

Wavelet transform for dimensionality reduction.

Wavelet transform for dimensionality reduction.

Introduction

Continuous wavelet transform

CWT vs STFT

Incertitude principle

Discrete wavelet transform (DWT)

Conclusion

What is a wavelet?

Scale and translation invariant

The continuous wavelet transform

Example using Mexican hat wavelet

A function $\psi(t)$, to be a wavelet must satisfy :

A function $\psi(t)$, to be a wavelet must satisfy :

1. The wavelet must be centered at 0 amplitude.

$$\int_{-\infty}^{\infty} \psi(t) dt = 0$$

A function $\psi(t)$, to be a wavelet must satisfy :

1. The wavelet must be centered at 0 amplitude.

$$\int_{-\infty}^{\infty} \psi(t) dt = 0$$

2. The wavelet must have a finite energy. Therefore it is localized in time (or space).

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty$$

A function $\psi(t)$, to be a wavelet must satisfy :

1. The wavelet must be centered at 0 amplitude.

$$\int_{-\infty}^{\infty} \psi(t) dt = 0$$

2. The wavelet must have a finite energy. Therefore it is localized in time (or space).

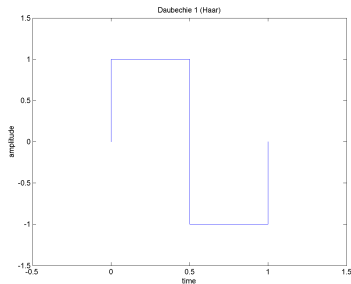
$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty$$

3. Sufficient condition for inverse wavelet transform

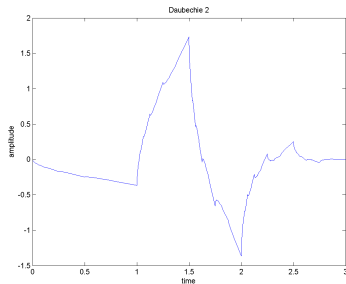
$$c \equiv \int_{-\infty}^{\infty} \frac{|\Psi(w)|^2}{|w|} dw$$

$$0 < c < \infty$$

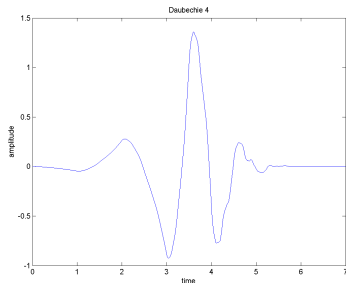
Popular wavelets which satisfy the previous conditions



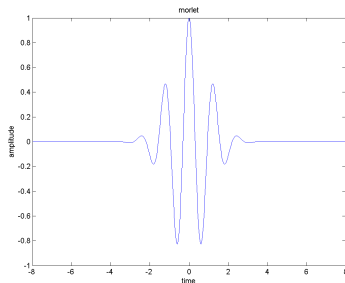
Popular wavelets which satisfy the previous conditions



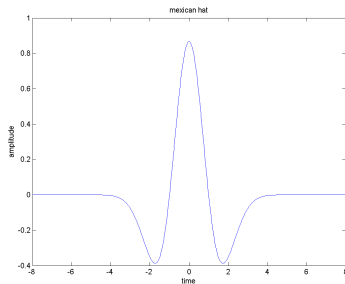
Popular wavelets which satisfy the previous conditions



Popular wavelets which satisfy the previous conditions



Popular wavelets which satisfy the previous conditions



we can move and stretch the mother wavelet

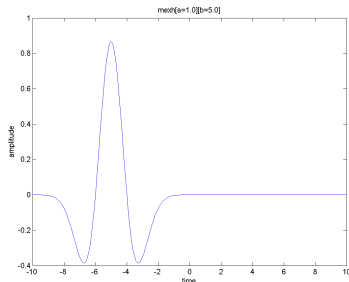
$$\psi_{a,b}(t) \equiv \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-a}{b}\right)$$

a is scale factor while b is a translation factor. $\frac{1}{\sqrt{|a|}}$ is a normalization factor to make sure the energy stay the same.

we can move and stretch the mother wavelet

$$\psi_{a,b}(t) \equiv \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-a}{b}\right)$$

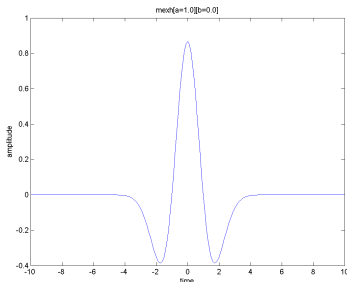
a is scale factor while b is a translation factor. $\frac{1}{\sqrt{|a|}}$ is a normalization factor to make sure the energy stay the same.



we can move and stretch the mother wavelet

$$\psi_{a,b}(t) \equiv \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-a}{b}\right)$$

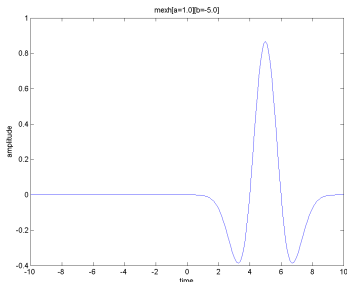
a is scale factor while b is a translation factor. $\frac{1}{\sqrt{|a|}}$ is a normalization factor to make sure the energy stay the same.



we can move and stretch the mother wavelet

$$\psi_{a,b}(t) \equiv \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-a}{b}\right)$$

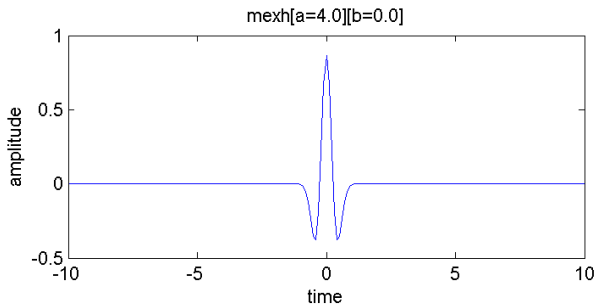
a is scale factor while b is a translation factor. $\frac{1}{\sqrt{|a|}}$ is a normalization factor to make sure the energy stay the same.



we can move and stretch the mother wavelet

$$\psi_{a,b}(t) \equiv \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-a}{b}\right)$$

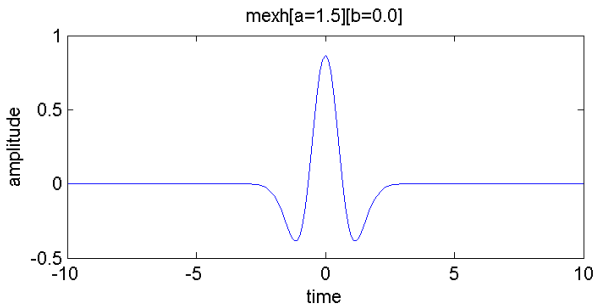
a is scale factor while b is a translation factor. $\frac{1}{\sqrt{|a|}}$ is a normalization factor to make sure the energy stay the same.



we can move and stretch the mother wavelet

$$\psi_{a,b}(t) \equiv \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-a}{b}\right)$$

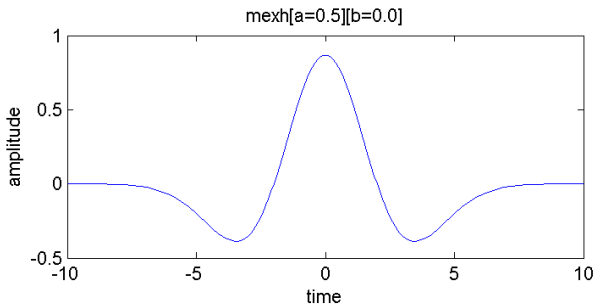
a is scale factor while b is a translation factor. $\frac{1}{\sqrt{|a|}}$ is a normalization factor to make sure the energy stay the same.



we can move and stretch the mother wavelet

$$\psi_{a,b}(t) \equiv \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-a}{b}\right)$$

a is scale factor while b is a translation factor. $\frac{1}{\sqrt{|a|}}$ is a normalization factor to make sure the energy stay the same.



The dot product for continuous function is defined as this :

$$\langle f, g \rangle \equiv \int_{-\infty}^{\infty} f(t) g(t)^* dt$$

The dot product for continuous function is defined as this :

$$\langle f, g \rangle \equiv \int_{-\infty}^{\infty} f(t) g(t)^* dt$$

which is similar to the discrete dot product :

$$u \cdot v \equiv \sum_{i=1}^n u_i v_i$$

The dot product for continuous function is defined as this :

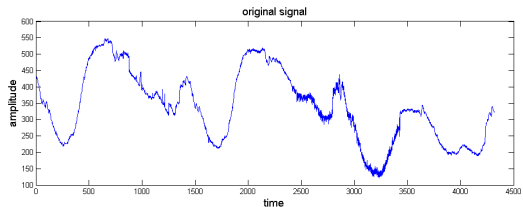
$$\langle f, g \rangle \equiv \int_{-\infty}^{\infty} f(t) g(t)^* dt$$

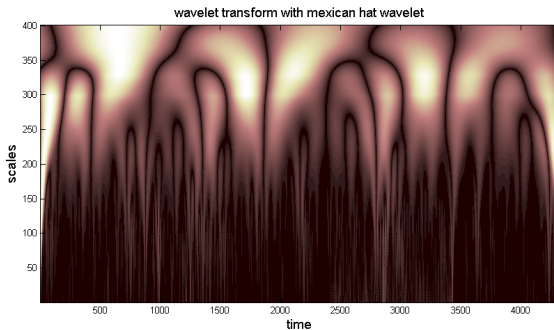
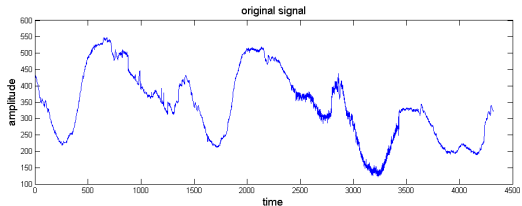
which is similar to the discrete dot product :

$$u \cdot v \equiv \sum_{i=1}^n u_i v_i$$

The wavelet transform is simply the dot product between the signal and the wavelet at each translation and each scale.

$$W_{a,b} \equiv \int_{-\infty}^{\infty} f(t) \psi_{a,b}^*(t) dt$$





- ▶ The STFT is the fourier transform computed for every time step.

- ▶ The STFT is the fourier transform computed for every time step.
- ▶ To isolate a particular time step, a window function is used.

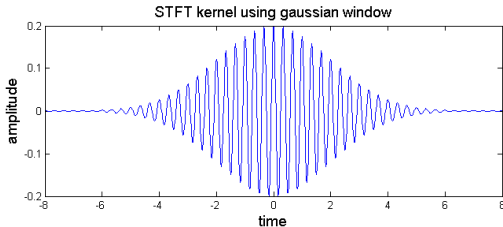
- ▶ The STFT is the fourier transform computed for every time step.
- ▶ To isolate a particular time step, a window function is used.
- ▶

$$STFT(t, \omega) = \int_{-\infty}^{\infty} x(\tau) W(\tau - t) e^{-j\omega\tau} d\tau$$

- ▶ The STFT is the fourier transform computed for every time step.
- ▶ To isolate a particular time step, a window function is used.
- ▶

$$STFT(t, \omega) = \int_{-\infty}^{\infty} x(\tau) W(\tau - t) e^{-j\omega\tau} d\tau$$

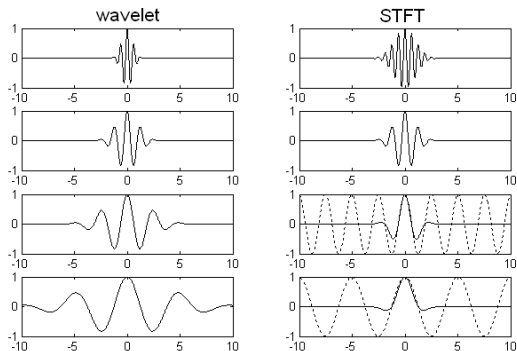
- ▶ The kernel have all the characteristics to be a wavelet



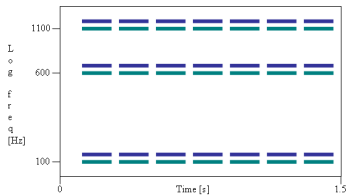
- ▶ If we choose a gaussian window for the SFTF, it is exactly the Morlet wavelet.

- ▶ If we choose a gaussian window for the SFTF, it is exactly the Morlet wavelet.
- ▶ The only difference, is when the frequency is changing.

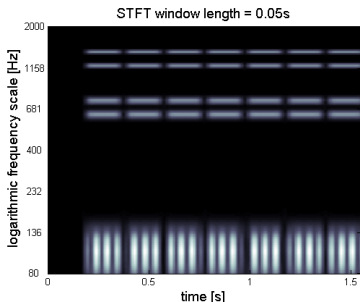
- ▶ If we choose a gaussian window for the SFTF, it is exactly the Morlet wavelet.
- ▶ The only difference, is when the frequency is changing.
- ▶ In the case of the wavelet, the widow width is adapting with the frequency, keeping the number of cycles constant inside the window.



Theoretical spectrogram of 2 notes repeated 7 times with 200ms of duration.

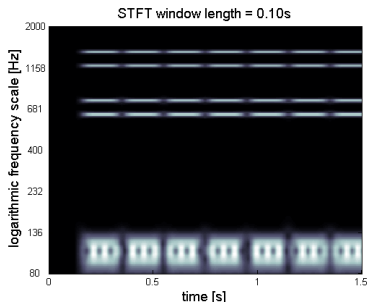
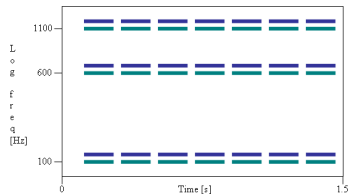


STFT with gaussian window of width 0.05ms.

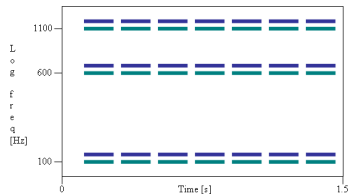


Theoretical spectrogram of 2 notes repeated 7 times with 200ms of duration.

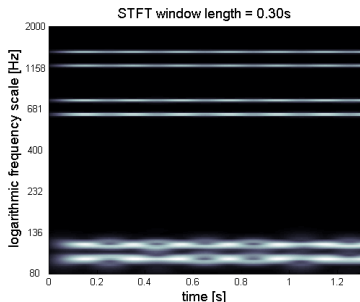
STFT with gaussian window of width 0.1ms.



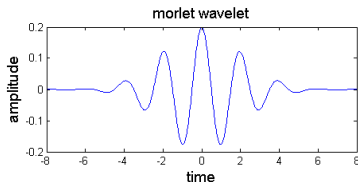
Theoretical spectrogram of 2 notes repeated 7 times with 200ms of duration.



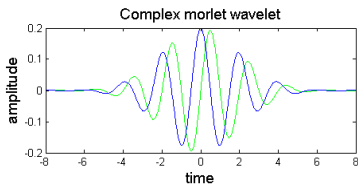
STFT with gaussian window of width 0.3ms.



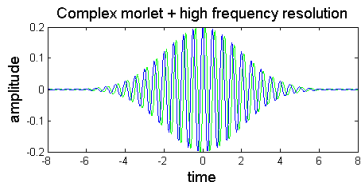
- ▶ Morlet wavelet is similar to the STFT kernel.



- ▶ Morlet wavelet is similar to the STFT kernel.
- ▶ To build a spectrogram, we need a complex wavelet.

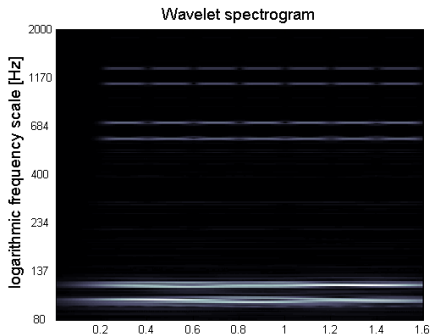
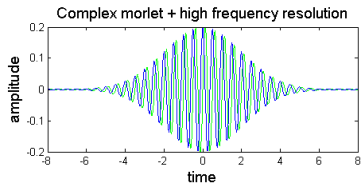


- ▶ Morlet wavelet is similar to the STFT kernel.
- ▶ To build a spectrogram, we need a complex wavelet.
- ▶ We also need to boost the frequency resolution.



- ▶ Morlet wavelet is similar to the STFT kernel.
- ▶ To build a spectrogram, we need a complex wavelet.
- ▶ We also need to boost the frequency resolution.

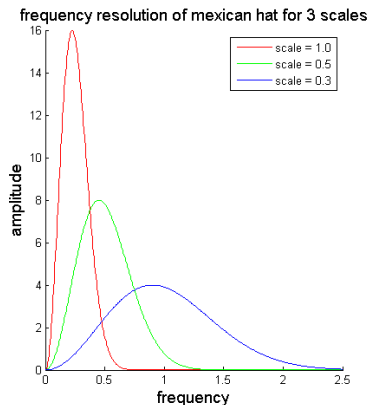
$$\psi(t) = \left(\frac{1}{\sqrt{\pi\sigma^2}} \right) e^{2i\pi f t} e^{-\frac{t^2}{\sigma^2}}$$



- ▶ Wavelet are like bandpass filters.

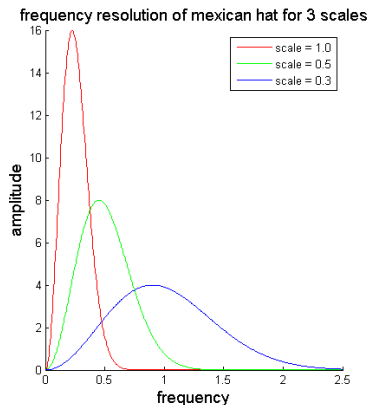
- ▶ Wavelet are like bandpass filters.
- ▶ By computing their fourier transform, we find their frequency response.

- ▶ Wavelet are like bandpass filters.
- ▶ By computing their fourier transform, we find their frequency response.



- ▶ Wavelet are like bandpass filters.
- ▶ By computing their fourier transform, we find their frequency response.
- ▶ The standard deviation of the frequency response gives the frequency resolution.

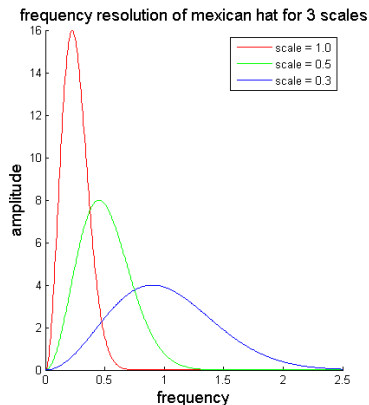
$$\Delta\omega = \sqrt{\frac{\int_{-\infty}^{\infty} (\omega - \omega_0)^2 |\Psi(\omega)|^2 d\omega}{\int_{-\infty}^{\infty} |\Psi(\omega)|^2 d\omega}}$$



- ▶ Wavelet are like bandpass filters.
- ▶ By computing their fourier transform, we find their frequency response.
- ▶ The standard deviation of the frequency response gives the frequency resolution.

$$\Delta\omega = \sqrt{\frac{\int_{-\infty}^{\infty} (\omega - \omega_0)^2 |\Psi(\omega)|^2 d\omega}{\int_{-\infty}^{\infty} |\Psi(\omega)|^2 d\omega}}$$

- ▶ similarly, we can find the time resolution.



- ▶ But the resolution depends of the mother wavelet and the scale.

- ▶ But the resolution depends of the mother wavelet and the scale.
- ▶ Time resolution

$$\Delta t \rightarrow \Delta t_{\psi}(a) = |a| \Delta t_{\psi}$$

- ▶ But the resolution depends of the mother wavelet and the scale.
- ▶ Time resolution

$$\Delta t \rightarrow \Delta t_{\psi}(a) = |a| \Delta t_{\psi}$$

- ▶ Frequency resolution

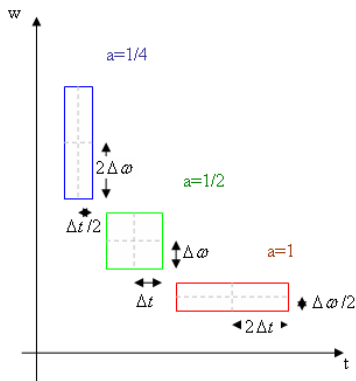
$$\Delta \omega \rightarrow \Delta \omega_{\psi}(a) = \Delta \omega_{\psi} / |a|$$

- ▶ But the resolution depends of the mother wavelet and the scale.
- ▶ Time resolution

$$\Delta t \rightarrow \Delta t_{\psi}(a) = |a| \Delta t_{\psi}$$

- ▶ Frequency resolution

$$\Delta \omega \rightarrow \Delta \omega_{\psi}(a) = \Delta \omega_{\psi} / |a|$$



- ▶ But the resolution depends of the mother wavelet and the scale.
- ▶ Time resolution

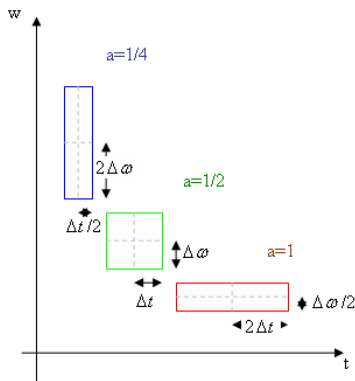
$$\Delta t \rightarrow \Delta t_{\psi}(a) = |a| \Delta t_{\psi}$$

- ▶ Frequency resolution

$$\Delta \omega \rightarrow \Delta \omega_{\psi}(a) = \Delta \omega_{\psi} / |a|$$

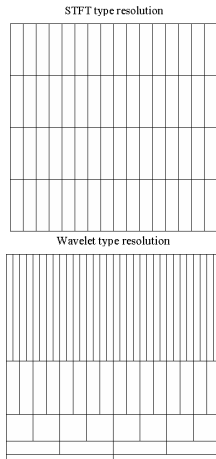
- ▶ Thus the area of the square is constant

$$\Delta t_{\psi}(a) \Delta \omega_{\psi}(a) = \Delta t_{\psi} \Delta \omega_{\psi} = c_{\psi}$$



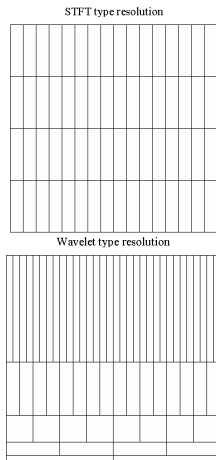
- ▶ Thus, the main difference between STFT and CWT is the tiling of the resolution.

- ▶ Thus, the main difference between STFT and CWT is the tiling of the resolution.
- ▶ One can tile the time-frequency space with any shape of rectangle, even with overlap.



- ▶ Thus, the main difference between STFT and CWT is the tiling of the resolution.
- ▶ One can tile the time-frequency space with any shape of rectangle, even with overlap.
- ▶ The rectangle can have a smaller area with a different filter, but there is a minimal area.

$$\Delta t_{\psi} \Delta \omega_{\psi} \geq 1/2$$



- ▶ It took about 10 minutes to generate the previous wavelet spectrogram (P4 1.8GHz with Matlab wavelet toolbox).

- ▶ It took about 10 minutes to generate the previous wavelet spectrogram (P4 1.8GHz with Matlab wavelet toolbox).
- ▶ For each of the m scales, the CWT perform a convolution on the raw signal of length n .

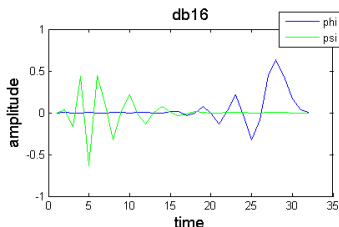
- ▶ It took about 10 minutes to generate the previous wavelet spectrogram (P4 1.8GHz with Matlab wavelet toolbox).
- ▶ For each of the m scales, the CWT perform a convolution on the raw signal of length n .
- ▶ The CWT return $m \cdot n$ coefficients in time $O(m \cdot n \log(n))$.

- ▶ It took about 10 minutes to generate the previous wavelet spectrogram (P4 1.8GHz with Matlab wavelet toolbox).
- ▶ For each of the m scales, the CWT perform a convolution on the raw signal of length n .
- ▶ The CWT return $m \cdot n$ coefficients in time $O(m \cdot n \log(n))$.
- ▶ There is a huge amount of redundancy and for higher scales, we could use a smaller sampling rate.

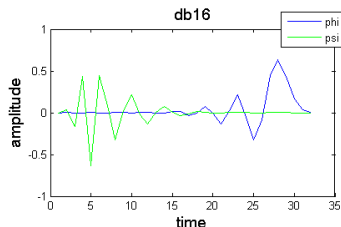
- ▶ To get a speed up in DWT, instead of stretching the wavelet to get to a bigger scale, we will compress the original signal.

- ▶ To get a speed up in DWT, instead of stretching the wavelet to get to a bigger scale, we will compress the original signal.
- ▶ For that we need a second wavelet, called the scaling function.

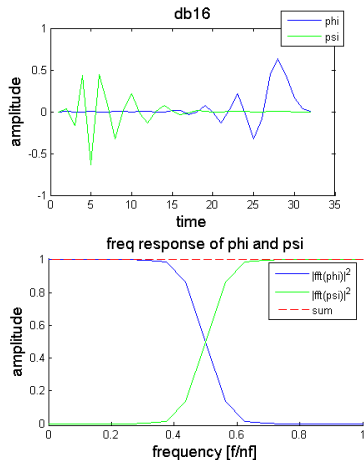
- ▶ To get a speed up in DWT, instead of stretching the wavelet to get to a bigger scale, we will compress the original signal.
- ▶ For that we need a second wavelet, called the scaling function.



- ▶ To get a speed up in DWT, instead of stretching the wavelet to get to a bigger scale, we will compress the original signal.
- ▶ For that we need a second wavelet, called the scaling function.
- ▶ This function is a low-pass filter with frequency cut half of Nyquist.



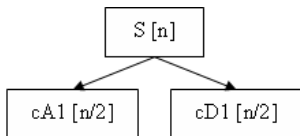
- ▶ To get a speed up in DWT, instead of stretching the wavelet to get to a bigger scale, we will compress the original signal.
- ▶ For that we need a second wavelet, called the scaling function.
- ▶ This function is a low-pass filter with frequency cut half of Nyquist.
- ▶ The wavelet is a complementary filter in that it is sensible for the rest of the frequency.



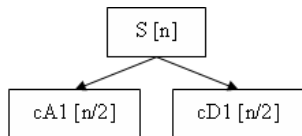
- ▶ To perform the DWT, we start from the signal.

$S[n]$

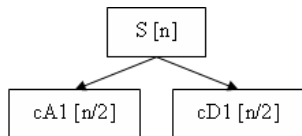
- ▶ To perform the DWT, we start from the signal.
- ▶ Then we split the signal in two part.



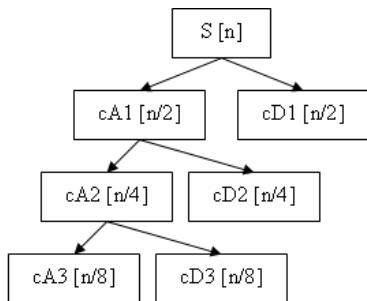
- ▶ To perform the DWT, we start from the signal.
- ▶ Then we split the signal in two part.
- ▶ Details, using the wavelet.



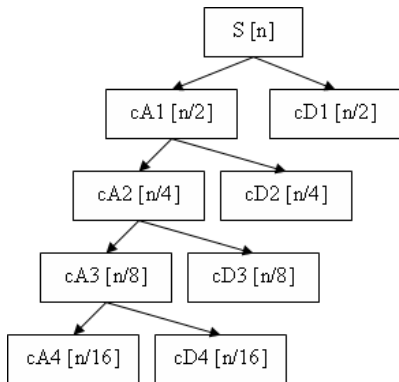
- ▶ To perform the DWT, we start from the signal.
- ▶ Then we split the signal in two part.
- ▶ Details, using the wavelet.
- ▶ Approximation, using the scaling function.



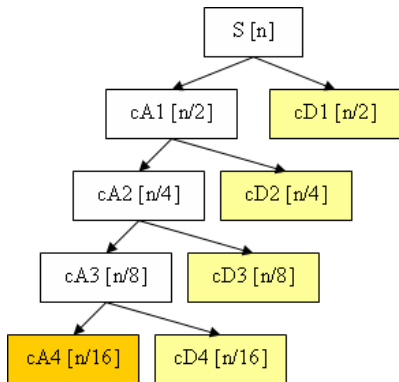
- ▶ To perform the DWT, we start from the signal.
- ▶ Then we split the signal in two part.
- ▶ Details, using the wavelet.
- ▶ Approximation, using the scaling function.
- ▶ We can start back the decomposition from the approximated signal.
- ▶ And again...



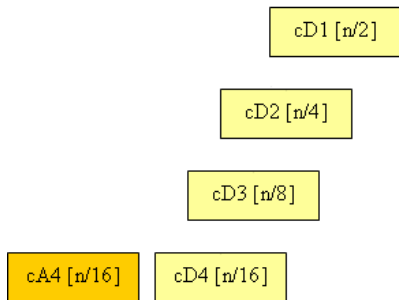
- ▶ To perform the DWT, we start from the signal.
- ▶ Then we split the signal in two parts.
- ▶ Details, using the wavelet.
- ▶ Approximation, using the scaling function.
- ▶ We can start back the decomposition from the approximated signal.
- ▶ And again...
- ▶ All the details is our wavelet transform. But we need to keep the last approximation for the inverse transform.



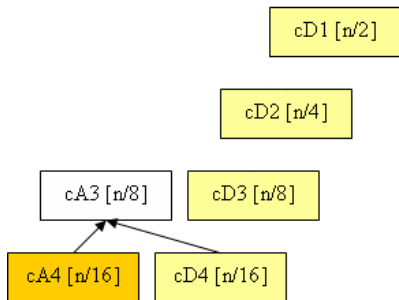
- ▶ To perform the DWT, we start from the signal.
- ▶ Then we split the signal in two parts.
- ▶ Details, using the wavelet.
- ▶ Approximation, using the scaling function.
- ▶ We can start back the decomposition from the approximated signal.
- ▶ And again...
- ▶ All the details are our wavelet transform. But we need to keep the last approximation for the inverse transform.



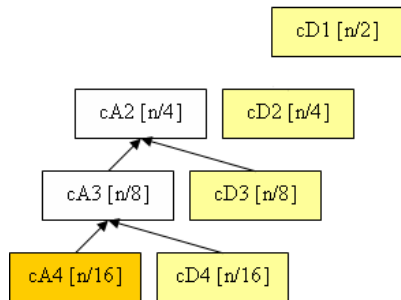
- ▶ To perform the inverse DWT, we start from the details coefficients and the last approximation.



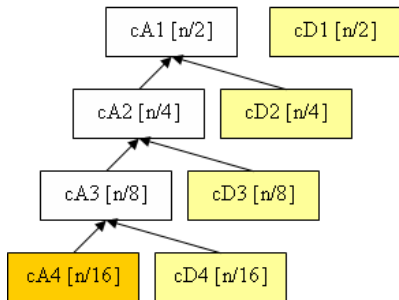
- ▶ To perform the inverse DWT, we start from the details coefficients and the last approximation.
- ▶ Then we combine the last approximation with the last details, to find the second last approximation.



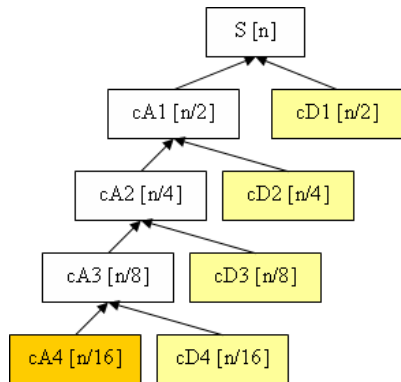
- ▶ To perform the inverse DWT, we start from the details coefficients and the last approximation.
- ▶ Then we combine the last approximation with the last details, to find the second last approximation.
- ▶ And we repeat...



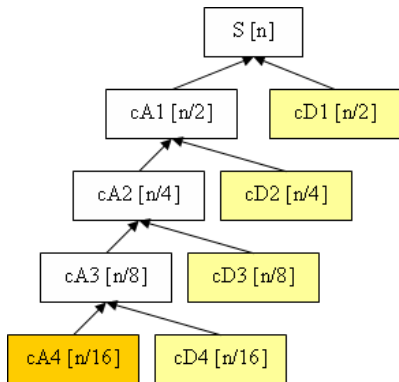
- ▶ To perform the inverse DWT, we start from the details coefficients and the last approximation.
- ▶ Then we combine the last approximation with the last details, to find the second last approximation.
- ▶ And we repeat...



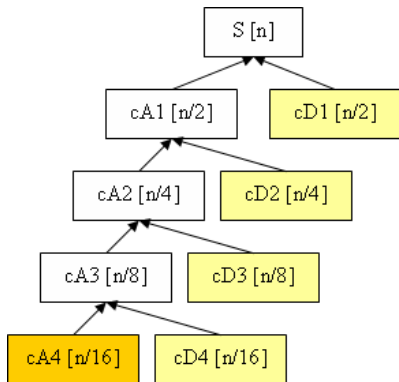
- ▶ To perform the inverse DWT, we start from the details coefficients and the last approximation.
- ▶ Then we combine the last approximation with the last details, to find the second last approximation.
- ▶ And we repeat...



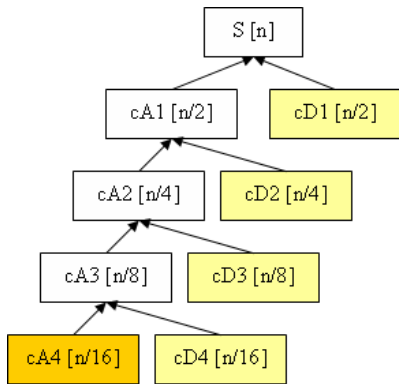
- ▶ To perform the inverse DWT, we start from the details coefficients and the last approximation.
- ▶ Then we combine the last approximation with the last details, to find the second last approximation.
- ▶ And we repeat...
- ▶ Both inverse and forward take $O(n)$



- ▶ To perform the inverse DWT, we start from the details coefficients and the last approximation.
- ▶ Then we combine the last approximation with the last details, to find the second last approximation.
- ▶ And we repeat...
- ▶ Both inverse and forward take $O(n)$
- ▶ It is thus faster than the fourier transform.



- ▶ To perform the inverse DWT, we start from the details coefficients and the last approximation.
- ▶ Then we combine the last approximation with the last details, to find the second last approximation.
- ▶ And we repeat...
- ▶ Both inverse and forward take $O(n)$
- ▶ It is thus faster than the fourier transform.
- ▶ But DWT restrict us to an octave of frequency resolution.



- ▶ We can compute the complete tree or choose a particular path (wavelet packet).
- ▶ We can compute the wavelet transform for n dimensions signals.
- ▶ By selecting the highest coefficients on each scales, we can keep only the most important details (image compression).

- ▶ We can compute the complete tree or choose a particular path (wavelet packet).
- ▶ We can compute the wavelet transform for n dimensions signals.
- ▶ By selecting the highest coefficients on each scales, we can keep only the most important details (image compression).
- ▶ By making statistics on wavelet coefficients, we can extract the global structure and use it as feature extraction for classification algorithms.

- ▶ Wavelet statistics are mainly used for texture classification in image processing.



- ▶ Wavelet statistics are mainly used for texture classification in image processing.
- ▶ But with a little of imagination, we can see texture everywhere.



- ▶ Wavelet statistics are mainly used for texture classification in image processing.
- ▶ But with a little of imagination, we can see texture everywhere.
- ▶ We have used wavelet statistics for genre classification.
- ▶ The texture of a small frame of sound represent the timbre.
- ▶ The texture of the temporal structure represent the rhythm...



that is all !!!
questions ?