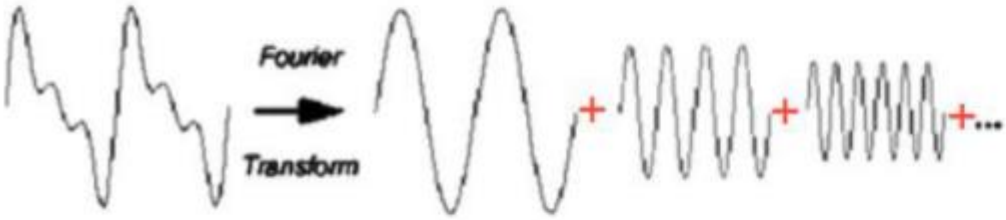
Fourier Transform & STFT & Wavelet Transform

Fourier Transform

: 푸리에 변환은 어떠한 형태의 신호든 다양한 주파수와 크기를 가진 사인 곡선을 합해서 표현할 수 있다고 가정한다.



[그림 1] Fourier Transform 참고 그림(출처 : [11])

위의 그림과 같이 여러 개의 사인 곡선으로 분해한 다음에 그 사인 곡선들의 주파수들을 통해 신호 내에 어떤 주파수들이 존재하는지를 나타낼 수 있다.

- 장점 : 신호 내에 존재하는 주파수들을 분석할 수 있다.
- 단점 : 시간에 대한 정보가 사라져 각 주파수가 시간적으로 언제 존재하는 것인지를 알 수 없다는 한계가 있다. 그래서 시간마다 변하는 주파수를 제대로 분석하기 어렵다.

  예를 들면 0~10초 10Hz, 10 ~ 20초 20Hz인 신호 1)과

  0~20초 10Hz와 20Hz가 공존하는 신호 2)를 구별할 수 없다.

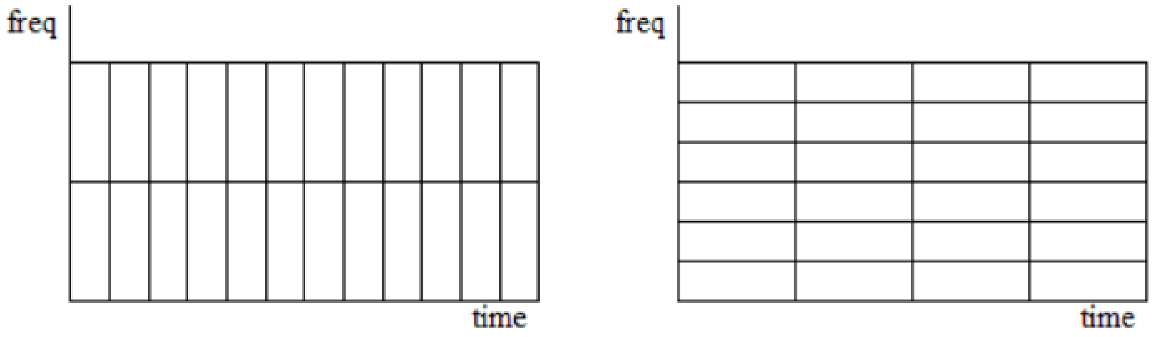
  단지, 10Hz와 20 Hz가 존재한다는 것만 알 수 있다. 이것은 악보에 음표가 시간에 따라 배열되어 있는 것이 아니라, 음의 높이(주파수)에 따라 배열되어 있는 것과 같은 것이다. **어떤 음들이 해당 곡에 들어있는지는 알 수 있지만, 실제로 이 음들로 어떤 곡이 탄생했는지는 알 수 없다.**

Short time Fourier Transform(STFT)

위의 Fourier Transform의 한계를 극복하기 위해서 단시간 푸리에 변환(STFT : Short Time Fourier Transform)이다. 시간에 따라 변화하는 **긴 신호를 짧은 시간 단위로 분할한 다음에 Fourier Transform을 적용**하는 것이다. 이렇게 하면 각 시간 구간마다 어떤 주파수들이 존재하는지 알 수 있다.

- 짧은 시간 단위로 신호를 분할할수록 어떤 '**시간**'에 어떤 주파수가 존재하는지를 알기 좋으며,
- 비교적 긴 시간 단위로 신호를 분할하면 어떤 '**주파수**'가 그 시간내에 존재하는지를 알기 좋아진다.

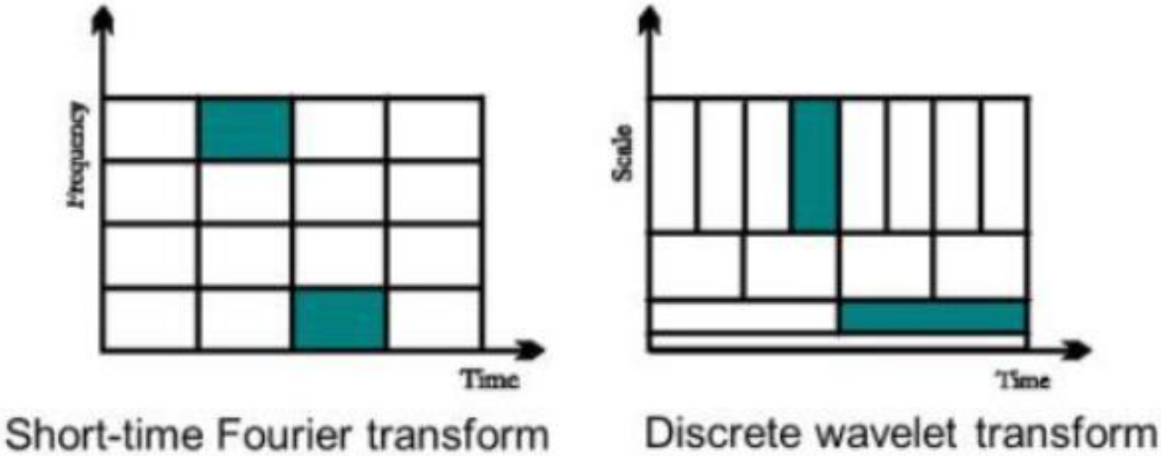
요약하자면, 신호를 분할하는 창(window)의 너비가 작아질수록 **시간 분해능**이 좋아진다는 것이고, 창의 너비가 넓어질수록 **주파수 분해능**이 좋아진다는 것이다.(분해능이 좋다는 것은 더 세밀하게 분석할 수 있다는 뜻)



[그림 2] 왼쪽이 짧은 시간 단위로 분할, 오른쪽이 비교적 긴 시간 단위로 분할(출처 : [8])

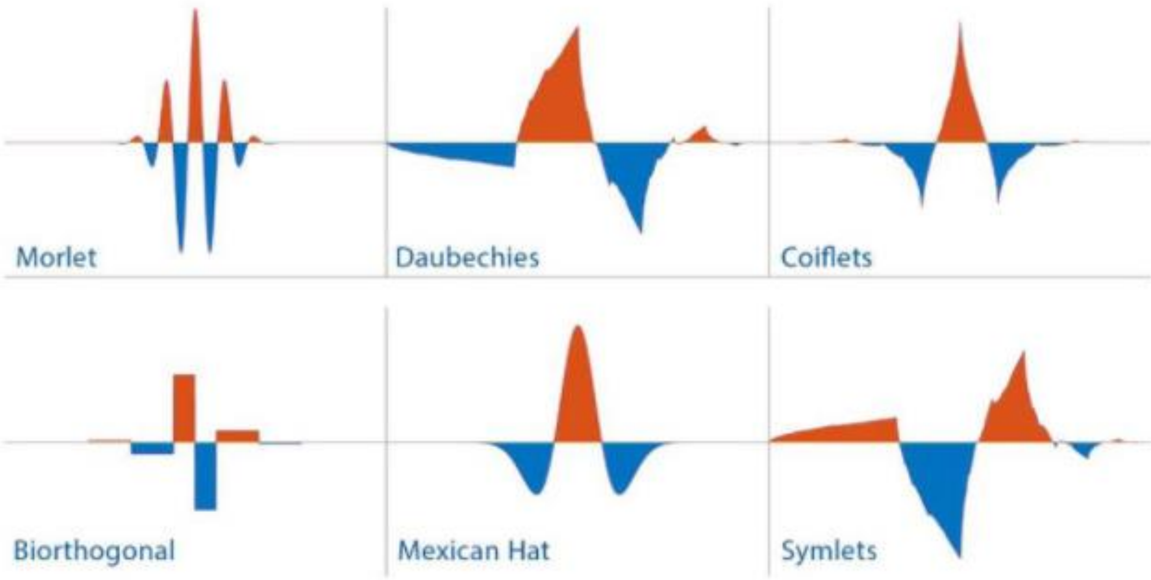
하지만 단시간 푸리에 변환의 경우 시간 분해능과 주파수 분해능은 tradeoff 관계이므로 둘 다 좋아지는 것은 불가능하다.

Wavelet Transform



[그림 3] STFT 와 Wavelet Transform 비교(출처 : [11])

오른쪽 그림과 같이 Wavelet Transform 은 **높은 Scale 의 신호에 대해서는 시간 해상도를 높이고 주파수 해상도를 낮추지만, 낮은 Scale 신호에 대해서는 주파수 해상도를 높이고 시간 해상도를 낮춘다.** STFT 가 시간 분해능과 주파수 분해능 둘 다를 잡기 힘들었다면, Wavelet Transform 은 둘 다 잡는다. 시간적으로 무한한 사인 곡선을 기본 함수로 사용하는 Fourier Transform 과 달리 Wavelet Transform 은 시간적으로 **한정**되어 있는 Wavelet Function 을 기본 함수로 사용한다. 또한 Wavelet 함수는 **여러 종류**가 있어 분야에 따라 선택해서 사용할 수 있다는 유연성을 가지고 있다.

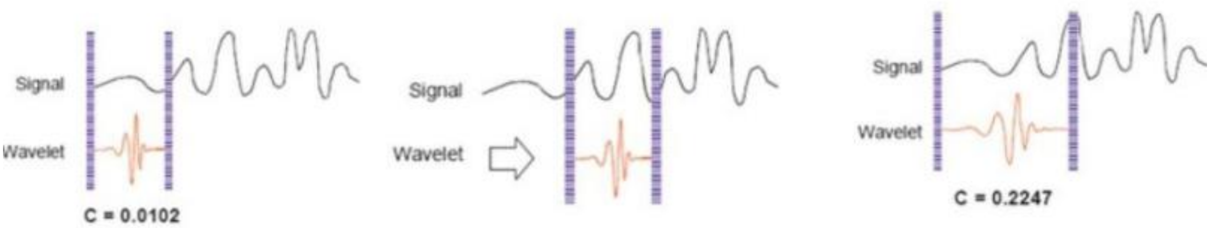


[그림 4] Wavelet Function 예시 (출처: https://www.mathworks.com/videos/understanding-wavelets-part-1-what-are-wavelets-121279.html)

위와 같은 Wavelet 을 Scaling(시간에 따라 확장 또는 축소)하고 Shifting(시간 축으로 이동)함으로써 신호를 분석해낼 수 있다는 것이 Wavelet Transform 의 핵심이다.

Wavelet 변환의 절차는 다음과 같다.

1) 작은 Scale 의 Wavelet 을 신호의 시작부터 끝까지 이동시키면서 비교한다. 신호와 웨이블릿의 유사도를 C 라는 계수에 담는다.
2) 신호의 끝까지 비교했다면, Wavelet 의 스케일을 좀 더 확장시킨 다음에 1)의 과정을 진행한다.
3) 설정한 스케일까지 이 과정을 반복한다. **작은 Scale 의 Wavelet 은 신호 내에서 급작스러운 변화와 가장 닮았기 때문에 그런 구간을 만날 때 큰 C 값을 준다. Wavelet 의 Scale 이 커질수록 신호 내에서 완만한 변화들과 가장 닮았기 때문에 그런 구간에 더 큰 C 값을 준다.**

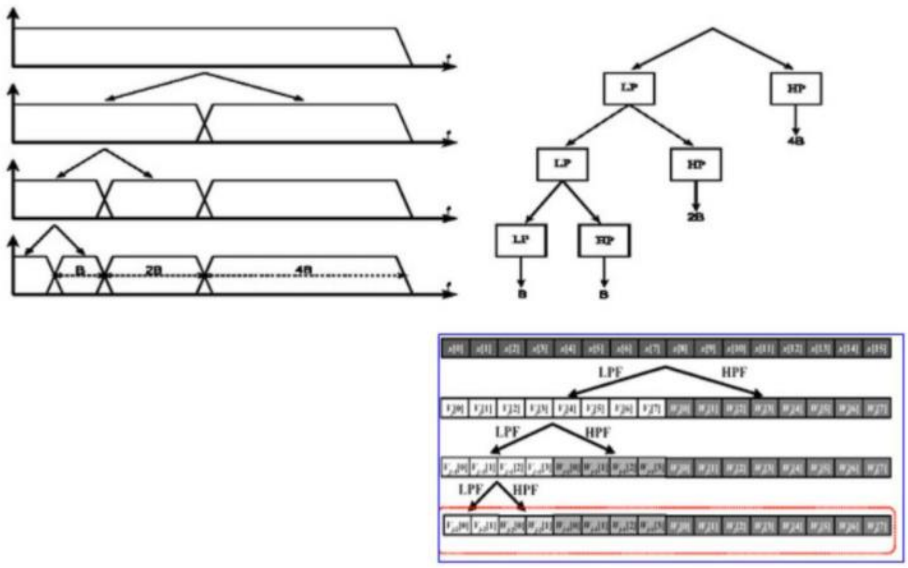


[그림 5] Wavelet Transform 절차 설명 (출처 : [11])

! Scale 은 || ||

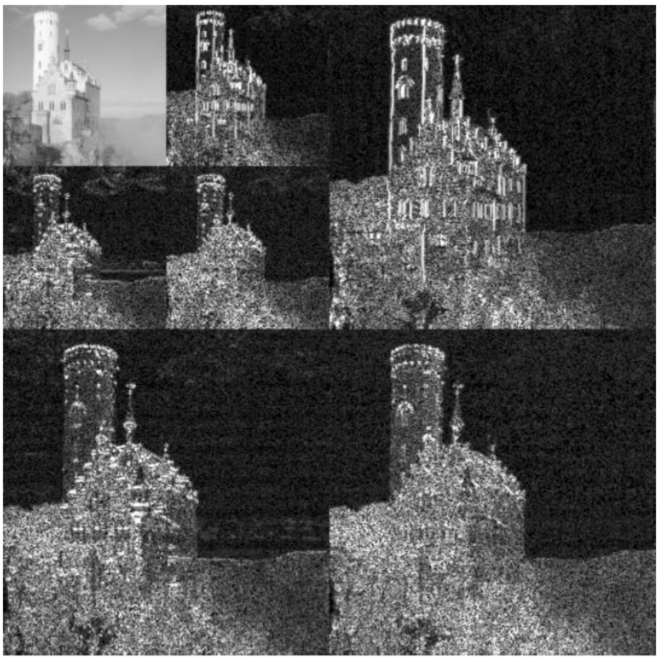
위의 폭이라고 생각하면 됨

Scale 이 작을 때 여러 구간으로 신호를 분리해서 분석하기 때문에 시간 분해능이 좋은 것이고, **스케일이 클 때 신호가 큼직하게 분리되지만 주파수에 대한 정보는 좀 더 잘 살펴볼 수 있기 때문에 주파수 분해능이 좋은 것**이다.



[그림 6] Wavelet Transform 은 신호를 근사값과 세부값으로 분해. 이 과정은 Multi Scale 로 반복될 수 있다.(출처 : [11])

결과적으로 Wavelet Transform 은 신호를 근사값(approximation)과 세부값(detail)로 분해한다. 근사값은 신호의 저주파 성분(전반적 내용)을 담고, 세부값은 신호의 고주파 성분(세부 사항)을 담는다. 이것은 사용자의 필요에 따라 원하는 만큼 반복될 수 있다.



[그림 7] 2 차원 영상의 Wavelet 분해의 예(출처 : https://en.wikipedia.org/wiki/Discrete_wavelet_transform)

2 차원 신호에 Wavelet Transform 를 적용하면 1 개의 근사값과 3 개의 세부 값(수평, 수직, 대각 세부값)으로 분해된다. Multi Scale 로 분해하면 근사값을 또 다시 1 개의 근사값과 3 개의 세부값으로 분해한다. 예를 들어, 400 x 400 이미지를 2 개의 스케일로 분해한다면 첫 단계에서는 200 x 200 사이즈의 근사값 이미지와 3 개의 세부값 이미지를 얻게 되고, 두번째 단계에서는 200 x 200 근사값 이미지가 다시 한번 분해되어 100 x 100 사이즈의 근사값 이미지와 3 개의 세부값 이미지로 분해된다.

영상 압축 기술인 JPEG2000 이 Wavelet Transform 을 활용한다. 방법은 Wavelet 분해 이후에 산출된 계수들을 양자화(Quantization)해서 계수의 개수를 줄인 뒤에 다시 합성하는 것이다. 결과적으로 이미지의 디테일들이 사라지기 때문에 그만큼 용량이 줄어들게 된다. 이 디테일들은 사람 눈에는 잘 보이지 않는 것들이므로 용량을 줄이기 위해서는 어느 정도 제거해도 괜찮다.

출처 : **https://bskyvision.com/404**

<참고자료>

[1] https://ko.wikipedia.org/wiki/%EC%9B%A8%EC%9D%B4%EB%B8%94%EB%A6%BF_%EB%B3%80%ED%99%98, 위키백과 웨이블릿 변환(한글)

[2] https://en.wikipedia.org/wiki/Wavelet_transform, 위키백과 웨이블릿 변환(영문)

[3] https://m.blog.naver.com/PostView.nhn?blogId=matlablove&logNo=221229281729&proxyReferer=https%3A%2F%2Fwww.google.co.kr%2F, Mathworks 가 제공하는 웨이블릿 변환에 관한 강의들.

[4] http://cfd.hanyang.ac.kr/index_kor.htm, 한양대 CFD 랩 웨이블릿 설명.

[5] http://clavez.tistory.com/54, clavez 님 티스토리, Wavelet 변환의 등장 배경 및 배경. Good!

[6] https://en.wikipedia.org/wiki/Fourier_transform, 위키백과 푸리에 변환(영문)

[7] https://ko.wikipedia.org/wiki/%ED%91%B8%EB%A6%AC%EC%97%90_%EB%B3%80%ED%99%98, 위키백과 푸리에 변환(한글)

[8] https://en.wikipedia.org/wiki/Short-time_Fourier_transform, 위키백과 단시간 푸리에 변환(영문)

[9] https://en.wikipedia.org/wiki/Multiresolution_analysis, 위키백과 멀티분해능 분석(영문)

[10] http://ryoo.tistory.com/attachment/pns3830.ppt, 홍정미, 문희윤님에 의해 작성된 웨이블릿 변환에 관한 PPT 자료. Good!

[11] https://slideplayer.com/slide/7537671/, 이 웨이블릿 변환에 관한 Alexander Kolesnikov 강의 PPT, Very Good!