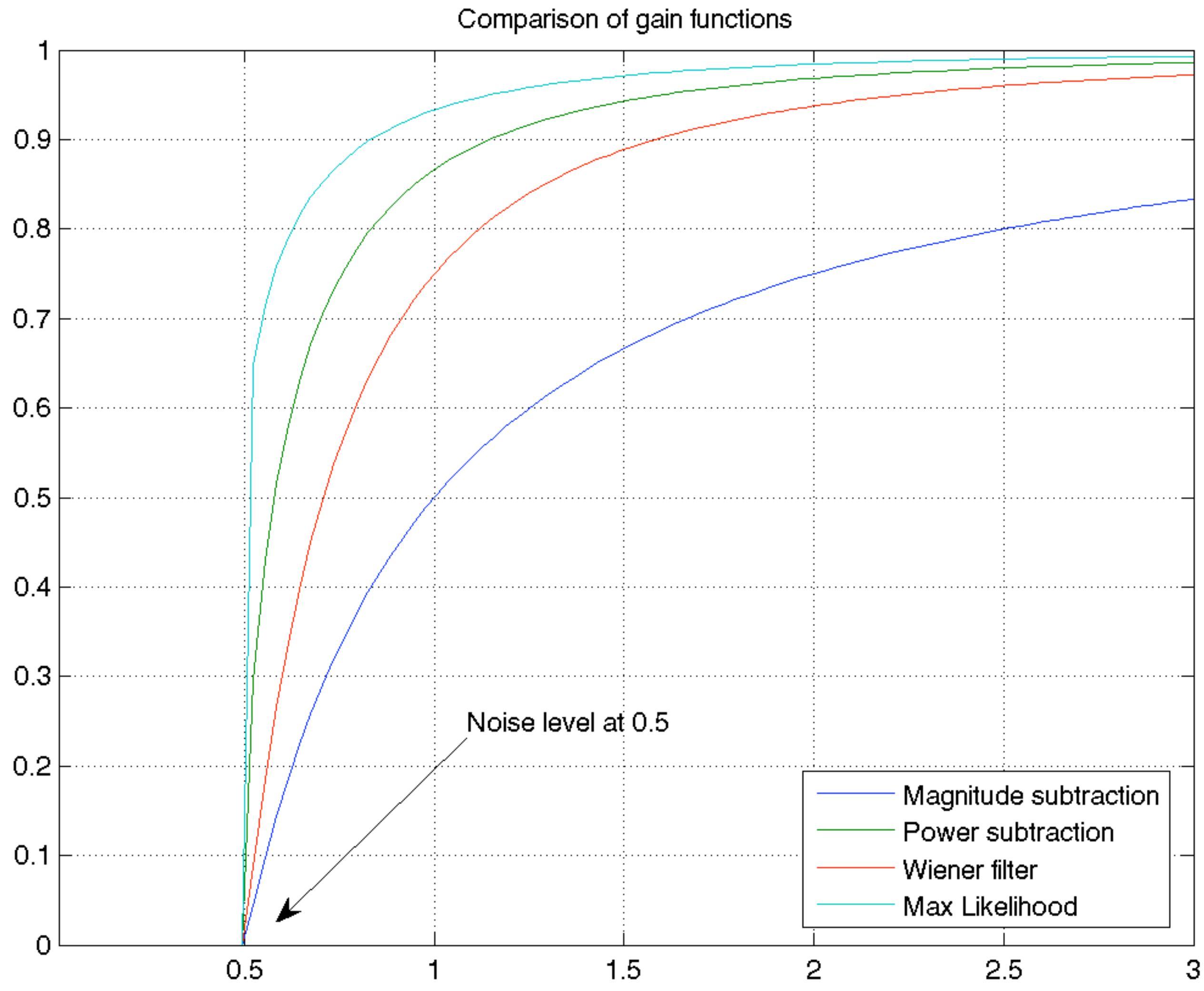
- For spectral subtraction the gain function is:

$$H(t,f) = 1 - \frac{\|N(t,f)\|}{\|X(t,f)\|}$$

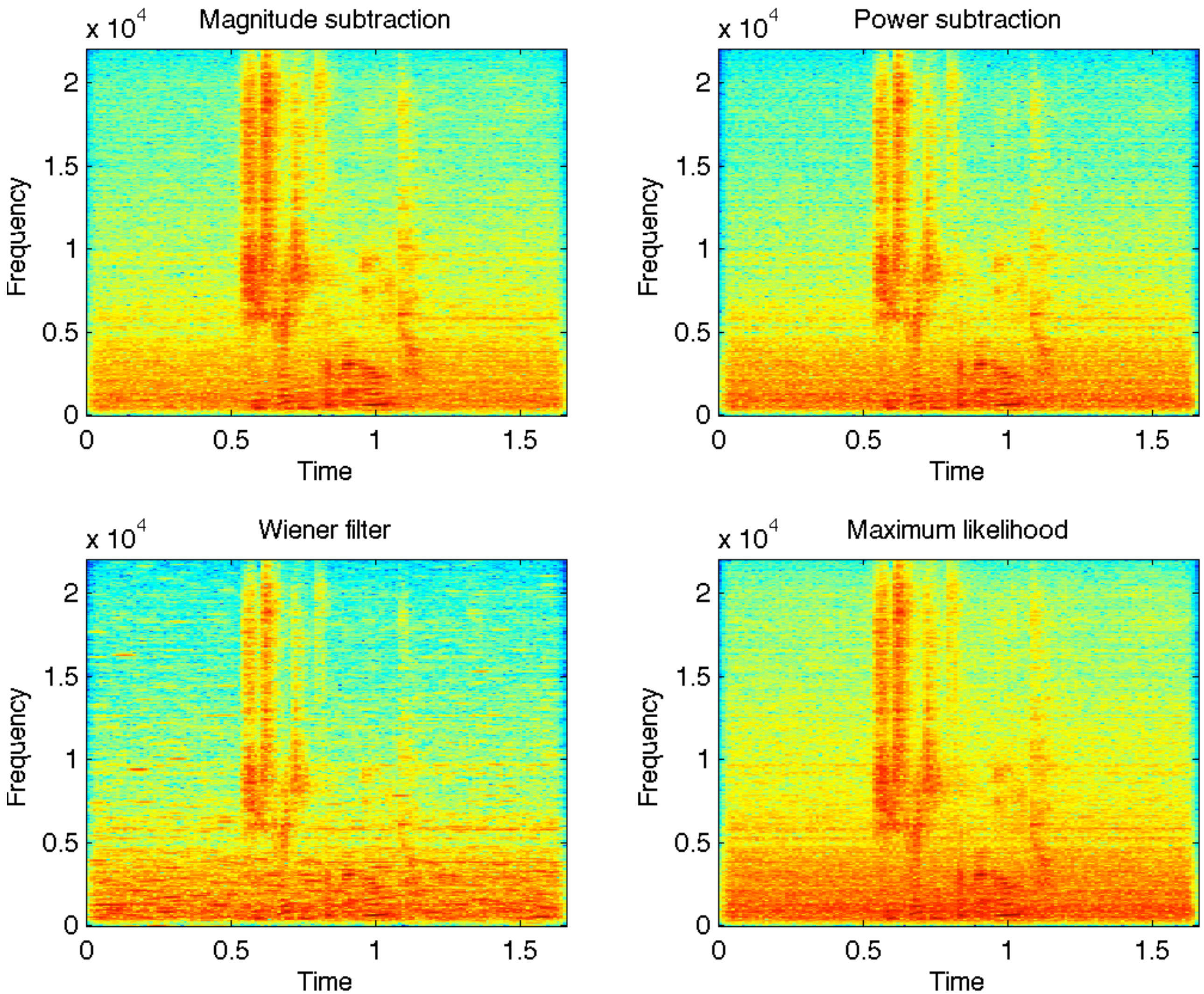
Filters for denoising

- Magnitude subtraction: $$H(f) = 1 - \frac{N(f)}{\|X(f)\|}$$
- Power subtraction: $$H(f) = \sqrt{1 - \frac{N^2(f)}{\|X(f)\|^2}}$$
- Wiener filter: $$H(f) = 1 - \frac{N^2(f)}{\|X(f)\|^2}$$
- Maximum likelihood: $$H(f) = \frac{1}{2}\left[1 + \sqrt{1 - \frac{N^2(f)}{\|X(f)\|^2}}\right]$$

Filter function comparison



Examples of various filter functions



Original

Magnitude subtraction

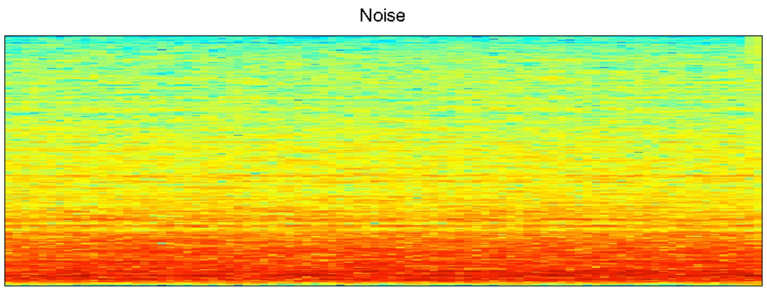
Power subtraction

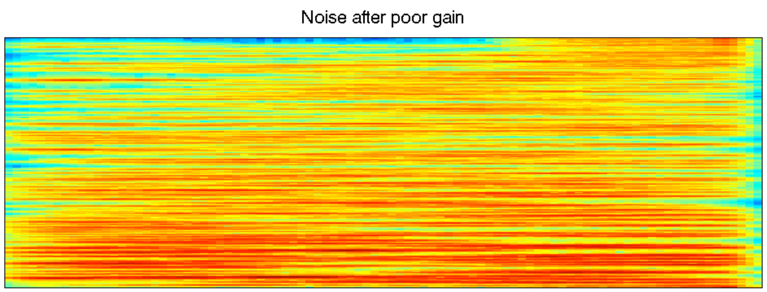
Wiener filter

Maximum likelihood

"Musical noise"

- What was that weirdness with the Wiener filter???
  - An artifact called *musical noise*
  - The other approaches had it too
- Takes place when the signal to noise ratio is small
  - Ends up on the steep part of the gain curve
    - Small fluctuations are then magnified
  - Results in complex or negative gain
    - An awkward situation!
- The result is sinusoids popping in and out
  - Hence the tonal overload





Noise reduced noise!
(lots of musical noise)

Reducing musical noise

- Thresholding

$$H'(f) = \begin{cases} H(f) \text{ if } \|X(f)\| > N(f) \\ 0 \quad \text{otherwise} \end{cases}$$

  - The gain curve is steeper on the negative side this removes effects in that area

- Scale the noise spectrum

  $N(f) = \alpha N(f), \alpha > 1$

  - (Linearly) increases gain in the new location

- Smoothing

  e.g. $H(t,f) = .5H(t,f) + .5H(t-1,f)$

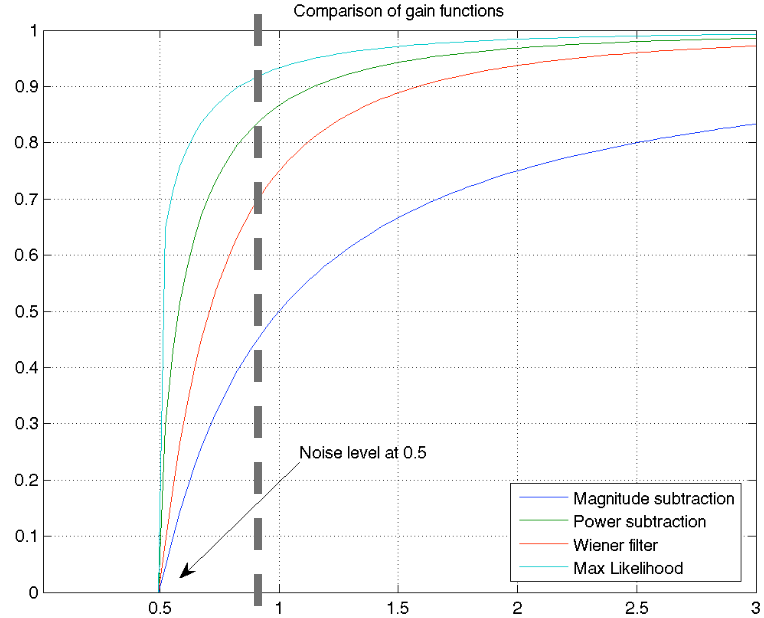
  - Or some other time averaging
  - Reduces sudden tone on/offs
  - But adds a slight echo

Wiener filter

With thresholding

With thresholding & smoothing

Reducing musical noise



Wiener filter

With thresholding and oversub

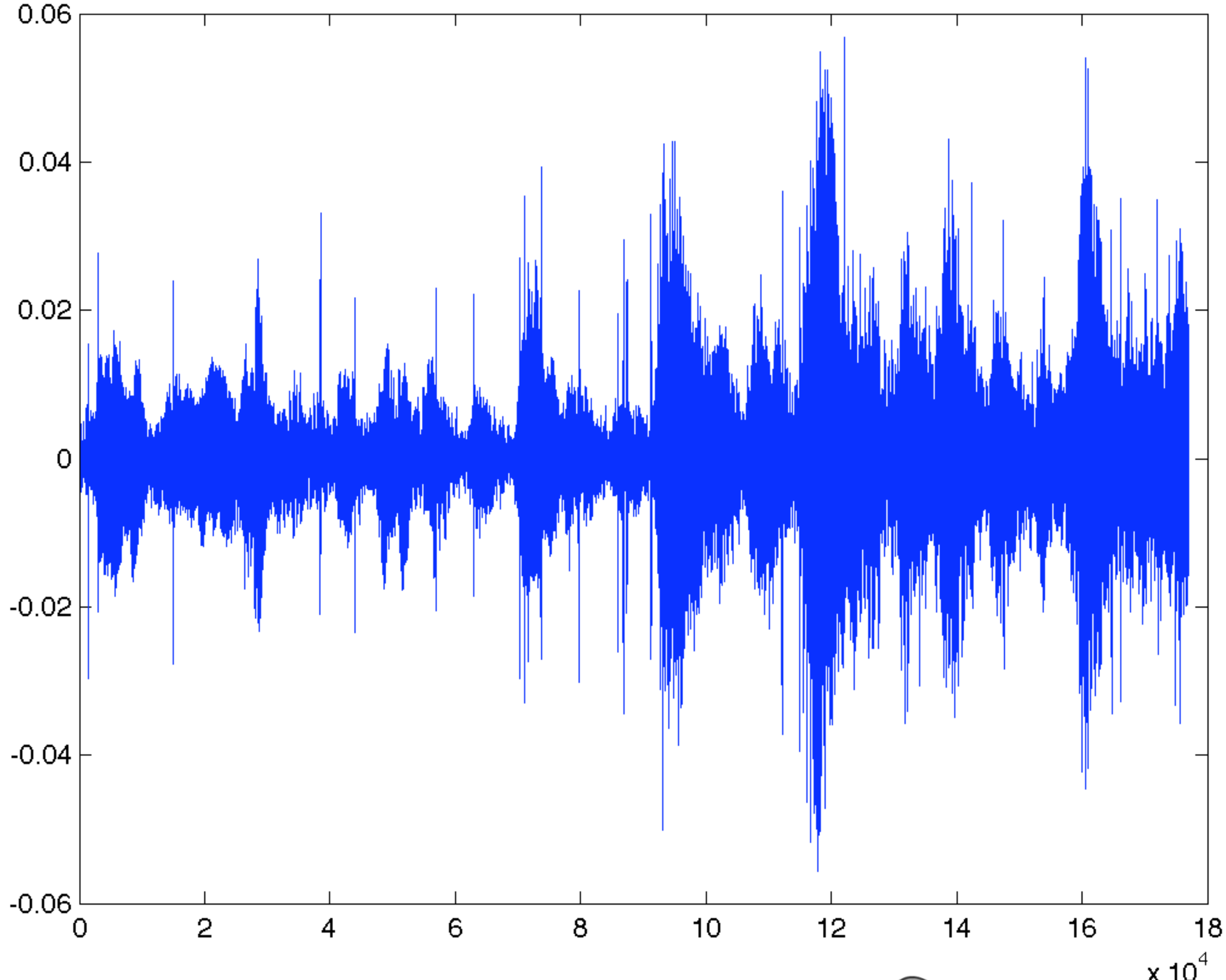
With thresholding, oversub, and smoothing

- Thresholding : Moves the operating point to a less sloped region of the curve
- Oversubtraction: Increases the slope in these regions for better differential gain
- Smoothing: $H(t,f) = 0.5H(t,f) + 0.5H(t-1,f)$
  - Adds an echo

Audio restoration II
Click/glitch/gap removal

- Two step process
  - Detection of abnormality
  - Replacement of corrupted data
- Detection stuff
  - Autoregressive modeling for abnormality detection
- Data replacement
  - Interpolation of missing data using autoregressive interpolation

Starting signal



- Can you spot the glitches?

Autoregressive (AR) models

- Predicting the next sample of a series using a weighted sum of the past samples

$$x(t) = \sum_{i=1}^{N} a(i)x(t-i) + e(t)$$

- The weights a can be estimated upon presentation of a training input
  - Least squares solution of above equation
  - Fancier/faster estimators, e.g. **`aryule`** in MATLAB

Matrix formulation

- Scalar version

$$x(t) = \sum_{i=1}^{N} a(i)x(t-i) + e(t)$$

- Matrix version

$$\mathbf{x} = \begin{bmatrix} a_{N-1} & \cdots & a_0 & 0 & 0 \\ 0 & \ddots & \cdots & a_0 & 0 \\ 0 & 0 & \ddots & \cdots & a_0 \\ 0 & 0 & 0 & \ddots & \cdots \\ 0 & 0 & 0 & 0 & a_{N-1} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_M \end{bmatrix}$$

Measuring prediction error

- As Convolution

$$\mathbf{e} = \mathbf{x} - \mathbf{a} * \mathbf{x}$$

- As matrix operation

$$\mathbf{e} = \begin{bmatrix} -a_N & \cdots & -a_1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -a_N & \cdots & -a_1 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots & \\ \cdots & 0 & 0 & -a_N & \cdots & -a_1 & 1 & 0 & 0 \\ 0 & \cdots & 0 & 0 & -a_N & \cdots & -a_1 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & -a_N & \cdots & -a_1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ \vdots \\ x_M \end{bmatrix}$$

- Overall error variance: $\mathbf{e}^T\mathbf{e}$

Measuring prediction error

- Convolution

  $$\mathbf{e} = \mathbf{x} - \mathbf{a} * \mathbf{x}$$

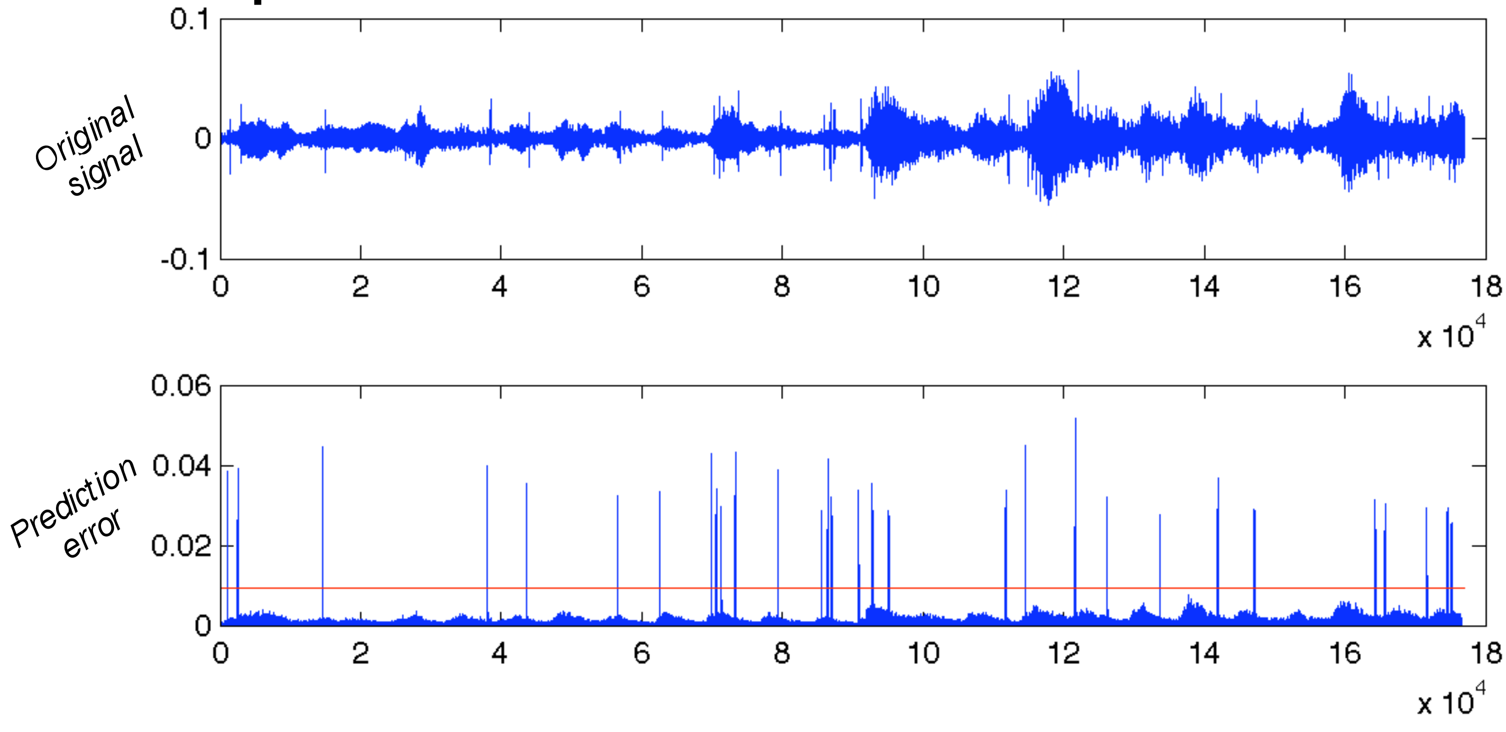
- Solution for $\mathbf{a}$ must minimize error variance: $\mathbf{e}^T\mathbf{e}$
  - While maintaining the Toeplitz structure of $\mathbf{a}$!
- A variety of solution techniques are available
  - The most popular one is the “Levinson Durbin” algorithm

Discovering abnormalities

- The AR models smooth and predictable things, e.g. music, speech, etc
- Clicks, gaps, glitches, noise are not very predictable (at least in the sense of a meaningful signal)
- Methodology
  - Learn an AR model on your signal type
  - Measure prediction error on the noisy data
  - Abnormalities appear as spikes in error

Glitch detection example

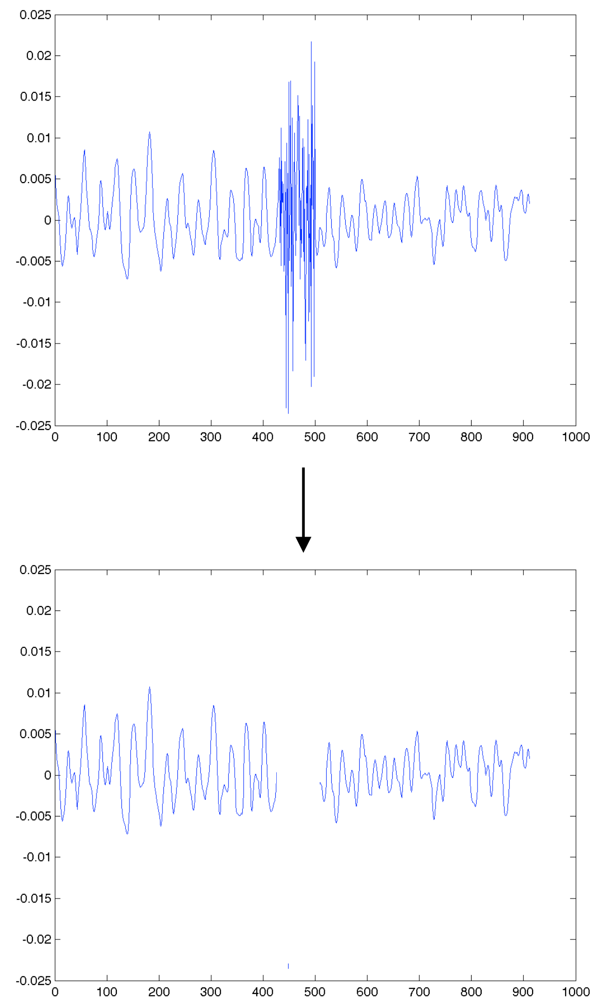
- Glitches are clearly detected as spikes in the prediction error



- Why? Glitches are unpredictable!

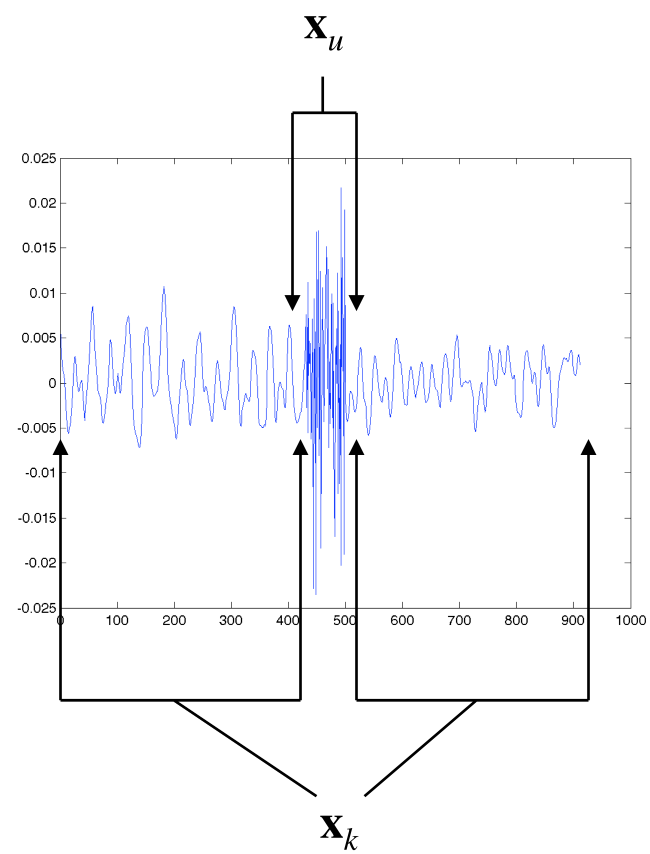
Now what?

- Detecting the glitches is only one step!
- How to we remove them?
- Information is lost!
  - We need to make up data!
- This is an interpolation problem
  - Filling in missing data
  - Hints provided from neighboring samples



Interpolation formulation

- Detection of spikes defines areas of missing samples
  - $\pm N$ samples from glitch point
- Group samples to known and unknown sets according to spike detection positions
  - $\mathbf{x}_k = \mathbf{K}\cdot\mathbf{x}$, $\mathbf{x}_u = \mathbf{U}\cdot\mathbf{x}$
  - $\mathbf{x} = (\mathbf{U}\cdot\mathbf{x} + \mathbf{K}\cdot\mathbf{x})$
  - Transforms $\mathbf{U}$ and $\mathbf{K}$ maintain only specific data (= unit matrices with appropriate missing rows)



Picking sets of samples

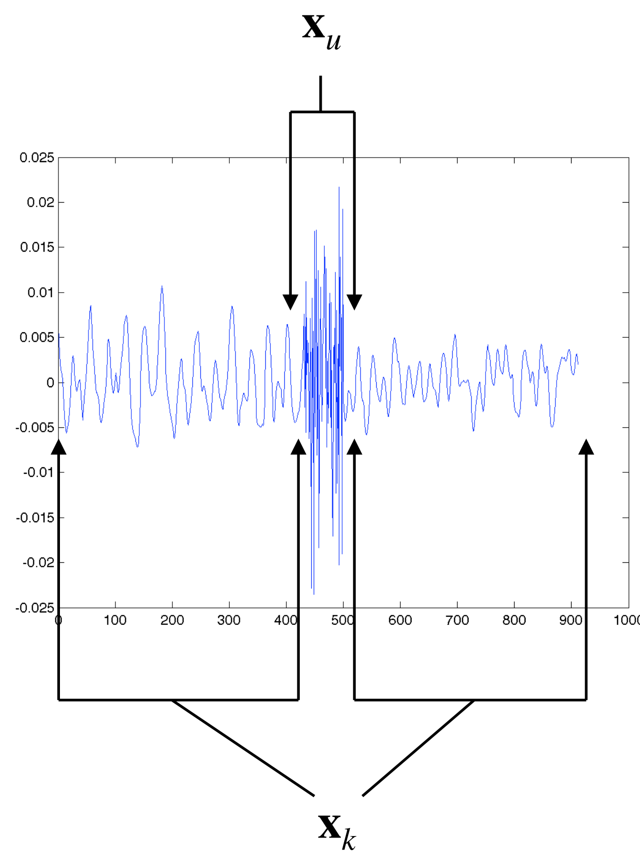
$$\mathbf{x} = \mathbf{U}\mathbf{x} + \mathbf{K}\mathbf{x} =$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} =$$

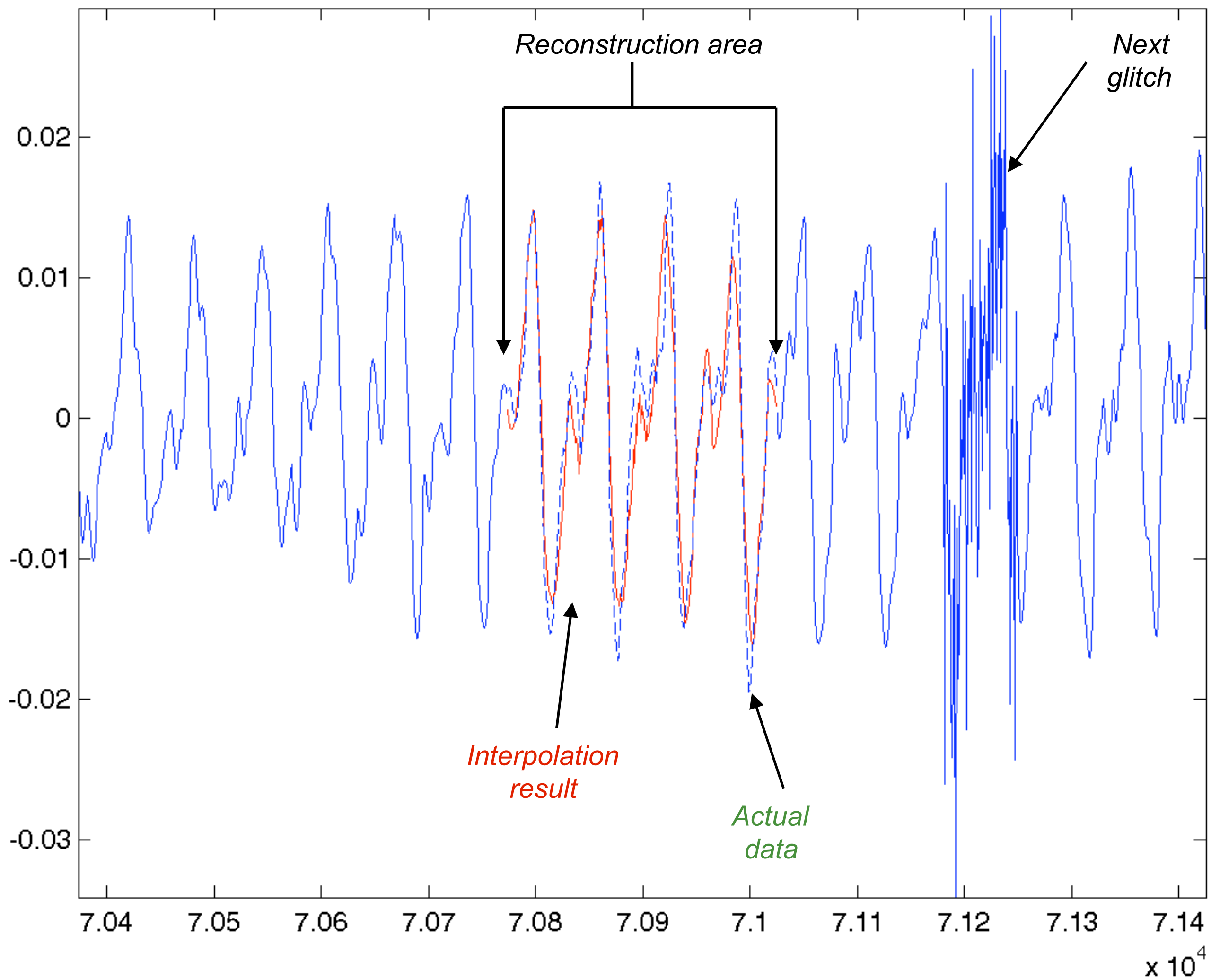
$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ x_2 \\ x_3 \\ 0 \end{bmatrix} + \begin{bmatrix} x_1 \\ 0 \\ 0 \\ x_4 \end{bmatrix}$$

Making up the data

- AR model error is
  - $\mathbf{e} = \mathbf{A} \cdot \mathbf{x} = \mathbf{A} \cdot (\mathbf{U} \cdot \mathbf{x}_u + \mathbf{K} \cdot \mathbf{x}_k)$
- We can solve for $\mathbf{x}_u$
  - Ideally $\mathbf{e}$ is 0
- Hence zero error estimate for missing data is:
  - $\mathbf{A} \cdot \mathbf{U} \cdot \mathbf{x}_u = -\mathbf{A} \cdot \mathbf{K} \cdot \mathbf{x}_k$
  - $\mathbf{x}_u = -(\mathbf{A} \cdot \mathbf{U})^+ \cdot \mathbf{A} \cdot \mathbf{K} \cdot \mathbf{x}_k$
  - $(\mathbf{A} \cdot \mathbf{U})^+$ is pseudo-inverse



Reconstruction zoom in



Distorted signal

Recovered signal

Restoration recap

- Constant noise removal
  - Spectral subtraction/Wiener filters
  - Musical noise and tricks to avoid it
- Click/glitch/gap detection
  - Music/speech is very predictable
  - AR models to detect abnormalities
- Missing sample interpolation
  - AR model for creating missing data